DECENTRALISED AND PRIVACY-PRESERVING MACHINE LEARNING APPROACH FOR DISTRIBUTED DATA RESOURCES

A THESIS SUBMITTED TO THE UNIVERSITY OF MANCHESTER FOR THE DEGREE OF DOCTOR OF PHILOSOPHY IN THE FACULTY OF SCIENCE AND ENGINEERING

2022

Mona Ghotaish Alkhozae

Department of Computer Science

Contents

Word Count: 51885

List of Tables

List of Figures

List of Publications

- Alkhozae, M. and Zeng, X., 2021, September. Decentralised and Privacy Preserving Machine Learning for Multiple Distributed Data Resources. In *UK Workshop on Computational Intelligence* (pp. 235-250). Springer, Cham.
- Accepted paper in UK Workshop on Computational Intelligence 2022 (UKCI2022): "Alkhozae, M. and Zeng, X., 2022. Nonlinear Model Combination Approach to Decentralised and Privacy-Preserving Classification".

List of Abbreviations

FL	Federated Learning
DP	Differential Privacy
HE	Homomorphic Encryption
SMC	Secure Multiparty Computation
SGD	Stochastic Gradient Descent
DT	Decision Tree
ANN	Artificial Neural Network
RF	Random forests
SVM	Support Vector Machine
LR	Logistic regression
KNN	K-Nearest Neighbor
NB	Naive Bayes
LR	Linear Regression
RBFNN	Radial Basis Function Neural Network
RBF	Radial Basis Function
LASSO	Least Absolute Shrinkage and Selection Operator
SVR	Support Vector Regressor
KNNR	K-Nearest Neighbor Regressor
RFR	Random Forest Regressor
DTR	Decision Tree Regressor
NNR	Neural Network Regressor
ROC	Receiver Operator Characteristic
AUC	Area Under the Curve
TP	True Positive
TN	True Negative
FP	False Positive
FN	False Negative
SSE	Sum of Squares Error
RMSE	Root Mean Squared Error

MAPE	Mean Absolute Percentage Error
DS	Dynamic Selection
DCS	Dynamic Classifier Selection
DES	Dynamic Ensemble Selection
OLA	Overall Local Accuracy
LCA	Local Classifier Accuracy
MLA	Modified Local Accuracy
MCB	Multiple Classifier Behavior
KNORA	K-Nearest-Oracles
KNORA-E	K-Nearest Oracles Eliminate
KNORA-U	K- Nearest Oracles Union
DES-P	Dynamic Ensemble Selection performance
RC	Random Classifier
KNOP	K-nearest output profiles
DES-KNN	Dynamic Ensemble Selection using K-Nearest Neighbors
SA	Simple Average
HMV	Hierarchical Majority Voting
HER	Electronic Health Records
SV	Shapley Value
IE	Information Entropy
UPDRS	Unified Parkinson's Disease Rating Scale

Abstract

DECENTRALISED AND PRIVACY-PRESERVING MACHINE LEARNING APPROACH FOR DISTRIBUTED DATA RESOURCES

Mona Ghotaish Alkhozae

A thesis submitted to The University of Manchester for the degree of Doctor of Philosophy, 2022

Distributed machine learning has become a significant approach due to the high demand for distributed and large-scale data processing. However, some issues related to distributed machine learning for distributed data resources, including data transfer restrictions, privacy, and communication and computation costs have not been properly addressed. Therefore, it brings challenges to tackle these issues when developing a distributed learning method without data sharing between the distributed sites, centralising the distributed data resources for central learning, or using complicated learning methods.

In this thesis, we addressed these issues by developing decentralised privacy-preserving learning approaches that allow distributed sites utilising distributed data resources to construct global and local combined prediction models without sharing, moving distributed data to a centralised database or using a central location for iterative communication or computation. Furthermore, the exchanged information between distributed sites is restricted to only trained local models and information about models performance to overcome data restriction issues, privacy concerns, and minimising data transformation costs. We focused on several model selection and combination strategies to achieve the optimal combined global and local models that maximise the combined models predictive performance. We selected and combined the best models using linear and nonlinear combination methods, stepwise models selection and combination method, and by using all possible sites sequence combinations approach.

The experimental evaluation conducted on different classification and regression datasets demonstrated that our approach performed comparably or better than the centralised learning approach or other existing distributed learning methods in most datasets. Furthermore, we overcame data privacy concerns and server issues by avoiding data sharing or centralisation or using a server for iterative learning or intermediate models updates sharing. This thesis contributes toward developing a simpler and effective machine learning approach and direction for decentralised privacy-preserving machine learning. It keeps data locally for each site and combines diverse and accurate models instead of complicated ways that increase communication and computational overheads without sacrificing predictive performance. Furthermore, it can be applied to large and distributed data resources that cannot be analysed in a single location, reduces coordination overhead for large-scale analyses, and reduces cost by avoiding a powerful central server requirement.

Declaration

No portion of the work referred to in the thesis has been submitted in support of an application for another degree or qualification of this or any other university or other institute of learning.

Copyright

Acknowledgements

First of all, I would like to express my sincere thanks and gratitude to my supervisor Professor Xiao-Jun Zeng for his continuous support and guidance during my research work. He enriched my research knowledge and experience with his invaluable knowledge, precious advice, and feedback. Additional thanks to him for reviewing and proofreading this thesis.

I would like to extend my deepest thankfulness to my beloved parents Ghotaish and Maisha, and my sisters and brothers, who are always proud of me. They always believe in me and encourage me to be strong in difficult times, support me when I am down and depressed, and understand and take care of my daughters. Without them and their endless love and prayers to keep my spirits and motivation high during these times, I would not have been able to complete this thesis.

I am thankful for my beloved daughters, my happiness, Reem and Maram, for their patience during my study and their understanding when I can't be with them on their occasions and other celebrations. Thanks for the lovely and motivational messages they sent and still send to me: “we are so proud of you”, “you are the best mum”, “we love you, and you mean everything to us”, and “be strong”. These messages keep me motivated and strong to complete my study. Special thanks to my friends for the good times we had, their kindness, inspiration, and support. I’m lucky to have them.

Chapter 1

Introduction

1.1 CHAPTER OVERVIEW

This chapter views the research scope of this thesis in section 1.2 and the motivations in section 1.3. Sections 1.4 and 1.5 display the research questions and our objectives, respectively. Then, our contributions are presented in section 1.6. Finally, the thesis organisation is shown in section 1.7.

1.2 BACKGROUND

There is significant progress in distributed machine learning research, and more advanced and practical learning strategies are required in the fast-growing distributed learning environment. The increasing necessity of distributed machine learning applications is attracting more attention from developers and researchers due to the high demand for large-scale and distributed data processing [1]. It helps to reduce communication and computational overheads, improve data processing robustness and scalability, and overcome data privacy issues [1 - 4]. In addition, it solves the algorithm complexity and memory limitation problem in large-scale machine learning [5, 9].

There are different methods for distributed learning, such as combining models to develop a shared/global model, using ensemble learning methods, combining models results using meta-learning, clustering local models, and moving the distributed data to a central site for a centralised learning process to build a global model [9, 10]. Ensemble learning is centralised machine learning that relies on data resampling, such as Bagging and Boosting. It builds a set of models from different subsets of the training data and decides by voting or averaging the models outcomes [8, 13]. However, it is not possible when the data resources are distributed, not exchangeable, and private. Also, ensemble learning methods increase the computational time and are not practically scalable and suitable for distributed environments since the distributed data should first be collected at a central server and may face single-point failure or data privacy issues [3, 5, 9, 11, 12, 14]. Therefore, data privacy-preserving solutions and learning computation improvements are required and should be considered where the data are distributed.

Several distributed machine learning methods were proposed to develop a generalised model for distributed sites and overcome centralised machine learning issues and distributed data resource restrictions. There are two main categories of distributed machine learning. The first one focuses on developing distributed processing, where centralised data resources are processed by distributed and parallel computation. Such distributed machine learning methods aim to improve prediction performance, scalability, processing speed computation performance, memory limitation problems, and algorithm complexity [11, 14, 43]. Well-known technologies such as Hadoop [129] and Spark [130] are developed to scale up learning algorithms in distributed machine learning, process large data sets, and reduce the capital cost of distributed learning [5, 17, 132].

The other category focuses on distributed data resources to overcome limited network bandwidth, data restriction issues, privacy concerns, and minimising data transformation costs. In particular, privacy-preserving is often one of the most critical issues to be addressed. The distributed machine learning methods in this category are also often called decentralised machine learning. In the second category of distributed or decentralised machine learning, several distributed learning methods were proposed to solve distributed data resources issues and to develop a generalised model for distributed sites. It is suitable for applications that deal with large and distributed datasets that cannot be analysed in a single location. Furthermore, it is used to increase the performance of the models, a solution for algorithm complexity, and overcome centralised storage problems [5, 9, 93, 98].

A distributed learning approach called federated learning (FL) provides a practical approach to learning from distributed data without sharing data while protecting private data with privacy-preserving techniques. Data privacy is protected where a global model is developed by combining models that have been trained locally on distributed sites without collecting the data on a central server [35, 90, 103]. First, the data at each site is used to compute an update of a received model from a central server without sharing data. Then, the model update is transmitted back to the server and combines these model updates to compute a new global model. The global model is sent back to the sites, and the communication rounds continue until a needed convergence is achieved. FL uses a server to coordinate the learning process and perform many model parameters update iterations between the server and sites [88, 89, 95, 167]. It is designed to provide a secure architecture for distributed learning and preserve data privacy for the participated sites by avoiding sending the data to a central server [166, 167]. Various methods are proposed to preserve data privacy in distributed learning environments while ensuring good prediction performance. These methods preserve the privacy of model output or intermediate statistical results by using differential privacy (DP), cryptographic approaches to protect the data or results, or secure multiparty computation protocol (MPC) [1, 14, 59, 62, 63, 89, 90, 188, 189].

Several learning approaches aim to reduce communication costs, decrease model update time, fault tolerance and server issues, and privacy concerns related to models updates information. Decentralised machine learning is proposed as a solution for these challenges. It does not rely on a server to control the learning process between the sites. Instead, the computations and learning process control in decentralised learning are distributed among multiple sites and include all the benefits of distributed computing without moving or sharing the distributed data. Furthermore, it preserves data privacy, reduces coordination overhead for large-scale analyses, can scale to multiple sites, and reduces cost by avoiding the requirement for a powerful central server [100].

1.3 MOTIVATIONS

There have been considerable efforts on distributed machine learning for distributed data resources. However, developing a distributed learning method that overcomes several issues related to data privacy and restrictions, communication and computation costs, and server concerns is still challenging. In this thesis, we focus on developing decentralised learning approaches for private data without sharing or centralising the distributed data or using a server for coordinating the learning process. Also, we propose model selection and combinations methods to overcome the following challenges:

1. Distributed machine learning methods that involve transferring, sharing, or combining distributed data to create a central data resource are time-consuming and incur concerns related to storage cost, communication and computational costs, and data privacy issues. In addition, it faces single point of failure issues and scalability issues [14, 32, 63, 89, 95, 97, 103, 108, 109, 111]. Furthermore, merging data to a central location is not possible when the distributed data resources are private and not exchangeable. Therefore, this motivates us to find alternative learning strategies to overcome the centralised machine learning issues, preserve data privacy, and improve learning performance with distributed computation.

2. Some distributed learning methods are centralised modelling, exchanging many information and intermediate results to get a general model, or using complex techniques to preserve data privacy. For example, in FL, there are challenges related to communication costs due to the learning process involving many communication rounds between the sites and the server to build a global model, and privacy concerns related to models updates information [14, 32, 63, 89, 95, 103]. Furthermore, the server coordinates the learning process and performs many model parameters update iterations between the server and sites. Since the server takes control during the learning process, federated learning faces risks of attacks to extract private information from model gradients or single points of failure [89, 97, 165, 187, 191]. On the other hand, decentralised machine learning preserves data privacy, can scale to multiple sites, reduces cost by avoiding the requirement for a powerful central server, and reduces coordination overhead for large-scale analyses

[100]. Our motivation is to develop decentralised machine learning without using a server to control the learning process to overcome iterative learning overhead and central server issues.

3. Despite the effectiveness of privacy-preserving learning techniques, there are some challenges. For example, MPC increases computation and communication overheads and requirements [14, 58]. In addition, Differential privacy involves adding noise (data alteration) and may not be suitable for applications where high-quality models are needed as it reduces the quality of data and hence its utility [14, 62, 88]. Also, the encrypted training data requires high computational requirements [14, 59, 60, 96]. Thus, we aim to find a simple privacy-preserving learning approach that overcomes communication and computational issues without exchanging intermediate computing updates between sites and carefully considers data privacy during the learning process.
4. Model combination strategy is a promising approach to build an optimal combined model for distributed sites to improve the prediction performance. Computation and communication costs and the amount of transferred information are important factors that must be considered for efficient decentralised machine learning [4, 87]. Thus, paying attention to robust decentralised model learning and combination strategies is desired to improve the combined model performance. This triggers our motivation to develop an optimal combined model for distributed sites using a decentralised version of model selection and combination strategies that will enhance prediction performance without using complicated learning process to minimise the computation and communication costs. Furthermore, the proposed method only exchanges models with minimal data information instead of raw data to overcome data privacy issues.

1.4 RESEARCH QUESTIONS

The research questions in this thesis can be stated as follows:

- How to develop a global model from multiple data resources that are distributed, private, and not exchangeable to be as accurate as the models learned from centralising these data resources to a central database?
- Can a site improve its local prediction model performance by utilising learning

outcomes from other sites data resources without sharing, centralising, or disclosing the privacy of these data resources?

1.5 AIMS AND OBJECTIVES

The thesis aimed to achieve the following objectives:

1) Computation effectiveness by avoiding the time and cost of data centralisation or data transferring between the distributed sites during the learning process.
2) Preserve data privacy for each site by avoiding data sharing between the distributed sites, combining the distributed datasets into central data, or exchanging many intermediate results and information. Each site avoids sharing its local data with other sites and performs the learning process locally; thus, it keeps the data private and not exposed to other sites.
3) Avoid using complex privacy-preserving methods or server for iterative learning or controlling the learning process to minimise communication and computation overheads and avoid single points of attacks or failure risks.
4) Maintain the learning performance at a similar level as centralised machine learning with less iterative learning process and exchanged information between the distributed sites. We aim to build a global model from distributed data resources, preserve data privacy for each site, and improve prediction performance comparable to centralised machine learning.
5) Optimise the local prediction performance for each site by efficiently utilising its local models and other local models learning from other sites to develop a local combined model.
6) Develop an efficient model combination strategy by designing simple, accurate, and scalable combination methods to achieve optimal combined model at global and local levels.

1.6 CONTRIBUTIONS

The contributions can be described briefly as the following:

- Develop a decentralised learning approach for distributed data resources without using centralised machine learning method or relying on a central location for iterative communication or computation. We build combined prediction models derived from local learning outcomes at global and local levels. Our privacy contribution is keeping data locally on each site; only the learning outcomes and minimum information (local models and model performance information) are exchangeable. In addition, each site overcomes its local model limitations by utilising other sites models with minimal communication and information transfer to build its local combined model. We use simple and efficient linear model combination models to utilise and combine the selected models from distributed sites to develop the local and global combined models. This approach is presented in Chapter 3.

- Develop a robust combined model using a decentralised version of model selection and nonlinear model combination strategies and try to achieve the best-combined model at global and local levels that could improve prediction performance for all distributed sites. Furthermore, the proposed method preserves data privacy by avoiding data sharing or centralisation and with fewer communication rounds than FL to develop the local and global combined models. In chapter 4, we present the proposed approach.
- Develop a decentralised alternative to the federated learning approach without using a server or exchanging intermediate computing updates to overcome iterative learning process issues. We propose model selection and updating strategies that make the final combined model optimal and valuable for all sites. This approach could minimise communication and computation overheads and preserve data privacy by only passing the models between the distributed sites and updating the models locally in each location without exchanging models updates information. It only selects and updates the best-performing models in all sites and discards the others to decrease model update time and overhead. We use a simple linear combination method to combine the best-updated models to develop combined models at global and local levels with less information sharing between the distributed sites. We show our proposed method in Chapter 5.
- Develop a decentralised learning approach that applies all possible sites sequence combinations and model selection approaches to achieve the

optimal global combined model without exchanging data or models updates information between the distributed sites. These strategies can contribute to preserve data privacy by performing models selection and updating methods locally and minimise communication and computation overheads. This approach is presented in Chapter 6.

1.7 THESIS OUTLINE

This thesis is organised as follows.

Chapter 1: presents the research background, motivations, and objectives. Then, the research questions, contributions, and thesis organisations are stated in this chapter.

Chapter 2: shows background on distributed machine learning and its challenges, privacy-preserving methods, machine learning techniques, and performance evaluation measures. In addition, it presents model selection and combination approaches and the research works proposed in the distributed learning field. Finally, this chapter views the research limitations and challenges, followed by our proposed methods to overcome these challenges.

Chapter 3: presents a proposed decentralised learning approach using a linear combination approach to develop global and local combined models for distributed sites. Then, it shows the analysis and discussion of the results.

Chapter 4: shows our proposed decentralised learning approach using a nonlinear model combination approach to build global and local combined models and presents our analyses and discussion of the results.

Chapter 5: presents a proposed decentralised learning strategy to build global and local models using Gossip learning method, stepwise model selection, and a linear model combination approach. Then, it views the analysis and discussion of the results.

Chapter 6: shows our proposed decentralised learning method for global combined model development using all possible sites sequence combinations approach. Also, it presents our discussion and analyses of the results.

Chapter 7: This chapter shows our conclusions for this thesis and outlines future research directions.

Along with these chapters, the thesis contains four appendices. Appendix A presents 5-fold and 10-fold Cross-Validation Results, and datasets distributions and Appendix B shows the detailed global and local level modelling results in chapter 3. Appendix C shows the detailed results of the decentralised learning approach in Chapter 5, and Appendix D illustrates the detailed results of the decentralised learning approach in Chapter 6.

Chapter 2

Background and Related Works

2.1 CHAPTER OVERVIEW

This chapter presents background information for the topics covered in the rest of the thesis. Sections 2.2 and 2.3 cover machine learning techniques and performance evaluation metrics, respectively. Distributed machine learning is presented in section 2.4, and section 2.5 shows the challenges in distributed machine learning. Section 2.6 shows privacy-preserving machine learning approaches, and model combination approaches are presented in Section 2.7. Several research works related to distributed machine learning, model selection, and combination methods for distributed data resources are shown in Section 2.8. Analysis and discussion of research challenges and limitations in distributed machine learning field are presented in Section 2.9. Finally, this chapter is summarised in Section 2.10.

2.2 MACHINE LEARNING TECHNIQUES

Machine learning techniques aim to develop models to perform different tasks such as classification, estimation, and prediction. Predictive modelling is developing a model that can predict an outcome based on given input variables. When choosing a machine learning technique, optimising the selection of predictive modelling method options can be beneficial for increased reliability and performance [113]. Classification and regression are the most common machine learning algorithms.

In classification, the learning function classifies the input data into one of several predefined output classes. Many learning algorithms are used to build classification models, such as Decision Tree (DT) [112, 145, 146], Artificial neural network technique (ANN) [112, 145], Random forests (RF) [145, 149, 150], K-Nearest Neighbor algorithm (KNN) [145, 148], Support Vector Machine algorithm (SVM) [112, 145], Logistic regression (LR) [145], and Naive Bayes (NB) algorithm [112, 145]. Regression is a supervised learning and statistical task used to define the relationships between inputs and continuous outputs from training data and to predict new input data [145]. The regression algorithms are Linear regression (LR) [145], Regression trees [147], A Radial Basis Function Neural Network (RBFNN) [114, 115], Selection Operator regression (LASSO) [151,172-174], Ridge [153, 172-174], ElasticNet [152, 171-174], Support Vector Regressor (SVR) [157, 173], K-Nearest Neighbor Regressor (KNNR) [148, 159], Random Forest Regressor (RFR) [149, 173], Decision Tree Regressor (DTR) [146, 173,174], and Neural Network Regressor (NNR) [158].

Data preprocessing and learning algorithms selection can significantly affect model performance. Data preprocessing is a common and important step for preparing raw data to a proper and understandable format for a learning algorithm. Basic data preprocessing techniques such as data cleaning, data transformation, and feature extraction and reduction make data adequate for machine learning tasks [41, 43, 113].

2.3 PERFORMANCE EVALUATION METRICS

Once a model is developed using a machine learning algorithm, it is essential to estimate its performance. Cross-validation is one of the most common approaches for evaluating a model performance by holding out a subset of the training data for evaluation and then repeating this process across several subsets/partitions. The result is calculated as the average of all process rounds results [112]. Cross-validation is the standard way to evaluate the robustness of the model and minimise overlap among dataset partitions [39, 125]. For reliable results, 10-fold cross-validation is suitable [39].

In classification tasks, Confusion Matrix, Receiver Operator Characteristic (ROC) curve, and Area Under the Curve (AUC) are used to measure the model performance and test a classification model quality [145]. As illustrated in Table 2.1, the confusion matrix is a table of prediction results to visualise model performance. The number of correct and incorrect classification predictions are summarised with count values and divided by each class label.

Table 2.1. Confusion Matrix

		Actual Class	
		Positive (P)	Negative (N)
Predicted Class	Positive (P)	True Positive (TP)	False Positive (FP)
	Negative (N)	False Negative (FN)	True Negative (TN)

The ROC curve is a popular validation method for assessing model performance [116], and AUC of ROC curve measures the model predictive capacity [117]. As shown in Figure 2.1, the ROC curve is a graph that plots True Positive Rate (Y-axis) against the False Positive Rate (X-axis). AUC is based on the ROC curve, and the AUC threshold values range from 0.5 to 1. 1 indicates the best performance, and 0.5 shows poor performance [112].

$$AUC = (\sum TP + \sum TN) / (P + N), \qquad (2.3)$$

where $\sum TP$ is the total number of positive examples that are correctly predicted, $\sum TN$ is the total number of negative examples that are correctly predicted, P is the total number of positive examples, and N is the total number of negative examples.

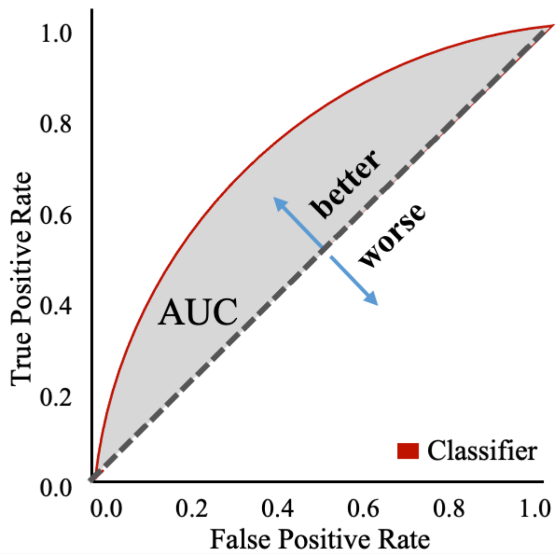


Figure 2.1: ROC curve and AUC

In addition, model performance is measured in terms of sensitivity (recall), specificity, precision, F- measure, and accuracy. Sensitivity (recall) is the ratio of true positives correctly predicted by model, whereas specificity is the ratio of true negatives correctly predicted. Precision is the correct prediction ratio of the positive cases to all positive cases, and accuracy is the ratio for the correct predictions [101, 112]. F-measure score is a harmonic mean of precision and recall and the most commonly used performance metric of machine learning, mainly when the data set is unevenly distributed. Since F-measure equally weights both false positives and false negatives, it offers a less biased metric than accuracy [155, 184]. The accuracy, sensitivity, specificity, precision, recall, and F-measure were calculated as follows [145, 155]:

$$\text{Accuracy} = (\text{TP} + \text{TN}) / (\text{TP} + \text{TN} + \text{FP} + \text{FN}) \qquad (2.4)$$

$$\text{Sensitivity (recall)} = \text{TP} / (\text{TP} + \text{FN}) \qquad (2.5)$$

$$\text{Specificity} = \text{TN} / (\text{FP} + \text{TN}) \qquad (2.6)$$

$$\text{Precision} = \text{TP} / (\text{TP} + \text{FP}) \qquad (2.7)$$

$$\mathrm{F-measure} = \frac{2 \times \mathrm{precesion} \times \mathrm{recall}}{\mathrm{precison} + \mathrm{recall}} = 2\mathrm{TP}/(2\mathrm{TP} + \mathrm{FP} + \mathrm{FN}), \qquad (2.8)$$

where TP is true positive, TN is true negative, FP is false positive, and FN is false negative examples.

For regression tasks, Sum of Squares Error (SSE), Root Mean Squared Error (RMSE), and Mean Absolute Percentage Error (MAPE) are simple and easy to compute to evaluate model performance. The prediction error is the difference between the actual and predicted values [101, 162]. The Root Mean Square Error (RMSE) metric is defined as a distance measure between the predicted and the actual value. The smaller RMSE value, the better is the predictive model performance. The mean absolute percentage error (MAPE) measures a model accuracy as a percentage [101].

$$\mathrm{SSE} = \sum_{i=1}^{m} |y_i - \hat{y}_i|^2 \qquad (2.9)$$

$$\text{RMSE}= \sqrt{\frac{1}{m} \sum_{i=1}^{m} |y_i - \hat{y_i}|^2} \tag{2.10}$$

$$\text{MAPE} = \frac{1}{m} \sum_{i=1}^{m} \left| \frac{y_i - \hat{y_i}}{y_i} \right| *100, \tag{2.11}$$

where y_i is the actual value and $\hat{y_i}$ is the predicted value for m data samples ($i = 1, \dots,$ m).

2.4 DISTRIBUTED MACHINE LEARNING

There is considerable progress in distributed machine learning research field. The objective of distributed machine learning is to perform the learning task based on the distributed resources, including data, processors, and machine learning algorithms. More advanced and practical distributed learning strategies are essential in the current fast-growing environments [2, 98]. There are two main categories of distributed machine learning. The first one focuses on distributed processing, where centralised data resources are processed by distributed and parallel computations. Such distributed machine learning methods aim to improve scalability, processing speed computation performance, memory limitation problems in large-scale environments, and algorithm complexity [11, 14, 43]. Well-known technologies such as Hadoop [129] and Spark [130] are developed to scale up learning algorithms in distributed machine learning. The other category focuses on distributed data resources to overcome limited network bandwidth, data restriction issues, privacy concerns, and minimising data transformation costs. Machine learning for distributed data resources mainly falls into three approaches: centralised machine learning, distributed/federated machine learning, and decentralised machine learning. In centralised machine learning, distributed data is transferred to a central location for the learning process. In contrast, distributed and decentralised learning is developed when data transmission between distributed sites is not allowed due to data privacy restrictions.

2.4.1 Centralised Machine Learning

In centralised machine learning, distributed data is moved to a central server for the learning process. As illustrated in Figure 2.2, all the computations are carried out on a server for central processing to train a general model that can be applied to all distributed sites [9, 14]. Ensemble learning approaches are under centralised learning process and aim to enhance the prediction performance, such as Bagging and Boosting. In ensemble learning, a learning algorithm is trained over a dataset and relies on data resampling to build a set of models. Then, combine these models outputs to get the final prediction [14]. However, the centralised learning approach is infeasible for most machine learning applications due to data privacy and protection rules, internal policies forced by some organisations, or limited network bandwidth [4, 140].

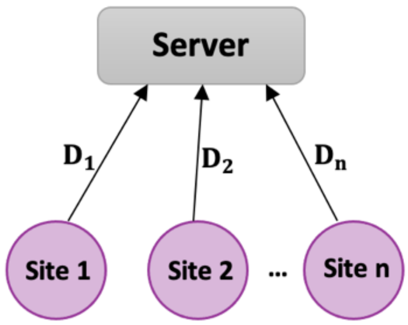


Figure 2.2: Centralised Machine Learning

For this reason, there is an increasing interest in building a general model using data distributed across different sites without transferring or moving data between distributed sites or to a central location [51, 140]. Distributed/Federated and decentralised learning methods deal with the above issues to learn from distributed data without moving or sharing the distributed data [9].

2.4.2 Distributed and Federated Machine Learning

Distributed machine learning is adopted due to the high demand for large-scale and distributed data processing. It is a way of scaling up learning algorithms by allocating the learning process on multiple sites and performing the learning algorithms over physically distributed datasets. In addition, it solves issues with algorithm complexity, improves processing speed and computation performance, and memory limitation problems in large-scale machine learning [5, 9, 93, 98]. In distributed learning, the computations are distributed to multiple sites, and the data are processed in a parallelised fashion, iteratively training a model on isolated datasets, and obtaining a shared model as if data were centralised [14]. Several open-source software projects enable distributed computing, such as Hadoop [129] and Spark [130].

Distributed learning approaches allow multiple sites to keep their datasets unexposed and collaborate on a learning objective by iterative local computation and message passing. Thus. It reduces network traffic, less network bandwidth, and preserves data privacy [1, 2, 4, 14, 92]. Federated learning (FL) is an alternative to allow for the collaborative training of models without sharing raw data and is proposed to solve data privacy and restriction issues [90]. As shown in Figure 2.3, several sites participate in the learning process, while the server is responsible for controlling the modelling process, and a global model is learned by aggregating models that have been trained locally on distributed sites [103]. FL approach preserves the data privacy for each site by performing collaborative learning that never requires the data in the distributed sites to be centralised. First, the server learns a global model based on its available data and then sends the model to the distributed sites. Secondly, each site trains the received model and computes model updates based on the local data. Next, the updated model parameters are sent back to the server. Finally, the server updates its global model based on the received model parameters to generate a new global model. This process is repeated until the model parameters desired convergence level is achieved [14, 95, 102, 140].

Several methods aim to reduce communication costs, decrease model update time, fault tolerance and server issues, and privacy concerns related to models updates information. Thus, decentralised machine learning is proposed as a solution for these challenges.

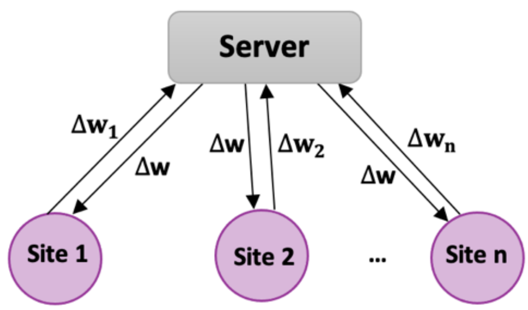


Figure 2.3: Distributed/Federated Machine Learning

2.4.3 Decentralised Machine Learning

As illustrated in Figure 2.4, both the computations and learning process control in decentralised machine learning are distributed among multiple sites and include all the benefits of distributed computing. Gossip learning is an example of decentralised learning for learning models from distributed data without central control.

I. Gossip Learning Approach

It is based on multiple models that move between distributed sites over the network. The models are trained on sites local data using online learning algorithm to improve the models performance, and then combined using combination learning methods. Online algorithm is a stochastic gradient descent (SGD) and can be applied as a learning algorithm to update the models using a continuous stream of data records. The models are updated when visiting a site using the local data. First, the site initialises a model and then sends the model to another site over the network. Then, each site updates the received model using its local data and combines it with its local model. Combining models is achieved by averaging model parameters based on the gradients number that contains the given parameter and updating it using the local data set. It is a scalable and robust learning process, and there is no single point of failure [139 - 141].

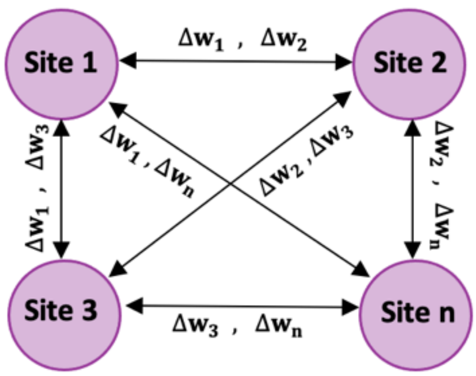


Figure 2.4: Decentralised Machine Learning

II. Stochastic Gradient Decent (SGD)

One of the most easily distributed learning algorithms is Stochastic Gradient Decent (SGD). Supervised learning can be considered as an optimisation problem, where we want to minimise the empirical risk $E(f)$ of the model $f(x)$[142,156]:

$$E(f) = \frac{1}{n} \sum_{i=1}^{n} Q\ (z_i, w) = \frac{1}{n} \sum_{i=1}^{n} \ell\ (f(x_i; w), y_i), \qquad (2.1)$$

where $Q\ (z_i, w) = \ell(f(x_i), y_i)$ is a loss function of model for training data example z_i , $z_i = (x_i, y_i)$, i= 1, …, n, and w is the model weight/update.

In SGD, the goal is to minimise the model loss function. The basic idea is that they iterate over the training examples in random order repetitively. For each training example, it calculates the gradient of the loss function, then modifies the model along this gradient to reduce model error on this example according to the following rule [142,156]:

$$w_{t+1} = w_t - \gamma_t \nabla_w Q(z_t, w_t), \qquad (2.2)$$

where γ_t is the learning rate at iteration t, w_{t+1} and w_t are the model weight/update, and $\nabla_w Q(z_t, w_t)$ is a gradient estimate of the loss function.

Update the model with the gradient of the sum of the loss functions of a few data examples using mini-batches instead of only one in each iteration can be used for fast distributed implementations and accelerated model convergence [143,161].

2.5 DISTRIBUTED MACHINE LEARNING CHALLENGES

Distributed machine learning methods that involve transferring data, sharing data, or combining distributed datasets to create a central data resource will incur serious concerns such as storage costs, data transformation costs, and data privacy issues [32, 108, 109, 111]. Pooling distributed data at a central location for learning is impractical for sensitive and private data. Centralised machine learning is not feasible in distributed environments and could increase bandwidth overhead and computational load, single point of failure issues, and scalability issues [3, 5, 9, 14, 51, 93, 100].

One solution is relying on a central server for coordination/computation without sharing data between distributed sites. Federated learning has increased interest from the research community for distributed machine learning. It preserves data privacy by learning a shared model without moving the data to a central server. The adoption of distributed and federated learning shows emerging approaches to manage privacy concerns and facilitate scientific research without directly sharing data [35, 100, 101].

Challenges in federated learning include communication costs between the sites and the server, learning process time, and privacy concerns related to models updates information [14, 32, 63, 89, 95, 103]. Therefore, it is crucial to minimise communication costs and provide privacy-preserving techniques for robust privacy protections [14, 102]. In addition, the server takes control during the learning process, so federated learning faces risks of a single point of failure or attacks [89, 97, 187, 191].

Decentralised machine learning does not rely on a central server to control the learning process between the sites. As a result, it preserves data privacy, can scale to multiple sites, reduces cost by avoiding the requirement for a powerful central server, and reduces coordination overhead for large-scale analyses [100].

2.6 PRIVACY-PRESERVING IN DISTRIBUTED MACHINE LEARNING

Distributed and federated learning approaches are proposed to preserve data privacy during the learning process without sharing the data between sites. Nevertheless, there is privacy concern related to models updates information exchanged between sites. Therefore, it is essential to provide privacy-preserving techniques for robust privacy protections [14, 102]. Privacy-preserving federated learning approaches aim to avoid inference over the messages exchanged between sites during the learning process while ensuring the final model has acceptable performance [90] and protecting model gradients during training from malicious attacks [187]. There are many techniques used to preserve data privacy in the federated learning approach, by using: Differential Privacy (DP), Cryptographic methods such as Homomorphic Encryption (HE), or Secure Multiparty Computation protocol (SMC). Differential Privacy (DP) preserves the privacy of models outputs or intermediate analysis and statistical results by adding noise into the gradients and data [1, 14, 63, 90]. HE allows secure computations over encrypted data or results without decryption on a computing platform [14, 59, 62, 89]. SMC enables distributed computations on encrypted data without decryption and excludes the need for a central trusted location for computations [59, 89, 90]. Each data is divided into several parts and allocated these parts to the sites. Then, the sites follow a protocol and compute a function on their inputs without revealing their information to other sites, and the final result is shared among them. However, SMC alone is not enough to preserve data privacy. Thus, a combination of SMC with other methods such as DP is required for better privacy results [14].

2.7 MODEL COMBINATION APPROACHES

There are two important issues in combining multiple models to build an optimal combined model. One is that the models must be diverse and accurate to perform better. The other issue is the combination method, which is regarding how to combine the outputs of individual models [123, 124].

2.7.1 Models Selection Strategies

A fundamental step in building an efficient combined model is the selection of models. However, a poor model selection strategy could lead to a combined model performing worse than the individual models' performance that formed the combined model [32]. Therefore, several selection strategies are used to build a combined model.

I. Static and Dynamic Model Selection Strategies

In static selection, the best model selection is defined during a training phase, such as Boosting and Bagging. While in dynamic model selection, the model selection is made during a testing phase. For each new sample, the most competent models are selected by some competency measures [52, 53]. In Dynamic Selection (DS), the key is how to choose the most competent models for any given new sample [67]. First, the competence of the models is estimated based on a local region of the feature space where the new sample is located. This region can be defined by different methods, such as applying the K-Nearest Neighbors technique, finding the neighbourhood of this new sample, or using clustering techniques. Then, the competence level of the base models is estimated, considering only the examples belonging to this local region according to a selection criterion; these include the accuracy of the models in this local region or ranking and probabilistic models. The model that achieved a definite competence level is selected [65, 67,104, 160].

Dynamic model selection is divided into Dynamic Classifier Selection (DCS) and Dynamic Ensemble Selection (DES). In DCS, only one model is selected for each test example, such as Overall Local Accuracy (OLA) [54], Local Classifier Accuracy (LCA) [54], Modified Local Accuracy (MLA) [55], Classifier Rank [56], and Multiple Classifier Behavior (MCB) [57]. In DES, an ensemble of models is selected for each test example. For a new test example, DES aims to select the competent models for the local region in the feature space where the test example is located [65, 160]. For example, META-DES assumes that the DES problem is a meta-problem. Meta-features are derived from the training data and represent different criteria to measure the model competence level for new examples classification. It is used to learn a meta-model to predict whether a model is competent to classify an input example or not [65]. First, the meta-features for each model are obtained from the input example and sent as input to the meta-model. Then, the meta-model estimates the competence level of the model for the new example classification. Finally, the models with a competence level higher than a defined threshold are selected [65]. In K-Nearest-Oracles (KNORA) based approaches, the accuracy metric is used as the selection criterion. It first finds K-Nearest Neighbors of the example to be classified and then selects a subset of models which can correctly classify all the neighbors. Finally, majority voting is usually used to combine the outputs of the selected models. K-Nearest Oracles Eliminate (KNORA-E) and K- Nearest Oracles Union (KNORA-U) are based on KNORA approach [160]. KNORA-E selects the models that identify all data examples belonging to the competence region and then applies the majority voting method.

In contrast, KNORA-U selects the models that correctly classify at least one example in the region of competence and perform a weighted voting method. The model weight is based on the number of correctly classified instances. In KNORA-E approach, the competence region size is reduced if no model is selected, and then start searching for competent models [66]. Dynamic Ensemble Selection performance (DES-P) selects all models that achieve a classification performance in the local region of competence higher than the random classifier (RC) performance. The performance of the random classifier is defined by RC = 1/C, where C is the number of classes in the problem [18]. K-nearest output profiles (KNOP) use accuracy metric to evaluate the models. It is similar to the KNORA-U approach, but KNORA-U works in the feature space, while KNOP works in the decision space to estimate the region of competence. K-Nearest Neighbors (DES-KNN) uses diversity and accuracy measures to select the models. First, the most accurate models in the region of competence are selected. Then, the more diverse models from the most accurate models are selected [67]. After the model selection step, it is essential to consider the selected models in a proper model combination approach to improve overall prediction performance.

II. Stepwise Model Selection Strategy

It aims to optimise combined model performance. It is adopted to add superior models or remove poor models from a given set of candidate models according to a specified performance threshold. The remaining models are used to develop the combined model [105, 106]. The stepwise model selection strategy selects desired surrogate models from a given set of candidate models within a sequence. It begins with a combination of models. Then, if a model was not in the combination but significantly affected the response positively, the model is added to the combination. If a model in the combination does not significantly affect the response, the model is removed from the combination. This process is repeated until no model needs to be added or removed from the combination. Finally, the final selected models are combined using combination methods [106].

2.7.2 Model Combination Methods

Since distributed learning aims to produce a learning result for distributed data resources, the learning outcomes from distributed sites must be appropriately combined. This can be achieved by combining local models predictions or local models [92, 121]. An optimal combination approach is required to develop an efficient combined model to improve distributed learning performance [4, 40]. The combination approach should take advantage of local models strengths while ignoring models weaknesses. Its success depends on how accurate, diverse, and independent the individual models are and how well the combination method can be determined [13, 83, 108, 125]. Diversity can be achieved by combining distributed and heterogeneous models, such that these models disagree with when predicting a new test data. Therefore, combining diverse models will balance the individual models weaknesses and produce a good combination approach, even with the simplest combination methods [18, 125, 126].

When the models outputs are discrete values, majority voting and weighted voting combination methods are commonly used [163]. The majority voting selects the final predicted value based on the most voted predicted class. The weighted voting method is a linear combination method used to improve model performance by combining models prediction results and selecting the highest vote based on the individual models weights. Therefore, it is required to define proper weight for each model, such as using average model performance in training data to assign the model weight. Averaging method takes the average sum of model performance to get the final prediction result [163]. If the models outputs are continuous, other model combination methods, such as the simple average, sum, weighted average, etc., can be used [121]. Linear and nonlinear combination methods are developed to combine the models outputs. Linear model combination methods combine models outputs in a linear fashion, such as the sum, mean, median, simple average, weighted average, etc. Nonlinear combination methods include rank-based combiners or using nonlinear learning algorithms such as decision tree, neural network, and support vector machine to combine models outputs as a high-level model (meta-model) [31, 120, 125].

1. Linear Combination Methods

In the linear combination methods, the final model result is a linear combination of several models. It is the simplest combination method, does not require learning procedures, and balances individual models overfitting [125]. A combination model can improve prediction performance if it combines models in an appropriate way. It is the most frequently used combination rule, and it has been shown that a linear model combination would give a smaller error variance than any single model. As a result, the accuracy could be improved for the model. Linear combination method uses the weighted-average method to predict the results obtained by all individual models [115, 123]. It was first proposed by [131]; the basic form of the linear combination model can be formulated as [131]:

$$f(x) = \sum_{i=1}^{m} w_i f_i(x), \qquad (2.12)$$

where w_i is the weight of the ith prediction model, f_i is the ith predictive value, m the number of prediction models, and $\sum_{i=1}^{m} w_i = 1$, $w_i \geq 0$; $1, 2, \ldots, m$.

This model combination approach success depends on how well the weights in model combination can be determined [13, 83, 108, 125]. The model weighting method can be defined as assessing the efficiency of each model and assigning an appropriate weight value that reflects model importance in the combined model. Each model weight coefficient is calculated by examining the actual and predicted values of training data. Thus, the coefficients are assigned to the models considering their prediction performance, and finally, the weighted predictions are combined. It would affect the model accuracy directly, so selecting a proper weighting coefficient is the core [122, 128, 135]. It can be estimated in different ways:

1) Simple Average Method (SA):

In SA, all models are equally weighted; the weight of each model is equal to the average models number.

$$\mathrm{w_i} = 1/n \text{, i= 1, 2, …, n,} \qquad (2.13)$$

where, $0 \leq \mathrm{w_i} \leq 1, \sum_{\mathrm{i=1}}^{\mathrm{n}} \mathrm{w_i} = 1$, and n is the number of models.

It is the simplest method, but it ignores the model performances in the combination and is affected by extreme values or outliers; therefore, the variation of model errors can be high [118]. Simple Average method is the optimal linearly combining method if the individual models show identical performances and error variance. Otherwise, assigning weight based on model performance can provide improved results [45, 83, 84, 86].

2) Error-based Method:

The weight for each model is inversely proportional to the model error. A small weight is assigned to the model with a large error and vice versa [118].

$$w_i = e_i^{-1} / \sum_{i=1}^{n} e_i^{-1}, \text{ i= 1,2, …, n}, \qquad (2.14)$$

where e_i denotes the error of the ith model, $0 \leq w_i \leq 1, \sum_{i=1}^{n} w_i = 1$ and n is the number of models.

3) Performance-based Method (Accuracy):

In the performance-based method, the weight for each model M_i is related to the model accuracy. The larger weight is assigned to the model with larger accuracy.

$$w_i = \frac{\text{Accuracy}(M_i)}{\sum_{i=1}^{n} \text{Accuracy}(M_i)}, \qquad (2.15)$$

where i=1, 2, …, n and $\sum_{i=1}^{n} w_i = 1$.

For example, the accuracy of the classification model is calculated by dividing the number of correct predictions by the total predictions number. The model accuracy for the regression model is based on the difference between the observed and predicted values. First, we calculate the Mean Average Percentage Error (MAPE), and then we compute the model accuracy.

$$\text{MAPE}(M_i) = \frac{1}{n} \sum_{i=1}^{n} \left| \frac{y_i - \hat{y_i}}{y_i} \right| * 100, \qquad (2.16)$$

where y_i is the actual value and $\hat{y_i}$ is the predicted value for n data samples.

$$\text{Accuracy}(M_i) = 100 - \text{MAPE}(M_i) \qquad (2.17)$$

Assigning model weight based on its performance is both a rational and advantageous approach and overcoming the limitations of the statistical combination approaches such as the simple average method [86].

4) Weighting Method using Shapley Value:

The Shapley Value is a mathematical method used to solve cooperation games with multiple players. It allows achieving an efficient and fair allocation of team total revenue among the several cooperative players. The aim of reducing the combined model prediction error can be considered as the goal of a cooperative game, where each model is a player in the process of combining the model. It calculates the model weight coefficient by evaluating the model contributions and corresponds to the average of the marginal contributions of the models associated with all possible models orders across all combinations. The marginal contribution is the difference between the prediction value when including that model, and the prediction reached without including that model. Every model weight can be determined according to the error distribution [127, 128, 133, 134]. Model weighting method using the Shapley Value steps as follow:

1) Suppose we have a set S of n models taking part in the combined model, S = {1, 2, …, n}, and e_i is the prediction error of model i, i= 1, 2, …, n.
2) Calculate total error E of all prediction models:

$$E = \frac{1}{n}\sum_{i=1}^{n} e_i \qquad (2.18)$$

3) Calculate the combined error $E(s_i)$ of a combination subset s_i:

$$E(s_i) = \frac{1}{m}\sum_{i=1}^{m} e_i, \qquad (2.19)$$

where s_i is a subset of S, and m is the number of the models in s_i.

4) Calculate the Shapley Value of the model i:

$$\varphi_i(v) = \sum_{si \in S} \frac{(n-|s_i|)!((|s_i|)-1)!}{n!} * [E(s_i) - E(s_i - \{i\})], \quad (2.20)$$

where $|s_i|$ is the number of models in the combination, and $s_i - \{i\}$ is a set obtained from s_i by removing i-th model from the combination.

5) Finally, calculate the weight of model i:

$$w_i = \frac{1}{n-1} * \frac{E - \varphi_i(v)}{E} \quad , \text{i}= 1,2, \dots, \text{n} \qquad (2.21)$$

The following is an example of the Shapley Value calculation for a model in a combination of 3 prediction models:

- If we have 3 models I = {1, 2, 3} and its errors $e_1 = 9$, $e_2 = 8.49$, and $e_3 = 8.92$
- All possible models subsets are: $\text{s}_1 = \{1\}$, $\text{s}_2 = \{2\}$, $\text{s}_3 = \{3\}$, $\text{s}_4 = \{1, 2\}$, $\text{s}_5 = \{2, 3\}$, $\text{s}_6 = \{1, 3\}$, and $\text{s}_7 = \{1, 2, 3\}$
- The models subsets that contain model 1 are $\text{s}_1, \text{s}_4, \text{s}_6$, and s_7.
- The total error E = 8.8
- The combined error of the combination subset s_i are: $E(s_1) = 9$, $E(s_4) = 8.7$, $E(s_6) = 8.9$, $E(s_7) = 8.8$
- Then, calculate the Shapley Value for model 1 in $\text{s}_1, \text{s}_4, \text{s}_6$, and s_7 as follows:

$$\varphi\_1\ (v) = (3-1)!\,(1-1)!/3! * [E(\{1\}) - E\ (\{1\} - \{1\})] + (3-1)!\ (1-1)!/3!\ * [E(\{1,2\}) - E\ (\{1,2\} - \{1\})] + (3-2)!\,(2-1)!/3!\ * [E(\{1,3\}) - E\ (\{1,3\} - \{1\})] + (3-3)!\,(3-1)!/3!\ * [E(\{1,2,3\}) - E\ (\{1,2,3\} - \{1\})]$$

$$= 2/6 * [E(\{1\})] + 1/6 * [E(\{1,2\}) - E\ (\{2\})] + 1/6 * [E(\{1,3\}) - E\ (\{3\})] + 2/6 * [E(\{1,2,3\}) - E\ (\{2,3\})]$$

$$= 2/6 * 9 + 1/6 * (8.7 - 8.49) + 1/6 * (8.9 - 8.92) + 2/6 * (8.8 - 8.7)$$

$$= 3.06$$

- Calculate the weight of model 1:

$$w_1 = \frac{1}{n-1} * \frac{E - \varphi_1(v)}{E} = \frac{1}{2} * \frac{8.8 - 3.06}{8.8} = 0.33$$

- By applying the same steps on model 2 and model 3, the Shapley Values are $\varphi_2(v) = 2.7$ and $\varphi_3(v) = 3.01$, and the models weights are $w_2 = 0.35$ and $w_3 = 0.32$.
- $\varphi_1(v) + \varphi_2(v) + \varphi_3(v) = 8.8 = E$; this indicated that the calculation of the shared error of each model is correct.

2. Nonlinear Combination Methods:

For models outputs $f_1(x), \dots, f_k(x)$ of k individual models, we can combine them as follows to get the nonlinear combination for y(x) [94]:

$$y\,(x) = g\,(f_1(x), f_2(x), \dots, f_k(x)), \qquad (2.22)$$

where g is a nonlinear function that used to combine the models outputs.

Nonlinear combination methods include using nonlinear learning algorithms to combine the base models outputs as a high-level model (meta-model), rank-based combiners such as majority voting, or hierarchical combination techniques [19, 31, 120, 125]. Nonlinear combination using meta-learning is performed by taking the outputs of the base model for the validation data as the input for the training to develop a meta-model and then applying the meta-model on the base models outputs for the test data to predict the result [13]. Meta-learning [72] deals with the problem of computing a global model from distributed data resources [28], aims to improve the quality of selection and the combination of models, and selects the reasonable model according to the relevance of different data sources [4]. Meta-learning performance not only depends on the base models but also on correct model selection for the high-level model [13]. Since meta-learning depends upon base models and a model which selects/weights those models, it is considered a two-level hierarchical model combination.

Another hierarchical model combination is based on a hierarchical strategy that filters out the best-performing models based on a model-encompassing test. First, models are ranked according to a performance measure and then selected for combination such that each model is not encompassed by any competing models. Thus, the hierarchical procedure represents a compromise between model selection and model averaging. Next, larger weights are assigned to models with higher performances. Finally, models with lower accuracy and are encompassed by all other models are discarded, as their weights will be insignificant [19].

2.8 RELATED WORKS

In the following content of this section, research works for distributed and decentralised machine learning methods are presented in section 2.8.1. Section 2.8.2 shows several proposed approaches using model selection and combination strategies to build a combined model.

2.8.1 Distributed and Decentralised Machine Learning

Several machine learning methods have been proposed and focused on improving the prediction performance of regression and classification models. Some of these proposed approaches are under centralised learning or data sharing methods [7, 37, 40, 44, 48, 91, 99, 177, 182]. For example, Kasturi et al. [177] proposed a distributed learning method for distributed data. First, each site sent its distribution parameters with its local model parameters to the server. Then the server regenerated the data and combined it to use it for global model building. Then the global model is sent to the distributed sites. The experiment results showed that the proposed method was similar to the federated and centralised learning approaches. The proposed learning approach in [182] used Spark environment for a real-time distributed learning method to predict heart disease. First, the datasets are streamed from distributed sites, and then the predicted results with the data streams are stored in a database for real-time monitoring. They evaluated their method in terms of execution time and accuracy, which showed improved results. In [37], they proposed a predictive method using five heterogeneous classification models and a weighted voting method to determine the final prediction, where weights are assigned based on model accuracy. Their approach provided a significant improvement compared to other classifiers. The proposed ensemble method in [48] used hierarchical majority voting (HMV) and multi-layer classification based on a combination of seven heterogeneous classifiers. They assessed their strategy on different datasets and showed that their proposed method obtained higher accuracy than other models and ensemble techniques.

Several distributed machine learning methods were proposed to overcome centralised machine learning issues and distributed data resource restrictions, and to develop a generalised model for distributed sites. For example, [23] introduced geo-distributed learning for geographically distributed datasets by implementing a communication-sparse learning algorithm to reduce bandwidth consumption costs and speed up the learning process. The results showed that their method could reduce the cost of communication bandwidth. Other distributed learning approaches have been used to optimise the learning process without using the centralised learning method. For example, [35] proposed a federated learning method and model considering the site models participation to the global model on the server and with an optimisation technique during server combination. They proposed an optimisation technique to measure selected sites importance and accelerate the learning process. Their federated learning approach reduced the algorithm complexity and communication cost. The Bayesian network model using distributed learning approach proposed in [41] showed prediction accuracy improvement in some hospitals and worse in others based on data from 5 hospitals in 3 countries. The hospitals iteratively communicate with a central site and aggregate models statistical results. Also, a federated learning approach using distributed SVM algorithm in [11] is used to predict hospitalisations for cardiac disease based on patients' medical history in Electronic Health Records (EHRs). They achieved similar accuracy and low communication cost compared with a centralised and alternative distributed algorithm.

Other studies applied privacy-preserving methods to overcome data privacy restrictions. For example, [90] developed a privacy-preserving federated learning method using differential privacy and secure multiparty computation and compared it with two different learning algorithms: decision trees and neural networks. Their method outperformed other approaches in accuracy and scalability. Also, the issue of site dropout from the network during FL has been addressed in [166] by developing a dropout-robust and an iterative secure global gradient computation protocol using homomorphic encryption (HE) to preserve the data and model privacy. The experimental results showed the feasibility of their method. Others proposed distributed learning methods in [96, 170, 176, 178 - 181] rely on encryption techniques to preserve the data or model privacy during the learning process.

Gossip learning is a decentralised alternative to federated learning and a competitive approach to the federated learning method [139-144]. For example, [139] is a proposed gossip-learning approach for linear classification models and a continuous combination of the models in the network using the weighted voting method. As a result, every site has at least one model available locally; thus, all the sites can perform a prediction. Their strategy showed the performance and robustness of the proposed approach and provided reasonable protection against uncovering private data.

2.8.2 Models Selection and Combination Approaches

Appropriate model combination methods can improve the predictive capabilities of the prediction models [77]. Many studies showed that combining or ensembling multiple models created unbiased models, achieved high prediction accuracy, and outperformed individual model predictions [13, 15, 16, 18-21, 36, 37, 39, 42, 44, 83, 85, 111, 119]. Distributed learning outcomes from distributed sites can be combined either by combining model prediction results or prediction models [111, 92]. The combination approach should take advantage of local models strengths while ignoring models weaknesses. Model combination success depends on how accurate, diverse, and independent the individual models are and how the models weighting strategy can be determined [13, 83, 108, 125]. Diversity can be achieved by combining distributed and heterogeneous models. Combining diverse models will balance the individual models weaknesses and produce a good combination approach, even with the simplest combination scheme [125, 126]. Dynamic model selection (DES) achieved better results in many studies [18, 57, 64-70].

Model combination using stepwise model selection obtained high accuracy compared with other combination approaches and showed the benefit of this selection method on the combined model performance [106, 107]. In [2], several global models built from different strategies are compared to the centralised learning method and other distributed learning methods. The main idea behind these different strategies is to understand the behavior of a global model constructed with classifiers copied from the local models. It could be seen that the accuracy of distributed learning methods is competitive compared to centralised learning methods. Also, the weighted classifier method got better accuracy when compared to other strategies.

Different model weighting approaches are used for linear model combination to make the final decision. Assigning weights to the models based on their performances is a rational and advantageous approach [76, 77, 81, 86]. It is a simple and optimal combination method for models and is often more efficient and reliable than other complicated techniques [33, 45-47, 86].

From [3] and [25], we concluded that learning outcome quality depends on model combination strategy at the global level of distributed learning, and model combination using a suitable selection strategy is more accurate than that formed from a single model.

An example of model selection and combination strategies is proposed in [6], which used statistical significance tests to select and combine accurate models built from a centralised dataset. They developed three effective voting methods to predict breast cancer: by the highest significance index (EV1), the lowest error rate models (EV2), and by the highest three significance index models (EV3). Their methods are comparable in accuracy to recent combining methods and have a low computational cost. Also, Rathore and Kumar investigated ensemble methods performance using three techniques to build ensemble methods with four combination rules for combining the ensemble outputs. The combination methods achieved better performance than the individual models, leading Rathore and Kumar to conclude that the heterogeneous approach outperformed homogeneous methods developed from a centralised dataset [31].

There are also other approaches been proposed for model selection and combination methods. For example, Li et al. [32] developed a distributed privacy-preserving combination method based on an ensemble method for developing robust prediction models from medical data by calculating the data distribution for the local models and combining the best models based on the distribution differences without disclosing sensitive data. Each site shares its local model with other sites and builds a specific combined model based on its specifications. It detects the data distribution difference and transfers useful knowledge. The proposed method performed better than the original local models constructed on each site data separately.

Similar model combination approaches using the F-measure for model weighting strategy are proposed in [36] and [49]. In [36], the model weight is based on the F-measure value of the training data set to develop a weighted voting method of five heterogeneous models for heart disease prediction. It achieved high diagnosis accuracy compared with individual models. Also, in [49], they used a weighted voting method for predicting heart disease using five heterogeneous models. The weight for each model is computed based on the F-measure of the training dataset, and the result of the ensemble model is the label with the highest weighted vote. They compared their method with individual and ensemble models, and the proposed method outperforms other approaches. In addition, F-measure showed higher accuracy when compared to combined models and randomly selected models in [185].

The proposed model selection strategy in [38] used a decision-making model to identify the prediction models superiority over others by estimating the models accuracy. It used confusion matrix and error measures to develop a performance ranking of classification models. The proposed method considered different information types from various performance metrics and provided firm rankings.

From the proposed research works in [127], [128], and [133], we conclude that the combined models developed by calculating the weight factor of each model using the Shapley Value (SV) method improved the prediction performance. For example, in [133], they compared a linear combination approach with two weighting methods, namely Information Entropy (IE) method and Shapley Value (SV) method. The Shapley Value method proved to be more capable of better combining individual models and improving the predictions for test data. Furthermore, the research showed improvements by using an appropriate linear combination of individual models.

Many studies showed that nonlinear combination methods achieved high prediction accuracy and outperformed individual model predictions and other ensemble methods [31, 71, 73 – 82]. For example, in [81], they proposed different nonlinear combination methods to predict diabetes and involved two steps: model selection and combination. They applied several nonlinear combination methods considering the best combiner method, which integrates the selected models results by utilising the prediction information obtained from the base models. The proposed method outperformed individual learning approaches and other ensemble methods.

Nonlinear model combination based on meta-learning improved prediction performance over individual models and other ensemble models. For example, in [77], they proposed a nonlinear combination method using random forest-based meta-learning for prostate cancer detection. It showed better performance than ensemble methods and other single learning algorithms. Also, in [169], they proposed heterogeneous classification models ensemble with fuzzy rule-based meta-model. Their strategy is competitive compared to several ensemble methods and individual learning algorithms. The proposed nonlinear combination model in [125] using diverse classification models predictions for meta-learning showed improved prediction performance over individual classification models. Also, [120] used decision tree forest (DTF) for meta-model learning for their proposed heterogeneous ensemble method with two linear and nonlinear combination methods. The nonlinear combination method outperformed the linear combination methods, and the combined methods outperformed other single models.

A similar meta-learning method using decision trees for the nonlinear combination was found in [31]. It is a nonlinear combination approach using decision tree and gradient boosting regression to predict the number of faults in a given software system. Experiment results showed that the proposed approach improved prediction accuracy over individual prediction models. Also, heterogeneous ensemble methods based on the nonlinear combination method outperformed homogeneous ensemble methods.

An artificial neural network (ANN) is used for nonlinear model combination in [136]. The experiment results showed that the proposed approach is superior to the common practice of linear combination and generally performed better than individual models.

Nonlinear model combination based on hierarchical model combination showed improved results over individual models. For example, in [138], they proposed a hierarchical model combination approach to tourism forecasting problem that combines linear and nonlinear combination methods predictions using the simple average approach. Their method outperformed individual learning models.

2.9 CRITICAL ANALYSIS

Although the existing research works have made considerable contributions to distributed machine learning research, there still exist some challenges and limitations to these learning methods. We summarise these limitations as follows:

1. Distributed machine learning methods in some studies mentioned above that involve sharing or combining distributed datasets to create a central data resource incur some serious issues and concerns related to communication, privacy issues, and data transformation costs. In addition, it is not possible when the distributed data resources are private and not exchangeable. Therefore, it is necessary to find alternative learning strategies to overcome the centralised machine learning issues. Such a strategy could contribute to preserving data privacy and improving learning performance.

2. Some studies dealing with distributed learning issues are centralised modelling, exchanging lots of information and intermediate results to get a general model or using complex methods to preserve data privacy. For example, Federated Learning (FL) is designed for distributed sites to preserve data privacy by avoiding sending the data to a central location. However, the learning process involves many communication rounds between the sites and the server to build a global model. Thus, there is a communication cost in Federated Learning. Furthermore, the server coordinates the learning process and performs many model parameters update iterations between the server and sites. Since the server takes control during the learning process, federated learning faces risks of single points of failure or attacks. Hence, it is highly desired to focus on decentralised machine learning to remove central location assumption for control and coordinate the learning process. Such an approach would allow us to overcome iterative learning overhead and central server issues.

3. Privacy-preserving federated learning techniques are proposed to preserve data privacy and models updates information. These methods use Differential Privacy (DP), which preserves the privacy of model output or intermediate statistical results, cryptographic approaches to protect the data or results, or

Secure Multiparty Computation protocol (SMC). Despite the effectiveness of these techniques, SMC causes huge computation time and requires extensive communication between the associated sites, and it has a considerable communication cost as its processes need to communicate encrypted data with each other across the network, and the communicating sites must remain online during joint computation [14, 58]. In addition, the learning speed on the encrypted training data is decreased due to the computation overheads, and it is not practical to be used in real applications [61, 88, 96]. Differential privacy involves adding noise (data alteration) and may not be suitable for applications where high-quality models are needed as it reduces the quality of data and hence its utility [14, 62, 88]. One main limitation is that existing federated learning approaches overall are still an iterative learning process and require iteratively exchanging the gradient information. Therefore, it is still computationally intensive and requires a lot of information sharing, which still inherits some privacy-preserving risk. Due to the above communication and computational issues, further efforts should be done to overcome these issues without using complex approaches for distributed learning and carefully considering data and model privacy during the learning process.

4. Model combination strategy is a promising approach to build an optimal combined model to improve the prediction performance and overcome individual model limitations. Exchanging models between the distributed sites instead of the raw data is a solution to overcome the data privacy and restriction issues [4, 50, 100. 108]. The challenge here is to develop a decentralised version of model selection and combination strategies to build an optimal combined model for distributed sites with less information sharing and communication and computation overheads. Thus, paying attention to model selection and combination strategies is desired to improve the performance of the combined model. This triggers our motivation to propose and develop a robust and optimal combined model using a decentralised version of model selection and combination strategies that could improve prediction performance for all distributed sites and preserve data privacy.

2.10 SUMMARY

This chapter presented an overview of machine learning techniques, distributed machine learning approaches, and followed by the challenges in the distributed learning area. Then, we showed privacy-preserving machine learning approaches and model combination approaches, and several research works. We have also analysed the research limitations and challenges of the research works and the possible approaches to tackle these challenges.

Chapter 3

Linear Model Combination Approach

3.1 CHAPTER OVERVIEW

This chapter presents our proposed decentralised learning approach using linear model combination method for distributed data sources to develop a global and combined model. Section 3.2 views our contribution and aims to develop a global combined model using decentralised learning and linear combination approaches. The proposed method for classification with its experiment results and discussion are shown in section 3.3 and regression algorithms in section 3.4. Finally, section 3.5 presents the chapter summary.

3.2 INTRODUCTION

We develop a decentralised machine learning approach to distributed, private, and un-exchangeable data resources without exchanging lots of intermediate information, using a centralised machine learning method, or using a central site for iterative communication or computation. This approach is proposed to overcome data restriction issues, preserve data privacy for each site, and avoid the iterative learning process overhead and server issues. We develop a global combined model derived from local learning outcomes from the distributed sites with minimal communication between the sites. Besides, we develop a local combined model in each site by utilising learning outcomes from other sites data resources and its local data to improve its local prediction performance. We use a linear combination approach of heterogeneous models to utilise and combine the selected models from distributed sites. We focus on models evaluation, selection, and combination strategies instead of complicated methods without revealing any information about data and only exchange the trained models and evaluation results with fewer computation and communication rounds than federated learning.

3.3 PROPOSED METHOD FOR CLASSIFICATION ALGORITHMS

The proposed method addresses individual model limitations by utilising distributed data resources to develop combined prediction models at global and local levels without data transformation between sites to preserve the privacy of local data resources. For this purpose, the related model names and definitions used in our methodology are first introduced in Table 3.1.

Table 3.1. Models Names and Descriptions

Model	Notation	Meaning
Local Model	M_{ij}	The local model that developed in site i using j classification algorithm
Received Model	$M_{i'j}$	A model that received from other sites i′
Best local model accuracy	M_{ij*}	The local model in site i that has the best accuracy developed by j ∗ classification algorithm
Best local model F-measure	M_{ij**}	The local model in site i that has the best F-measure developed by j ∗∗ classification algorithm
Best Global Average Model Accuracy	M^{G}_{ij*}	The best global average accuracy in site i after evaluating the models M_{ij} in all sites and calculating the average accuracy
Best Global Average Model F-measure	M^{G}_{ij**}	The best global average F-measure in site i after evaluating the models M_{ij} in all sites and calculating the average F-measure
Global Combined Model (1)	M^{G*}	The final global combined model that combines the best global average model accuracy M^{G}_{ij*} from each site
Global Combined Model (2)	M^{G**}	The final global combined model that combines the best global average model F-measure M^{G}_{ij**} from each site
Best Model Accuracy	$M_{i'j*}$	The selected model from other sites i′ which is better than or equal to the best local model accuracy M_{ij*}
Best Model F-measure	$M_{i'j**}$	The selected model from other sites i′ which is better than or equal to the best local model F-measure M_{ij**}

List of the best accuracy models	M_{Acc}	List of the best local model accuracy M_{ij*} and the selected models of the best accuracy from other sites $M_{i'j*}$ that will linearly be combined to build the local combined model.
List of the best F-measure models	M_F	List of the best local model F-measure $M_{ij**}$and the selected models of the best F-measure from other sites $M_{i'j**}$ that will linearly be combined to build the local combined model.
Local Combined Model (1)	M_i^{L*}	The final local combined model in site i that combines the best local model accuracy of the site i with the best models accuracy from other sites
Local Combined Model (2)	M_i^{L**}	The final local combined model in site i that combine the best local model F-measure of the site i with the best models F-measure from other sites

3.3.1 Global-level Modelling Approach

We aim to build a global combined model at the central server by combining the best global average model from each site. Figure 3.1 shows that each site builds local and heterogeneous models on their local data using different classification algorithms. We develop heterogeneous models because a single model may perform well in one dataset and not fit well in others. Therefore, we build diverse models to find a suitable model for all distributed sites. Second, each site takes advantage of its local models strength while ignoring the models weaknesses by selecting the best local model using 10-fold cross-validation technique. We selected 10-fold cross-validation based on our experiments using 5-fold and 10-fold cross-validation, and 10-fold cross-validation showed better results than 5-fold cross-validation on different classification and regression datasets (Appendix A). Next, each site shares its local models with other sites for general evaluation to find good models as candidates for future selection and combination to build a well-generalised/global model for all sites. Then, when a site receives other sites local models, it first calculates the prediction accuracy of these models based on its local data and then sends back the models with its evaluation results to the sites, and with the local data size used for evaluation. Each site should not get more information about the received model during evaluation than the prediction accuracy of its local data, which will preserve the models' privacy. Then, each site will receive its local models evaluation results from other sites, calculate the average accuracy of its local models and send the best global average model with its average accuracy to the server. Finally, the server combines the models using a linear combination method by weighting the models based on its average accuracy to develop the global combined model. This approach develops a global model without centralising the data for the learning process. This will overcome the risks of a centralised architecture, such as computational load and single point of failure. In addition, the proposed method does not expose the data resource and hence preserves data privacy and uses minimum exchanged information between the distributed sites and the server to mitigate the computation and communication overhead between the server and sites.

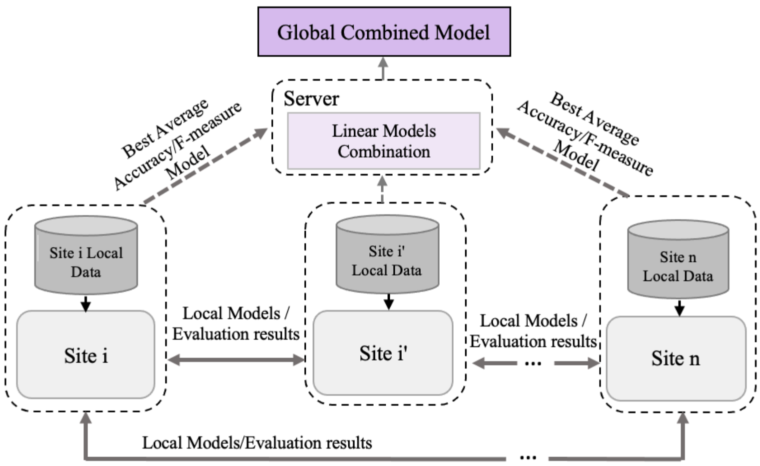


Figure 3.1. The Proposed Method for Global-Level Modelling using Linear Combination Approach

The following steps implement the above idea:

1) As illustrated in Figure 3.2, for each site S_i , where $i = 1,2, \dots, n$:
   - Apply different j learning algorithms, where $j = 1,2, \dots, m$ to build local models M_{ij}.
   - Use 10-fold cross-validation results to evaluate the local models M_{ij} based on the local data in site S_i and calculates the accuracy $Acc\left(M_{ij}\right)$ and F-measure $F\left(M_{ij}\right)$ using confusion matrix.

$$Acc\left(M_{ij}\right) = TP + TN/(TP + TN + FP + FN) \qquad (3.1)$$

$$F\left(M_{ij}\right) = 2TP/(2TP + FP + FN) \qquad (3.2)$$

, where TP is true positive, TN is true negative, FP is false positive, and FN is false negative examples.

- Select the best local accuracy model M_{ij*} and the best local F-measure model M_{ij**}.

2) Each site sends its local models M_{ij} to other sites for global evaluation to see which model got the best performance globally and select the best average models as follows:
   - Each site will receive models from other sites $M_{i' j}$, then start to evaluate these models over its local dataset and calculate the accuracy $Acc(M_{i' j})$ and $F\left(M_{i' j}\right), (i' = 1, .., i-1, i+1, \dots, n)$.
   - Send the evaluated models back to the sites with the evaluation results.
   - Each site S_i:
     a) Receive the evaluation results $Acc\ (M_{ij})$ and $F\ (M_{ij})$ of its local models M_{ij} from other sites with the number of data samples that used for evaluation.
     b) Calculate the global average accuracy and the global average F-measure for each local model.

$$Acc\ (M_{ij}) = \sum_{k=1}^{n} \frac{D_k}{D} * Acc\ (M_{ij}) \text{ in } S_k \qquad (3.3)$$

$$F\ (M_{ij}) = \sum_{k=1}^{n} \frac{D_k}{D} * F\ (M_{ij}) \text{ in } S_k \qquad (3.4)$$

, where k is sites number, D_k is the number of samples of site k, and D is all sites' data samples number

     c) Select the best global average accuracy M_{ij*}^{G} and the best global average F-measure M_{ij**}^{G}.
     d) Send the selected best global average models with its evaluation results $Acc\ (M_{ij*}^{G})$ and $F\ (M_{ij**}^{G})$ to the server.

3) As shown in Figure 3.3, the server combines the received models by linear combination method to develop global combined models. We develop two global combined models M^{G*} and M^{G**}. M^{G*} is the global combined model that combines the best global average model accuracy from each site, and M^{G**} is the global combined model that combines the best global

average model F-measure from each site. The linear combination method implementation is as follows:

a) The server receives two models from each site, the best global average accuracy M^{G}_{ij*} with its global average accuracy Acc $\left(M^{G}_{ij*}\right)$, and the best global average F-measure M^{G}_{ij**} with its F-measure F (M^{G}_{ij**}).

b) The server combines the best global average accuracy models by weighting the models based on its global average accuracy $Acc\left(M^{G}_{ij*}\right)$, and combines the best global average F-measure models by weighting the models based on its global average F-measure $F\left(M^{G}_{ij**}\right)$. We calculate the weighted average to get an unbiased model weight obtained by weighting each model based on its global accuracy and F-measure [30, 33, 86]. The most accurate model will get higher weight, and the less accurate model will get low weight. Models' weights are constrained such that their sum is equal to one.

$$w_{M^{G}_{ij*}} = \frac{Acc\ (M^{G}_{ij*})}{\sum_{i=1}^{n} Acc\ (M^{G}_{ij*})} \quad , i=1, 2, \ldots, n \qquad (3.5)$$

where, $0 <= w_{M^{G}_{ij*}} <= 1$ and $\sum_{i=1}^{n} w_{M^{G}_{ij*}} = 1$

$$w_{M^{G}_{ij**}} = \frac{F\ (M^{G}_{ij**})}{\sum_{i=1}^{n} F\ (M^{G}_{ij**})} \quad , i=1, 2, \ldots, n \qquad (3.6)$$

where, $0 <= w_{M^{G}_{ij**}} <= 1$ and $\sum_{i=1}^{n} w_{M^{G}_{ij**}} = 1$

c) The server linearly combines the models to develop the global models by using the weighted voting method. We use the weighted voting method as the linear combination strategy because it is the most popular strategy and an advantageous approach and has a significant impact on the prediction results in the combination [33, 34, 84, 86].

$$M^{G*}\ (x) = \max \sum_{i=1}^{n} w_{M^{G}_{ij*}}\ M^{G}_{ij*}\ (x) \qquad (3.7)$$

$$M^{G**}\ (x) = \max \sum_{i=1}^{n} w_{M^{G}_{ij**}}\ M^{G}_{ij**}\ (x) \qquad (3.8)$$

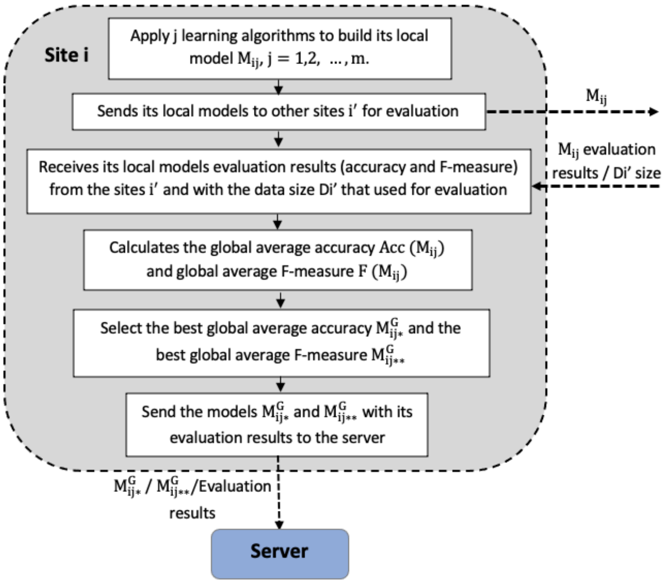


Figure 3.2. Global-Level Modelling Method

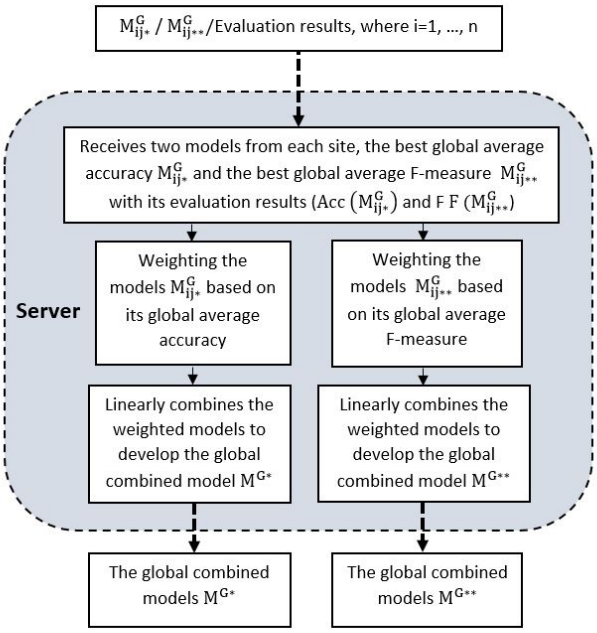


Figure 3.3. Linear Model Combination Approach

3.3.2 Local-level Modelling Approach

In the local-level modelling, the basic idea is, for each data site, to find the best local combined model by utilising the local data resource and the local prediction models from the other sites. The main advantages of such an idea are that, firstly, only the local prediction models from the other sites are used, and therefore we save the cost of data transformation from one site to another. As we know, data transformation is time-consuming and costly if the datasets are large. Therefore, such an approach greatly improves the computation effectiveness and efficiency; secondly, there is no data sharing or transformation, whereas the only information exchanged between different sites are local models and the evaluation results and data size. Such an approach does not disclose the data resource and therefore preserves data privacy. Each site tries to find the best local combined model by utilising the best local models from the other sites. According to the results of the received models from other sites that evaluated based on sites local data, each site will compare these results with its own best local model result and select the best models. Then, each site combines the selected models using linear combination methods to build the local combined model. The proposed approach for local-level modelling is illustrated in Figure3.4.

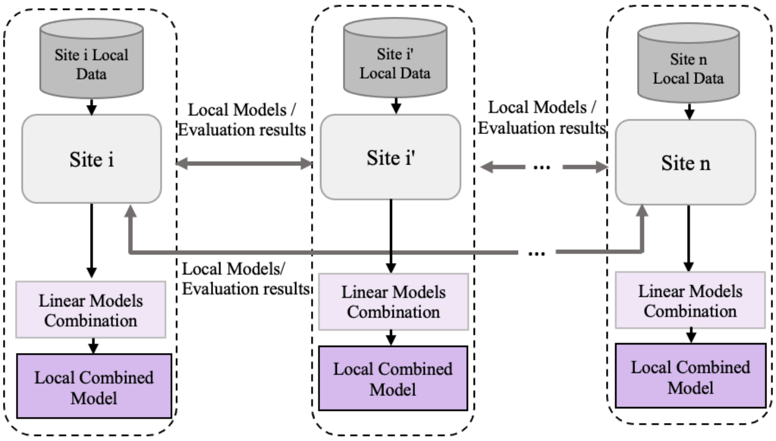


Figure 3.4. The Proposed Method For Local-Level Modelling Using Linear Combination

The proposed local-level modelling is illustrated in Figure 3.5 and implemented by the following steps:

1) When a site i receives models from other sites $M_{i'j}$, and evaluate these models over its local dataset and calculate the model accuracy $Acc(M_{i'j})$ and F-measure $F(M_{i'j})$:
   a) Compare the model accuracy Acc $(M_{i'j})$ and F-measure F $(M_{i'j})$ of the received models with its best local model accuracy Acc (M_{ij*}) and best local model F-measure F (M_{ij**}).
   b) Select the best model accuracy $M_{i'j*}$ and the best model F-measure $M_{i'j**}$ from each site as follows:

- **Method L1:**
  - Select the best model accuracy $M_{i'j*}$ from site i' if:

$$Acc\ (M_{i'j}) \geq Acc\ (M_{ij*})$$

  - Select the best model F-measure $M_{i'j**}$ from site i' if:

$$F\ (M_{i'j}) \geq F\ (M_{ij**})$$

  The aim of this method is to utilise the best models learned from other data resources to build an accurate local combined model.

  Alternatively:

- **Method L2:** Select the best model from each site, even if it is not better than the best local model. We proposed this method if the best local model performance is better than the received models, but there is not much difference between the models results. We apply this selection method for the best model accuracy and F-measure.

2) In each site i, we apply linear combination method to develop two local combined models M_i^{L*} and M_i^{L**}. M_i^{L*} is the final local combined model that combines the best local model accuracy M_{ij*} and the selected best model accuracy from other sites $M_{i'j*}$. M_i^{L**} is the final local combined model that combines the best local model F-measure M_{ij**} and the selected best model F-measure from other sites $M_{i'j**}$. We aim to reduce model biases and errors in individual models when combining the models rather than selecting an individual model. Each site calculates and assigns weights for the best local model and the selected models from other sites to perform the linear combination as follows:
   a) Site i has a list of the best models accuracy M_{Acc}, where M_{Acc} is the best local model accuracy M_{ij*} and the selected models of the best accuracy from other sites $M_{i'j*}$, M_{Acc}=$\{M_{ij*}\ ,\ M_{i'j*}, \ldots\ ,\ M_{nj*}\}$, where i = 1, 2, …, n ,and its prediction accuracy results. Also, the site i has a list of the best models F-measure M_F, where M_F is the best local model F-measure M_{ij**} and the selected models of the best F-measure from other sites $M_{i'j**}$, $M_F = \{\ M_{ij**}, M_{i'j**}, \ldots\ ,\ M_{nj**}\}$ and its F-measure results.
   b) For each model in M_{Acc}, calculate model weight based on its average accuracy and average F-measure:

$$w_{M_{ij*}} = \frac{\mathrm{Acc}\ (M_{ij*})}{\sum_{i=1}^{n} \mathrm{Acc}\ (M_{ij*})}, \quad (3.9)$$

where, i=1, 2, ..., n, $0 <= w_{M_{ij*}} <= 1$ and $\sum_{i=1}^{n} w_{M_{ij*}} = 1$

$$w_{M_{ij**}} = \frac{F\ (M_{ij**})}{\sum_{i=1}^{n} F\ (M_{ij**})}, \quad (3.10)$$

where, i=1, 2, ..., n, $0 <= w_{M_{ij**}} <= 1$ and $\sum_{i=1}^{n} w_{M_{ij**}} = 1$

c) Combine the models to develop the final local combined model of the best models' accuracy M_i^{L*} , and the final local combined model of the best models F-measure M_i^{L**} by using the weighted voting method to predict x.

$$M_i^{L*}(x) = \max \sum_{i=1}^{n} w_{M_{ij*}} M_{ij*}\ (x) \quad (3.11)$$

$$M_i^{L**}(x) = \max \sum_{i=1}^{n} w_{M_{ij**}} M_{ij**}\ (x) \quad (3.12)$$

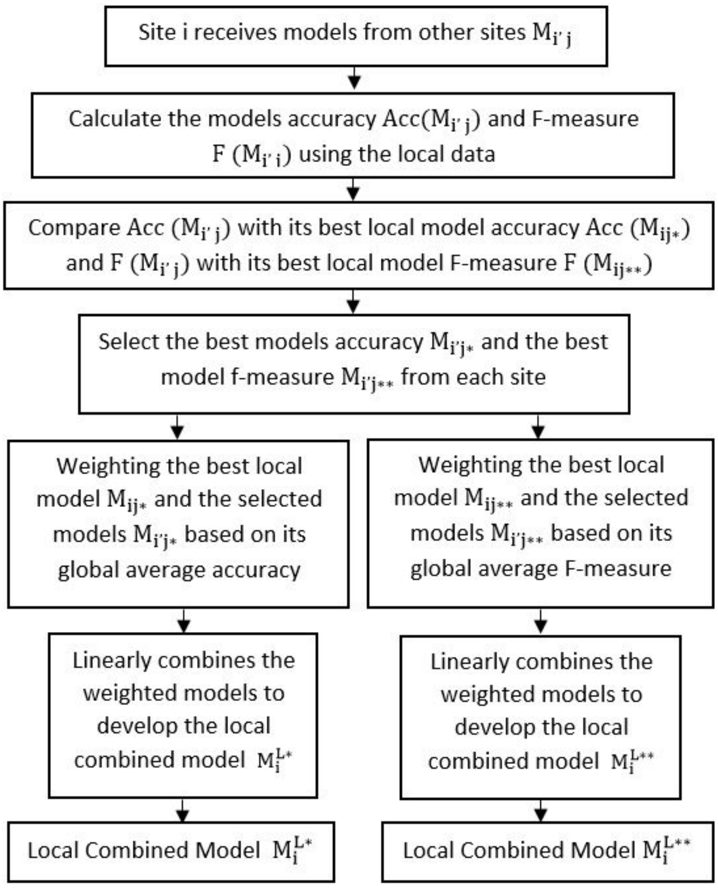


Figure 3.5. Local-Level Modelling Method

3.3.3 Experimental Study

The experiments are conducted to evaluate the performance of our proposed methods for the global and local combined model.

I. Datasets

We used eight databases: blood transfusion, liver disease, diabetes, heart disease, lower back pain (spine disease), breast cancer Wisconsin (Diagnostic), breast cancer Wisconsin (Original) [29], and cardiovascular diseases [27]. Before performing the experiments, we preprocessed the datasets to a suitable data format. The datasets are with a binary target value, 0 and 1. All datasets variables except blood transfusion dataset are patient information and medical diagnosis or measurement values. Table 3.2 describes the datasets that used to train the models and evaluate the combined models. The datasets size except for cardiovascular disease are small, we could not find large and useful datasets publicly available to apply our proposed methods, and we will investigate this issue in future research.

Table 3.2. Datasets Descriptions.

Datasets	Data Size	No. of attributes
Diabetes	768 Positive: 268 (34.9%) Negative 500 (65.1%)	9
Liver disease	579 Positive: 414 (71.5%) Negative 165 (28.5%)	11
Breast Cancer Wisconsin (Diagnostic)	569 Positive: 212 (37.3%) Negative 357 (62.7%)	31
Blood transfusion	748 Positive: 178 (23.8%) Negative 570 (76.2%)	5
Heart disease	303 Positive: 165 (54.5%) Negative 138 (45.5%)	14

Lower back pain (Spine disease)	310 Positive: 210 (67.8%) Negative 100 (32.2%)	13
Breast Cancer Wisconsin (Original)	683 Positive: 239 (35%) Negative 444 (65%)	10
Cardiovascular disease	68783 Positive: 34041 (49.5%) Negative 34742 (50.5%)	12

II. Simulating Distributed Data

We applied the proposed methods using two dataset partitioning strategies to mimic a real-world scenario for distributed datasets (independent datasets) for distributed sites. The strategies are (1) random data partitioning approach and (2) non-random data partitioning approach. For non-random data partitioning approach, we partitioned the data based on patients' age; we assumed each partition is from a hospital whose patients are in a specific age range to simulate the distributed data. The data are liver disease, diabetes, and heart disease datasets. We assume that the distributed sites are homogeneous, and the data are independent and identically distributed in this study. Also, the sites agreed on the learning algorithms that will be used for learning models and have the same data attributes and target. Therefore, we divided each dataset into different parts as distributed sites site1, site2, and site3. In each site, the dataset is divided into local and validation data. The local data partition is used to develop and evaluate the local models and assess the received models. The validation data is used to evaluate the final global and local combined model. We assume all the sites will strictly follow the proposed approach, and sites collusion will not happen between the sites, which means the sites only share the local models and evaluation results with others during the learning process. Besides, we do not consider the outside attack cases in the distributed environment. Furthermore, the data for each site is private and un-exchangeable and will not be directly shared or exposed to other sites.

III. Models Building and Evaluation

Well-known classification algorithms are used to build the local prediction models in each site: K-Nearest Neighbor, Logistic Regression, Neural Network, Support Vector Machine, Random Forests, Decision tree, and Naïve Bayes. We built diverse binary-class prediction models to predict blood transfusion, liver disease, diabetes, heart disease, lower back pain (spine disease), Breast Cancer Wisconsin (Diagnostic), Breast Cancer Wisconsin (Original), and cardiovascular diseases datasets (positive or negative) depending on the patients' diagnosis. In each site, the local models are trained on the local training dataset from its local dataset. We used 10-fold cross-validation to prevent selection-biased results from being drawn from a single partition of the local data into training and test sets and maintained the same class distribution in each subset. K-fold cross-validation is a form of bootstrapping by random sampling with replacement. We used accuracy and F-measure metrics in model evaluation, selection, and weighting strategies. We used the F-measure metric for model evaluation because some datasets, such as liver disease, blood transfusion, and lower back pain (Spine disease), are imbalanced. It is the most commonly used performance metric of machine learning, mainly when the data set is imbalanced. F-measure is the harmonic mean of precision and recall and a less biased metric than accuracy. It equally weights both false positives and false negatives and provides more weight to correctly classified samples in the minority class [155, 184].

IV. Combined Model Evaluation Methods

1) Global Combined Model Evaluation:

a) **Testing accuracy:** in each site, we evaluated the global combined models $\mathrm{M}^{\mathrm{G}*}$ and $\mathrm{M}^{\mathrm{G}**}$ based on the site local validation data instead of the server because the sites will not share their validation data with the server and only send the global combined model results and number of validation data samples to the server. Then, the server calculated the global average accuracy of the global combined models.

b) **Training accuracy:** each site evaluated the final global combined model based on its local data that used to train the local models. Then, the sites send the evaluation results to the server with the number of local data samples to calculate the average training accuracy.

2) **Local Combined Model Evaluation:** we calculated the final local combined models accuracy based on the local validation data and compared it with other combination methods, average accuracy, and majority voting methods. In addition, we compared our proposed methods with the best local model accuracy M_{ij*} and the best local model F-measure M_{ij**}.

In addition, we evaluated the global combined model and compared the model with other combination methods, average accuracy, and majority voting [163]. Besides, we compared it with a technique that if each site sends the best local model accuracy and F-measure to the server instead of sending the best global average model (**Best Local Models Combination**). In addition, the proposed method is compared with the centralised learning approach and **Single Best Model**. Single Best Model is the model of the best prediction result from the selected models that formed the global combined model. Furthermore, we evaluated the local combined model. We compared the model with the dynamic ensemble selection (DES) methods: KNORA-U, KNORA-E, DES-P, META-DES, KNOP, and DES-KNN dynamic ensemble methods. We applied DES methods in each site on the received models from other sites using the local data and then evaluating the selected models. DES works by estimating the competence level of each model from a pool of models during the classification or testing phase. The most competent models are selected for each new sample by different competency measures [52, 53].

V. Experiment Results and Analysis

1) Random Data Partitioning Approach:

As shown in Table 3.3, for each site, we split the datasets into two main parts, local and validation data. The local data is used to build the local models and evaluate the received models from other sites. The validation data is used to evaluate the combined models of the global and local level modelling methods. Datasets distributions are shown in Appendix A.

Table 3.3. Datasets Partitions

Datasets	Site 1		Site 2		Site 3	
	Local data	Validation data	Local data	Validation data	Local data	Validation data
Diabetes	300	40	150	35	218	25
Liver disease	248	25	148	40	103	15
Breast Cancer Wisconsin (Diagnostic)	200	20	150	30	150	19
Blood transfusion	250	35	150	30	248	35
Heart disease	100	15	93	20	60	15
Lower back pain (Spine disease)	100	20	50	14	110	16
Breast Cancer Wisconsin (Original)	156	60	240	70	107	50
Cardiovascular disease	13800	6900	20066	10034	11988	5995

I. Global-level Modelling Results:

The global-level modelling approach detailed results are illustrated in Appendix B. Table 3.4 shows evaluation results for blood transfusion dataset and compares the combined models with the other model combination methods, centralised learning approach, and the single best model. Single Best Model is the model of the best prediction result from the selected models that formed the global combined model. For example, the global combined model combines three models, LR model from site 1, SVM model from site 2, and LR model from site 3. When we evaluated these models individually, LR model from site 3 was the best model performance. The Single Best Model is slightly better than the other methods, and our proposed method got close results compared to the centralised learning approach.

Table 3.4. The Global Combined Model and Centralised Learning Approach Evaluation for Blood Transfusion Dataset

Models	Selection Metric	Combination Method	Accuracy
Global-level modelling	Accuracy	Weighted Voting	59%
		Average Accuracy	60%
		Majority Voting	59%
	F-measure	Weighted Voting	59%
		Majority Voting	59%
	Single Best Model (LR model – S3)		**62%**
Best Local Models Combination	Accuracy	Weighted Voting	51%
		Average Accuracy	60%
		Majority Voting	48%
	F-measure	Weighted Voting	61%
		Majority Voting	61%
	Single Best Model (LR model – S3)		61%
Centralised Learning Approach (LR model)			60%

Table 3.5 illustrates the global-level modelling results for breast cancer Wisconsin (Diagnostic) dataset. The results of the single best models are slightly better than the other methods, and our proposed method is not far from the centralised learning method result.

Table 3.5. Global Combined Model and Centralised Learning Approach Evaluation for Breast Cancer Wisconsin (Diagnostic) Dataset

Models	Selection Metric	Combination Method	Accuracy
Global-level modelling	Accuracy	Weighted Voting	94%
		Average Accuracy	93%
		Majority Voting	94%
	F-measure	Weighted Voting	93%
		Majority Voting	93%
	Single Best Model (LR model – S1)		**97%**
Best Local Models Combination	Accuracy	Weighted Voting	91%
		Average Accuracy	93%
		Majority Voting	91%
	F-measure	Weighted Voting	94%
		Majority Voting	94%
	Single Best Model (RF model – S2)		95%
Centralised Learning Approach (RF model)			96%

For diabetes dataset, Table 3.6 shows that our proposed approach got similar and close results compared to the centralised learning approach.

Table 3.6. Global Combined Model and Centralised Learning Approach Evaluation for Diabetes Dataset

Models	Selection Metric	Combination Method	Accuracy
Global-level modelling	Accuracy	Weighted Voting	**78%**
		Average Accuracy	75%
		Majority Voting	**78%**
	F-measure	Weighted Voting	77%
		Majority Voting	76%
	Single Best Model (RF model – S2)		**78%**
Best Local Models Combination	Accuracy	Weighted Voting	76%
		Average Accuracy	76%
		Majority Voting	77%
	F-measure	Weighted Voting	**78%**
		Majority Voting	**78%**
	Single Best Model (RF model – S2)		**78%**
Centralised Learning Approach (LR model)			**78%**

In Tables 3.7 and 3.9, the centralised learning approach shows better performance than our proposed method for liver and spine disease datasets, but it is not far from our proposed method result in Table 3.8 for heart disease dataset.

Table 3.7. Global Combined Model and Centralised Learning Approach Evaluation for Liver Disease Dataset

Models	Selection Metric	Combination Method	Accuracy
Global-level modelling	Accuracy	Weighted Voting	65%
		Average Accuracy	66%
		Majority Voting	65%
	F-measure	Weighted Voting	65%
		Majority Voting	65%
	Single Best Model (SVM model - S1)		68%
Best Local Models Combination	Accuracy	Weighted Voting	66%
		Average Accuracy	69%
		Majority Voting	67%
	F-measure	Weighted Voting	65%
		Majority Voting	65%
	Single Best Model (LR model– S3)		75%
Centralised Learning Approach (NB model)			**78%**

Table 3.8. Global Combined Model and Centralised Learning Approach Evaluation for Heart Disease Dataset

Models	Selection Metric	Combination Method	Accuracy
Global-level modelling	Accuracy	Weighted Voting	86%
		Average Accuracy	84%
		Majority Voting	86%
	F-measure	Weighted Voting	86%
		Majority Voting	86%
	Single Best Model (RF model - S3)		86%

Best Local Models Combination	Accuracy	Weighted Voting	88%
		Average Accuracy	87%
		Majority Voting	90%
	F-measure	Weighted Voting	81%
		Majority Voting	80%
	Single Best Model (RF model-S2)		**97%**
Centralised Learning Approach (LR model – NB model)			92%

Table 3.9. Global Combined Model and Centralised Learning Approach Evaluation for Spine Disease Dataset

Models	Selection Metric	Combination Method	Accuracy
Global-level modelling	Accuracy	Weighted Voting	59%
		Average Accuracy	60%
		Majority Voting	59%
	F-measure	Weighted Voting	58%
		Majority Voting	60%
	Single Best Model (RF model- S3)		**62%**
Best Local Models Combination	Accuracy	Weighted Voting	58%
		Average Accuracy	60%
		Majority Voting	60%
	F-measure	Weighted Voting	54%
		Majority Voting	58%
	Single Best Model (RF model -S3)		65%
Centralised Learning Approach (RF model – KNN model)			**66%**

For breast cancer Wisconsin (Original) and cardiovascular diseases datasets, Tables 3.10 and 3.11 show that the results of the proposed method and the centralised learning method got equal performance.

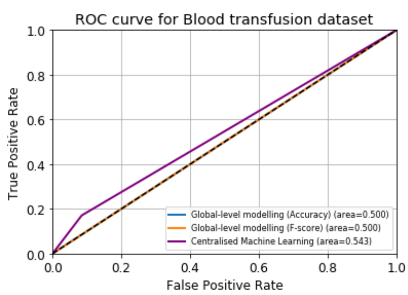
Table 3.10. Global Combined Model and Centralised Learning Approach Evaluation for Breast Cancer Wisconsin (Original) Dataset

Models	Selection Metric	Combination Method	Accuracy
Global-level modelling	Accuracy	Weighted Voting	98%
		Average Accuracy	98%
		Majority Voting	98%
	F-measure	Weighted Voting	98%
		Majority Voting	98%
	Single Best Model (RF model-S1)		98%
Best Local Models Combination	Accuracy	Weighted Voting	98%
		Average Accuracy	98%
		Majority Voting	**99%**
	F-measure	Weighted Voting	**99%**
		Majority Voting	**99%**
	Single Best Model (KNN model – S3)		**99%**
Centralised Learning Approach (LR model – RF model – NN model)			98%

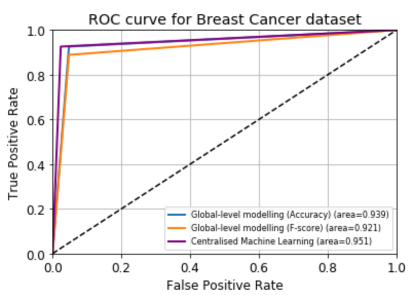
Table 3.11. Global Combined Model and Centralised Learning Approach Evaluation for Cardiovascular Disease Dataset

Models	Selection Metric	Combination Method	Accuracy
Global-level modelling	Accuracy	Weighted Voting	73%
		Average Accuracy	73%
		Majority Voting	73%
	F-measure	Weighted Voting	73%
		Majority Voting	73%
	Single Best Model (LR-S1)		73%
Best Local Models Combination	Accuracy	Weighted Voting	73%
		Average Accuracy	73%
		Majority Voting	73%
	F-measure	Weighted Voting	73%
		Majority Voting	73%
	Single Best Model (LR-S1)		73%
Centralised Learning Approach (LR model)			73%

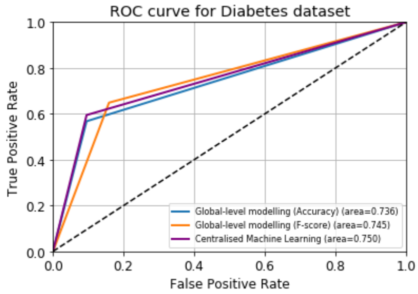
Figure 3.6 shows the ROC Curve and AUC of the global-level modelling developed by the two-selection metrics and compared with the centralised machine learning. It shows that the proposed method and centralised learning approach got comparable or close results. In Figures (a) and (e), the AUC is equal to 0.5 for bold transfusion and liver disease datasets, respectively, which means the global combined model is unable to predict the classes correctly. It ranks a randomly selected positive example higher than a negative example 50% of the time. Thus, the model is not working, as its predictive ability is not better than random guessing. This issue may occur because the dataset used for testing is small and insufficient to represent the overall global model performance, and we will investigate this problem in future.



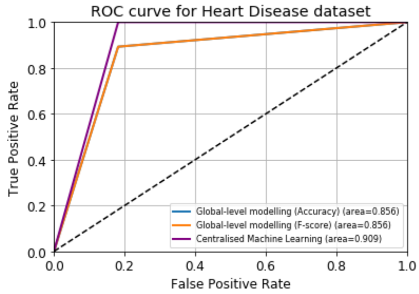
(a) Blood Transfusion



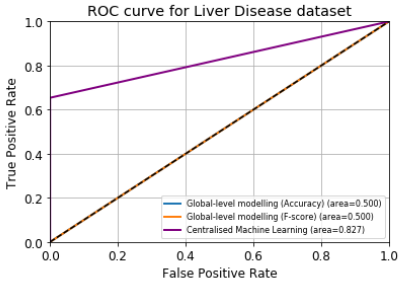
(b) Breast Cancer Wisconsin (Diagnostic



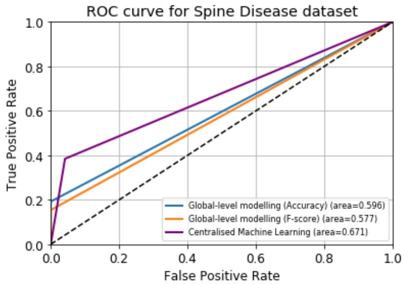
(c) Diabetes



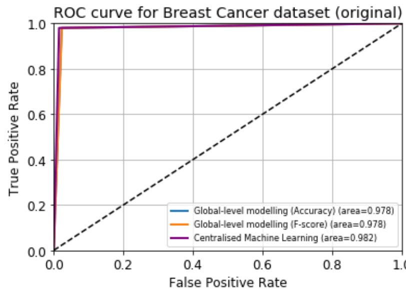
(d) Heart Disease



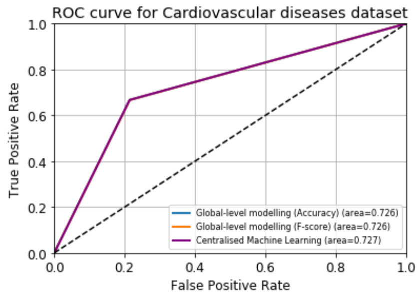
(e) Liver disease



(f) Spine disease



(g) Breast Cancer Wisconsin (original)



(h) Cardiovascular disease

Figure 3.6. ROC Curve Analysis

Table 3.12 presents the training and testing accuracy for the proposed method and centralised learning approach. In diabetes, heart disease, liver disease, and Breast Cancer Wisconsin (Original) datasets, the training accuracy for the centralised machine learning approach is lower than testing accuracy. This may happen because the testing data is small or similar to the training data after splitting the data into training and testing datasets, and the model unexpectedly performed better on the test data than the training data. Furthermore, the centralised machine learning approach result is from a single model. Generally, a model with fewer parameters is less likely to overfit than a combined model. However, the difference between the testing and training accuracy is insignificant.

Table 3.12. The Training and Testing Accuracy for the Proposed Method and Centralised Learning Approach

Dataset	Models	Selection Metric	Testing Accuracy	Training accuracy
Blood Transfusion	Global-level modelling	Accuracy	59%	80%
		F-measure	59%	80%
	Centralised learning approach		60%	79%
Breast Cancer Wisconsin (Diagnostic)	Global-level modelling	Accuracy	94%	97%
		F-measure	93%	96%
	Centralised learning approach		96%	97%
Diabetes	Global-level modelling	Accuracy	78%	83%
		F-measure	77%	78%
	Centralised learning approach		78%	77%
Heart Disease	Global-level modelling	Accuracy	86%	86%
		F-measure	86%	86%
	Centralised learning approach		92%	82%
Liver Disease	Global-level modelling	Accuracy	65%	74%
		F-measure	65%	70%
	Centralised learning approach		78%	72%
Spine Disease	Global-level modelling	Accuracy	59%	87%
		F-measure	58%	92%
	Centralised learning approach		66%	87%
Breast Cancer Wisconsin (Original)	Global-level modelling	Accuracy	98%	96%
		F-measure	98%	96%
	Centralised learning approach		98%	97%
Cardiovascular Disease	Global-level modelling	Accuracy	73%	73%
		F-measure	73%	73%
	Centralised learning approach		73%	73%

Table 3.13 shows the global-level modelling results for the randomly partitioned data and compares the proposed global combined model with the related works [6, 48, 96, 166, 170, 176-182]. The proposed approaches in [6] and [48] are based on centralised machine learning. In [96], the proposed method is a two-party collaborative classification using an encryption technique. The proposed approaches in [166], [170], and [176-181] are federated learning using encryption method, and in [182] is based on distributed processing using Spark. The proposed methods outperformed most proposed research works in diabetes, heart disease, and breast cancer Wisconsin (Original) datasets. In liver disease dataset, the proposed approach in [48] outperformed the other methods, and the proposed methods in [96] is better than the other approaches in diabetes dataset.

Table 3.13. Global Combined Model Evaluation Compared with Research Works

Models	Selection Metric	Combination Method	Breast Cancer Wisconsin (Diagnostic)	Diabetes	Heart Disease	Liver Disease	Breast Cancer Wisconsin (Original)
Global-level modelling	Accuracy	Weighted Voting	94%	78%	**86%**	65%	**98%**
		Average Accuracy	93%	75%	84%	66%	**98%**
		Majority Voting	94%	78%	**86%**	65%	**98%**
	F-measure	Weighted Voting	93%	77%	**86%**	65%	**98%**
		Majority Voting	93%	76%	**86%**	65%	**98%**
	Single Best Model		**97%**	78%	**86%**	68%	**98%**
Tsoumakas et al. [6] - EV1			-	77%	84%	**-**	97%
Tsoumakas et al. [6] - EV2			-	77%	83%	**-**	97%
Tsoumakas et al. [6] - EV 3			-	77%	85%	**-**	97%
Bashir et al. [48]			-	77%	84%	**71%**	97%
Zhang et al. [96]			-	**80%**	-	-	-
Mandal et al. [166]			96%	76%	-	-	-
Wang et al. [170]			-	77%	-	-	96%
Gao et al. [176]			-	-	72%	-	95%
Kasturi et al. [177]			-	-	-	-	96%
Ma et al. [178]			-	-	-	-	96%
Haque et al. [179]			-	78%	82%	-	**98%**
Sav et al. [180]			-	-	-	-	97%
Froelicher et al. [181]			-	78%	-	-	96%
Ed-daoudy and Maalmi [182]			-	-	82%	-	-

II. Local-level Modelling Results:

Table 3.14 shows the local combined model evaluation for blood transfusion dataset; we compared **Method L1** and **Method L2** methods with average accuracy and majority voting combination methods (MV) for the selected models based on accuracy and F-measure performance metrics. Besides, we compared the results of the proposed methods with the best local model in each site and DES methods. Our method outperformed the best local model in all sites. Furthermore, it is better than the DES methods in site 1 and comparable results in the other sites; thus, we conclude that the distributed sites can utilise other sites models to improve the prediction accuracy.

Table 3.14. Blood Transfusion Prediction Accuracy for Local Combined Model

Methods	Selection metric	Combination method	Site1	Site2	Site3
Method L1	Accuracy	Weighted Voting	**57%**	**60%**	-
		Average Accuracy	51%	55%	-
		Majority Voting	51%	**60%**	-
	F-measure	Weighted Voting	51%	**60%**	60%
		Majority Voting	51%	**60%**	63%

Method L2	Accuracy	Weighted Voting	-	-	63%
		Average Accuracy	-	-	58%
		Majority Voting	-	-	63%
	F-measure	Weighted Voting	-	-	60%
		Majority Voting	-	-	63%
The Best Local Model			51%	40%	54%
Dynamic Ensemble Selection	KNORA-U		51%	**60%**	**66%**
	KNORA-E		43%	43%	60%
	DES-P		51%	**60%**	57%
	META-DES		43%	**60%**	60%
	KNOP		34%	**60%**	**66%**
	DES-KNN		46%	40%	46%

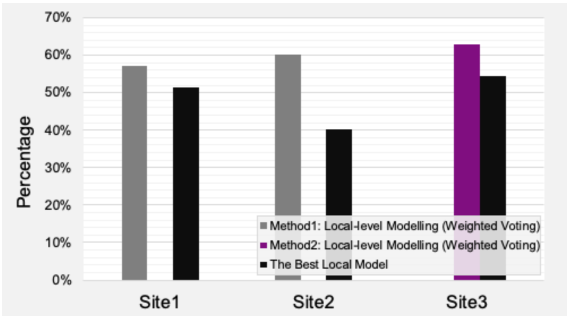


Figure 3.7. Local-Level Modelling Accuracy vs The Best Local Model For Blood Transfusion Dataset

For breast cancer Wisconsin (Diagnostic) dataset in Table 3.15, the local combined model results outperformed the best local model in site 3 and are in par with the best local model results in site 1 and site 2. In addition, the local combined model results are better or similar to DES methods performance in site 1 and 3.

Table 3.15. Breast Cancer Wisconsin (Diagnostic) Prediction Accuracy for Local Combined Model

Methods	Selection metric	Combination method	Site1	Site2	Site3
Method L1	Accuracy	Weighted Voting	90%	93%	89%
		Average Accuracy	93%	93%	92%
		Majority Voting	90%	93%	89%
	F-measure	Weighted Voting	90%	93%	-
		Majority Voting	90%	93%	-
Method L2	Accuracy	Weighted Voting	**95%**	-	**95%**
		Average Accuracy	93%	-	**95%**
		Majority Voting	**95%**	-	**95%**
	F-measure	Weighted Voting	**95%**	-	89%
		Majority Voting	**95%**	-	**95%**
The Best Local Model			**95%**	93%	89%
Dynamic Ensemble Selection	KNORA-U		90%	**97%**	**95%**
	KNORA-E		90%	**97%**	89%
	DES-P		90%	**97%**	**95%**
	META-DES		**95%**	**97%**	89%
	KNOP		**95%**	**97%**	89%
	DES-KNN		**95%**	**97%**	89%

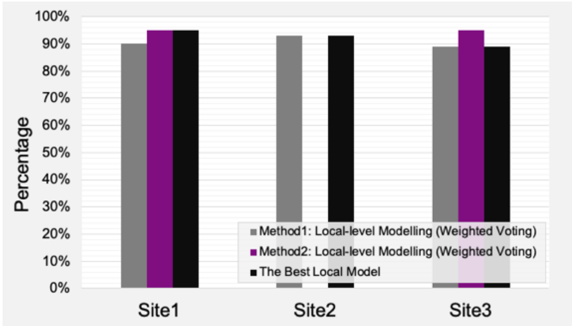


Figure 3.8. Local-Level Modelling Accuracy vs The Best Local Model for Breast Cancer Wisconsin (Diagnostic)

In Table 3.16, the proposed local level modelling results for diabetes dataset showed that each site could utilised from other sites models to improve its prediction performance. Moreover, the local combined model in site 3 is better than DES methods and got close results in site 2.

Table 3.16. Diabetes Prediction Accuracy for Local Combined Model

Methods	Selection metric	Combination method	Site1	Site2	Site3
Method L1	Accuracy	Weighted Voting	75%	80%	72%
		Average Accuracy	67%	77%	75%
		Majority Voting	68%	80%	**76%**
	F-measure	Weighted Voting	70%	69%	72%
		Majority Voting	75%	69%	**76%**
Method L2	Accuracy	Weighted Voting	78%	-	**76%**
		Average Accuracy	69%	-	75%
		Majority Voting	78%	-	72%
	F-measure	Weighted Voting	78%	74%	-
		Majority Voting	78%	74%	-
The Best Local Model			68%	74%	68%
Dynamic Ensemble Selection	KNORA-U		78%	**83%**	68%
	KNORA-E		72%	74%	68%
	DES-P		80%	80%	64%
	META-DES		68%	80%	64%
	KNOP		72%	**83%**	72%
	DES-KNN		**82%**	**83%**	68%

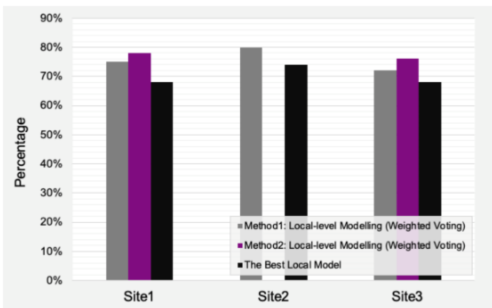


Figure 3.9. Local-Level Modelling Accuracy vs The Best Local Model for Diabetes Dataset

As shown in Table 3.17 for heart diseases dataset, the proposed local combined model in site 2 and site 3 is better than the best local model and got close results in site 1. The local combined model and some DES methods got similar or comparable results in all sites.

Table 3.17. Heart Disease Prediction Accuracy for Local Combined Model

Methods	Selection metric	Combination method	Site1	Site2	Site3
Method L1	Accuracy	Weighted Voting	87%	**95%**	87%
		Average Accuracy	90%	91%	85%
		Majority Voting	93%	**95%**	87%
	F-measure	Weighted Voting	87%	**95%**	87%
		Majority Voting	73%	90%	80%
Method L2	Accuracy	Weighted Voting	87%	-	-
		Average Accuracy	89%	-	-
		Majority Voting	87%	-	-
	F-measure	Weighted Voting	87%	-	-
		Majority Voting	87%	-	-
The Best Local Model			93%	85%	80%
Dynamic Ensemble Selection	KNORA-U		87%	**95%**	87%
	KNORA-E		**99%**	60%	67%
	DES-P		87%	85%	87%
	META-DES		80%	85%	87%
	KNOP		87%	90%	**93%**
	DES-KNN		87%	80%	80%

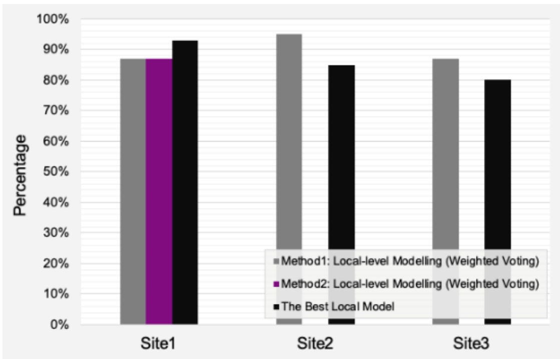


Figure 3.10. Local-level modelling accuracy vs the best local model for Heart disease dataset

For liver disease dataset, the local combined model results in Table 3.18 are in par with the best local models results in all sites. In addition, most dynamic ensemble selection methods are similar to or better than our proposed combined model.

Table 3.18. Liver Disease Prediction Accuracy for Local Combined Model

Methods	Selection metric	Combination method	Site1	Site2	Site3
Method L1	Accuracy	Weighted Voting	-	-	67%
		Average Accuracy	-	-	62%
		Majority Voting	-	-	60%
	F-measure	Weighted Voting	-	65%	60%
		Majority Voting	-	65%	60%
Method L2	Accuracy	Weighted Voting	68%	68%	-
		Average Accuracy	68%	67%	-
		Majority Voting	68%	68%	-
	F-measure	Weighted voting	68%	-	-
		Majority Voting	68%	-	-
The Best Local Model			68%	68%	67%
Dynamic Ensemble Selection	KNORA-U		68%	72%	60%
	KNORA-E		68%	65%	67%
	DES-P		68%	72%	67%
	META-DES		68%	80%	**73%**
	KNOP		68%	**75%**	60%
	DES-KNN		**72%**	72%	67%

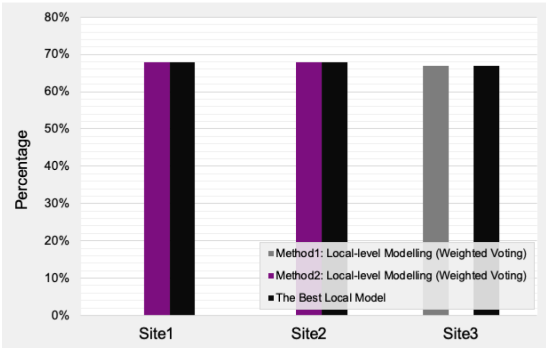


Figure 3.11. Local-Level Modelling Accuracy vs The Best Local Model for Liver Disease Dataset

In Table 3.19, the local combined model results got better results than the best local model and DES methods in site 1, and similar results in site 2 for spine disease dataset.

Table 3.19. Spine Disease Prediction Accuracy for Local Combined Model

Methods	Selection metric	Combination method	Site1	Site2	Site3
Method L1	Accuracy	Weighted Voting	**-**	50%	-
		Average Accuracy	**-**	54%	-
		Majority Voting	-	50%	-
	F-measure	Weighted Voting	-	50%	-
		Majority Voting	-	50%	-
Method L2	Accuracy	Weighted Voting	**65%**	50%	69%
		Average Accuracy	55%	52%	69%
		Majority Voting	55%	**57%**	69%
	F-measure	Weighted Voting	60%	-	69%
		Majority Voting	60%	-	**75%**
The Best Local Model			55%	50%	**75%**

Dynamic Ensemble Selection	KNORA-U	55%	50%	69%
	KNORA-E	50%	50%	**75%**
	DES-P	55%	50%	69%
	META-DES	55%	50%	**75%**
	KNOP	55%	50%	**75%**
	DES-KNN	55%	50%	69%

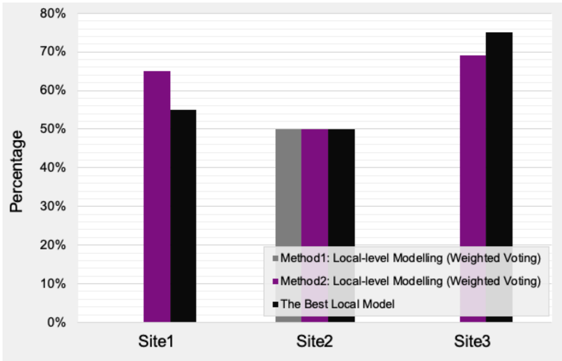


Figure 3.12. Local-Level Modelling Accuracy vs The Best Local Model for Spine Disease Dataset

For Breast Cancer Wisconsin (Original) dataset, Table 3.20 shows that the proposed local combined model, the best local model, and DES methods got close results.

Table 3.20. Breast Cancer Wisconsin (Original) Prediction Accuracy for Local Combined Model

Methods	Selection metric	Combination method	Site1	Site2	Site3
Method L1	Accuracy	Weighted Voting	-	**99%**	**99%**
		Average Accuracy	-	98%	98%
		Majority Voting	-	**99%**	**99%**
	F-measure	Weighted Voting	-	-	**99%**
		Majority Voting	-	-	**99%**
Method L2	Accuracy	Weighted Voting	97%	-	-
		Average Accuracy	97%	-	-
		Majority Voting	97%	-	-
	F-measure	Weighted Voting	97%	97%	-
		Majority Voting	97%	97%	-
The Best Local Model			**98%**	**99%**	**99%**
Dynamic Ensemble Selection	KNORA-U		**98%**	97%	98%
	KNORA-E		**98%**	**99%**	98%
	DES-P		**98%**	97%	98%
	META-DES		**98%**	97%	98%
	KNOP		**98%**	97%	98%
	DES-KNN		**98%**	**99%**	98%

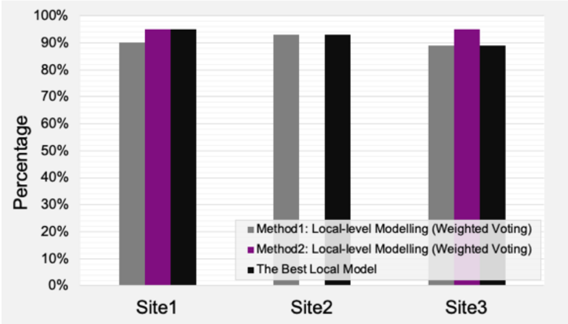


Figure 3.13. Local-Level Modelling Accuracy vs The Best Local Model for Breast Cancer Wisconsin (Original) Dataset

In Table 3.21, the local combined model for cardiovascular disease dataset got similar results in site 2 and site 3 compared with the best local model. The best local model in site 1 is slightly better than the proposed method. In addition, some DES methods results are similar to or close to our approach and the best local model results.

Table 3.21. Cardiovascular Diseases Prediction Accuracy for Local Combined Model

Methods	Selection metric	Combination method	Site1	Site2	Site3
Method L1	Accuracy	Weighted Voting	73%	**73%**	**72%**
		Average Accuracy	73%	**73%**	**72%**
		Majority Voting	73%	**73%**	**72%**
	F-measure	Weighted Voting	73%	**73%**	**72%**
		Majority Voting	73%	**73%**	**72%**
Method L2	Accuracy	Weighted Voting	-	-	-
		Average Accuracy	-	-	-
		Majority Voting	-	-	-
	F-measure	Weighted Voting	-	-	**72%**
		Majority Voting	-	-	**72%**
The Best Local Model			**74%**	**73%**	**72%**
Dynamic Ensemble Selection	KNORA-U		73%	**73%**	**72%**
	KNORA-E		69%	67%	68%
	DES-P		72%	72%	71%
	META-DES		66%	71%	65%
	KNOP		66%	**73%**	65%
	DES-KNN		71%	72%	71%

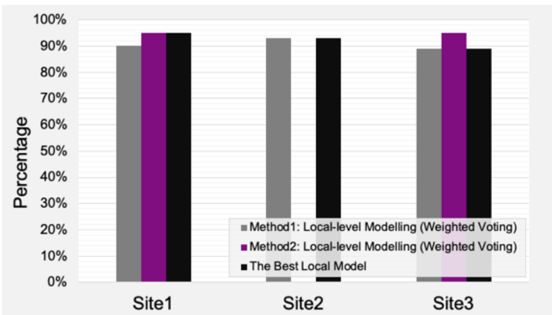


Figure 3.14. Local-Level Modelling Accuracy vs The Best Local Model for Cardiovascular Diseases Dataset

2) Non-random Data Partitioning Approach:

We split the data by age and tried to get different data distributions to mimic a real-world scenario for distributed datasets for distributed sites. Table 3.22 shows our partition strategy for the three datasets between the sites, and Table 3.23 shows the datasets partitions size. Local data is used to build the local models in each site and evaluate the received models from other sites. The second partition is used to evaluate the combined models of the global and local level modelling methods. Datasets distributions are illustrated in Appendix A.

Table 3.22. Datasets Partitioning Scenarios

Datasets	Site1	Site2	Site3
Diabetes	Age: less than 30	Age: bigger than or equal 30 and less than or equal 45	Age: bigger than 45
Liver Disease	Age: less than or equal to 30	Age: bigger than 30 and less than 60	Age: bigger than or equal to 60
Heart Disease	Age: less than or equal to 45	Age: bigger than 45 and less than 60	Age: bigger than or equal to 60

Table 3.23. Datasets Partitions

Datasets	Site 1		Site 2		Site 3	
	Local data	Validation data	Local data	Validation data	Local data	Validation data
Diabetes	349	47	211	43	87	31
Liver disease	94	20	299	33	116	17
Heart disease	44	20	116	33	68	22

I. Global-level Modelling Results:

The detailed results for the global-level modelling are shown in Appendix B. For diabetes dataset, Table 3.24 shows that the global level modelling and the centralised learning approach results are close.

Table 3.24. Global Combined Model and Centralised Learning Approach Evaluation for Diabetes Dataset

Models	Selection Metric	Combination Method	Accuracy
Global-level modelling	Accuracy	Weighted Voting	66%
		Average Accuracy	65%
		Majority Voting	68%
	F-measure	Weighted Voting	67%
		Majority Voting	68%
	Single Best Model (LR model – S2)		67%
Best Local Models Combination	Accuracy	Weighted Voting	68%
		Average Accuracy	66%
		Majority Voting	**73%**
	F-measure	Weighted Voting	67%
		Majority Voting	**73%**
	Single Best Model (NB model -S2)		72%
Centralised Learning Approach (LR model)			69%

Table 3.25 and 3.26 shows that our proposed method results are close to the centralised learning approach results. For heart and liver disease datasets, respectively.

Table 3.25. Global Combined Model and Centralised Learning Approach Evaluation for Heart Disease Dataset

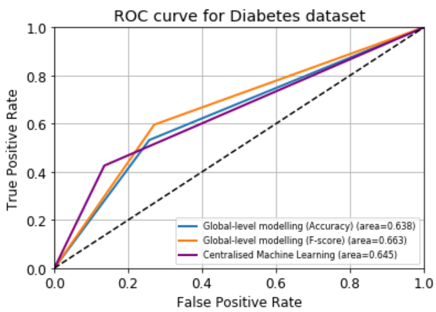
Models	Selection Metric	Combination Method	Accuracy
Global-level modelling	Accuracy	Weighted Voting	87%
		Average Accuracy	83%
		Majority Voting	85%
	F-measure	Weighted Voting	85%
		Majority Voting	85%
	Single Best Model (NB model – S3)		**88%**
Best Local Models Combination	Accuracy	Weighted Voting	87%
		Average Accuracy	83%
		Majority Voting	85%
	F-measure	Weighted Voting	84%
		Majority Voting	85%
	Single Best Model (NB model – S3)		**88%**
Centralised Learning Approach (DT model)			**88%**

Table 3.26. Global Combined Model and Centralised Learning Approach Evaluation for Liver Disease Dataset

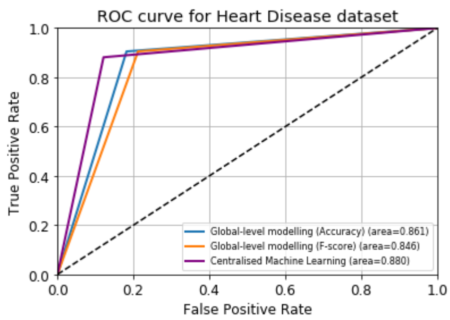
Models	Selection Metric	Combination Method	Accuracy
Global-level modelling	Accuracy	Weighted Voting	72%
		Average Accuracy	72%
		Majority Voting	72%
	F-measure	Weighted Voting	72%
		Majority Voting	72%
	Single Best Model (SVM model- S2)		74%
	Accuracy	Weighted Voting	73%

Best Local Models Combination		Average Accuracy	69%
		Majority Voting	67%
	F-measure	Weighted Voting	72%
		Majority Voting	72%
	Single Best Model (SVM model -S2)		**76%**
Centralised Learning Approach (SVM model)			74%

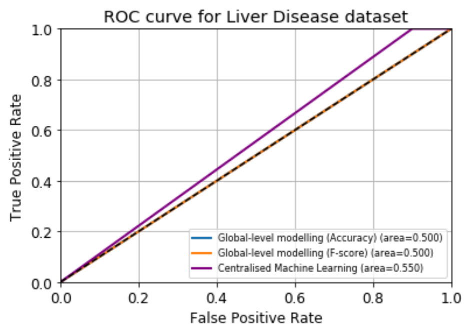
Figure 3.15 illustrates the ROC curve analysis for the proposed method and the centralised learning methods for diabetes, heart disease, and liver disease dataset. The proposed method results are close to the centralised learning approach. Table 3.27 shows the training and testing accuracy for the proposed method and the centralised learning approach.



(a) Diabetes



(b) Heart disease



(c) Liver disease

Figure 3.15. ROC Curve Analysis

Table 3.27. The Training and Testing Accuracy for The Proposed Method and Centralised Learning Approach

Dataset	Models	Selection Metric	Testing Accuracy	Training accuracy
Diabetes	Global-level modelling	Accuracy	66%	78%
		F-measure	67%	**79%**
	Centralised learning approach		**69%**	78%
Heart Disease	Global-level modelling	Accuracy	87%	**85%**
		F-measure	85%	84%
	Centralised learning approach		**88%**	83%
Liver Disease	Global-level modelling	Accuracy	72%	**73%**
		F-measure	72%	**73%**
	Centralised learning approach		**74%**	72%

Table 3.28 shows the global level modelling results compared with the related works [6, 48, 96, 166, 170, 176, 179, 181, 182]. Our method got better results in heart disease and liver disease datasets.

Table 3.28. Global Combined Model Evaluation Compared with Research Works

Models	Selection Metric	Combination Method	Diabetes	Heart Disease	Liver Disease
Global-level modelling	Accuracy	Weighted Voting	66%	87%	72%
		Average Accuracy	65%	83%	72%
		Majority Voting	68%	85%	72%
	F-measure	Weighted Voting	67%	85%	72%
		Majority Voting	68%	85%	72%
	Single Best Model		67%	**88%**	**74%**
Tsoumakas et al. [6] - EV1			77%	84%	-
Tsoumakas et al. [6] - EV2			77%	83%	-
Tsoumakas et al. [6] - EV 3			77%	85%	-
Bashir et al. [48]			77%	84%	71%
Zhang et al. [96]			**80%**	-	-
Mandal et al. [166]			76%	-	-
Wang et al. [170]			77%	-	-
Gao et al. [176]			-	72%	-
Haque et al. [179]			78%	82%	-
Froelicher et al. [181]			78%	-	-
Ed-daoudy and Maalmi [182]			-	82%	-

II. Local-Level Modelling Results:

In Table 3.29, the best local model result for diabetes dataset in site 2 is slightly better than our method, while in sites 1 and 3, the results of our proposed method are better than the best local model. In addition, the proposed local combined model got better results than most DES methods in sites 1 and 2.

Table 3.29. Diabetes Prediction Accuracy for Local Combined Model for Diabetes Dataset

Methods	Selection metric	Combination method	Site1	Site2	Site3
Method L1	Accuracy	Weighted Voting	-	65%	-
		Average Accuracy	-	64%	-
		Majority Voting	-	63%	-
	F-measure	Weighted Voting	**77%**	56%	-
		Majority Voting	72%	63%	-
Method L2	Accuracy	Weighted Voting	72%	65%	58%
		Average Accuracy	71%	62%	59%
		Majority Voting	70%	60%	58%
	F-measure	Weighted Voting	74%	-	58%
		Majority Voting	72%	-	58%
The Best Local Model			70%	**67%**	52%
Dynamic Ensemble Selection	KNORA-U		74%	56%	**77%**
	KNORA-E		68%	53%	61%
	DES-P		72%	58%	**77%**
	META-DES		68%	51%	71%
	KNOP		72%	56%	74%
	DES-KNN		**77%**	58%	74%

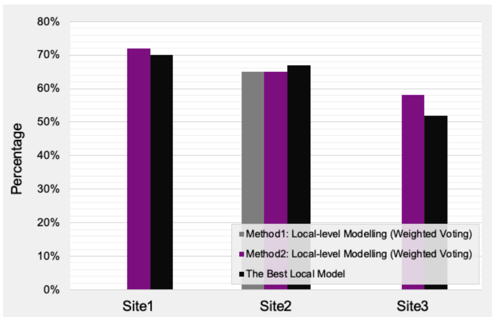


Figure 3.16. Local-Level Modelling Accuracy vs The Best Local Model for Diabetes Dataset

For heart disease dataset, Table 3.30 shows that the local combined model outperformed the best local model in site 1 and got a similar performance in site 2. While in site 3, the best local model result is better than the proposed local combined model. Some DES methods results are better than the proposed method and the best local model in site 1 and site 2.

Table 3.30. Heart Disease Prediction Accuracy for Local Combined Model

Methods	Selection metric	Combination method	Site1	Site2	Site3
Method L1	Accuracy	Weighted Voting	85%	85%	82%
		Average Accuracy	83%	84%	80%
		Majority Voting	85%	88%	82%
	F-measure	Weighted Voting	85%	88%	82%
		Majority Voting	85%	88%	82%
Method L2	Accuracy	Weighted Voting	85%	-	82%
		Average Accuracy	84%	-	78%
		Majority Voting	85%	-	77%

	F-measure	Weighted Voting	85%	-	86%
		Majority Voting	85%	-	82%
The Best Local Model			75%	88%	**91%**
Dynamic Ensemble Selection	KNORA-U		**90%**	88%	86%
	KNORA-E		80%	**91%**	**91%**
	DES-P		**90%**	**91%**	86%
	META-DES		85%	82%	82%
	KNOP		85%	**91%**	86%
	DES-KNN		85%	82%	**91%**

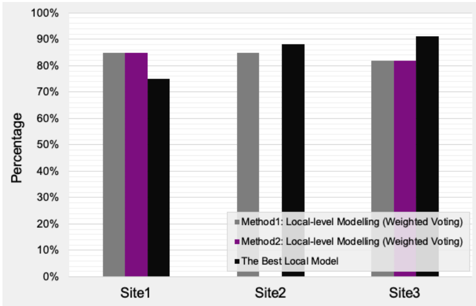


Figure 3.17. Local-Level Modelling Accuracy vs The Best Local Model for Heart Disease Dataset

As shown in Table 3.31, the local combined model got better accuracy than the best local model and DES methods in site 1 for liver disease dataset. In site 2 and site 3, the best local model has the best accuracy but is not far from our results. Some DES methods results are similar to or better than the proposed method and the best local model in site 2 and site 3.

Table 3.31. Liver Disease Prediction Accuracy for Local Combined Model

Methods	Selection metric	Combination method	Site1	Site2	Site3
Method L1	Accuracy	Weighted Voting	-	-	71%
		Average Accuracy	-	-	73%
		Majority Voting	-	-	71%
	F-measure	Weighted Voting	**65%**	-	71%
		Majority Voting	**65%**	-	71%
Method L2	Accuracy	Weighted Voting	40%	76%	71%
		Average Accuracy	52%	77%	72%
		Majority Voting	40%	76%	71%
	F-measure	Weighted voting	-	76%	71%
		Majority Voting	-	76%	71%
The Best Local Model			50%	**79%**	76%
Dynamic Ensemble Selection	KNORA-U		50%	67%	76%
	KNORA-E		50%	76%	76%
	DES-P		40%	58%	71%
	META-DES		45%	76%	76%
	KNOP		55%	73%	**82%**
	DES-KNN		45%	61%	71%

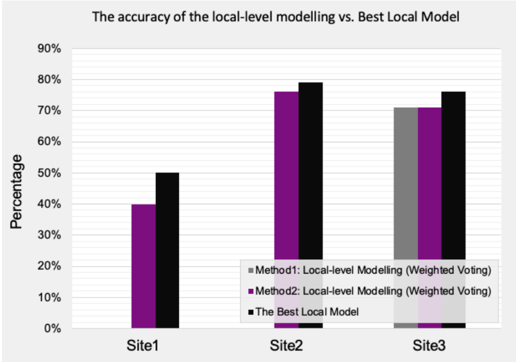


Figure 3.18. Local-Level Modelling Accuracy vs The Best Local Model for Liver Disease Dataset

3.3.4 Discussion and Evaluation

We evaluated the reliability and prediction performance of the global combined model for the data that was partitioned randomly and non-randomly. We compared the proposed global combined model for the randomly partitioned datasets with the centralised learning approach that moves all distributed data to a centralised database. Our method results are in par with the centralised learning approach in diabetes, cardiovascular disease, and breast cancer Wisconsin (Original) datasets and close performance in blood transfusion and breast cancer Wisconsin (Diagnostic) datasets. The combined models selected by accuracy metric are similar to or slightly better than F-measure selection metric, and no significant differences in results between the combination methods, weighted voting, majority voting, and average accuracy. For the non-randomly partitioned datasets, we compared the global combined model with the centralised learning approach. The proposed method results are close to the centralised learning approach in all datasets. The global combined model selected by accuracy metric is better than the global combined model selected by F-measure selection metric in the heart disease dataset, and the combined model selected by accuracy and F-measure metrics got similar results in the liver disease dataset. In diabetes dataset, the combined model selected by F-measure is slightly better than accuracy selection metric. The weighted model combination method is slightly better than other methods in the heart disease dataset, and similar to majority voting and average accuracy in the liver disease dataset. In the diabetes dataset, the majority voting is the best but not far from the other combination methods.

We developed a global model that performs similarly or close to the centralised learning approach without sharing the data between distributed sites to preserve data privacy, send the data to a central location, or use a server to control the learning process to avoid server issues and overheads. We saved the cost and time of data transformation from one site to another or a central location. We developed the combined models with fewer communication rounds, and minimal information exchanged between distributed sites. The proposed linear combination approach did not expose the data resource and preserved data privacy. It showed its efficiency and applicability in the distributed environment. We compared the proposed local combined model results with the best local model and DES methods, and we achieved improved performance compared with the best local model in each site. Furthermore, we proved that the distributed sites could utilise other sites models to improve the prediction accuracy without exchanging the data between sites, which will preserve data privacy. In addition, the proposed linear combination approach is easy to implement in distributed environments and could be applied to solve issues related to large data, such as memory limitation and huge data transformation costs and time. However, the proposed method exchanged the models instead of data to preserve data privacy and there is a possibility for malicious attacks on the trained models to retrieve training data or reveal meaningful information. We did not consider model attacks case in the distributed environment. It is beyond the scope of our thesis, and we will consider these issues to analyse the possible malicious attacks on distributed sites and exchanged models in future research.

3.4 PROPOSED METHOD FOR REGRESSION ALGORITHMS

The proposed method addresses individual model limitations by utilising distributed data resources to develop combined prediction models at global and local levels without data transformation between sites to preserve the privacy of local data resources. For this purpose, the related model names and definitions used in our methodology are first introduced in Table 3.32.

Table 3.32. Models Names and Descriptions.

Model	Notation	Meaning
Local Model	M_{ij}	The local model that developed in site i using j regression algorithm
Received Model	$M_{i'j}$	Model that received from other sites i'
Best local model MAPE	M_{ij*}	The local model in site i that has the lowest MAPE and developed by j* regression algorithm
Best local model RMSE	M_{ij**}	The local model in site i that has the lowest RMSE and developed by j** regression algorithm
Best Model MAPE	$M_{i'j*}$	The selected model from other sites i' which is lower than or equal to the best local model MAPE
Best Model RMSE	$M_{i'j**}$	The selected model from other sites i' which is lower than or equal to the best local model RMSE
Best Global Average Model MAPE	M^G_{ij*}	The best average MAPE model in site i after global evaluation and average MAPE calculation
Best Global Average Model RMSE	M^G_{ij**}	The best average RMSE model in site i after global evaluation and average RMSE calculation
Global Combined Model (1)	M^{G*}	The final global combined model at the server that combine the best global average model MAPE from each site M^G_{ij*}
Global Combined Model (2)	M^{G**}	The final global combined model at the server that combine the best global average model RMSE from each site M^G_{ij**}
List of the best models MAPE	M_{MAPE}	List of the best local model MAPE M_{ij*} and the selected models of the best MAPE from other sites $M_{i'j*}$ that will linearly be combined to build the local combined model
List of the best models RMSE	M_{RMSE}	List of the best local model RMSE $M_{ij**}$and the selected models of the best RMSE from other sites $M_{i'j**}$ that will linearly be combined to build the local combined model
Local Combined Model (1)	M^{L*}_i	The final local combined model in site i that combine the best local model MAPE of the site i with the best models MAPE from other sites
Local Combined Model (2)	M^{L**}_i	The final local combined model in site i that combine the best local model RMSE of the site i with the best models RMSE from other sites

3.4.1 Global-level Modelling Approach

We aim to build a global combined model at the central server by combining the best global average model RMSE and MAPE from each site. We apply the same approach steps that used in section 3.3.1, but we use RMSE and MAPE metrics for model evaluation, selection, and combination approaches.

1) For each site S_i , where $i = 1,2, \ldots , n$:

- Apply different j learning algorithms, where $j = 1,2, \ldots, m$ to build local models M_{ij}.
- Use 10-fold cross-validation results to evaluate the local models M_{ij} based on the local data in site S_i.
- Calculates the RMSE (M_{ij}) and MAPE (M_{ij}) based on local data error results.

$$\text{RMSE}(M_{ij}) = \sqrt{\frac{1}{m} \sum_{i=1}^{m} |y_i - \hat{y_i}|^2} \qquad (3.13)$$

$$\text{MAPE}(M_{ij}) = \frac{1}{m} \sum_{i=1}^{m} \left| \frac{y_i - \hat{y_i}}{y_i} \right| * 100 \qquad (3.14)$$

, where y_i is the actual value and $\hat{y_i}$ is the predicted value for m data samples.

2) Each site sends its local models M_{ij} to other sites for global evaluation as follows:
   - Each site i will receive models from other sites $M_{i'j}$, then start to evaluate these models over its local dataset and calculate MAPE and RMSE for each received model MAPE ($M_{i'j}$) and RMSE ($M_{i'j}$)
   - Send the evaluated models back to the sites with its evaluation results.
   - In each site S_i:
     a) Receive the evaluation results MAPE (M_{ij}) and RMSE (M_{ij}) of its local models M_{ij} from other sites with sites data samples number that used for evaluation.
     b) Calculate the global average MAPE and global average RMSE for each local model.

$$\text{MAPE}(M_{ij}) = \sum_{k=1}^{n} \frac{D_k}{D} * \text{MAPE}(M_{ij}) \text{ in } S_k \qquad (3.15)$$

$$\text{RMSE}(M_{ij}) = \sum_{k=1}^{n} \frac{D_k}{D} * \text{RMSE}(M_{ij}) \text{ in } S_k \qquad (3.16)$$

, where k is the number of sites, D_k is the number of samples of site k and D is all sites' samples number.

     c) Select the best model(s) based on the best global average MAPE (M_{ij*}^{G}) and the best global average RMSE(M_{ij**}^{G}).
     d) Send the selected global average models with its evaluation

results, MAPE (M_{ij}^{G*}), RMSE (M_{ij}^{G*}), MAPE (M_{ij}^{G**}), and RMSE (M_{ij}^{G**}) to the server.

3) The server combines the received models to develop two final global combined models M^{G*} and M^{G**} using linear combination methods, M^{G*} is the final global combined model at the server that combines the best global average model MAPE from each site, and M^{G**} The final global combined model at the server that combines the best global average model RMSE from each site, the server performs the linear combination method as follows:
   a) The server receives two models from each site, the best global average MAPE M_{ij*}^{G} and the best global average RMSE M_{ij**}^{G} with its global average results, MAPE (M_{ij}^{G*}), RMSE (M_{ij}^{G*}), MAPE (M_{ij}^{G**}), and RMSE (M_{ij}^{G**}).
   b) The server combines the best global average MAPE model M_{ij*}^{G} by calculating the weight of each model based on its global average results. Also, it combines the best global average model RMSE M_{ij**}^{G} by weighting the models based on its global average results.
   c) We apply four different weighting methods to develop the final global combined models M^{G*} and M^{G**}, the most accurate model will get higher weight, and the less accurate model will get low weight, and models' weights are constrained such that their sum is equal to one. The weighting methods are:

i. Simple Weight Average Method:

$$w_{M_{ij*}^{G}} = \frac{1}{n} \qquad (3.17)$$

, where 0 <= $w_{M_{ij*}^{G}}$ <=1, $\sum_{i=1}^{n} w_{M_{ij*}^{G}} = 1$, and n is the number of models.

$$w_{M_{ij**}^{G}} = \frac{1}{n} \qquad (3.18)$$

, where 0 <= $w_{M_{ij**}^{G}}$<=1, $\sum_{i=1}^{n} w_{M_{ij**}^{G}} = 1$, and n is the number of models.

ii. **Error-based (RMSE):** weight for each model M_i is taken to be inversely proportional to the model error.

$$w_{M_i} = \frac{error_{M_i}^{-1}}{\sum_{i=1}^{n} error_{M_i}^{-1}} \tag{3.19}$$

We use RMSE for models error, higher RMSE is given a smaller weight.

$$w_{M_{ij*}^G} = \frac{\frac{1}{RMSE\ M_{ij*}^G}}{\sum_{i=1}^{n} \frac{1}{RMSEM_{ij*}^G}} \tag{3.20}$$

, where $\sum_{i=1}^{n} w_{M_{ij*}^G} = 1$

$$w_{M_{ij**}^G} = \frac{\frac{1}{RMSE\ M_{ij**}^G}}{\sum_{i=1}^{n} \frac{1}{RMSE\ M_{ij**}^G}} \tag{3.21}$$

, where $\sum_{i=1}^{n} w_{M_{ij**}^G} = 1$

iii. **Performance-based (Accuracy):** Calculates the accuracy for each local model M_{ij*}^G and M_{ij**}^G.

$$Acc\ (M_{ij*}^G) = 100 - MAPE\ (M_{ij*}^G) \tag{3.22}$$

$$Acc\ (M_{ij**}^G) = 100 - MAPE\ (M_{ij**}^G) \tag{3.23}$$

Then,

$$w_{M_{ij*}^G} = \frac{Acc\ (M_{ij*}^G)}{\sum_{i=1}^{n} Acc\ (M_{ij*}^G)} \tag{3.24}$$

, where i=1, 2, …, n and $\sum_{i=1}^{n} w_{M_{ij*}^G} = 1$

$$w_{M_{ij**}^G} = \frac{Acc\ (M_{ij**}^G)}{\sum_{i=1}^{n} Acc(\ M_{ij**}^G)} \tag{3.25}$$

, where i=1, 2, …, n and $\sum_{i=1}^{n} w_{M_{ij**}^G} = 1$

iv. **Shapley Value:**

a) Calculate total RMSE of the models.

$$RMSE^* = \frac{1}{n} \sum_{i=1}^{n} RMSE\ M_{ij*}^G \tag{3.26}$$

$$\text{RMSE}^{**} = \frac{1}{n}\sum_{i=1}^{n} \text{RMSE } M^G_{ij**} \qquad (3.27)$$

, where n is the number of models.

b) Calculate Shapley value:

$$\varphi_{w_{M^G_{ij*}}}(v) = \sum_{si \in s} w\,(|s_i|) * [\text{RMSE}\,(s_i) - \text{RMSE}\,(s_i - \{i\})] \quad (3.28)$$

$$\varphi_{w_{M^G_{ij**}}}(v) = \sum_{si \in s} w\,(|s_i|) * [\text{RMSE}\,(s_i) - \text{RMSE}\,(s_i - \{i\})] \quad (3.29)$$

$$\text{where, } w(|s_i|) = \frac{(n-|s_i|)!((|s_i|)-1)!}{n!} \qquad (3.30)$$

s_i is the set containing the best global average model, $|s_i|$ is the number of models in the combination, RMSE (s_i) is the combined RMSE of this combination subset, and $s_i - \{i\}$ is a set obtained from s_i by removing i-th model in the combination.

c) Calculate the weight of each model in the combination:

$$w_{M^G_{ij*}} = \frac{1}{n-1} * \frac{\text{RMSE}^{*} - \varphi_{w_{M^G_{ij*}}}(v)}{\text{RMSE}^{*}} \text{, i= 1, 2, …, n} \qquad (3.31)$$

$$w_{M^G_{ij**}} = \frac{1}{n-1} * \frac{\text{RMSE}^{**} - \varphi_{w_{M^G_{ij**}}}(v)}{\text{RMSE}^{**}} \text{, i= 1, 2, …, n} \qquad (3.32)$$

d) The server linearly combines the models to develop the global model to predict x.

$$M^{G*}(x) = \sum_{i=1}^{n} w_{M^G_{ij*}} M^G_{ij*}(x) \qquad (3.33)$$

$$M^{G**}(x) = \sum_{i=1}^{n} w_{M^G_{ij**}} M^G_{ij**}(x) \qquad (3.34)$$

3.4.2 Local-level Modelling Approach:

In the local-level modelling, each data site tries to find the best local combined model by utilising the local data resource and the local prediction models from the other sites. We apply the same approach that used in section 3.3.2 but by using RMSE and MAPE evaluation metrics for model evaluation, selection, and combination methods.

1) When a site i receives models from other sites $M_{i'j}$, evaluate these models over its local dataset, and calculate the $MAPE(M_{i'j})$ and $RMSE(M_{i'j})$.
   a) Compare MAPE $(M_{i'j})$ and RMSE $(M_{i'j})$ of the received models with its best local model MAPE (M_{ij*}) and best local model RMSE (M_{ij**}).
   b) Select the best model MAPE $(M_{i'j*})$ and the best model RMSE $(M_{i'j**})$ from each site, we select two models to develop two local combined models M_i^{L*} and M_i^{L**}, M_i^{L*} is the final local combined model that combines the best local model MAPE M_{ij*} and the selected best model MAPE from other sites $M_{i'j*}$. And M_i^{L**} is the final local combined model that combines the best local model RMSE M_{ij**} and the selected best model RMSE from other sites $M_{i'j**}$. The aim of this method is to utilise the best models learned from other data resources to build an accurate local combined model.
2) Apply linear combination method to develop two local combined models M_i^{L*} and M_i^{L**} in each site i. Each site i calculates and assigns weights for the best local model and the selected models from other sites to perform the linear combination as follows:
   a) Each site i has a list of the best models MAPE M_{MAPE}, where M_{MAPE} is the best local model MAPE M_{ij*} and the selected models of the best MAPE from other sites $M_{i'j*}$, $M_{MAPE}= \{M_i^*, M_{i'}^*, \dots, M_n^*\}$, where i = 1, 2, …, n ,and its evaluation results, MAPE (M_i^*) and RMSE (M_i^*). Also, the site i has a list of the best models RMSE M_{RMSE}, where M_{RMSE} is the best local model RMSE M_{ij**} and the selected models of the best RMSE from other sites $M_{i'j**}$, $M_{RMSE} = \{M_i^{**}, M_{i'}^{**}, \dots, M_n^{**}\}$ and its evaluation results, MAPE (M_i^{**}) and RMSE (M_i^{**}).
   b) Calculate models weights for the selected models in M_{MAPE} and M_{RMSE}

using four weighting methods:

i. Simple Weight Average Method:

$$w_{M_i^*} = \frac{1}{n} \tag{3.35}$$

, where $0 <= w_{M_i^*} <= 1$, $\sum_{i=1}^{n} w_{M_i^*} = 1$, and n is the number of models.

$$w_{M_i^{**}} = \frac{1}{n} \tag{3.36}$$

, where $0 <= w_{M_i^{**}} <= 1$, $\sum_{i=1}^{n} w_{M_i^{**}} = 1$, and n is the number of models.

ii. Error-based (RMSE): The weight for each model M_i is taken to be inversely proportional to the model error.

$$w_{M_i} = \frac{\text{error}_{M_i}^{-1}}{\sum_{i=1}^{n} \text{error}_{M_i}^{-1}} \tag{3.37}$$

We use RMSE for models error; higher RMSE is given a smaller weight.

$$w_{M_i^*} = \frac{\frac{1}{\text{RMSE } M_i^*}}{\sum_{i=1}^{n} \frac{1}{\text{RMSE } M_i^*}} \tag{3.38}$$

, where $\sum_{i=1}^{n} w_{M_i^*} = 1$

$$w_{M_i^{**}} = \frac{\frac{1}{\text{RMSE } M_i^{**}}}{\sum_{i=1}^{n} \frac{1}{\text{RMSE } M_i^{**}}} \tag{3.39}$$

, where $\sum_{i=1}^{n} w_{M_i^{**}} = 1$

iii. Performance Based (Accuracy): Calculates the accuracy for each model M_i^* and M_i^{**}

$$\text{Acc}\,(M_i^*) = 100 - \text{MAPE}\,(M_i^*) \tag{3.40}$$

$$\text{Acc}\,(M_i^{**}) = 100 - \text{MAPE}\,(M_i^{**}) \tag{3.41}$$

Then,

$$w_{M_i^*} = \frac{\text{Acc } M_i^*}{\sum_{i=1}^{n} \text{Acc } M_i^*} \tag{3.42}$$

$$w_{M_i^{**}} = \frac{\text{Acc } M_i^{**}}{\sum_{i=1}^{n} \text{Acc } M_i^{**}} \quad (3.43)$$

iv. Shapley Value:

a) Calculate total RMSE of the models.

$$\text{RMSE}^{*} = \frac{1}{n} \sum_{i=1}^{n} \text{RMSE } M_i^{*} \quad (3.44)$$

$$\text{RMSE}^{**} = \frac{1}{n} \sum_{i=1}^{n} \text{RMSE } M_i^{**} \quad (3.45)$$

, where n is the number of models

b) Calculate Shapley value:

$$\varphi_{M_i^{*}}(v) = \sum_{si \in s} w(|s_i|) * [\text{RMSE}(s_i) - \text{RMSE}(s_i - \{i\})] \quad (3.46)$$

$$\varphi_{M_i^{**}}(v) = \sum_{si \in s} w(|s_i|) * [\text{RMSE}(s_i) - \text{RMSE}(s_i - \{i\})] \quad (3.47)$$

$$\text{where } w(|s_i|) = \frac{(n-|s_i|)!((|s_i|)-1)!}{n!} \quad (3.48)$$

s_i is the set containing the best models, $|s_i|$ is the number of models in the combination, $\text{RMSE}(s_i)$ is the combined RMSE of this combination subset, and $s_i - \{i\}$ is a set obtained from s_i by removing i-th model in the combination.

c) Calculate the weight of each model in the combination:

$$w_{M_i^{*}} = \frac{1}{n-1} * \frac{\text{RMSE}^{*} - \varphi_{M_i^{*}}(v)}{\text{RMSE}^{*}}, \; i = 1, 2, \ldots, n \quad (3.49)$$

$$w_{M_i^{**}} = \frac{1}{n-1} * \frac{\text{RMSE}^{**} - \varphi_{M_i^{**}}(v)}{\text{RMSE}^{**}}, \; i = 1, 2, \ldots, n \quad (3.50)$$

3) Combine the models to develop our proposed local combined model M_i^{L*} to predict x.

$$M_i^{L*}(x) = \sum_{i=1}^{n} w_{M_i^{*}} M_i^{*}(x) \quad (3.51)$$

$$M_i^{L**}(x) = \sum_{i=1}^{n} w_{M_i^{**}} \, M_i^{**}(x) \qquad (3.52)$$

3.4.3 Experimental Study

The experiments are performed to evaluate the proposed method performance and compared it with related work and a centralised learning approach.

I. Datasets

We used three databases: Parkinson disease, Boston housing, and Abalone datasets [29]. Before conducting the experiments, the databases are first preprocessed to a suitable data format. Parkinson disease data features are patient age and biomedical voice measurement with two target values, motor Unified Parkinson's Disease Rating Scale (UPDRS) and total Unified Parkinson's Disease Rating Scale (UPDRS). The target values show the measurement of presence and severity of Parkinson disease. Total-UPDRS ranges between 0–176, 0 reflecting healthy status and 176 indicating total disability. Motor-UPDRS, which denotes to the motor section, the range is between 0–108, 0 indicates healthy status and 108 severe case. Boston housing dataset is from several suburbs in Boston and includes economic, demographic, and land use features, and the median price of houses is the target value. Abalone dataset features are physical measurements that are used to predict the age of Abalone. We replaced the target value **rings** with **age** (rings + 1.5 = Abalone age in years). Table 3.33 describes the datasets that are used to train and test the models.

Table 3.33. Datasets Descriptions.

Datasets	Data size	No. of attributes
Parkinson	5875	18
Boston housing	506	14
Abalone	4166	8

II. Simulating Distributed Data

We applied the proposed methods using two dataset partitioning strategies: (1) random data partitioning approach and (2) non-random data partitioning approach. For the non-random data partitioning approach, we partitioned the Parkinson disease dataset by patient age and Boston housing dataset by per capita crime rate by town attribute to simulate that each data comes from different regions. Therefore, we divided each dataset into different parts as distributed sites site1, site2, and site3. In each site, the dataset is divided into local data and validation datasets. The local data partition is used to develop and evaluate the local models and evaluate the received models. The validation data is used to evaluate the final global and local combined model.

III. Models Building and Evaluation

Well-known regression algorithms are used to build the local prediction models: Linear Regression (LR), Random Forest Regressor (RFR), Radial Basis Function Neural Network (RBFNN), K-Nearest Neighbor Regressor (KNNR), Decision tree regression (DTR), Support Vector Regressor (SVR), Neural Network Regressor (NNR), Lasso, ElasticNet, and Ridge. In each site, the local models are trained on the local training dataset from its local dataset. We used RMSE and MAPE metrics in model evaluation, selection, and weighting strategies.

IV. Combined Model Evaluation

1) Global Combined Model Evaluation

a) Testing error: we evaluated the final global combined models in each site based on the local validation data of the site. We sent the global combined model SSE and MAPE results with the number of validation data samples to the server. Then, the server calculated the global average MAPE and RMSE of the global combined model. To calculate the global RMSE of the global combined models, each site i:

i. Compute SSE of the global combined models $\mathrm{SSE_i}(\mathrm{M^{G*}})$ and $\mathrm{SSE_i}(\mathrm{M^{G**}})$.

$$\mathrm{SSE_i}\ (\mathrm{M^{G*}})= \sum_{t=1}^{n} \left|y_t^{M^{G*}} - \hat{y}_t^{M^{G*}}\right|^2 \qquad (3.53)$$

, where t is the sample number t=1, …, n, $y_t^{M^{G*}}$ is the actual value, and $\hat{y}_t^{M^{G*}}$ is the predicted value for n data samples.

$$SSE_i\ (M^{G**}) = \sum_{t=1}^{n} \left| y_t^{M^{G**}} - \hat{y}_t^{M^{G**}} \right|^2 \qquad (3.54)$$

, where t is the sample number t=1, …, n, $y_t^{M^{G**}}$ is the actual value, and $\hat{y}_t^{M^{G**}}$ is the predicted value for n data samples.

ii. Send the evaluation results with the number of local validation data samples to the server. The server will calculate the average RMSE by dividing the sum of SSE from all sites by the number of validation data samples of all sites D, then get the square root error to get RMSE (M^{G*}) and RMSE (M^{G**}).

$$RMSE\ (M^{G*}) = \sqrt{\frac{\sum_{i=1}^{n} SSE_i(M^{G*})}{D}} \qquad (3.55)$$

$$RMSE\ (M^{G**}) = \sqrt{\frac{\sum_{i=1}^{n} SSE_i(M^{G**})}{D}} \qquad (3.56)$$

, where D is the total number of data samples from all sites, and n is the number of sites

Also, calculate the average MAPE (M^{G*}) and MAPE (M^{G**}).

$$MAPE\ (M^{G*}) = \sum_{i=1}^{n} \frac{D_i}{D} * MAPE_i(M^{G*}) \qquad (3.57)$$

$$MAPE\ (M^{G**}) = \sum_{i=1}^{n} \frac{D_i}{D} * MAPE_i(M^{G**}) \qquad (3.58)$$

, where D_i is the number of samples of site i and D is all sites' samples number.

b) **Training error:** each site evaluated the final global combined model based on its local data that used to train the local models. Then, each site sends the evaluation results SSE and MAPE to the server with the number of local data samples to calculate the average training MAPE and RMSE.

2) **Local Combined Model Evaluation:** we calculated the local combined models MAPE and RMSE based on the local validation data and compared with the best local model RMSE M_{ij*} and the best local model MAPE M_{ij**}.

V. Experiment Results and Analysis

1) Random Data Partitioning Approach:

As shown in Table 3.34, for each site, we split the datasets into two main parts. The first one is used as the local dataset used to build the local models and evaluate the received models from other sites, and the second part is used to evaluate the combined models of the global and local level modelling methods. Datasets distributions are shown in Appendix A.

Table 3.34. Datasets Partitions

Datasets	Site 1		Site 2		Site 3	
	Local data	Validation data	Local data	Validation data	Local data	Validation data
Parkinson	1402	515	1004	443	1846	665
Boston housing	160	49	94	21	140	42
Abalone	998	403	1293	589	599	284

I. Global-level Modelling Results:

The detailed results for the global-level modelling are in Appendix B. Table 3.35 shows the proposed method evaluation results for Parkinson disease (Total UPDRS) dataset. It compares the results with a technique that if each site sends the best local model MAPE and RMSE to the server instead of sending the best global average model (**Best Local Models Combination**). Besides, the proposed method results are compared with the centralised learning approach. The centralised learning approach result is slightly better than the proposed method. Table 3.36 shows the training and testing error for the proposed method and the centralised learning approach.

Table 3.35. Global-level Modelling and Centralised Learning Approach Evaluation for Parkinson Disease (Total UPDRS)

Methods	Selection metric	Weighting method	MAPE	RMSE
Global-level modelling	MAPE	Simple average	28.38	9.10
		Error-based (RMSE))	28.24	8.98
		Performance-based (Accuracy)	28.36	9.03
		Shapley value	28.27	8.97
	RMSE	Simple average	28.38	9.10
		Error-based (RMSE)	28.24	8.98
		Performance-based (accuracy)	28.36	9.03
		Shapley value	28.27	8.97
Best Local Models Combination	MAPE	Simple average	27.97	8.47
		Error-based (RMSE)	27.84	8.54
		Performance-based (accuracy)	28.04	8.46
		Shapley value	27.91	8.54
	RMSE	Simple average	27.97	8.47
		Error-based (RMSE)	27.84	8.54
		Performance-based (accuracy)	28.04	8.46
		Shapley value	27.91	8.54
Centralised Learning Approach (RFR model)			**24.24**	**7.53**

Table 3.36. Global Combined Model and Centralised Learning Approach Training and Testing Error for Parkinson Disease (Total UPDRS) Dataset

Methods	Selection metric	Weighting method	Testing error		Training error	
			MAPE	RMSE	MAPE	RMSE
Global-level modelling	MAPE	Simple average	28.38	9.10	27.41	9.72
		Error-based (RMSE)	28.24	8.98	27.53	9.61
		Performance-based (Accuracy)	28.36	9.03	27.49	9.65
		Shapley value	28.27	8.97	27.60	9.61
	RMSE	Simple average	28.38	9.10	27.41	9.72
		Error-based (RMSE)	28.24	8.98	27.53	9.61
		Performance-based (accuracy)	28.36	9.03	27.49	9.65
		Shapley value	28.27	8.97	27.60	9.61
Centralised Learning Approach			**24.24**	**7.53**	**23.53**	**8.30**

As shown in Table 3.37, the global-level modelling and centralised learning approach results for Parkinson disease (Motor UPDRS) dataset are close. In addition, it shows the global-level modelling results using MAPE selection metric are slightly better than RMSE selection metric, and the RMSE results for the weighting method using error-based approach are slightly better than the other weighting methods. Table 3.38 shows the training and testing errors for the proposed method and centralised learning approach.

Table 3.37. Global-level Modelling and Centralised Learning Approach Evaluation for Parkinson Disease (Motor UPDRS)

Methods	Selection metric	Weighting method	MAPE	RMSE
Global-level modelling	MAPE	Simple average	36.64	7.36
		Error-based (RMSE)	36.49	7.30
		Performance-based (Accuracy)	36.50	7.34
		Shapley value	36.60	7.32
	RMSE	Simple average	38.48	7.48
		Error-based (RMSE))	38.62	7.46
		Performance-based (accuracy)	38.65	7.48
		Shapley value	38.42	7.48
Best Local Models Combination	MAPE	Simple average	34.70	7.03
		Error-based (RMSE)	34.63	7.01
		Performance-based (accuracy)	34.62	7.45
		Shapley value	**34.26**	6.99
	RMSE	Simple average	34.70	7.03
		Error-based (RMSE)	34.63	7.01
		Performance-based (accuracy)	34.62	7.45
		Shapley value	**34.26**	6.99
Centralised Leanring Approach (RFR model)			35.47	**6.98**

Table 3.38. Global Combined Model and Centralised Learning Approach Training and Testing Error for Parkinson Disease (Motor UPDRS)

Methods	Selection metric	Weighting method	Testing error		Training error	
			MAPE	RMSE	MAPE	RMSE
Global-level modelling	MAPE	Simple average	36.64	7.36	27.20	6.22
		Error-based (RMSE)	36.49	7.30	27.26	6.19
		Performance-based (Accuracy)	36.50	7.34	27.12	6.24
		Shapley value	36.60	7.32	27.33	6.19
	RMSE	Simple average	38.48	7.48	32.73	7.34
		Error-based (RMSE)	38.62	7.46	33.07	7.33
		Performance-based (accuracy)	38.65	7.48	32.98	7.34
		Shapley value	38.42	7.48	32.68	7.33
Centralised Learning Approach			**35.47**	**6.98**	**24.78**	**6.15**

Table 3.39 illustrates the global combined model evaluation compared with centralised learning approach for Abalone dataset. Our method using Shapley value for model combination approach got the best RMSE and MAPE results. The training and testing error for the proposed method and the centralised learning approach is shown in Table 3.40.

Table 3.39. Global-level Modelling and Centralised Learning Approach Evaluation for Abalone Dataset

Methods	Selection metric	Weighting method	MAPE	RMSE
Global-level modelling	MAPE	Simple average	14.01	2.77
		Error-based (RMSE)	13.86	2.71
		Performance-based (Accuracy)	13.92	2.72
		Shapley value	13.90	2.71
	RMSE	Simple average	13.24	2.39
		Error-based (RMSE)	13.24	2.39
		Performance-based (accuracy)	13.35	2.37
		Shapley value	**13.14**	**2.36**
Best Local Models Combination	MAPE	Simple average	13.97	2.79
		Error-based (RMSE)	13.96	2.78
		Performance-based (accuracy)	13.86	2.75
		Shapley value	13.86	2.75
	RMSE	Simple average	13.26	2.40
		Error-based (RMSE))	13.31	2.39
		Performance-based (accuracy)	13.36	2.38
		Shapley value	13.32	2.38
Centralised Learning Approach (NNR model)			13.69	2.39

Table 3.40. Global Combined Model and Centralised Learning Approach Training and Testing Error for Abalone Dataset

Methods	Selection metric	Weighting method	Testing error		Training error	
			MAPE	RMSE	MAPE	RMSE
Global-level modelling	MAPE	Simple average	14.01	2.77	**12.30**	2.25
		Error-based (RMSE)	13.86	2.71	12.46	2.22
		Performance-based (Accuracy)	13.92	2.72	12.48	2.23
		Shapley value	13.90	2.71	12.47	2.22
	RMSE	Simple average	13.24	2.39	12.62	2.06
		Error-based (RMSE)	13.24	2.39	12.62	2.06
		Performance-based (accuracy)	13.35	2.37	12.88	**2.05**
		Shapley value	**13.14**	**2.36**	12.89	**2.05**
Centralised Learning Approach			13.69	2.39	13.28	2.12

For Boston housing dataset in Table 3.41, the proposed method is slightly better RMSE than the centralised learning approach, and Table 3.42 shows the training and testing error for the global combined model and centralised learning approach.

Table 3.41. Global-Level Modelling and Centralised Learning Approach Evaluation for Boston Housing Dataset

Methods	Selection metric	Weighting method	MAPE	RMSE
Global-level modelling	MAPE	Simple average	14.72	3.43
		Error-based (RMSE)	13.94	3.17
		Performance-based (Accuracy)	14.89	3.42
		Shapley value	13.94	3.17
	RMSE	Simple average	15.32	3.25
		Error-based (RMSE)	14.32	3.10
		Performance-based (accuracy)	14.41	3.45
		Shapley value	14.32	3.10
Best Local Models Combination	MAPE	Simple average	16.64	3.87
		Error-based (RMSE)	13.63	**3.04**
		Performance-based (accuracy)	17.90	4.29
		Shapley value	13.63	**3.04**
	RMSE	Simple average	16.64	3.87
		Error-based (RMSE)	13.63	**3.04**
		Performance-based (accuracy)	17.90	4.29
		Shapley value	13.63	**3.04**
Centralised Learning Approach (SVR model)			**11.96**	3.20

Table 3.42. Global Combined Model and Centralised Learning Approach Training and Testing Error for Boston Housing Dataset

Methods	Selection metric	Weighting method	Testing error		Training error	
			MAPE	RMSE	MAPE	RMSE
Global-level modelling	MAPE	Simple average	14.72	3.43	13.57	4.11
		Error-based (RMSE)	13.94	3.17	**11.98**	3.86
		Performance-based (Accuracy)	14.89	3.42	13.32	3.98
		Shapley value	13.94	3.17	**11.98**	3.86
	RMSE	Simple average	15.32	3.25	14.53	3.87
		Error-based (RMSE)	14.32	**3.10**	12.56	**3.80**
		Performance-based (accuracy)	14.41	3.45	15.07	4.46
		Shapley value	14.32	**3.10**	12.56	**3.80**
Centralised Learning Approach			**11.96**	3.20	17.39	5.45

In Table 3.43, we compared our method RMSE results with a proposed method RMSE result obtained from [166] for Boston housing dataset, and our proposed method outperformed the proposed method in [166].

Table 3.43. Datasets Evaluation for Global-level Modelling for Boston Housing Dataset

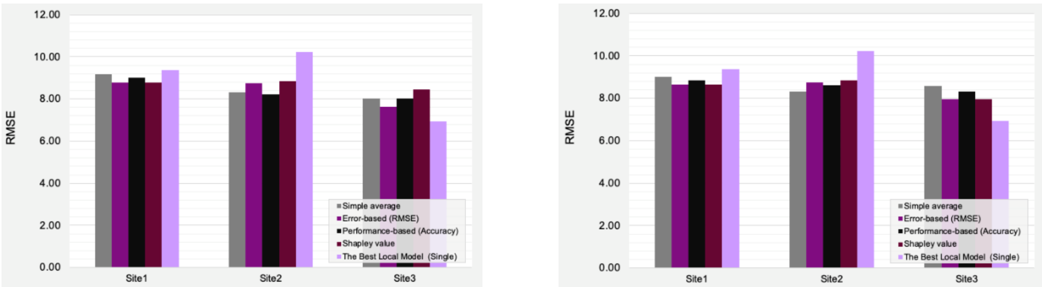
Methods	Selection metric	Weighting method	Boston Housing
Global Combined Model	MAPE	Simple average	3.43
		Error-based (RMSE)	3.17
		Performance-based (Accuracy)	3.42
		Shapley value	3.17
	RMSE	Simple average	3.25
		Error-based (RMSE)	**3.10**
		Performance-based (accuracy)	3.45
		Shapley value	**3.10**
Mandal et al. [166]			4.91

II. Local-level Modelling Results:

As shown in Table 3.44, the proposed local combined model for Parkinson disease (Total UPDRS) dataset performs better than the best local model in site 1 and site 2.

Table 3.44. Parkinson Disease (Total UPDRS) Dataset Evaluation Results for Local-Level Modelling

Method	Selection Metric	Weighting Method	Site1		Site2		Site3	
			RMSE	MAPE	RMSE	MAPE	RMSE	MAPE
Local-Level Modelling	MAPE	Simple average	9.17	22.02	8.33	30.06	8.01	29.29
		Error-based (RMSE)	8.78	22.64	8.76	31.37	7.61	28.37
		Performance-based (Accuracy)	9.02	22.25	**8.23**	**29.7**	8.02	29.92
		Shapley value	8.77	22.67	8.86	31.7	8.46	30.55
The Best Local Model (using MAPE for model selection)			9.36	26.69	10.23	36.18	**6.95**	**23.71**
Local-Level Modelling	RMSE	Simple average	9.01	**21.75**	8.33	30.06	8.57	31.69
		Error-based (RMSE)	**8.64**	22.30	8.76	31.37	7.95	29.98
		Performance-based (Accuracy)	8.83	21.94	8.61	30.88	8.32	31.21
		Shapley value	**8.64**	22.34	8.86	31.7	7.94	30.02
The Best Local Model (using RMSE for model selection)			9.36	26.69	10.23	36.18	**6.95**	**23.71**



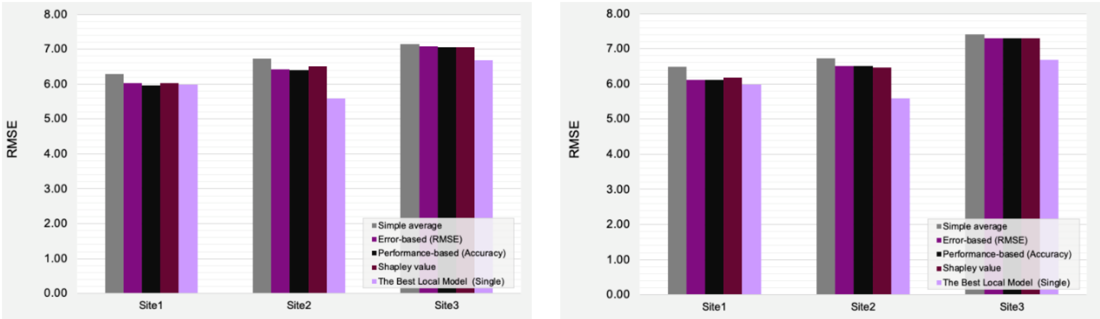
(a) (b)

Figure 3.19. Parkinson Disease (Total UPDRS) Dataset Local-Level Modelling Evaluation and The Best Models Selected By, (a) MAPE, and (b) RMSE

Table 3.45 shows the local-level modelling results for Parkinson Disease (Motor UPDRS). The best local model results are better than the proposed methods in site 2 and site 3, while in site 1, our method RMSE result is slightly better.

Table 3.45. Local-level Modelling Evaluation for Parkinson Disease (Motor UPDRS)

Method	Selection Metric	Weighting Method	Site1		Site2		Site3	
			RMSE	MAPE	RMSE	MAPE	RMSE	MAPE
Local-Level Modelling	MAPE	Simple average	6.28	27.31	6.73	35.49	7.14	39.35
		Error-based (RMSE)	6.03	26.37	6.42	33.87	7.09	39.44
		Performance-based (Accuracy)	**5.96**	**26.12**	6.40	33.43	7.06	38.94
		Shapley value	6.03	26.37	6.51	34.49	7.06	38.94
The Best Local Model (using MAPE for model selection)			5.99	**26.12**	**5.58**	**23.84**	**6.68**	**33.43**
Local-Level Modelling	RMSE	Simple average	6.48	28.38	6.72	36.05	7.42	42.29
		Error-based (RMSE)	6.12	26.99	6.52	35.07	7.31	41.64
		Performance-based (Accuracy)	6.12	26.99	6.52	35.07	7.31	41.64
		Shapley value	6.19	27.28	6.46	34.64	7.31	41.64
The Best Local Model (using RMSE for model selection)			5.99	26.14	**5.58**	**23.84**	**6.68**	**33.43**



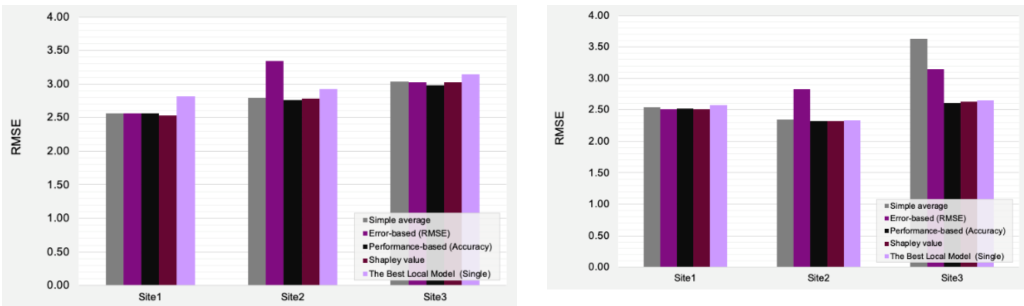
(a) (b)

Figure 3.20. Parkinson Disease (Motor UPDRS) Local-Level Modelling Evaluation And The Best Models Selected By, (a) MAPE, and (b) RMSE

For Abalone dataset, Table 3.46 shows the local combined model results are slightly better than the best local models in all sites.

Table 3.46. Local-level Modelling and Centralised Learning Approach Evaluation for Abalone dataset

Method	Selection Metric	Weighting Method	Site1		Site2		Site3	
			RMSE	MAPE	RMSE	MAPE	RMSE	MAPE
Local-Level Modelling	MAPE	Simple average	2.56	**13.11**	2.79	14.14	3.04	15.69
		Error-based (RMSE)	2.56	**13.11**	3.34	16.88	3.03	15.66
		Performance-based (Accuracy)	2.56	**13.11**	2.76	14.13	2.98	15.53
		Shapley value	2.53	13.16	2.78	14.13	3.03	15.66
The Best Local Model (using MAPE for model selection)			2.82	14.72	2.93	14.55	3.14	16.29
Local-Level Modelling	RMSE	Simple average	2.54	13.65	2.35	12.73	2.63	**14.45**
		Error-based (RMSE)	**2.51**	13.83	2.83	14.39	3.14	16.43
		Performance-based (Accuracy)	2.52	13.76	**2.32**	12.79	**2.61**	14.47
		Shapley value	**2.51**	13.77	**2.32**	12.82	2.63	**14.45**
The Best Local Model (using RMSE for model selection)			2.57	14.79	2.33	**12.55**	2.65	15.19



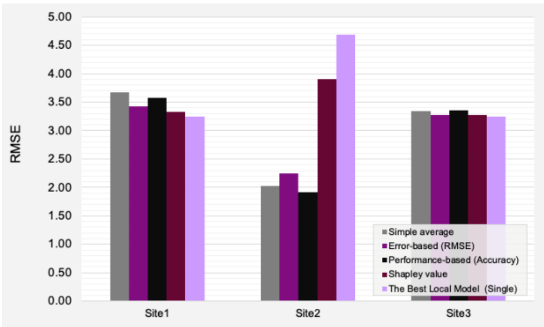
(a) (b)

Figure 3.21. Abalone Local-Level Modelling Evaluation and The Best Models Selected By, (a) MAPE, and (b) RMSE

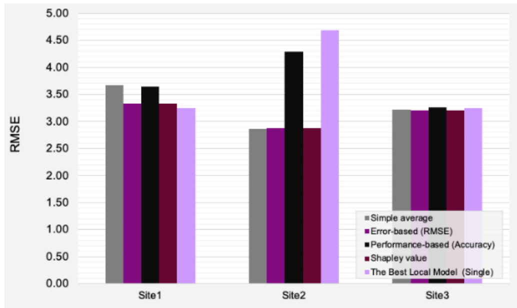
The local combined model results in Table 3.47 for Boston housing dataset show better results than the best local model in site 2 and site 3 and close results in site 1.

Table 3.47. Local-level Modelling and Centralised Learning Approach Evaluation for Boston Housing Dataset

Method	Selection Metric	Weighting Method	Site1		Site2		Site3	
			RMSE	MAPE	RMSE	MAPE	RMSE	MAPE
Local-Level Modelling	MAPE	Simple average	3.67	18.31	2.02	5.99	3.34	16.16
		Error-based (RMSE)	3.42	16.64	2.25	6.05	3.27	16.00
		Performance-based (Accuracy)	3.58	17.80	**1.91**	**5.93**	3.35	16.24
		Shapley value	3.33	15.90	3.91	11.73	3.27	16.00
The Best Local Model (using MAPE for model selection)			**3.25**	**14.01**	4.69	12.75	3.25	16.08
Local-Level Modelling	RMSE	Simple average	3.67	18.31	2.86	6.94	3.22	**15.34**
		Error-based (RMSE)	3.33	15.90	2.88	7.09	**3.21**	15.55
		Performance-based (Accuracy)	3.64	18.17	4.29	12.39	3.26	15.59
		Shapley value	3.33	15.90	2.88	7.09	**3.21**	15.55
The Best Local Model (using RMSE for model selection)			**3.25**	**14.01**	4.69	12.75	3.25	16.08



(a)



(b)

Figure 3.22. Boston Housing Local-Level Modelling Evaluation and The Best Models Selected By, (a) MAPE, and (b) RMSE

2) Non-random data partitioning Approach:

We partitioned the Parkinson disease dataset by patient age and Boston housing dataset by crime rate attribute to simulate that each data comes from different regions. Table 3.48 shows our partition strategy for the datasets between the sites, and Table 3.49 shows the datasets partitions size. The local data is used to build the local models in each site and evaluate the received models from other sites. The second partition is used to evaluate the combined models of the global and local level modelling methods. Datasets distributions are illustrated in Appendix A.

Table 3.48. Datasets Partitioning Scenarios

Datasets	Site1	Site2	Site3
Parkinson	Age: less than or equal to 60	Age: bigger than 60 and less than or equal to 70	Age: bigger than 70
Boston housing	Crime rate: less than 0.1	Crime rate: bigger than or equal to 0.1 and less than 0.99	Crime rate: bigger than 1

Table 3.49. Dataset Partitions

Datasets	Site 1		Site 2		Site 3	
	Local data	Validation data	Local data	Validation data	Local data	Validation data
Parkinson	1457	576	1460	511	1416	455
Boston housing	112	40	130	50	133	41

I. Global-level Modelling Results:

The detailed results for the global level modelling are in Appendix B. Table 3.50 compares the global combined model with the centralised learning approach for Parkinson disease (Total UPDRS) dataset. RMSE results for all methods are close, and MAPE of our method using Shapley value for model weighting is slightly better than other methods. The training and testing error results are shown in Table 3.51.

Table 3.50. Global-Level Modelling and Centralised Learning Approach Evaluation for Parkinson Disease (Total UPDRS) Dataset

Methods	Selection metric	Weighting method	MAPE	RMSE
Global-level modelling	MAPE	Simple average	25.18	8.20
		Error-based (RMSE)	25.26	8.20
		Performance-based (Accuracy)	25.37	8.21
		Shapley value	25.09	8.19
	RMSE	Simple average	24.94	8.44
		Error-based (RMSE)	24.75	8.46
		Performance-based (accuracy)	25.21	8.42
		Shapley value	**24.71**	8.46
Best Local Models Combination	MAPE	Simple average	27.12	8.48
		Error-based (RMSE)	27.23	8.49
		Performance-based (accuracy)	27.81	8.55
		Shapley value	27.12	8.49
	RMSE	Simple average	27.12	8.48
		Error-based (RMSE)	27.23	8.49
		Performance-based (accuracy)	27.81	8.55
		Shapley value	27.12	8.49
Centralised Learning Approach (LR model)			25.22	**8.17**

Table 3.51. Global Combined Model and Centralised Learning Approach Training and Testing Error for Parkinson Disease (Total UPDRS) Dataset

Methods	Selection metric	Weighting method	Testing error		Training error	
			MAPE	RMSE	MAPE	RMSE
Global-level modelling	MAPE	Simple average	25.18	8.20	34.18	10.42
		Error-based (RMSE)	25.26	8.20	34.14	10.44
		Performance-based (Accuracy)	25.37	8.21	**33.43**	**10.37**
		Shapley value	25.09	8.19	34.21	10.40
	RMSE	Simple average	24.94	8.44	41.69	11.58
		Error-based (RMSE)	24.75	8.46	42.13	11.56
		Performance-based (accuracy)	25.21	8.42	41.13	11.59
		Shapley value	**24.71**	8.46	42.18	11.56
Centralised Learning Approach			25.22	**8.17**	40.04	11.13

For Parkinson disease (Motor UPDRS) dataset, our global combined model RMSE result in Table 3.52 is slightly better than the centralised learning approach, while MAPE result for the centralised learning approach is the best result. The training and testing error for the proposed method and the centralised learning approach is illustrated in Table 3.53.

Table 3.52. Global-Level Modelling and Centralised Learning Approach Evaluation for Parkinson Disease (Motor UPDRS) Dataset

Methods	Selection metric	Weighting method	MAPE	RMSE
Global-level modelling	MAPE	Simple average	34.1	7.2
		Error-based (RMSE)	34.5	7.2
		Performance-based (Accuracy)	33.4	**7.1**
		Shapley value	34.5	7.2
	RMSE	Simple average	37.4	7.5
		Error-based (RMSE)	37.7	7.5
		Performance-based (accuracy)	37.1	7.4
		Shapley value	37.8	7.5
Best Local Models Combination	MAPE	Simple average	35.6	7.5
		Error-based (RMSE)	36.4	7.6
		Performance-based (accuracy)	36.6	7.6
		Shapley value	36.2	7.6
	RMSE	Simple average	35.6	7.5
		Error-based (RMSE)	36.4	7.6
		Performance-based (accuracy)	36.6	7.6
		Shapley value	36.2	7.6
Centralised Learning Approach (LR model)			**30.68**	7.3

Table 3.53. Global Combined Model and Centralised Learning Approach Training and Testing Error for Parkinson Disease (Motor UPDRS) Dataset

Methods	Selection metric	Weighting method	Testing error		Training error	
			MAPE	RMSE	MAPE	RMSE
Global-level modelling	MAPE	Simple average	34.1	7.2	38.4	7.7
		Error-based (RMSE)	34.5	7.2	38.6	7.7
		Performance-based (Accuracy)	33.4	**7.1**	37.5	7.7
		Shapley value	34.5	7.2	38.6	7.7
	RMSE	Simple average	37.4	7.5	45.7	8.5
		Error-based (RMSE)	37.7	7.5	46.1	8.5
		Performance-based (accuracy)	37.1	7.4	45.5	8.5
		Shapley value	37.8	7.5	46.1	8.5
Centralised Learning Approach			**30.68**	7.3	**34.4**	**6.9**

Table 3.54 shows that the centralised learning approach is slightly better than our proposed method for Boston housing dataset. Table 3.55 illustrates the training and testing error results for the proposed method and the centralised learning approach.

Table 3.54. Global-level Modelling and Centralised Learning Approach Evaluation for Boston Housing Dataset

Methods	Selection metric	Weighting method	MAPE	RMSE
Global-level modelling	MAPE	Simple average	15.92	3.68
		Error-based (RMSE)	16.02	4.19
		Performance-based (Accuracy)	16.02	4.19
		Shapley value	16.20	3.67
	RMSE	Simple average	20.35	3.77
		Error-based (RMSE)	20.35	3.77
		Performance-based (accuracy)	20.28	3.76
		Shapley value	18.31	4.02
Best Local Models Combination	MAPE	Simple average	20.35	3.77
		Error-based (RMSE)	19.01	4.13
		Performance-based (accuracy)	20.88	3.83
		Shapley value	21.41	3.97
	RMSE	Simple average	23.76	4.35
		Error-based (RMSE)	22.73	4.63
		Performance-based (accuracy)	21.15	4.34
		Shapley value	25.65	4.73
Centralised Learning Approach (RFR model)			**13.31**	**3.38**

Table 3.55. Global Combined Model and Centralised Learning Approach Training and Testing Error for Boston Housing Dataset

Methods	Selection metric	Weighting method	Testing error		Training error	
			MAPE	RMSE	MAPE	RMSE
Global-level modelling	MAPE	Simple average	15.92	3.68	13.29	4.26
		Error-based (RMSE)	16.02	4.19	14.32	4.96
		Performance-based (Accuracy)	16.02	4.19	14.32	4.96
		Shapley value	16.20	3.67	13.33	4.24
	RMSE	Simple average	20.35	3.77	15.04	4.08

		Error-based (RMSE)	20.35	3.77	15.04	4.08
		Performance-based (accuracy)	20.28	3.76	14.96	4.07
		Shapley value	18.31	4.02	13.92	4.52
Centralised Learning Approach			**13.31**	**3.38**	**11.51**	**3.27**

For Boston housing dataset, Table 3.56 shows that our proposed method RMSE result using MAPE selection and model combination using Shapley value is better than the proposed approach in [166]. Besides, all model weighting methods got close RMSE results.

Table 3.56. Datasets Evaluation for Global-level Modelling for Boston housing dataset

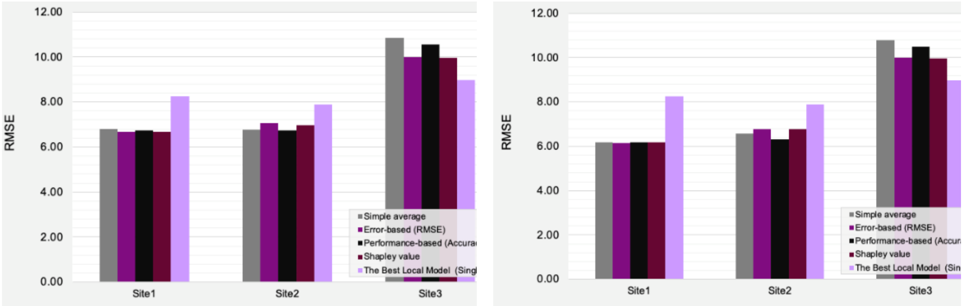
Methods	Selection metric	Weighting method	Boston Housing
Global Combined Model	MAPE	Simple average	3.68
		Error-based (RMSE)	4.19
		Performance-based (Accuracy)	4.19
		Shapley value	**3.67**
	RMSE	Simple average	3.77
		Error-based (RMSE)	3.77
		Performance-based (accuracy)	3.76
		Shapley value	4.02
Mandal et al. [166]			4.91

II. Local-level Modelling Results:

In Table 3.57 for Parkinson disease (Total UPDRS) dataset, the local combined model results in site 1 and site 2 are better than the best local model.

Table 3.57. Local-level Modelling and Centralised Learning Approach Evaluation for Parkinson Disease (Total UPDRS) dataset

Method	Selection Metric	Weighting Method	Site1		Site2		Site3	
			RMSE	MAPE	RMSE	MAPE	RMSE	MAPE
Local-Level Modelling	MAPE	Simple average	6.79	23.37	6.78	30.57	10.84	24.05
		Error-based (RMSE)	6.67	22.38	7.07	31.72	9.99	22.83
		Performance-based (Accuracy)	6.75	22.24	6.74	30.14	10.54	23.66
		Shapley value	6.67	22.30	6.98	31.29	9.95	22.78
The Best Local Model (using MAPE for model selection)			8.26	22.56	7.89	31.77	**8.96**	**20.09**
Local-Level Modelling	RMSE	Simple average	6.19	22.21	6.56	28.95	10.79	24.05
		Error-based (RMSE)	**6.16**	21.28	6.78	29.98	9.98	22.83
		Performance-based (Accuracy)	6.19	**21.02**	**6.32**	**27.67**	10.50	23.67
		Shapley value	6.17	21.27	6.78	29.98	9.96	22.82
The Best Local Model (using RMSE for model selection)			8.25	22.56	7.89	31.77	**8.96**	**20.09**



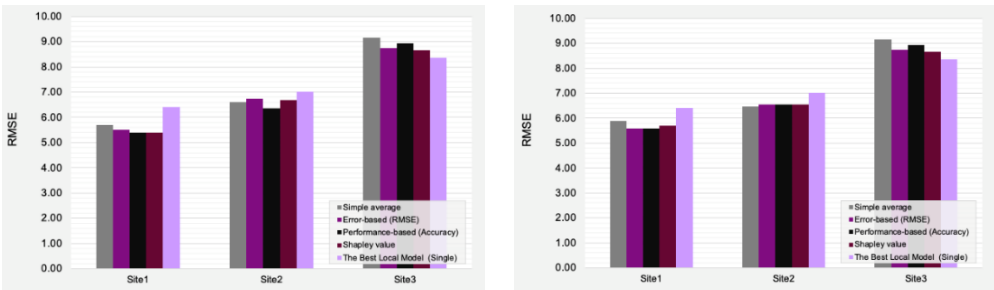
(a) (b)

Figure 3.23. Parkinson Disease (Total UPDRS) Dataset Local-Level Modelling Evaluation and The Best Models Selected By, (a) MAPE, and (b) RMSE

In Table 3.58, our proposed method results in site 1 and site 2 are better than the best local model RMSE results for Parkinson disease (Motor UPDRS) dataset, while the best local model is slightly better MAPE than our method in site 2 and site 3.

Table 3.58. Local-Level Modelling and Centralised Learning Approach Evaluation for Parkinson Disease (Motor UPDRS) Dataset

Method	Selection Metric	Weighting Method	Site1		Site2		Site3	
			RMSE	MAPE	RMSE	MAPE	RMSE	MAPE
Local-Level Modelling	MAPE	Simple average	5.7	31.96	6.59	43.26	9.16	25.97
		Error-based (RMSE)	5.5	29.13	6.73	44.01	8.73	24.94
		Performance-based (Accuracy)	**5.4**	27.52	**6.36**	40.88	8.94	25.48
		Shapley value	**5.4**	28.97	6.68	43.71	8.67	24.78
The Best Local Model (using MAPE for model selection)			6.4	**23.17**	7.02	**40.64**	**8.36**	**22.63**
Local-Level Modelling	RMSE	Simple average	5.9	33.88	6.46	42.66	9.16	25.97
		Error-based (RMSE)	5.6	29.17	6.56	43.17	8.73	24.94
		Performance-based (Accuracy)	5.6	29.17	6.56	43.17	8.94	25.48
		Shapley value	5.7	30.71	6.56	43.04	8.67	24.79
The Best Local Model (using RMSE for model selection)			6.4	**23.17**	7.02	**40.64**	**8.36**	**22.63**



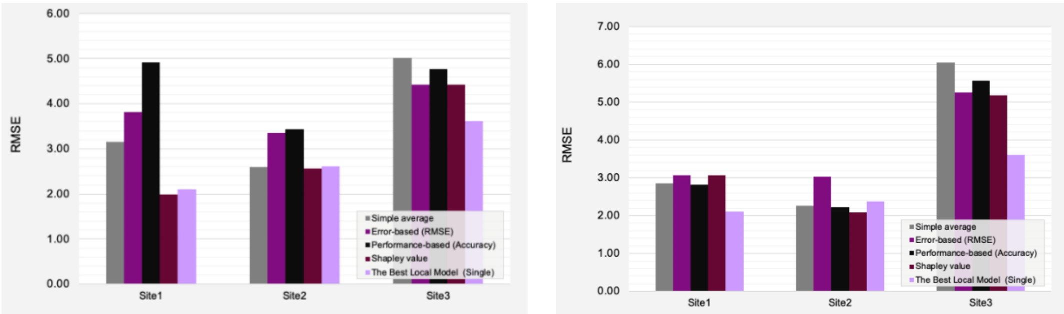
(a) (b)

Figure 3.24. Parkinson Disease (Motor UPDRS) Dataset Local-Level Modelling Evaluation And The Best Models Selected By, (a) MAPE, and (b) RMSE

For Boston housing dataset in Table 3.59, we found that the local combined model got better RMSE results in site 1 and site 2 using Shapley value for the model weighting method.

Table 3.59. Local-level Modelling and Centralised Learning Approach Evaluation for Boston Housing dataset

Method	Selection Metric	Weighting Method	Site1		Site2		Site3	
			RMSE	MAPE	RMSE	MAPE	RMSE	MAPE
Local-Level Modelling	MAPE	Simple average	3.16	6.86	2.60	11.45	5.02	30.07
		Error-based (RMSE)	3.82	10.60	3.35	14.24	4.42	28.32
		Performance-based (Accuracy)	4.91	12.53	3.44	14.15	4.77	29.74
		Shapley value	**1.98**	**5.43**	2.56	11.29	4.43	29.13
The Best Local Model (using MAPE for model selection)			2.11	5.99	2.62	11.42	**3.61**	**24.04**
Local-Level Modelling	RMSE	Simple average	2.86	9.32	2.26	9.98	6.05	47.41
		Error-based (RMSE)	3.07	8.50	3.02	12.18	5.25	41.74
		Performance-based (Accuracy)	2.81	9.17	2.23	9.93	5.56	43.44
		Shapley value	3.07	8.50	**2.09**	9.49	5.18	40.86
The Best Local Model (using RMSE for model selection)			2.11	5.99	2.38	**9.32**	**3.61**	**24.04**



(a) (b)

Figure 3.25. Boston Housing Local-level Modelling evaluation and the best models selected by, (a) MAPE, and (b) RMSE

3.4.4 Discussion and Evaluation

We evaluated the global combined model prediction performance and compared it with a research study and a centralised learning approach that moves all distributed data to a centralised database. Our proposed method for the randomly partitioned datasets is slightly better than the centralised learning method in Abalone and Boston housing datasets. In Parkinson disease dataset, the centralised learning approach performed slightly better. In addition, the model weighting method using Shapley values is slightly better than other methods. For the non-randomly partitioned datasets, our proposed global combined model is slightly better than the centralised learning approach in Parkinson disease (Motor UPDRS) dataset, and close results in Parkinson disease (Total UPDRS) and Boston Housing datasets. We found that the proposed method could perform comparable to or better than the centralised learning method, overcoming the centralised learning issues and overheads and preserving data privacy. We saved the cost and time of data transformation from one site to another or a central location. We developed the combined models with fewer communication rounds, and minimal information exchanged between distributed sites. In the local level modelling approach, we proved that the distributed sites could utilise other sites models to improve the prediction performance with fewer communication rounds and without sharing data between the sites to preserve data privacy. In addition, the proposed approach could be applied to solve issues related to large data, such as memory limitation and huge data transformation costs. However, as discussed in section 3.3.4, there is a possibility for malicious attacks on the trained models and we will consider this issue in future research.

3.5 SUMMARY

This chapter presented our proposed global and local level modelling using the linear combination method. We simulated distributed sites using different dataset partitioning scenarios and developed local and global combined models using different models selection and combining approaches. The final model weights used in the global combined model are calculated based on average accuracy using all sites datasets. This will contribute to developing an unbiased and generalised global model for all distributed sites. We evaluated the performance of the proposed local and global level modelling methods on different classification and regression datasets with different models selection and combination strategies. It showed its efficiency and applicability in the distributed environment. The global-level modelling for classification datasets got similar results with the centralised learning method in most datasets, and the single best model of the proposed method got similar results with the centralised learning method in several datasets. The results of the global combined model for regression datasets are slightly better or close to the centralised learning approach. In the local-level modelling approach, all sites are utilised from other sites models and improved the prediction performance without requiring data sharing, and there are no poor results obtained by models combining methods.

We used simple model evaluation and selection strategies instead of complicated methods that need more communication and computational cost and less information exchange than FL. We developed a global combined model without moving the local data to another site or a central server. The proposed method did not expose the data resource and hence preserved data privacy. We developed a combined local model for each site by utilising the learning outcomes from other local data resources. As a result, we saved the cost and time of data transformation from one site to another, improved computation effectiveness and efficiency, and preserved data privacy. We developed well-generalised combined models by weighting the final model weights used in global combined models based on average accuracy using all sites datasets. Part of this chapter is published in [22].

Chapter 4

Nonlinear Model Combination Approach

4.1 CHAPTER OVERVIEW

This chapter presents our proposed decentralised machine learning method to build global and local combined models using a nonlinear model combination approach. First, section 4.2 views our contribution and aims to develop the combined models using the decentralised learning approach. Then, the proposed method for classification with its experiment results and discussion are shown in section 4.3 and for regression algorithms in section 4.4. Finally, section 4.5 presents the chapter summary.

4.2 INTRODUCTION

We propose a decentralised machine learning method based on the nonlinear model combination approach, which allows distributed sites to build combined prediction models at global and local levels without sharing or disclosing distributed data resources. This method is proposed to address several issues such as data privacy, data transfer restrictions, and communication and computation costs. We avoid exchanging lots of intermediate information, using a centralised machine learning method, or using a central site for iterative learning process to minimise communication or computation overheads. The proposed method restricts the exchanged information between sites with only the local models learned from the local data and therefore shares minimal information. We aim to build a global combined model derived from local learning outcomes and an optimal local combined model by utilising learning outcomes from other sites data resources and its local data.

Furthermore, using a nonlinear combination approach of heterogeneous models to utilise and combine the selected models from distributed sites. The local base models between distributed sites are diverse, and it is sensible to consider using them in combination to overcome individual models limitations and improve the prediction performance. Moreover, with much less information sharing and without the iterative computing process between local sites and the central server, the proposed approach can achieve strong performance, which often is as good as centralised machine learning. Therefore, the proposed method leads toward to a simpler and new direction for decentralised privacy-preserving machine learning.

4.3 PROPOSED METHOD FOR CLASSIFICATION ALGORITHMS

We develop a decentralised version of nonlinear combination approach for distributed and private data resources. The proposed decentralised learning and nonlinear combination methods address individual model limitations by utilising distributed data to develop global and local combined prediction models without data transformation between distributed sites to preserve data privacy. The exchanged information between distributed sites is only the models, accuracy/F-measurement, and data size. We use different classification algorithms to develop heterogeneous prediction models. We develop a global combined model at the central server by nonlinearly combining the best average accuracy/F-measure model from each distributed site. Also, build a local combined model in each site by utilising and nonlinearly combining the best local models from the other sites. For this purpose, the related symbols and definitions used in our approach are first introduced in Table 4.1.

Table 4.1. Models Names and Descriptions

Model	Notation	Meaning
Local Data	D_i	Local data in site i that used for local models learning, evaluate the received models from other sites, and local meta-models learning.

Local Model	M_{ij}	A local model that developed in site i using j classification algorithm
Best local model	M_{ij*} and M_{ij**}	The best local model in site i, which have the best accuracy using j* classification algorithm, and the best F-measure using j** classification algorithm, respectively
Received Model	$M_{i'j}$	Model in site i that received from another site i'
Selected Model	$M_{i'j*}$ and $M_{i'j**}$	Selected model in site i that selected from other sites i' which is better than the best local model accuracy M_{ij*} and better than the local model F-measure M_{ij**} respectively
Best average accuracy and F-measure model	M_{ij*}^{G} and M_{ij**}^{G}	The best average accuracy model and the best average F-measure model in site i after global evaluation will nonlinearly combined to build the global combined model
List of the best models	M_{G-Acc} and M_{G-F}	List of the best average accuracy models and the best average F-measure models from all sites
Meta-model	$M_{ij}^{G-Meta-Acc}$ and $M_{ij}^{G-Meta-F}$	Meta-model which developed from M_{G-Acc} models list outputs and from M_{G-F} models list outputs in site i using j learning algorithms respectively
Local test data	D_{i}^{Ts}	Local test data in site i used for meta-models evaluation
The best meta-model	$M_{ij*}^{Best\ G-Meta-Acc}$ and $M_{ij**}^{Best\ G-Meta-F}$	The best meta-model in site i from $M_{ij}^{G-Meta-Acc}$ models and $M_{ij}^{G-Meta-F}$ models, respectively
The global combined model	$M_{G-META-Acc}$ and $M_{G-META-F}$	The global combined model at the server that combine the best meta-models $M_{ij*}^{Best\ G-Meta-Acc}$ and the best meta-models $M_{ij**}^{Best\ G-Meta-F}$, respectively
List of the best accuracy models	M_{Acc}	List of the best local model accuracy M_{ij*} and the selected models of the best accuracy from other sites $M_{i'j*}$ that will nonlinearly be combined
List of the best F-measure models	M_{F}	List of the best local model F-measure M_{ij**} and the selected models of the best F-measure from other sites $M_{i'j**}$ that will nonlinearly be combined
Local Meta-model	$M_{ij}^{L-Meta-Acc}$ and $M_{ij}^{L-Meta-F}$	Meta-model in site i which developed from M_{Acc} models list outputs and M_{F} models list outputs, respectively

The best meta-model (Local combined Model)	$M_{ij*}^{Best\ L-Meta-Acc}$ and $M_{ij**}^{Best\ L-Meta-F}$	The best meta-model in site i and selected form $M_{ij}^{L-Meta-Acc}$ models and $M_{ij}^{L-Meta-F}$, respectively

4.3.1. Global-level Modelling Approach

We aim to build a global combined model at the central server by nonlinearly combining the best average accuracy/F-measure model from each distributed site. First, each site builds heterogeneous local models on their local data using different classification algorithms. Our aim for developing several models not only an individual model, because a model may perform well in a dataset and not in other datasets and building a single model may not fit well for particular datasets. So, we develop different models to find a good model for all sites. Second, each site selects its best local model using the 10-fold cross-validation technique. Next, each site shares its local models with other sites for general evaluation to find the best models as candidates for selection and combination to build a global model for all sites. Then, when a site receives local models from other sites, it first evaluates the models based on its own local data, then sends the models back with its evaluation results to the sites and with the local data size that used for evaluation. Each site should not get more information about the received model during evaluation than the evaluation results of its local data, which will preserve the models privacy. Then, each site will receive its local models evaluation results from other sites and calculate the average accuracy/F-measure of its local models. Finally, combine the best average accuracy/F-measure model from each site using a nonlinear combination method to develop the global combined model. The proposed approach is illustrated in Figure 4.1.

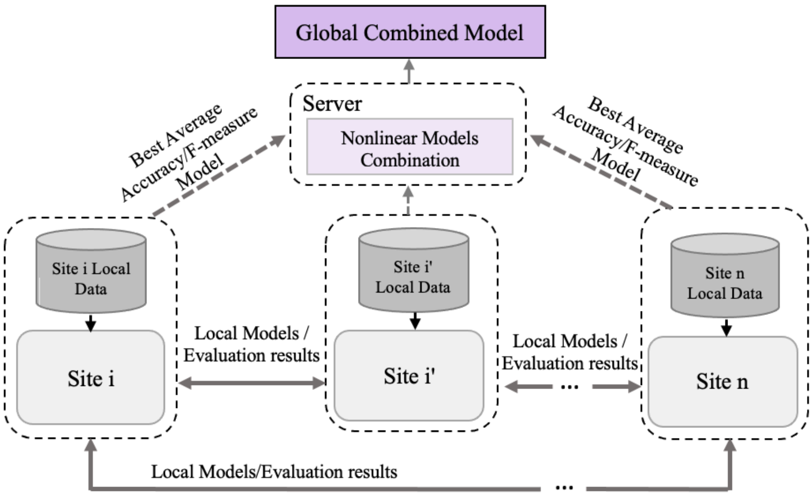


Figure 4.1. The Proposed Method for Global-level Modelling Approach

Figure 4.2 shows steps 1 and 2 for local model building and best average accuracy and F-measure models calculations, and it is similar to chapter 3 steps in section 3.3.1. The following steps implement the above idea:

1) Each site S_i , where $i = 1,2, \ldots, n$:
   iii. Apply different j learning algorithms, where $j = 1, 2, \ldots, m$ to build local models M_{ij}.
   iv. Use 10-fold cross-validation results to evaluate the local models M_{ij} based on the local data D_i and calculates the accuracy Acc (M_{ij}) and F-measure F (M_{ij}) using confusion matrix.

$$\mathrm{Acc}\left(M_{ij}\right) = TP + TN/(TP + TN + FP + FN) \quad (4.1)$$

$$F\left(M_{ij}\right) = 2TP/(2TP + FP + FN) \quad (4.2)$$

, where TP is true positive, TN is true negative, FP is false positive, and FN is false negative examples.

2) After building the local models, each site sends its local models M_{ij} to other sites for global evaluation to see which model performs best globally, and select the best average models as follows:
   1. Each site i will receive models from other sites $M_{i'j}$, then start to evaluate these models over its local dataset D_i and calculate the accuracy $\mathrm{Acc}(M_{i'j})$ and $F(M_{i'j})$, where $i' = 1, \ldots, i-1, i+1, \ldots, n)$ and $j = 1, 2, \ldots, m$
   2. Send the evaluated models back to the sites with the evaluation results.

3. Each site:
   a) Receive the evaluation results $\mathrm{Acc}\,(M_{ij})$ and $F\,(M_{ij})$ of its local models M_{ij} from other sites with the number of data samples that used for evaluation.
   b) Calculate the average accuracy and average F-measure for each local model.

$$\mathrm{Acc}\,(M_{ij}) = \sum_{k=1}^{n} \frac{D_k}{D} * \mathrm{Acc}\,(M_{ij}) \text{ in } S_k \qquad (4.3)$$

$$F\,(M_{ij}) = \sum_{k=1}^{n} \frac{D_k}{D} * F\,(M_{ij}) \text{ in } S_k \qquad (4.4)$$

, where k is sites number, D_k is the number of samples of site k, and D is all sites' data samples number

   c) Select the best average accuracy M^G_{ij*} and the best average F-measure M^G_{ij**}.

3) Each site sends its best average accuracy model M^G_{ij*} and best average F-measure model M^G_{ij**} to the other sites to start the nonlinear combination approach. We implement the nonlinear combination method in each site instead of the server because there is no data at the server, and the sites will not share their local data with the server. We perform two nonlinear combination scenarios to build the global combined models; the first scenario is for the best average accuracy model M^G_{ij*} and the other one is for the best average F-measure model M^G_{ij**}. Figures 4.3 and 4.4 show the nonlinear combination scenario for the best average accuracy models. The nonlinear combination method is as follows:

- In each site i:
  a) We have a list of the best average accuracy models M_{G-Acc} from all n sites, $M_{G-Acc}= \{M^G_{ij*}, M^G_{i'j*}, \dots, M^G_{nj*}\}$, and a list of the best average F-measure models M_{G-F} from all n sites, $M_{G-F}= \{M^G_{ij**}, M^G_{i'j**}, \dots, M^G_{nj**}\}$.
  b) Apply each model over the local dataset D_i to generate two meta-datasets $\{x'_t, y_t\}$ and $\{x''_t, y_t\}$, where

$$x'_t = \{M^G_{ij*}(x_t), M^G_{i'j*}(x_t), \dots, M^G_{nj*}(x_t)\} \qquad (4.5)$$

$$x''_t = \{M^G_{ij**}(x_t), M^G_{i'j**}(x_t), \dots, M^G_{nj**}(x_t)\} \qquad (4.6)$$

  , where t = 1, …, p, p is the local dataset samples number, and y_t is the predicted value. The prediction results by each model as independent variables and their corresponding actual prediction result as the dependent variable.

c) Build several meta-models by applying j different learning algorithms on the generated meta-datasets, the meta-models $M_{ij}^{G-Meta-Acc}$ are developed from the meta-dataset that generated from M_{G-Acc} model list outputs, and $M_{ij}^{G-Meta-F}$ are developed from the meta-dataset that generated from M_{G-F} model list outputs, j=1, …, m. The meta-models inputs are M_{G-Acc} or M_{G-F} models predictions, and the output is the actual predicted value.

d) After building the meta-models, select the best meta-models $M_{ij*}^{Best\ G-Meta-Acc}$ and $M_{ij**}^{Best\ G-Meta-F}$ as follow:

   i. Apply the models in M_{G-Acc} over D_i^{Ts} and use the models outputs as input features to meta-models $M_{ij}^{G-Meta-Acc}$. Also, apply the models in M_{G-F} over D_i^{Ts} and use the models outputs as input features to meta-models $M_{ij}^{G-Meta\_F}$.

   ii. Apply the meta-models on the input data and predict the result.

   iii. Evaluate the meta-models using the local test data, then select the best meta-model $M_{ij*}^{Best\ G-Meta-Acc}$ and $M_{ij**}^{Best\ G-Meta-F}$.

e) Send the best meta-models to the server.

4) As shown in Figure 4.4, the server receives the best meta-models from each site and combines the meta-models $M_{ij*}^{Best\ G-Meta-Acc}$ to build the global model $M_{G-META-Acc}$, and the meta-models $M_{ij**}^{Best\ G-Meta-F}$ to build the global model $M_{G-META-F}$.

5) The final global combined models $M_{G-META-Acc}$ and $M_{G-META-F}$ is meta-models combination and used to predict a result y of x, the predicted result y is the meta-models average results.

$$y(x) = M_{G-META-Acc}[M_{ij*}^{Best\ G-Meta-Acc}(x), \dots, M_{nj*}^{Best\ G-Meta-Acc}(x)] \quad (4.7)$$

$$y(x) = M_{G-META-F}\ [M_{ij**}^{Best\ G-Meta-F}(x), \dots, M_{nj**}^{Best\ G-Meta-F}(x)] \quad (4.8)$$

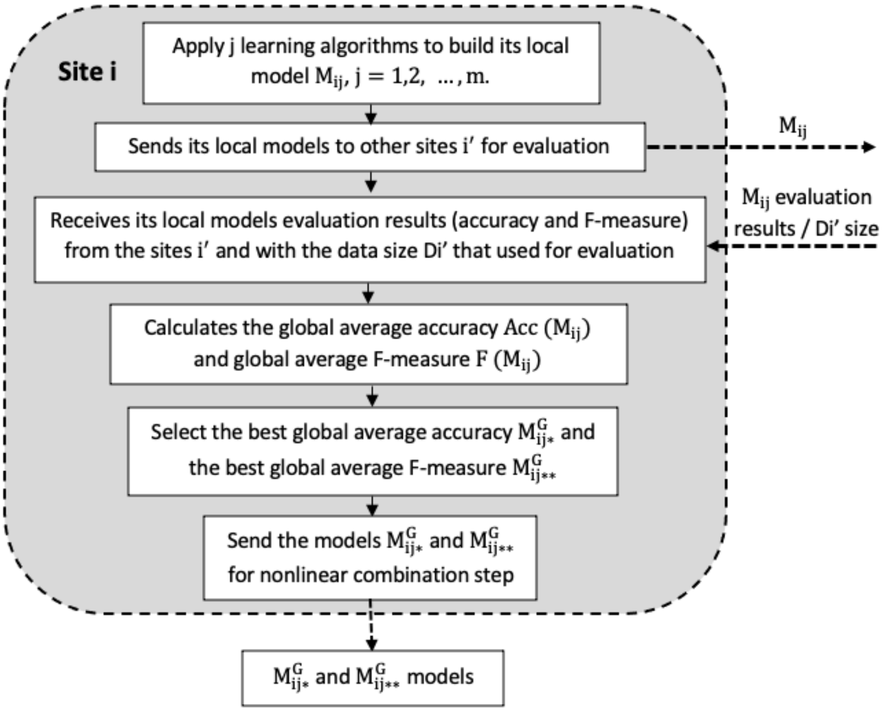


Figure 4.2. Local Model Building and Best Average Models Calculations

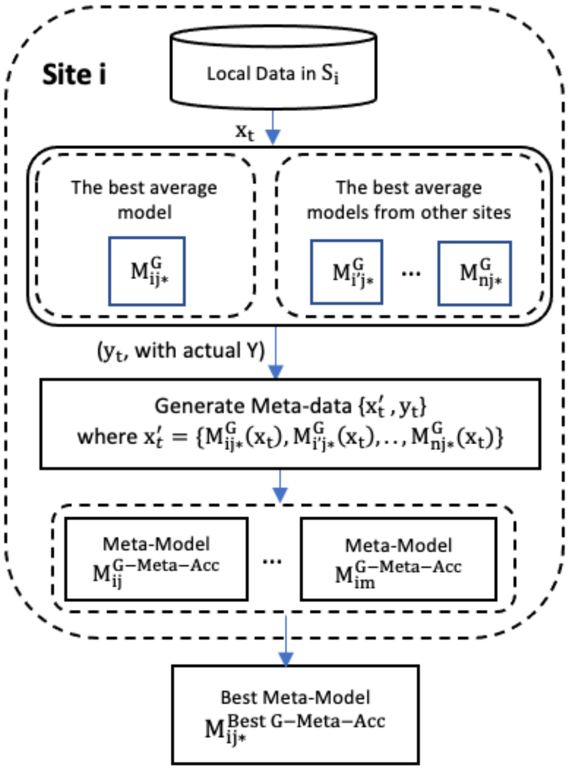


Figure 4.3. Meta-learning Method in Global-level Modelling

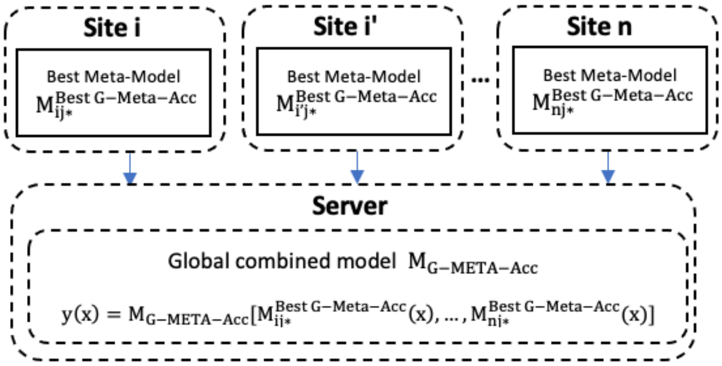


Figure 4.4. Global Combined Model at The Server

4.3.2. Local-level Modelling Approach

The basic idea for each site is to find the best local combined model by utilising other sites local models. Thus, there is no data sharing or transformation, and the only information exchanged between distributed sites are local models and the evaluation results; such a method does not disclose the data resource and preserves data privacy. Each site tries to find the best local combined model by utilising the best local models from the other sites. Then, the sites combine the selected best models using the nonlinear combination method.

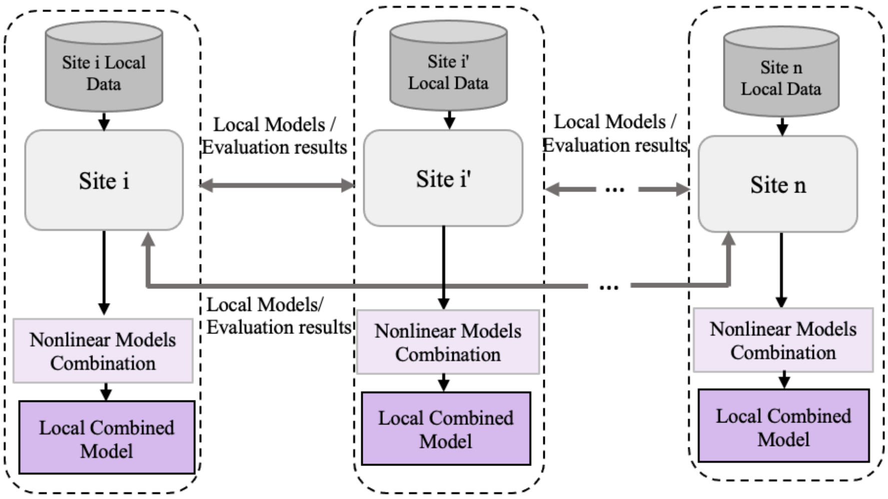


Figure 4.5. Local-level Modelling Approach

The following steps implement local-level modelling:

1) When a site i receives models from other sites $M_{i'j}$, evaluate these models over its local dataset D_i, and calculate the accuracy $Acc(M_{i'j})$ and F-

measure $F(M_{i'j})$:

a) Compare the accuracy $Acc(M_{i'j})$ and F-measure $F(M_{i'j})$ of the received models with its best local model accuracy $Acc(M_{ij*})$ and best local model F-measure $F(M_{ij**})$.

b) Select the best model accuracy $M_{i'j*}$ and the best model F-measure $M_{i'j**}$ from each site as follows:

- **Method L1:**
  - Select the best model accuracy $M_{i'j*}$ from site i' if:

    $$Acc(M_{i'j}) >= Acc(M_{ij*})).$$

  - Select the best model F-measure $M_{i'j**}$ from site i' if:

    $$F(M_{i'j}) >= F(M_{ij**}).$$

  This method utilises the best models learned from other data resources to build an accurate local combined model.

- **Method L2:** Select the best model from each site even if the selected model not performed better than the best local model. We proposed this method if the best local model result is better than the received models, but there is little difference between the best local model and the received models results. We apply this selection method for the best model accuracy and F-measure.

2) In each site i, we apply two nonlinear combination methods to build two local combined models. The first local combined model is by combining the best local model accuracy M_{ij*} and the selected models from other sites $M_{i'j*}$. The other local combined model is by combining the best local model F-measure M_{ij**} and the selected models from other sites $M_{i'j**}$. The nonlinear combination scenario for the best accuracy models as follows:

   a) Site i has a models list of the best accuracy M_{Acc}, where M_{Acc} is the list of the best local model accuracy and the selected models of the best accuracy from other sites $M_{Acc} = \{M_{ij*}, M_{i'j*}, \dots, M_{nj*}\}$, where I = 1, 2, …, n. Also, the site i has a models list of the best F-measure M_F, where M_F is list of the best local model F-measure and the selected models of the best F-measure from other sites, $M_F = M_{ij**}, M_{i'j**}, \ldots, M_{nj**}\}$.

b) Apply each model over the local dataset D_I to generate two meta-data $\{x'_t, y_t\}$ and $\{x''_t, y_t\}$, where

$$x'_t = \{M_{ij*}(x_t), M_{i'j*}(x_t), \ldots., M_{nj*}(x_t)\} \quad (4.9)$$

$$x''_t = \{M_{ij**}(x_t), M_{i'j**}(x_t), \ldots., M_{nj**}(x_t)\} \quad (4.10)$$

, where t = 1, …, p, p is the local dataset samples number, and y_t is the predicted value. The prediction results by each model as independent variables and their corresponding actual prediction result as the dependent variable.

c) Build several meta-models by applying j different learning algorithms on the generated meta-datasets, the meta-models $M_{ij}^{L-Meta-Acc}$ are developed from the meta-dataset that generated from M_{Acc} models outputs, and $M_{ij}^{L-Meta-F}$ are developed from the meta-data that generated from M_F models outputs, j=1, …, m. The meta-models inputs are M_{Acc} or M_F models predictions, and the output is the actual predicted value.

d) After building the meta-models, select the best meta-models $M_{ij*}^{Best\ L-Meta-Acc}$ and $M_{ij**}^{Best\ L-Meta-F}$ as follows:

  1. Apply the models M_{Acc} over D_i^{Ts} and use the models outputs as input features to meta-models $M_{ij}^{L-Meta-Acc}$. Also, apply the models M_F over D_i^{Ts} and use the models outputs as input features to meta-models $M_{ij}^{L-Meta-F}$. The meta-models inputs are M_{Acc} or M_F models predictions, and the output is the actual predicted value.
  2. Apply the meta-models on the input data and predict the result.
  3. Evaluate the prediction performance of the meta-models using the local test data and select the best meta-models $M_{ij*}^{Best\ L-Meta-Acc}$ and $M_{ij**}^{Best\ L-Meta-F}$.

e) The final local combined model of a site i to predict a result y of x that used to make a prediction on the test data is:

$$y(x) = M_{ij*}^{Best\ L-Meta-Acc}\ [M_{ij*}(x), M_{i'j*}(x), \ldots, M_{nj*}(x)] \quad (4.11)$$

$$y(x) = M_{ij**}^{Best\ L-Meta-F}\ [M_{ij**}(x), M_{i'j**}(x), \ldots, M_{nj**}(x)] \quad (4.12)$$

Figure 4.6 shows the local level modelling and nonlinear combination in each site i. Here we present the scenario for the best models that selected by accuracy metric to develop the local combined model $M_{ij*}^{Best\ L-Meta-Acc}$.

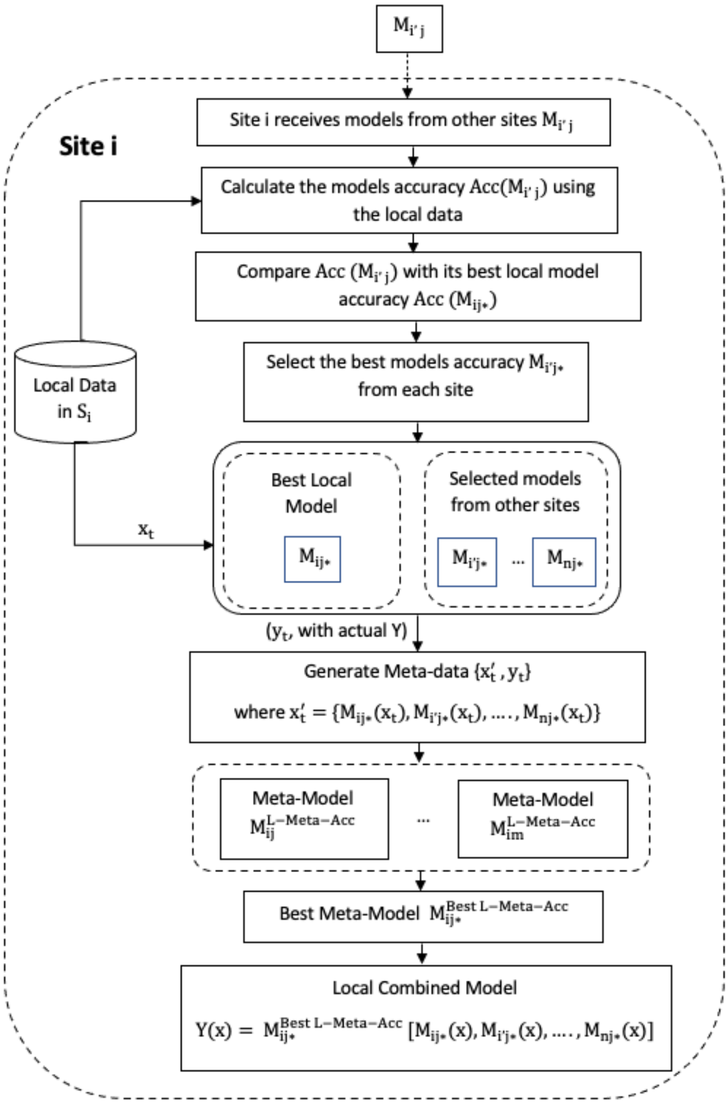


Figure 4.6. Local Level Modelling and Nonlinear Combination In Each Site

4.3.3. Experimental Study

The experiments are conducted to evaluate the performance of our proposed methods. We evaluated the prediction performance, compared it with different studies, and centralised learning approach that moves all distributed data to a centralised database.

I. Datasets

We used the eight databases that used in chapter 3 (Section 3.3.3): blood transfusion, liver disease, diabetes, heart disease, lower back pain (spine disease), breast cancer Wisconsin (Diagnostic), breast cancer Wisconsin (Original) [29], and cardiovascular diseases [27]. We partitioned the data in each site into three splits; the first one is used as the local dataset used to develop the local models, evaluate the received models from other sites, and develop the meta-models, and the second part (local test data) is used for meta-models evaluation. The last part is used to assess the local and global combined models.

II. Simulating distributed data

We used the two dataset partitioning strategies that used in chapter 3 (Section 3.4.3): (1) random data partitioning approach and (2) non-random data partitioning approach. We divided each dataset into different parts as distributed sites: site1, site2, and site3.

III. Models building and evaluation

We applied the classification algorithms that used in chapter 3 (Section 3.3.3) to develop the local models: K-Nearest Neighbor, Logistic Regression, Neural Network, Support Vector Machine, Random Forests, Decision tree, and Naïve Bayes. We built diverse binary-class prediction models to predict diabetes, heart disease, blood transfusion, liver disease, lower back pain (spine disease), cardiovascular diseases, and breast cancer datasets (positive or negative) depending on the patients' diagnosis. Different local models are trained in each site on the local training dataset from its local dataset. We used 10-fold cross-validation to prevent selection-biased results from being drawn from a single split of the local data into training and test sets. We applied two model selection strategies that select the best models based on accuracy and F-measure.

IV. Combined Model Evaluation Methods

1) **Global Combined Model Evaluation:**

a) **Testing accuracy:** We evaluated the global combined models $M_{G-META-Acc}$ and $M_{G-META-F}$ in each site instead of the server because the sites will not share their validation data with the server. Each site applies the best global average models over the validation data and uses the models outputs as input features to the meta-models (the global combined model is meta-models combination). Then, apply the meta-models on the input data, predict the result, average the results of the meta-models, and send it to the server with the number of validation data samples. Next, the server receives the sites evaluation results and then calculates the global weighted average accuracy of the global combined models.

b) **Training accuracy:** each site evaluated the global combined models output for the x that used as input to develop meta-data to train the meta-model. Finally, average the meta-models results and send it to the server with the number of local data samples to calculate the global weighted average accuracy.

2) **Local Combined Model Evaluation:** Each site calculates the accuracy of the final local combined models $M_{ij*}^{Best\ L-Meta-Acc}$ and $M_{ij**}^{Best\ L-Meta-F}$ based on the local validation data and compare it with the best local models accuracy M_{ij*} and M_{ij**}.

In addition, we evaluated and compared the global combined model with a technique that if nonlinearly combines the best local model from each site instead of the best global average model (**Best Local Models Combination**). Furthermore, we evaluated and compared the local combined model with the dynamic ensemble selection (DES) methods: KNORA-U, KNORA-E, DES-P, META-DES, KNOP, and DES-KNN dynamic ensemble methods.

V. Experiment Results and Analysis

1) Random data partitioning Approach:

i. Global-level modelling results:

In Tables 4.2 and 4.3, we compared the proposed methods with the centralised learning approach and the proposed **Best Local Models Combination method**. The results of our method using the two model selection metrics are similar to or better than the other method in most datasets. In addition, our method got similar results with the centralised learning in breast cancer and cardiovascular diseases datasets and close results in blood transfusion, spine disease, and heart disease datasets. In diabetes, the proposed method outperformed the centralised learning approach. As we discussed, the centralised learning approach has many issues and is inadequate for private and un-exchangeable data in a distributed environment.

Table 4.2.Global Combined Method and Centralised Learning Approach Evaluation (1)

Methods	Selection Metric	Blood transfusion	Breast Cancer Wisconsin (Diagnostic)	Diabetes	Heart Disease
Global-level modelling	Accuracy	57%	94%	**80%**	88%
	F-measure	57%	94%	79%	89%
Best Local Models Combination	Accuracy	53%	95%	78%	88%
	F-measure	54%	95%	74%	90%
Centralised Learning		**60%**	**96%**	78%	**92%**

Table 4.3.Global Combined Method and Centralised Learning Approach Evaluation (2)

Methods	Selection Metric	Spine Disease	Liver Disease	Breast Cancer Wisconsin (Original)	Cardiovascular diseases
Global-level modelling	Accuracy	62%	68%	98%	73%
	F-measure	62%	68%	98%	73%
Best Local Models Combination	Accuracy	61%	68%	98%	73%
	F-measure	59%	68%	98%	73%
Centralised Learning		**66%**	**78%**	98%	73%

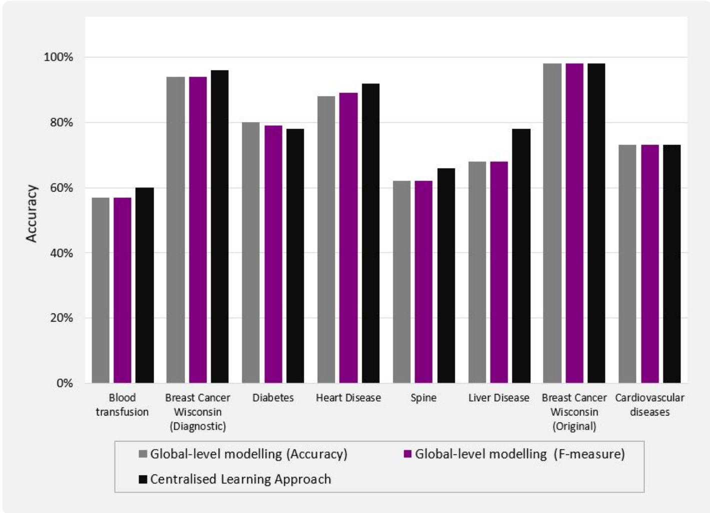


Figure 4.7. Prediction Accuracy for Global Combined Model and Centralised Machine Learning

Table 4.4 shows the training and testing accuracy for the proposed method and centralised learning approach.

Table 4.4. Training and Testing Accuracy for the Proposed Method and Centralised Learning Approach

Datasets	Models	Selection Metric	Testing Accuracy	Training accuracy
Breast Cancer Wisconsin (Diagnostic)	Global-level modelling	Accuracy	94%	95%
		F-measure	94%	95%
	Centralised Learning Approach		**96%**	**97%**
Blood Transfusion	Global-level modelling	Accuracy	57%	**83%**
		F-measure	57%	**83%**
	Centralised Learning Approach		**60%**	79%
Diabetes	Global-level modelling	Accuracy	**80%**	**84%**
		F-measure	79%	**84%**
	Centralised Learning Approach		78%	77%
Heart Disease	Global-level modelling	Accuracy	88%	**86%**
		F-measure	89%	**86%**
	Centralised Learning Approach		**92%**	82%
Liver Disease	Global-level modelling	Accuracy	68%	**79%**
		F-measure	68%	**79%**
	Centralised Learning Approach		**78%**	72%
Spine Disease	Global-level modelling	Accuracy	62%	**90%**
		F-measure	62%	**90%**
	Centralised Learning Approach		**66%**	87%
Breast Cancer Wisconsin (Original)	Global-level modelling	Accuracy	**98%**	97%
		F-measure	**98%**	97%
	Centralised Learning Approach		**98%**	97%
Cardiovascular diseases	Global-level modelling	Accuracy	**73%**	**73%**
		F-measure	**73%**	**73%**
	Centralised Learning Approach		**73%**	**73%**

Table 4.5 shows the prediction accuracy of our global-level modelling method for the randomly partitioned datasets and compares it with different research works [6, 48, 96, 166, 170, 176-182]. The proposed methods outperformed most of the related works except liver disease dataset and got close results in breast cancer Wisconsin (Diagnostic) dataset.

Table 4.5. Global Combined Model and Research Works Evaluation

Methods	Selection Metric	Breast Cancer Wisconsin (Diagnostic)	Diabetes	Heart Disease	Liver Disease	Breast Cancer Wisconsin (Original)
Global-level modelling	Accuracy	94%	**80%**	88%	68%	**98%**
	F-measure	94%	79%	**89%**	68%	**98%**
Tsoumakas et al. [6] - EV1		-	77%	84%	-	97%
Tsoumakas et al. [6] - EV2		-	77%	83%	-	97%
Tsoumakas et al. [6] - EV 3		-	77%	85%	-	97%
Bashir et al. [48]		-	77%	84%	71%	97%
Zhang et al. [96]		-	**80%**	-	-	-
Mandal et al. [166]		96%	76%	-	-	-
Wang et al. [170]		-	77%	-	-	96%
Gao et al. [176]		-	-	72%	-	95%
Kasturi et al. [177]		-	-	-	-	96%
Ma et al. [178]		-	-	-	-	96%
Haque et al. [179]		-	78%	82%	-	**98%**
Sav et al. [180]		-	-	-	-	97%
Froelicher et al. [181]		-	78%	-	-	96%
Ed-daoudy and Maalmi [182]		-	-	82%	-	-

Furthermore, we compared our nonlinear combination method with our proposed decentralised learning approach using linear model combination method in chapter 3 (section 3.3) and published in [22]. Both methods followed similar model building and selection strategies but differed in model combination approach. In [22], the selected models are combined in a linear combination method using the weighted combination method. As shown in Table 4.6, our nonlinear combination method is slightly better than the other method in diabetes, heart disease, spine disease, and liver disease datasets. In breast cancer and cardiovascular disease datasets, both methods got similar results. In blood transfusion dataset, the linear method is slightly better than the nonlinear method.

Table 4.6. Linear And Nonlinear Combination Methods Evaluation Results

Dataset	Linear Combination Method	Nonlinear Combination Method
Blood transfusion	**59%**	57%
Breast Cancer Wisconsin (Original)	**98%**	**98%**
Breast Cancer Wisconsin (Diagnostic)	**94%**	**94%**
Diabetes	78%	**80%**
Heart Disease	86%	**89%**
Spine Disease	59%	**62%**
Liver Disease	65%	**68%**
Cardiovascular disease	**73%**	**73%**

ii. Local-level modelling results:

Tables 4.7 – 4.14 show the local combined model results for all datasets. We compared **Method L1** and **Method L2** with the best local model in each site. Our approach got better or similar accuracy than the best local model in most of the datasets except in heart disease dataset (site1). The local combined models results using the two model selection metrics are similar or close. Furthermore, in each site, we applied dynamic ensemble selection (DES) methods to the received models from other sites. Our methods outperformed the best local model in most datasets sites and got better or similar results with DES methods in most datasets sites. We achieved that the distributed sites could utilise other sites models to improve the prediction performance without sharing data.

Table 4.7. Local Combined Model Evaluation for Blood Transfusion Dataset

Methods	Selection metric	Site1	Site2	Site3
Method L1	Accuracy	**54%**	40%	-
	F-measure	51%	40%	**66%**
Method L2	Accuracy	-	-	60%
	F-measure	-	-	60%
The Best Local Model		51%	40%	54%
Dynamic Ensemble Selection	KNORA-U	51%	**60%**	**66%**
	KNORA-E	43%	43%	60%
	DES-P	51%	**60%**	57%
	META-DES	43%	**60%**	60%
	KNOP	34%	**60%**	**66%**
	DES-KNN	46%	40%	46%

Table 4.8. Local Combined Model Evaluation for Diabetes Dataset

Methods	Selection metric	Site1	Site2	Site3
Method L1	Accuracy	73%	**86%**	68%
	F-measure	75%	66%	**76%**
Method L2	Accuracy	78%	-	68%
	F-measure	73%	83%	-
The Best Local Model		68%	74%	68%
Dynamic Ensemble Selection	KNORA-U	78%	83%	68%
	KNORA-E	72%	74%	68%
	DES-P	80%	80%	64%
	META-DES	68%	80%	64%
	KNOP	72%	83%	72%
	DES-KNN	**82%**	83%	68%

Table 4.9. Local Combined Model Evaluation for Heart Disease Dataset

Methods	Selection metric	Site1	Site2	Site3
Method L1	Accuracy	80%	**95%**	80%
	F-measure	73%	**95%**	80%
Method L2	Accuracy	80%	-	-
	F-measure	80%	-	-
The Best Local Model		93%	85%	80%
Dynamic Ensemble Selection	KNORA-U	87%	**95%**	87%
	KNORA-E	**99%**	60%	67%
	DES-P	87%	85%	87%
	META-DES	80%	85%	87%
	KNOP	87%	90%	**93%**
	DES-KNN	87%	80%	80%

Table 4.10. Local Combined Model Evaluation for Liver Disease Dataset

Methods	Selection metric	Site1	Site2	Site3
Method L1	Accuracy	-	-	67%
	F-measure	-	65%	**73%**
Method L2	Accuracy	68%	70%	-
	F-measure	68%	-	-
The Best Local Model		68%	68%	67%
Dynamic Ensemble Selection	KNORA-U	68%	72%	60%
	KNORA-E	68%	65%	67%
	DES-P	68%	72%	67%
	META-DES	68%	80%	**73%**
	KNOP	68%	**75%**	60%
	DES-KNN	**72%**	72%	67%

Table 4.11. Local Combined Model Evaluation for Spine Disease Dataset

Methods	Selection metric	Site1	Site2	Site3
Method L1	Accuracy	-	50%	-
	F-measure	-	50%	-
Method L2	Accuracy	**65%**	50%	**75%**
	F-measure	60%	-	**75%**

The Best Local Model		55%	50%	**75%**
Dynamic Ensemble Selection	KNORA-U	55%	50%	69%
	KNORA-E	50%	50%	**75%**
	DES-P	55%	50%	69%
	META-DES	55%	50%	**75%**
	KNOP	55%	50%	**75%**
	DES-KNN	55%	50%	69%

Table 4.12. Local Combined Model Evaluation for Breast Cancer Wisconsin (original) Dataset

Methods	Selection metric	Site1	Site2	Site3
Method L1	Accuracy	-	**99%**	**99%**
	F-measure	-	-	**99%**
Method L2	Accuracy	**98%**	-	-
	F-measure	**98%**	97%	-
The Best Local Model		**98%**	**99%**	**99%**
Dynamic Ensemble Selection	KNORA-U	**98%**	97%	98%
	KNORA-E	**98%**	**99%**	98%
	DES-P	**98%**	97%	98%
	META-DES	**98%**	97%	98%
	KNOP	**98%**	97%	98%
	DES-KNN	**98%**	**99%**	98%

Table 4.13. Local Combined Model Evaluation for Breast Cancer Wisconsin (diagnostic) Dataset

Methods	Selection metric	Site1	Site2	Site3
Method L1	Accuracy	95%	93%	89%
	F-measure	95%	**97%**	-
Method L2	Accuracy	**99%**	-	89%
	F-measure	**99%**	-	89%
The Best Local Model		95%	93%	89%
Dynamic Ensemble Selection	KNORA-U	90%	**97%**	**95%**
	KNORA-E	90%	**97%**	89%
	DES-P	90%	**97%**	**95%**
	META-DES	95%	**97%**	89%
	KNOP	95%	**97%**	89%
	DES-KNN	95%	**97%**	89%

Table 4.14. Local Combined Model Evaluation for Cardiovascular Diseases Dataset

Methods	Selection metric	Site1	Site2	Site3
Method L1	Accuracy	**74%**	**73%**	72%
	F-measure	**74%**	**73%**	72%
Method L2	Accuracy	-	-	-
	F-measure	-	-	**73%**
The Best Local Model		**74%**	**73%**	72%
	KNORA-U	73%	**73%**	72%
	KNORA-E	69%	67%	68%
	DES-P	72%	72%	71%

Dynamic Ensemble Selection	META-DES	66%	71%	65%
	KNOP	66%	**73%**	65%
	DES-KNN	71%	72%	71%

2) Non-random data partitioning Approach:

i. Global-level Modelling Results:

Table 4.15 shows that our combined model selected by accuracy metric performed slightly better or similar to F-measure model selection metric in most sites. We compared the prediction accuracy of the global combined model with a method that if each site sends the best local model to the server instead of sending the best global average model (**Best Local Models Combination**). The results of the global combined model got similar or close results to the best local model combination method. In addition, we got close results with the centralised learning approach, and the global combined model results using the two model selection metrics are similar.

Table 4.15. Global Combined Model and Centralised Learning Approach Evaluation

Models	Selection Metric	Diabetes	Heart Disease	Liver Disease
Global-level modelling	Accuracy	67%	83%	72%
	F-measure	65%	83%	72%
Best Local Models Combination	Accuracy	65%	83%	70%
	F-measure	65%	83%	72%
Centralised Learning Approach		**69%**	**88%**	**74%**

Table 4.16. Training and Testing accuracy for the Proposed Method and Centralised Learning Approach

Datasets	Models	Selection Metric	Testing Accuracy	Training accuracy
Diabetes	Global-level modelling	Accuracy	67%	73%
		F-measure	65%	**80%**
	Centralised Learning Approach		**69%**	78%
Heart Disease	Global-level modelling	Accuracy	83%	82%
		F-measure	83%	82%
	Centralised Learning Approach		**88%**	**83%**
Liver Disease	Global-level modelling	Accuracy	72%	**81%**
		F-measure	72%	**81%**
	Centralised Learning Approach		**74%**	72%

Besides, we compared the proposed decentralised learning approach using nonlinear model combination method with the related works [6, 48, 69, 166,170, 176, 179,181,182]; Table 4.17 shows that the research works are better than our proposed method in diabetes dataset, and our method is slightly better than some approaches in heart disease and liver disease datasets.

Table 4.17. Global Combined Model and Related Works Evaluation

Models	Selection Metric	Diabetes	Heart Disease	Liver Disease
Global-level modelling	Accuracy	67%	83%	**72%**
	F-measure	65%	83%	**72%**
Tsoumakas et al. [6] - EV1		77%	84%	-
Tsoumakas et al. [6] - EV2		77%	83%	-
Tsoumakas et al. [6] - EV 3		77%	**85%**	-
Bashir et al. [48]		77%	84%	71%
Zhang et al. [96]		**80%**	-	-
Mandal et al. [166]		76%	-	-
Wang et al. [170]		77%	-	-
Gao et al. [176]		-	72%	-
Haque et al. [179]		78%	82%	-
Froelicher et al. [181]		78%	-	-
Ed-daoudy and Maalmi [182]		-	82%	-

We compared our nonlinear combination method with our proposed decentralised learning approach using linear model combination method in chapter 3 (section 3.3) and published in [22]. As presented in Table 4.18, the proposed nonlinear and linear combination methods got similar results in diabetes and liver disease datasets, while in heart disease dataset, the linear combination method is slightly better than the other method.

Table 4.18. Linear And Nonlinear Combination Methods Evaluation Results

Dataset	Linear Combination Method	Nonlinear Combination Method
Diabetes	**67%**	**67%**
Heart Disease	87%	83%
Liver Disease	**72%**	**72%**

ii. Local-level modelling results:

As shown in Tables 4.19- 4.21, the local combined model is better or similar to the best local models only in liver disease dataset in site 3. Therefore, the local combined model using the two model selection metrics results are similar or close. Also, we compared the local combined model with DES methods. It shows that some DES methods results are better than our method.

Table 4.19. Local Combined Model Evaluation for Diabetes Dataset

Methods	Selection metric	Site1	Site2	Site3
Method L1	Accuracy	-	72%	-
	F-measure	74%	**77%**	-
Method L2	Accuracy	74%	**77%**	53%
	F-measure	74%	-	53%
The Best Local Model		70%	67%	52%
Dynamic Ensemble Selection	KNORA-U	74%	56%	**77%**
	KNORA-E	68%	53%	61%
	DES-P	72%	58%	**77%**
	META-DES	68%	51%	71%
	KNOP	72%	56%	74%
	DES-KNN	**77%**	58%	74%

Table 4.20. Local Combined Model Evaluation for Heart Disease Dataset

Methods	Selection metric	Site1	Site2	Site3
Method L1	Accuracy	80%	88%	**91%**
	F-measure	80%	85%	**91%**
Method L2	Accuracy	80%	-	**91%**
	F-measure	80%	-	**91%**
The Best Local Model		75%	88%	**91%**
Dynamic Ensemble Selection	KNORA-U	**90%**	88%	86%
	KNORA-E	80%	**91%**	**91%**
	DES-P	**90%**	**91%**	86%
	META-DES	85%	82%	82%
	KNOP	85%	**91%**	86%
	DES-KNN	85%	82%	**91%**

Table 4.21. Local Combined Model Evaluation for Liver Disease Dataset

Methods	Selection metric	Site1	Site2	Site3
Method L1	Accuracy	-	-	71%
	F-measure	**65%**	-	71%
Method L2	Accuracy	40%	**79%**	71%
	F-measure	-	**79%**	71%
The Best Local Model		50%	**79%**	76%
Dynamic Ensemble Selection	KNORA-U	50%	67%	76%
	KNORA-E	50%	76%	76%
	DES-P	40%	58%	71%
	META-DES	45%	76%	76%
	KNOP	55%	73%	**82%**
	DES-KNN	45%	61%	71%

4.3.4. Discussion and Evaluation

The proposed method results for randomly partitioned datasets showed that the global model performance is in par with the centralised learning approach in breast cancer Wisconsin (Original) and cardiovascular diseases datasets, and got close results in blood transfusion, breast cancer Wisconsin (Diagnostic), heart disease, and spine disease datasets. Our method is slightly better than the centralised learning approach in diabetes dataset. In non-random partitioned data, the proposed method and the centralised learning approach results are close. The results showed that we could improve the prediction performance as similar as or better than the centralised learning approach without using a server for models learning, controlling the learning process, or sharing the data. Our method overcomes central location issues and overheads and preserves data privacy. Compared to federated learning, which requires sharing gradient information and iterative learning communication, the proposed nonlinear combination approach provides an effective alternative with much less information sharing and reduced computation cost. Also, it could be applied to solve issues related to large data, such as memory limitation and huge data transformation costs and time. The results of the proposed local level modelling approach showed that the distributed sites could improve the prediction performance using other sites models without sharing data. However, there is a possibility for malicious attacks on the trained models to retrieve training data or reveal meaningful information. It is beyond the scope of our thesis, and in future research, we will consider these issues to analyse the possible malicious attacks on distributed environments.

4.4 PROPOSED METHOD FOR REGRESSION ALGORITHMS

We apply the global and local level modelling approaches proposed in sections 4.3.1 and 4.3.2, but we use RMSE and MAPE evaluation metrics instead of accuracy and F-measure metrics that used for classification algorithms.

The final global combined models at the server are $M_{G-META-RMSE}$ and $M_{G-META-MAPE}$, where $M_{G-META-RMSE}$ combines the best meta-models $M_{ij*}^{Best\ G-Meta-RMSE}$ from each site, and $M_{G-META-MAPE}$ combines the best meta-models $M_{ij**}^{Best\ G-Meta-MAPE}$ that used to predict a result y of x as follows:

$$y(x) = M_{G-META-RMSE}[M_{ij*}^{Best\ G-Meta-RMSE}(x), \dots, M_{nj*}^{Best\ G-Meta-RMSE}(x)] \quad (4.13)$$

$$y(x) = M_{G-META-MAPE}\ [M_{ij**}^{Best\ G-Meta-MAPE}(x), \dots, M_{nj**}^{Best\ G-Meta-MAPE}(x)] \quad (4.14)$$

The local combined models in each site i are the best meta-models $M_{ij*}^{Best\ L-Meta-RMSE}$ and $M_{ij**}^{Best\ L-Meta-MAPE}$:

$$y(x) = M_{ij*}^{Best\ L-Meta-RMSE}\ [M_{ij*}(x), M_{i'j*}(x), \dots., M_{nj*}(x)] \quad (4.15)$$

$$y(x) = M_{ij**}^{Best\ L-Meta-MAPE}\ [M_{ij**}(x), M_{i'j**}(x), \dots., M_{nj**}(x)] \quad (4.16)$$

, where M_{ij*} are the best local model RMSE in site i and the selected model from other sites, and M_{ij**} are the best local model MAPE in site i and the selected model from other sites, i=1,..,n.

4.4.1. Experimental Study

We evaluated the reliability and prediction performance of our proposed decentralised learning approach using nonlinear model combination method and compared it with different studies and a centralised learning approach that moves all distributed data to a centralised database.

I. Datasets

We used the three databases used in chapter 3 (Section 3.4.3): Parkinson disease, Boston housing, and Abalone datasets [29]. We partitioned the data in each site into three main splits; the first one is the local dataset that is used to develop the local models, evaluate the received models from other sites, and develop the meta-models, the second part (local test data) is used for meta-models evaluation, and the last part is used to evaluate the final combined local and global combined models.

II. Simulating Distributed Data

We used the two dataset partitioning strategies that applied in chapter 3 (Section 3.4.3): (1) random data partitioning approach and (2) non-random data partitioning approach. Therefore, we divided each dataset into different parts as distributed sites site1, site2, and site3.

III. Models Building and Evaluation

We applied the regression algorithms that used in chapter 3 (Section 3.4.3) to develop the local models: Linear Regression (LR), Random Forest Regressor (RFR), Radial Basis Function Neural Network (RBFNN), K-Nearest Neighbor Regressor (KNNR), Decision tree regression (DTR), Support Vector Regressor (SVR), Neural Network Regressor (NNR), Lasso, ElasticNet, and Ridge. Different local models are trained in each site on the local training dataset from its local dataset. We used 10-fold cross-validation results to evaluate the local models and applied two model selection strategies that select the best models based on RMSE and MAPE.

IV. Combined Model Evaluation Methods

1) Global Combined Model Evaluation

a) Testing error: we evaluated the final global combined models $M_{G-META-RMSE}$ and $M_{G-META-MAPE}$ in each site based on the local validation data of the site. We evaluated the global combined model in each site instead of the server because the sites will not share their validation data with the server. Each site applies the best global average models over the validation data and uses the models outputs as input features to the meta-models (the global combined model is meta-models combination). Then, apply the meta-models on the input data and predict the result. Finally, it sends the averaged global combined model evaluation results SSE and MAPE with the number of validation data samples to the server. Then, the server calculates the average RMSE by dividing the sum of averaged SSE from all sites by the number of validation data samples of all sites, then gets the square root error to get the RMSE. Also, it calculates the average MAPE. We compare the global combined models with the centralised learning approach and other research works.

b) **Training error:** each site evaluated the global combined models output for the x that used as input to develop meta-data to train meta-model. Finally, average the meta-models results SSE and MAPE and send it to the server with the number of local data samples to calculate the global average RMSE and MAPE.

2) **Local Combined Model Evaluation:** Each site calculated the MAPE and RMSE of the local combined models based on the local validation data and compared it with the best local model M_{ij}^{*} and M_{ij}^{**}. Also, we compare the local models with DES methods.

In addition, we evaluated and compared the global combined model with a technique that if nonlinearly combines the best local model from each site instead of the best global average model (**Best Local Models Combination**).

V. Experiment Results and Analysis

1) Random Data Partitioning Approach:

i. Global-level Modelling results

Table 4.22 shows the global model evaluation compared with the centralised learning approach and the proposed **Best Local Models Combination approach**. For Parkinson Disease (Total UPDRS) dataset, the results show that the centralised learning approach outperformed the other methods. Also, it shows that the prediction results for Parkinson Disease (Motor UPDRS) dataset and the centralised learning approach results are the best but not far from our method results. For Boston housing and Abalone datasets, the centralised learning approach and our method RMSE results are close. The global combined model results using RMSE metric for model selection method are slightly better than MAPE selection metric in most datasets.

Table 4.22. Global Combined Model and Centralised Learning Approach Evaluation

Dataset	Methods	Selection metric	MAPE	RMSE
Parkinson Disease (Total UPDRS)	Global-level modelling	MAPE	33.05	10.29
		RMSE	32.51	10.16
	Best Local Models Combination	MAPE	34.77	11.33
		RMSE	33.45	10.64
	Centralised Learning Approach		**24.24**	**7.53**
Parkinson Disease (Motor UPDRS)	Global-level modelling	MAPE	38.13	8.35
		RMSE	40.56	8.22
	Best Local Models Combination	MAPE	37.80	8.16
		RMSE	37.93	8.21
	Centralised Learning Approach		**35.47**	**6.98**
Boston Housing	Global-level modelling	MAPE	17.35	3.98
		RMSE	15.48	3.55
	Best Local Models Combination	MAPE	15.24	3.81
		RMSE	15.14	3.56
	Centralised Learning Approach		**11.96**	**3.20**
Abalone	Global-level modelling	MAPE	13.88	2.47
		RMSE	13.76	2.43
	Best Local Models Combination	MAPE	14.74	2.62
		RMSE	**13.66**	2.42
	Centralised Learning Approach		13.69	**2.39**

Table 4.23 presents the training and testing error for the proposed method and centralised learning approach.

Table 4.23. Training and Testing Error for the Proposed Method and Centralised Learning Approach

Dataset	Methods	Selection metric	Testing error		Training error	
			MAPE	RMSE	MAPE	RMSE
Parkinson Disease (Total UPDRS)	Global-level modelling	MAPE	33.05	10.29	30.06	10.54
		RMSE	32.51	10.16	29.80	10.74
	Centralised Learning Approach		**24.24**	**7.53**	**23.53**	**8.30**
Parkinson Disease (Motor UPDRS)	Global-level modelling	MAPE	38.13	8.35	28.29	7.85
		RMSE	40.56	8.22	31.84	7.78
	Centralised Learning Approach		**35.47**	**6.98**	**24.78**	**6.15**
Boston Housing	Global-level modelling	MAPE	17.35	3.98	14.32	4.86
		RMSE	15.48	3.55	**13.61**	**4.50**
	Centralised Learning Approach		**11.96**	**3.20**	17.39	5.45
Abalone	Global-level modelling	MAPE	13.88	2.47	13.33	2.13
		RMSE	13.76	2.43	**13.04**	**2.08**
	Centralised Learning Approach		**13.69**	**2.39**	13.28	2.12

Table 4.24 shows RMSE results for the proposed approach using nonlinear model combination method for the randomly partitioned datasets compared with the proposed linear model combination method in chapter 3 (section 3.4), the centralised learning approach, and the proposed method in [166]. It shows that our method using the linear model combination approach is slightly better than the other methods in Boston housing and Abalone datasets. For Parkinson disease dataset, the centralised learning method outperformed the nonlinear combination method but it is not far from the linear combination method RMSE results.

Table 4.24. Global Combined Model Methods and Centralised Learning Approach Evaluation for Random Partitioned Data

Dataset	Linear Combination Method	Nonlinear Combination Method	Centralised Learning Approach	Mandal et al. [166]
Parkinson Disease (Total UPDRS)	8.97	10.16	**7.53**	-
Parkinson Disease (Motor UPDRS)	7.30	8.22	**6.98**	-
Boston Housing	**3.10**	3.55	3.20	4.91
Abalone	**2.36**	2.43	2.39	-

ii. Local-level Modeling result

Tables 4.25 – 4.28 show the local combined evaluation results in each site, compared with the best local model for Parkinson Disease (Total UPDRS), Parkinson Disease (Motor UPDRS), Boston housing, and Abalone datasets. The results show that our method results are better than the best local model in all datasets except in site1 in Abalone dataset, the best local model RMSE result is slightly better than the proposed method. We conclude that each site could improve the prediction performance by utilising other sites models without sharing the data. The global combined model results using RMSE method are slightly better than MAPE in most distributed sites.

Table 4.25. Local Combined Model Evaluation for Parkinson Disease (Total UPDRS) Dataset

Method	Selection metric	Site1		Site2		Site3	
		RMSE	MAPE	RMSE	MAPE	RMSE	MAPE
Local-Level Modelling	MAPE	8.29	23.22	**8.47**	**27.24**	6.83	19.88
The Best Local Model		9.36	26.69	10.23	36.18	6.95	23.71
Local-Level Modelling	RMSE	**8.27**	**22.97**	10.33	36.53	**6.72**	**19.39**
The Best Local Model		9.36	26.69	10.23	36.18	6.95	23.71

Table 4.26. Local Combined Model Evaluation for Parkinson Disease (Motor UPDRS) Dataset

Method	Selection metric	Site1		Site2		Site3	
		RMSE	MAPE	RMSE	MAPE	RMSE	MAPE
Local-Level Modelling	MAPE	5.92	25.36	5.63	23.86	**6.59**	**31.02**
The Best Local Model		5.99	26.14	5.58	23.84	6.68	33.43
Local-Level Modelling	RMSE	**5.76**	**24.81**	**5.56**	**20.61**	6.60	33.24
The Best Local Model		5.99	26.14	5.58	23.84	6.68	33.43

Table 4.27. Local Combined Model Evaluation for Boston Housing Dataset

Method	Selection metric	Site1		Site2		Site3	
		RMSE	MAPE	RMSE	MAPE	RMSE	MAPE
Local-Level Modelling	MAPE	**2.79**	12.99	**2.31**	8.84	3.55	17.37
The Best Local Model		3.25	14.01	4.69	12.75	3.25	16.08
Local-Level Modelling	RMSE	2.85	**12.20**	2.62	**8.76**	**2.99**	**13.63**
The Best Local Model		3.25	14.01	4.69	12.75	3.25	16.08

Table 4.28. Local Combined Model Evaluation for Abalone Dataset

Method	Selection metric	Site1		Site2		Site3	
		RMSE	MAPE	RMSE	MAPE	RMSE	MAPE
Local-Level Modelling	MAPE	2.63	14.35	2.73	14.19	**2.59**	**14.87**
The Best Local Model		2.82	14.72	2.93	14.55	3.14	16.29
Local-Level Modelling	RMSE	2.63	**14.34**	**2.18**	**12.33**	2.62	14.96
The Best Local Model		**2.57**	14.79	2.33	12.55	2.65	15.19

2) Non-random Data Partitioning Approach:

i. Global-level Modeling result

Table 4.29 shows the global combined model and centralised learning approach results and the testing and training errors for Parkinson disease (Total UPDRS), Parkinson disease (Motor UPDRS), and Boston housing datasets. Our method outperformed the centralised learning approach in Parkinson Disease (Total UPDRS) dataset and got close results in Parkinson disease (Motor UPDRS) and Boston housing datasets. The evaluation results using the two model selection methods are close.

Table 4.29. Global Combined Model and Centralised Learning Approach Evaluation

Dataset	Methods	Selection metric	MAPE	RMSE
Parkinson Disease (Total UPDRS)	Global-level modelling	MAPE	**22.26**	**6.94**
		RMSE	23.22	7.14
	Best Local Models Combination	MAPE	22.45	7.01
		RMSE	22.46	7.02
	Centralised Learning Approach		25.22	8.17
Parkinson Disease (Motor UPDRS)	Global-level modelling	MAPE	41.52	9.51
		RMSE	38.15	8.51
	Best Local Models Combination	MAPE	41.29	9.43
		RMSE	40.85	9.29
	Centralised Learning Approach		**30.68**	**7.3**
Boston Housing	Global-level modelling	MAPE	16.9	4.26
		RMSE	16.2	4.01
	Best Local Models Combination	MAPE	23.3	4.87
		RMSE	22.6	4.69
	Centralised Learning Approach		**13.31**	**3.38**

Table 4.30. Training and Testing Error for for the Proposed Method and Centralised Learning Approach

Dataset	Methods	Selection metric	Testing error		Training error	
			MAPE	RMSE	MAPE	RMSE
Parkinson Disease (Total UPDRS)	Global-level modelling	MAPE	**22.26**	**6.94**	**24.62**	**7.52**
		RMSE	23.22	7.14	24.97	7.91
	Centralised Learning Approach		25.22	8.17	40.04	11.13
Parkinson Disease (Motor UPDRS)	Global-level modelling	MAPE	41.52	9.51	40.27	9.57
		RMSE	38.15	8.51	45.14	9.81
	Centralised Learning Approach		**30.68**	**7.3**	**34.4**	**6.91**
Boston Housing	Global-level modelling	MAPE	16.9	4.26	16.6	5.28
		RMSE	16.2	4.01	16.3	5.01
	Centralised Learning Approach		**13.31**	**3.38**	**11.51**	**3.27**

In Table 4.31, our proposed approach using nonlinear combination method is better than the linear combination method and centralised learning approach in Parkinson Disease (Total UPDRS). The centralised learning approach result in the Boston housing dataset is slightly better than the linear combination method.

Table 4.31. Global Combined Model Methods and Centralised Learning Approach Evaluation for Non-random Partitioned Data

Dataset	Linear Combination Method	Nonlinear Combination Method	Centralised Learning Approach	Mandal et al. [166]
Parkinson Disease (Total UPDRS)	8.19	6.94	8.17	-

Parkinson Disease (Motor UPDRS)	7.1	8.51	7.3	**-**
Boston Housing	3.67	4.01	**3.38**	4.91

ii. Local-level Modeling Result

Tables 4.32 – 4.34 show the local combined models RMSE results in each site compared with the best local model for Parkinson disease (Total UPDRS), Parkinson disease (Motor UPDRS), and Boston housing datasets. Our method is better than the best local models in most datasets sites, and the results using the two model selection metrics are close.

Table 4.32. Local Combined Model Evaluation for Parkinson Disease (Total UPDRS) Dataset

Method	Selection metric	Site1		Site2		Site3	
		RMSE	MAPE	RMSE	MAPE	RMSE	MAPE
Local-Level Modelling	MAPE	**5.82**	**16.96**	7.48	**27.77**	**7.80**	21.77
The Best Local Model		8.26	22.56	7.89	31.77	8.96	**20.09**
Local-Level Modelling	RMSE	5.86	17.12	**7.38**	28.12	7.88	22.10
The Best Local Model		8.25	22.56	7.89	31.77	8.96	**20.09**

Table 4.33. Local Combined Model Evaluation for Parkinson Disease (Motor UPDRS) Dataset

Method	Selection metric	Site1		Site2		Site3	
		RMSE	MAPE	RMSE	MAPE	RMSE	MAPE
Local-Level Modelling	MAPE	**6.32**	**22.66**	**6.76**	**36.99**	**6.50**	22.84
The Best Local Model		6.4	23.17	7.02	40.64	8.36	**22.63**
Local-Level Modelling	RMSE	6.39	23.26	6.81	37.67	7.07	23.17
The Best Local Model		6.4	23.17	7.02	40.64	8.36	**22.63**

Table 4.34. Local Combined Model Evaluation for Boston Housing Dataset

Method	Selection metric	Site1		Site2		Site3	
		RMSE	MAPE	RMSE	MAPE	RMSE	MAPE
Local-Level Modelling	MAPE	1.84	**4.97**	2.20	9.35	4.27	24.58
The Best Local Model		2.11	5.99	2.62	11.42	3.61	24.04
Local-Level Modelling	RMSE	**1.81**	5.32	**2.08**	**8.77**	**3.18**	**22.34**
The Best Local Model		2.11	5.99	2.38	9.32	3.61	24.04

4.4.2. Discussion and Evaluation

The proposed global level modelling using nonlinear model combination method for random partitioned data showed that the centralised learning approach results are better than our method in Parkinson disease dataset and close results in Abalone and Boston housing datasets. On the other hand, in non-random partitioned data, the centralised learning approach RMSE results are better but not far from the global combined model. The global combined model results using RMSE method are slightly better than MAPE in most distributed sites. We proposed a decentralised learning approach for distributed datasets to build a global combined model without data sharing between the sites, centralising the data to a central database, or using a server to control the learning process or iterative communication. We avoid server issues and overheads and data transformation cost. Also, we preserved data privacy and avoided using complex privacy-preserving methods. The proposed method could be applied to solve issues related to large data, such as memory limitation and huge data transformation costs and time. The local level modelling results in randomly partitioned datasets showed that our method results are better than the best local model in all datasets except in site1 in Abalone dataset. For non-random partitioned datasets, the local combined model outperformed the best local models in most datasets sites, and the results using the two model selection metrics are close. We conclude that each site could improve the prediction performance by utilising other sites models without sharing the data. As discussed in 4.3.4, there is a possibility for malicious attacks on the trained models of the distributed sites, and we will consider this issue in future.

4.5 SUMMARY

This chapter presented our proposed decentralised learning using nonlinear model combination methods for classification and regression algorithms. We developed global and local-level modelling approaches for distributed data resources to build combined global and local prediction models without sending the local data between the distributed sites or centralising the datasets to a central location. In global-level modelling, we used simple model evaluation and selection strategies instead of complicated approaches that need more communication between the distributed sites and less information exchange than federated learning. Furthermore, the proposed method did not expose the data and hence preserved data privacy. We developed a combined local model for each site by utilising the learning outcomes from local data resources from other sites.

The experiments showed similar or close results to the centralised learning approach when using classification algorithms. When using regression algorithms, our method showed better performance than the centralised learning approach in Parkinson disease (Total UPDRS) dataset that partitioned non-randomly, and close or similar results in the other dataset. The local combined model showed better prediction performance than the best local model in most datasets, proving that each site can utilise other sites models to improve the prediction performance without sharing the data between sites to preserve data privacy. Compared with the popular privacy-preserving approach such as federated learning which requires sharing gradient information and iterative learning communication, the proposed nonlinear combination approach provides an effective alternative with much less information sharing and reduced computation cost. We saved the time and cost of data transformation between sites, improved computation efficiency, and preserved data privacy.

Chapter 5

Model Combination Method using Stepwise Model Selection Approach

5.1 CHAPTER OVERVIEW

This chapter presents our proposed decentralised learning method and model combination method using stepwise model selection approach. The approach performs stepwise model selection and updating in each site locally and in a decentralised learning fashion using Gossip learning method. Section 5.2 views our contribution and aims to develop decentralised learning using stepwise model selection method. Then, the proposed decentralised methods for classification with the experiment results and discussion are shown in section 5.3 and for regression algorithms in section 5.4. Finally, this chapter is summarised in section 5.5.

5.2 INTRODUCTION

We propose a decentralised learning approach and stepwise model selection strategy to distributed, private, and un-exchangeable data resources to develop global and local combined models without using a central site to control the learning process. We use stepwise model selection strategy to optimise the performance of the combined models by selecting the superior models within a sequence from a set of models according to a specified performance threshold. We perform stepwise model selection and updating approach in each site locally and in a decentralised learning fashion using Gossip learning method. Gossip learning is a decentralised learning alternative to federated learning without central control and a competitive approach to the federated learning method. It is based on models transferred between distributed sites and updated on sites local data using Stochastic Gradient Descent (SGD) to improve the models performance, and then combined the models using combination learning methods. Models combining is performed using a linear combination approach. We aim to preserve data privacy and model updates for each site, mitigate the iterative learning process overhead and communication cost, and overcome the centralised learning issues. In addition, our method only exchanges the trained models with minimal information and fewer communication rounds than federated learning.

5.3 PROPOSED METHOD FOR CLASSIFICATION ALGORITHMS

We develop a combined model at global and local levels using Gossip learning and stepwise model selection methods. First, we apply stepwise model selection method and update the selected models using mini-batch stochastic gradient descent. Finally, apply a linear combination method to develop the combined model for the final selected models.

5.3.1. Global-level Modelling Approach

We develop a decentralised alternative to the federated learning approach without using a server or exchanging intermediate computing updates to overcome iterative learning process issues. We perform model selection strategy using stepwise model selection method, then update the selected models locally in each site using Gossip learning approach to make the final combined model optimal and valuable for all sites. The stepwise model selection strategy is used to select the superior models within a sequence from a set of models and discard the poor performance models. Then, update the selected models using mini-batch SGD approach using Gossip learning method, then combine the updated models using linear combination learning method. We aim to minimise communication and computation overheads and preserve data privacy by only passing the models between the distributed sites and updating the models locally in each location without exchanging models updates information. We used a simple linear combination method to combine the best-updated models to develop global combined model with less information sharing between the distributed sites. We apply the proposed decentralised learning approach as follows:

1. Each site builds different classification local models using different classification algorithms and then passes these models to other sites.
2. When a site receives local models from other sites, it first evaluates the models based on its local data, then sends the models back with the evaluation results to the sites with the local data size that used for evaluation.
3. Each site will receive its local models evaluation results from other sites and calculate the average accuracy of its local models.
4. For each classification model, for example, SVM model, we perform the stepwise model selection and Gossip learning strategies as follows:
    a) In all sites, we compare the average accuracy for the models then select the best average accuracy model and discard the other models.
    b) Send the selected model to the other sites for updating process using mini-batch SGD without losing the previous site data information. Each site updates the model using its local data and keeps the previous models updates from other sites data.
    c) After the sites finish the model updating process, each site has an updated model. So, we send these updated models to all sites for evaluation and calculate the average accuracy.
    d) Select the best average accuracy model and send it to the other sites for the updating process. We send the model to the sites that still not update the selected model.
    e) We repeat steps (c) and (d) until all the sites update the model. We aim to get a final updated model that performs better globally by only selecting and updating the best average accuracy model.
5. Apply step 4 to all classification models that used to build the local models.
6. When all classification models are updated using all sites data, we send these updated models to all sites to recalculate the final average accuracy. Then, each site sends the evaluated models with the average accuracy and data size to the server.

7. The server selects and combines the best average accuracy models using the linear combination method to develop the global combined model. The weight for each model is calculated based on its average accuracy.

This approach does not expose the data and hence preserves data privacy. In addition, there is no computation overhead between the distributed sites and the server, avoid central control for the learning or updating processes, and use minimum exchanged information between the sites and the server. The related model names and definitions used in our methodology are introduced in Table 5.1.

Table 5.1. Models Names and Descriptions (1).

Model	Notation	Meaning
Local Model	M_{ij}	The local model that developed in site i using j classification algorithm
Best Average Accuracy Model	M_{i*j}	The best average accuracy model among all sites for each j learning algorithm
Updated Model	$M_{i*j}^{U-i'}$	The updated model M_{i*j} using site i' local data
Best Average Accuracy Updated Model	$M_{i*j}^{U-i'*}$	The best average accuracy model during updating process and after evaluation using all sites data
Best Average Accuracy Final Updated Model	$M_{i*j*}^{U-i'*}$	The best average accuracy of the final updated model after updating process and after average accuracy calculation using all sites data
Global Combined Model	M^{G*}	The final global combined model at the server that combines the best average accuracy updated models using the linear model combination method

The following steps implement the above idea:

1) Similar to our methodology for local models building and average accuracy calculations in chapter 3 (section 3.3.1), each site S_i apply j learning algorithms to build its local model M_{ij} where $i = 1,2, \dots, n$ and $j = 1,2, \dots, m$, send its local models to other sites for evaluation, receive its local models evaluation results from other sites, and calculate the average accuracy for the local models $Acc\,(M_{ij})$.
2) As illustrated in Figure 5.1, for each model developed by a learning algorithm j in all sites i, where $j = 1,2, \dots, m$:
   a) Compare the average accuracy of the models $Acc\left(M_{ij}\right)$ and select the best average accuracy model M_{i*j}.
   b) Send the selected model M_{i*j} to other sites i', $(i' = 1, \dots, i-1, i+1, \dots, n)$.

c) Start stepwise model selection and updating process as follows:

1. Each site i′ update the received model M_{i*j} based on its local data using mini-batch stochastic gradient descent method as follows:
   - Divide the local data $D_{i'}$ into batches b.
   - For each batch of p examples x_t with corresponding target value y_t, t = 1,2,…, p:
     i. Compute the gradient estimate of the loss function [143,161]:

$$g = \frac{1}{p}\nabla_{w}\sum_{t=1}^{p}\ell\left((M_{i*j}(x_t\,; w), y_t\right) \qquad (5.1)$$

     ii. Update model parameter $w = w - \gamma g$, where γ is the learning rate, and w is the model parameter.

2. After updating the model M_{i*j} using the sites i′ local data, we have different updated model versions $M_{i*j}^{U-i'}$. Therefore, we send the updated models to all sites for evaluation, then calculate the average accuracy Acc ($M_{i*j}^{U-i'}$),

$$\mathrm{Acc}\,(M_{i*j}^{U-i'}) = \sum_{k=1}^{n}\frac{D_k}{D} * \mathrm{Acc}\,(M_{i*j}^{U-i'})\ \text{in}\ S_k, \qquad (5.2)$$

   where k is the sites number, D_k is the number of samples of site k, and D is all sites data samples number.

3. Select the best average accuracy model $M_{i*j}^{U-i'*}$ and discard the other models.
4. Send the selected model $M_{i*j}^{U-i'*}$ to the other sites for updating process. The sites that do not update the selected model will receive the model. Our aim for sending the updated models to all sites is to calculate the average accuracy and then select the best model for the next updating step because we want to update the model that performs well for all sites.

3) Repeat step c until the selected model is updated using all sites local data.
4) After updating all the learning models j, send the final updated models $M_{i*j}^{U-i'*}$ to all sites for final evaluation.
5) Each site evaluates the models $M_{i*j}^{U-i'*}$ and sends the models with its evaluation results and the data size that is used for models evaluation to the server.

6) As illustrated in Figure 5.2, the server receives the evaluated models $M_{i*j}^{U-i'*}$ from all sites with its evaluation results and the data size that is used for evaluation.
7) Calculates the global average accuracy Acc $\left(M_{i*j}^{U-i'*}\right)$.

$$\mathrm{Acc}\ (M_{i*j}^{U-i'*}) = \sum_{k=1}^{n} \frac{D_k}{D} * \mathrm{Acc}\ (M_{i*j}^{U-i'*})\ \mathrm{in}\ S_k, \qquad (5.3)$$

where k is sites number, D_k is the number of samples of site k, and D is all sites data samples number.

8) Compare the average accuracy for all updated j models Acc ($M_{i*j}^{U-i'*}$) and select the best updated models $M_{i*j*}^{U-i'*}$.
9) The server assigns weight to the selected models $M_{i*j*}^{U-i'*}$ based on its average accuracy. We weight each model based on its average accuracy to get an unbiased model weight [30, 33, 86]. The most accurate model will get higher weight, and the less accurate model will get low weight. Models weights are constrained such that their sum is equal to one.

$$w_{M_{i*j*}^{U-i'*}} = \frac{\mathrm{Acc}\ (M_{i*j*}^{U-i'*})}{\sum_{i=1}^{n} \mathrm{Acc}\ (M_{i*j*}^{U-i'*})} \qquad (5.4)$$

, where i=1, 2, …, n, 0 <= $w_{M_{i*j*}^{U-i'*}}$ <=1 , and $\sum_{i=1}^{n} w_{M_{i*j*}^{U-i'*}} = 1$

10) The server linearly combines the models to develop the global model M^{G*}:

$$M^{G*}\ (x) = \max \sum w_{M_{i*j*}^{U-i'*}} M_{i*j*}^{U-i'*}\ (x) \qquad (5.5)$$

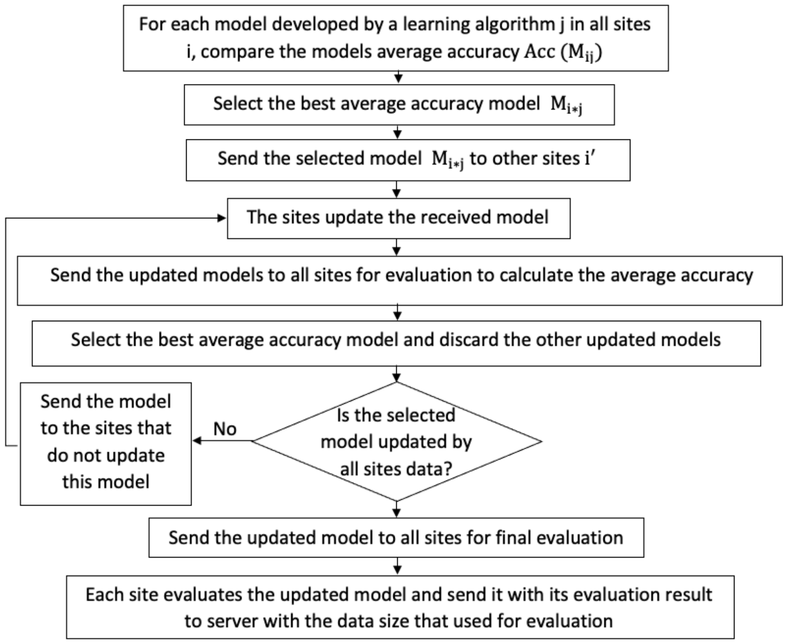


Figure 5.1. Stepwise model selection and updating methods

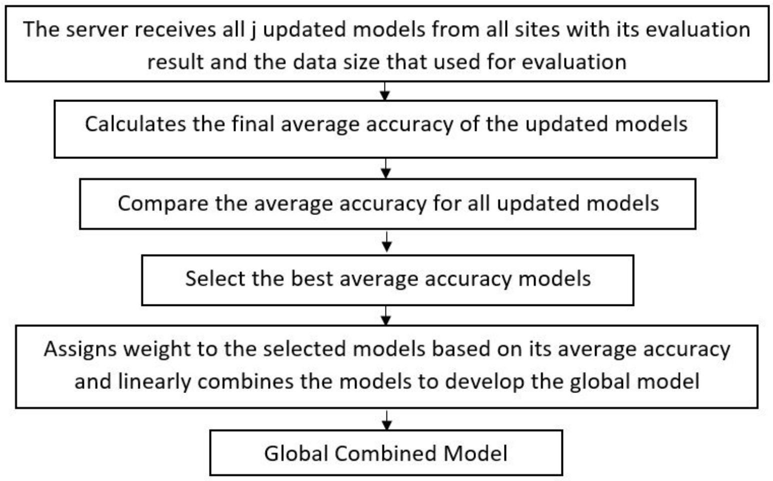


Figure 5.2. Final Models Selection and Combination Methods

5.3.2. Local-level Modelling Approach

We develop model selection and updating strategies to build a local combined model in each site. Our aim is to preserve data privacy by only passing the models between the distributed sites and updating these models locally in each site without exchanging models updates information. We used a simple linear combination method to combine the best-updated models to develop the local combined model with less information sharing between the distributed sites.

In the local-level modelling, each site tries to find the best local combined model by utilising the local data resource and the prediction models from other sites. Each site evaluates other sites models based on its local data, compares the results with its best local model result that selected using 10-fold cross-validation, and selects the best models as candidates for the updating process and discards the other models. We perform the model selection approach for the best models before the model updating process instead of updating all received models to avoid models update computation overhead and decrease computation time. Then, the site updates the selected models using mini-batches stochastic gradient descent, evaluates the updated models, then selects the best-updated models to combine with its best local model using linear combination methods to build the local combined model. We aim to reduce model biases and errors in individual models when combining the models rather than selecting a single model.

The main advantages of such an idea are that, firstly, only the local models from the other sites are used, and therefore we save the cost of data transformation from one site to another. As we know, data transformation is time-consuming and costly if the datasets are large. Therefore, such an approach greatly improves computation effectiveness and efficiency. Secondly, as there is no data sharing or transformation, whereas the only information exchanged between different sites are local models and the evaluation results and data size, such an approach does not disclose the data resource and therefore preserves data privacy. In addition, it addresses individual model limitations by utilising distributed data resources to develop a combined prediction model and preserve the privacy of local data resources. The related model names and definitions used in our methodology are presented in Table 5.2.

Table 5.2. Models Names and Descriptions (2)

Model	Notation	Meaning
Best Accuracy Local Model	$\mathrm{M_{ij*}}$	The local model in site i that has the best accuracy using j* learning algorithm
Received Model	$\mathrm{M_{i'j}}$	Model in site i that received from other sites i′
Best Accuracy Model	$\mathrm{M_{i'j*}}$	The selected model from other sites i′ which is better than or equal to the best local model accuracy $\mathrm{M_{ij*}}$
Updated Model	$\mathrm{M_{i'j*}^{U-i}}$	The updated model $\mathrm{M_{i'j*}}$ using the site i local data

List of the Best Accuracy Models	M	A list of the best local model accuracy M_{ij*} and the selected updated models of the best accuracy from other sites $M_{i'j*}^{U-i}$ that will linearly be combined to build the local combined model
Local Combined Model	M_i^{L*}	The final local combined model in site i that combine the best local model accuracy of the site i with the best updated models from other sites

Each site locally performs stepwise model selection and model updating methods. As shown in Figure 5.3, each site i receives models from other sites $M_{i'j}$, evaluate these models over its local dataset D_i, and calculate the accuracy $Acc(M_{i'j})$, then:

1) Compare the accuracy Acc $(M_{i'j})$ of the received models with its best local model accuracy Acc (M_{ij*}).
2) Select the best model accuracy $M_{i'j*}$ from each site and discard the other models as follows:
   - **Method L1:** Select the best model accuracy $M_{i'j*}$ from site i′ if:

     $$\text{Acc } (M_{i'j}) >= \text{Acc } (M_{ij*})$$

   - **Method L2:** Select the best model from each site even if the selected model not performed better than the best local model. We proposed this method if the best local model result is better than the received models, but there is little difference between the best local model and the received model results.
3) After the selection process, update the selected models $M_{i'j*}$ using its local data by performing mini-batch SGD method as follows:
   a) Divide the local data D_i into batches b
   b) For each batch of p examples x_t with corresponding target value y_t, t = 1,2,3, …, p:
      1. Compute the gradient estimate of the loss function [143,161]:

         $$g = \frac{1}{p} \nabla_w \sum_{t=1}^{p} \ell \left(M_{i'j*} (x_t ; w), y_t \right) \qquad (5.6)$$

      2. Update model parameter $w = w - \gamma g$, where γ is the learning rate and w is the model parameter
4) Evaluate the updated models $M_{i'j*}^{U-i}$ and select the best-performed updated models and discard the other updated models using a defined model

performance threshold.

5) Combine the best local model M_{ij*} and the selected updated models $M_{i'j*}^{U-i}$ using the linear combination to develop the local combined model M_i^{L*} as follows:

   a) Site i has the best models M, where M is the list of the best local model and the selected updated models from other sites M_i^*, M= $\{M_1^*, M_2^*, \ldots, M_n^*\}$ and their accuracy Acc (M_i^*).

   b) Calculate models weights for the models M_i^*, the model weight is computed based on its average accuracy [30, 86]:

$$w_{M_i^*} = \frac{\text{Acc } M_i^*}{\sum_{i=1}^{n} \text{Acc } M_i^*}, \qquad (5.7)$$

   where i=1, 2,… ,n, 0 <= $w_{M_i^*}$ <=1 and $\sum_{i=1}^{n} w_{M_i^*} = 1$

   c) Combine the models to develop the proposed local combined model M_i^{L*} by using the weighted voting method to predict x.

$$M_i^{L*}(x) = \max \sum_{i=1}^{n} w_{M_i^*} M_{i\,(x)}^* \qquad (5.8)$$

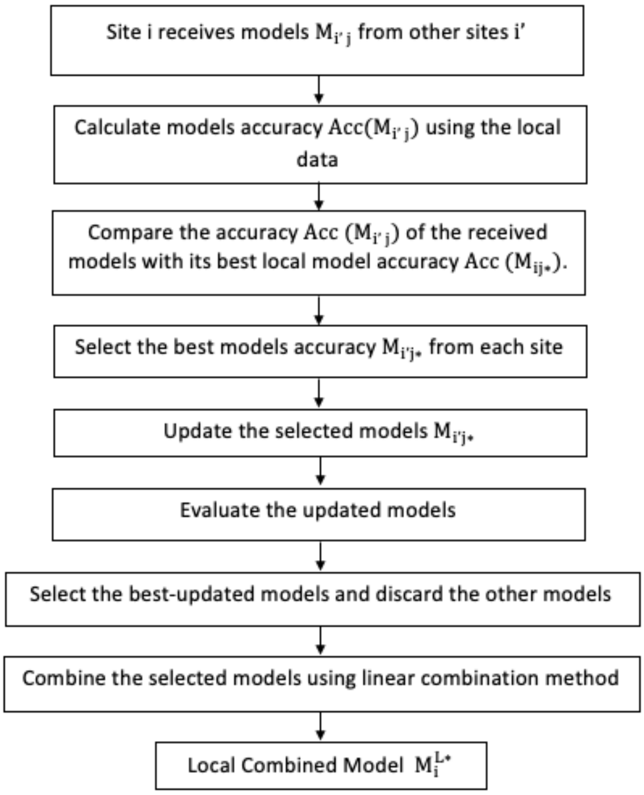


Figure 5.3. Local-level Modelling Method

5.3.3. Experimental Study

We evaluated and compared the prediction performance of the proposed global and local level modelling methods with different studies and centralised learning approach that moves all distributed data to a centralised database.

I. Distributed Data Simulation

We applied our methods to the datasets that used for the proposed approach using linear combination method in Chapter 3 (section 3.3.3): blood transfusion, liver disease, diabetes, heart disease, lower back pain (spine disease), Breast Cancer Wisconsin (Diagnostic), Breast Cancer Wisconsin (Original) [29], and cardiovascular diseases [27]. Also, we applied the data partitioning strategies that used in chapter 3 (section 3.3.3) to simulate distributed site data: (1) random data partitioning and (2) non-random data partitioning. Therefore, we divided each dataset into different parts as distributed sites site1, site2, and site3. At each site, the dataset is divided into local and validation data. The local data partition is used to develop the local models, evaluate the received models, and update the selected models. The validation data is used to evaluate the final global and local combined model.

II. Models Building and Evaluation

We applied different learning methods that allow for frequent model updating processes when new data is available. The learning algorithms that are used for model building and updating methods are Support Vector Machine (linear and nonlinear SVM), Neural Network (NN), Naïve Bayes (NB), Random Forest (RF), Decision Tree (DT), and Logistic Regression (LR). We used **SGD Classifier** in [174, 175] to implement Stochastic Gradient Descent learning for SVM and LR. For the model update approach, some Python methods have been used in DT, RF, linear and nonlinear SVM, NN, and LR algorithms to reuse a trained model in its previous state to start the model updating process [175]. We updated the model weights using mini-batch SGD in linear and nonlinear SVM, NN, LR, and NB algorithms. Not all learning algorithms use gradient descent algorithm for model training and updating methods, such as decision tree algorithms. In DT and RF, the model update approach adds successive trees to the previous model. In DT, we used Gradient Boosting to produce a prediction model in the form of an ensemble of prediction models [174, 175]. Each model tries to improve on its predecessor by reducing the errors. But instead of fitting a model on the data at each iteration, it fits a new model to the residual errors made by the previous model. It builds trees one by one in a sequential method, and each tree needs the results of the previous tree. In each step, the models fit on the negative gradient of the loss function.

In addition, we observed the effects of the learning rate, iterations number, and mini-batches SGD updates on model performance during updating process in blood transfusion dataset. The learning rate controls how much to change the model according to the estimated model error each time the model weights are updated. Figure 5.4 shows the impact of learning rate on DT model performance of site 1 during updating process using site 2 local data. In our method, we chose the learning rate value = 0.1.

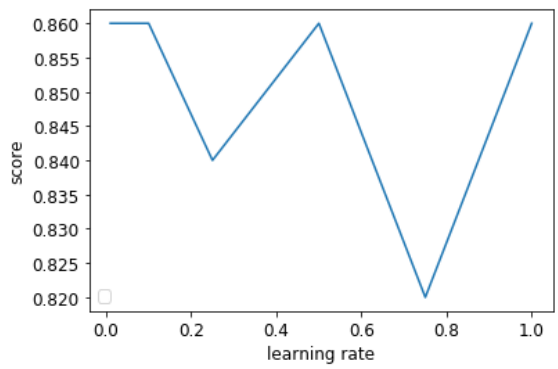
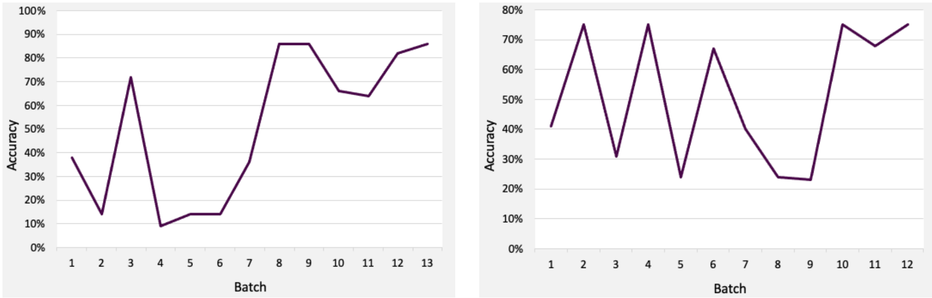


Figure 5.4. Learning Rate Effect on DT Model performance

Figure 5.5 shows site 1 NN model accuracy during updating process using mini-batch stochastic gradient descent using: (a) site 2 local data and (b) site 3 local data. As shown, NN model performance is optimised during mini-batch SGD updates.



(a) using site 2 local data (b) using site 3 local data

Figure 5.5. NN model accuracy during updating process using mini-batch stochastic gradient descent

The iterations number in SGD is the number of complete passes of the entire training dataset during the learning process. Figure 5.6 illustrates the effect of iteration number in LR model using SGD in site 1. The best accuracy when we used 10 iterations in site1 = 76 %, and the worst accuracy was 52 % when we chose 300 iterations.

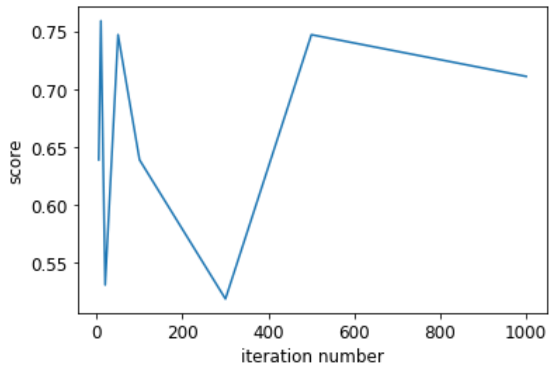


Figure 5.6. Effect of Iteration Number in LR Model Using SGD

In NN algorithm, we used adaptive learning rate method with the initial learning rate value = 0.001. The adaptive learning method keeps the learning rate fixed to the specified learning rate (initial learning rate) as long as model loss decreases or performance improves. If not, it divides the current learning rate by 5 [175]. As shown in Figure 5.7, adaptive performed better than constant and invscaling learning rate methods. Constant learning rate method keeps the learning rate fixed, while in invscaling, the learning rate at each time step gradually decreases through a learning process. In addition, we used adaptive learning rate method in LR and SVR algorithms [175].

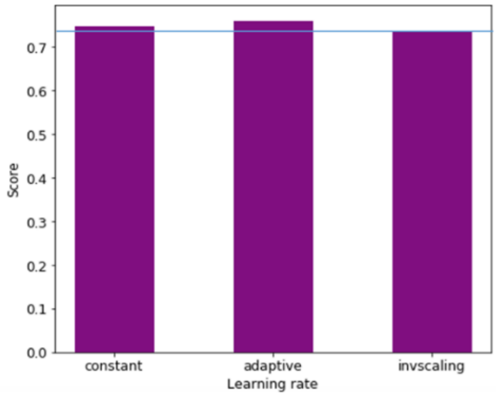


Figure 5.7. Effect of Learning Rate on NN Model using SGD in Site 1

III. Experiments Results

We compared the global combined model results with other combination methods: average accuracy and majority voting. Also, compared with the best-updated model (Single Best Model). For each site, we compared the local combined model with the best local model.

1) Random Data Partitioning:

i. Global-level modelling results:

The detailed results for the global-level modelling are illustrated in Appendix C. We evaluated the global combined models in each site instead of the server because the sites will not share their validation data with the server. The evaluation results are sent to the server with the test data size. Finally, the server received the evaluation results from the sites and calculated the global average accuracy of the global combined models. Our method outperformed the other combination methods. Table 5.3 shows the global combined model evaluation results for heart disease, breast cancer Wisconsin (Diagnostic), diabetes, liver disease, blood transfusion, lower back pain (spine disease), breast cancer Wisconsin (Original), and cardiovascular diseases datasets and compared to average accuracy and majority voting combination methods and with the best-updated model. Our proposed method outperformed the others in all datasets except in Breast Cancer Wisconsin (Diagnostic), Diabetes, and cardiovascular disease datasets; majority voting got similar results with our method.

Table 5.3. Global Combined Model Evaluation for Random Partitioned Data

Dataset	Global Combined Model			Single Best Model
	Weighted Voting	Average Accuracy	Majority Voting	
Heart Disease	**90%**	84%	88%	86% (NN-S213)
Breast Cancer Wisconsin (Diagnostic)	**98%**	90%	**98%**	86% (SVM linear-S321)
Diabetes	**81%**	73%	**81%**	76% (DT-S231)
Liver Disease	**83%**	69%	79%	65% (DT-S312)
Blood Transfusion	**63%**	47%	59%	59% (RF-S231)
Lower back pain (spine disease)	**71%**	56%	66%	54% (DT-S231)
Breast Cancer Wisconsin (Original)	**97%**	91%	94%	94% (LR-S231)
Cardiovascular disease	**73%**	70%	**73%**	**73%** (RF-S231, DT-S213)

In Table 5.4, we compared the proposed method using stepwise model selection and Gossip learning approaches for the randomly partitioned datasets with our proposed linear and nonlinear combination methods in chapters 3 and 4, respectively, and with the centralised learning approach. Our proposed method using stepwise model selection strategy outperformed the centralised learning in blood transfusion, breast cancer Wisconsin (Diagnostic), diabetes, lower back pain (spine disease), and liver disease datasets, close results in breast cancer Wisconsin (Original) and heart disease datasets, and a similar result in cardiovascular disease dataset. Furthermore, compared to the proposed linear and nonlinear methods, our method is the best in blood transfusion, Breast Cancer Wisconsin (Diagnostic), diabetes, heart disease, spine disease, liver disease datasets, a similar result in cardiovascular disease dataset, and close result in Breast Cancer Wisconsin (Original) dataset.

Table 5.4. All Proposed Methods Evaluation for Random Partitioned Data

Dataset	Linear combination method	Nonlinear combination method	Stepwise Model Selection Approach	Centralised learning
Blood transfusion	59%	57%	**63%**	60%
Breast Cancer Wisconsin (Diagnostic)	94%	94%	**98%**	96%

Breast Cancer Wisconsin (Original)	**98%**	**98%**	97%	**98%**
Diabetes	78%	80%	**81%**	78%
Heart Disease	86%	89%	90%	**92%**
Lower back pain (spine disease)	59%	62%	**71%**	66%
Liver Disease	65%	68%	**83%**	78%
Cardiovascular disease	**73%**	**73%**	**73%**	**73%**

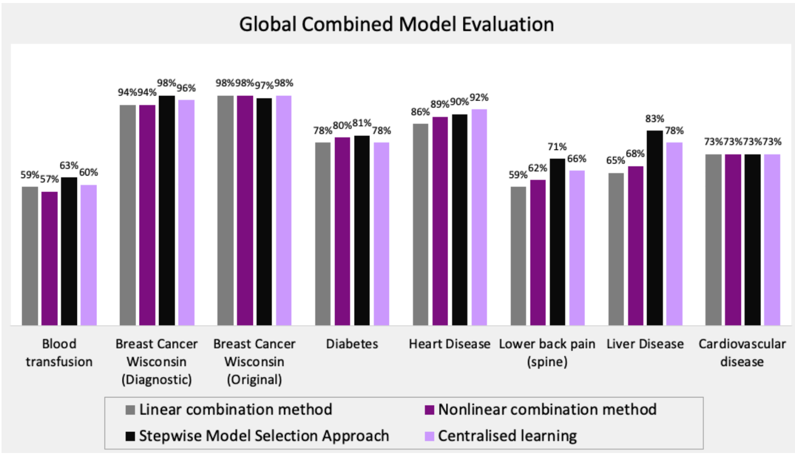


Figure 5.8. Global Combined Model Evaluation vs Centralised Learning for Random Partitioned Data

Table 5.5 shows the prediction accuracy of our proposed method using stepwise model selection for the data partitioned randomly and compared with the proposed linear and nonlinear methods and with different distributed learning methods proposed in [6, 48, 96, 166, 170, 176-182]. The proposed method using stepwise model selection outperformed the related works in breast cancer Wisconsin (Diagnostic), diabetes, heart disease, and liver disease datasets and got close result in breast cancer Wisconsin (Original) dataset.

Table 5.5. Distributed Learning Methods Evaluation for Random Partitioned Data

Method	Breast Cancer Wisconsin (Diagnostic)	Breast Cancer Wisconsin (Original)	Diabetes	Heart Disease	Liver Disease
Linear Combination Method	94%	**98%**	78%	86%	65%
Nonlinear Combination Method	94%	**98%**	80%	89%	68%
Stepwise Model Selection Approach	**98%**	97%	**81%**	**90%**	**83%**
Tsoumakas et al. [6] - EV1	-	97%	77%	84%	-
Tsoumakas et al. [6] - EV2	-	97%	77%	83%	-
Tsoumakas et al. [6] - EV 3	-	97%	77%	85%	-
Bashir et al. [48]	-	97%	77%	84%	71%

Zhang et al. [96]	-	-	80%	-	-
Mandal et al. [166]	96%	-	76%	-	-
Wang et al. [170]	-	96%	77%	-	-
Gao et al. [176]	-	95%	-	72%	-
Kasturi et al. [177]	-	96%	-	-	-
Ma et al. [178]	-	96%	-	-	-
Haque et al. [179]	-	**98%**	78%	82%	-
Sav et al. [180]	-	97%	-	-	-
Froelicher et al. [181]	-	96%	78%	-	-
Ed-daoudy and Maalmi [182]	-	-	-	82%	-

ii. Local-level modelling results:

The local-level modelling detailed results are illustrated in Appendix C. Tables 5.6 - 5.13 present the evaluation results for blood transfusion, breast cancer Wisconsin (Diagnostic), diabetes, heart disease, liver disease, lower back pain (spine disease), breast cancer Wisconsin (Original), and cardiovascular diseases datasets. It compares it with average accuracy, majority voting methods, and the best local model. Our method performance is better or similar to the best local model in most of the datasets, except in spine disease dataset in site 1 and site 3, and in breast cancer Wisconsin (Diagnostic) dataset in site 2. Thus, we conclude that the distributed sites can utilise other sites models to improve the prediction accuracy.

Table 5.6.Local Combined Model Evaluation for Blood Transfusion Dataset

Methods	Combination method	Site1	Site2	Site3
Local-Level Modelling	Weighted Voting	**51%**	**60%**	**66%**
	Average Accuracy	**51%**	58%	**66%**
	Majority Voting	**51%**	57%	**66%**
The Best Local Model		**51%**	**60%**	**66%**

Table 5.7. Local Combined Model Evaluation for Breast Cancer Wisconsin (Diagnostic) dataset

Methods	Combination method	Site1	Site2	Site3
Local-Level Modelling	Weighted Voting	**95%**	**99%**	**84%**
	Average Accuracy	**95%**	90%	82%
	Majority Voting	**95%**	**99%**	**84%**
The Best Local Model		**95%**	**99%**	**84%**

Table 5.8. Local Combined Model Evaluation for Diabetes dataset

Methods	Combination method	Site1	Site2	Site3
Local-Level Modelling	Weighted Voting	**75%**	**80%**	**60%**
	Average Accuracy	71%	78%	59%
	Majority Voting	**75%**	**80%**	56%
The Best Local Model		70%	**80%**	56%

Table 5.9. Local Combined Model Evaluation for Heart disease dataset

Methods	Combination method	Site1	Site2	Site3
Local-Level Modelling	Weighted Voting	**73%**	**90%**	73%
	Average Accuracy	71%	**90%**	**75%**
	Majority Voting	**73%**	**90%**	73%
The Best Local Model		60%	**90%**	67%

Table 5.10. Local Combined Model Evaluation for Lower back pain (spine disease) dataset

Methods	Combination method	Site1	Site2	Site3
Local-Level Modelling	Weighted Voting	55%	**71%**	62%
	Average Accuracy	62%	67%	75%
	Majority Voting	**65%**	64%	69%
The Best Local Model		**65%**	**71%**	**88%**

Table 5.11. Local Combined Model Evaluation for Breast Cancer Wisconsin (Original) dataset

Methods	Combination method	Site1	Site2	Site3
Local-Level Modelling	Weighted Voting	97%	96%	**98%**
	Average Accuracy	92%	97%	63%
	Majority Voting	**98%**	96%	90%
The Best Local Model		97%	**99%**	**98%**

Table 5.12. Local Combined Model Evaluation for Liver disease dataset

Methods	Combination method	Site1	Site2	Site3
Local-Level Modelling	Weighted Voting	**72%**	**75%**	67%
	Average Accuracy	71%	73%	70%
	Majority Voting	**72%**	72%	**73%**
The Best Local Model		**72%**	68%	67%

Table 5.13. Local Combined Model Evaluation for Cardiovascular disease dataset

Methods	Combination method	Site1	Site2	Site3
Local-Level Modelling	Weighted Voting	**74%**	**73%**	**72%**
	Average Accuracy	**74%**	**73%**	51%
	Majority Voting	**74%**	**73%**	**72%**
The Best Local Model		**74%**	**73%**	**72%**

Table 5.14 shows the evaluation results for our proposed local level modelling method compared with our proposed linear and nonlinear combination methods for the randomly partitioned datasets. The results are compared with the best local model in each site. The proposed method using stepwise model selection got better or similar results compared with the best local model results, except in spine disease dataset in two sites. The nonlinear combination method performed better than other methods in most sites datasets. In diabetes, the local combined model of the proposed linear and nonlinear methods is better than the best local model in all sites. Our proposed method using stepwise model selection proved that we could utilise other sites models to build an accurate combined model without sharing the data between the sites to preserve the data privacy for each site.

Table 5.14. Local Combined Model Evaluation for Random Partitioned Data

Dataset	Linear combination method	Nonlinear combination method	Stepwise Model Selection Approach
Blood transfusion	**The best in all sites**	Better in two sites and similar in one site	Similar in all sites
Breast Cancer Wisconsin (Diagnostic)	Similar in two sites	Better in two sites and similar in one site	Similar in all sites
Breast Cancer Wisconsin (Original)	Similar in two sites	Similar in all sites	Similar in two sites
Diabetes	**The best in all sites**	**The best in all sites**	Better in two sites and similar in one site
Heart Disease	Better in two sites	Better in one site and similar in one site	Better in two sites and similar in one site
Lower back pain (spine disease)	Better in one site and similar in one site	Better in one site and similar in two sites	Similar in one site
Liver Disease	Similar in all sites	Better in two sites and similar in one site	Better in one site and similar in two sites
Cardiovascular disease	Similar in two sites	Better in one site and similar in two sites	Similar in all sites

2) **Non-random Data Partitioning:**

i. **Global-level Modelling Results:**

Table 5.15 shows the global combined model evaluation results for diabetes, heart disease, and liver disease datasets that partitioned non-randomly and compared with other model combination methods and the best-updated model. The results of our approach are not far from the best results. The best-updated model (Single Best Model) is the best in heart disease and liver disease datasets. In diabetes dataset, the majority voting method is the best.

Table 5.15. Global Combined Model Evaluation for Non-random Partitioned Data

Dataset	Global Combined Model			Single Best Model
	Weighted Voting	Average Accuracy	Majority Voting	
Diabetes	67%	59%	**68%**	60% (NN-S123)
Heart Disease	82%	68%	82%	**85%** (NN-S213)
Liver Disease	71%	71%	70%	**72%** (NN-S231)

Table 5.16 shows the evaluation results for our proposed method using stepwise model selection compared with the proposed linear and nonlinear model combination methods and the centralised learning approach. The centralised learning approach is slightly better than our methods, but its results are close to our proposed methods. As we discussed, the centralised learning approach has many issues and is inadequate for private and un-exchangeable data in a distributed environment. Compared to the proposed linear and nonlinear methods, the results are similar in diabetes dataset and close results in heart and liver disease datasets.

Table 5.16. Global Combined Model and Centralised Learning Evaluation for Non-random Partitioned Data

Dataset	Linear combination method	Nonlinear combination method	Stepwise Model Selection Approach	Centralised learning
Diabetes	67%	67%	67%	**69%**
Heart Disease	87%	83%	82%	**88%**
Liver Disease	72%	72%	71%	**74%**

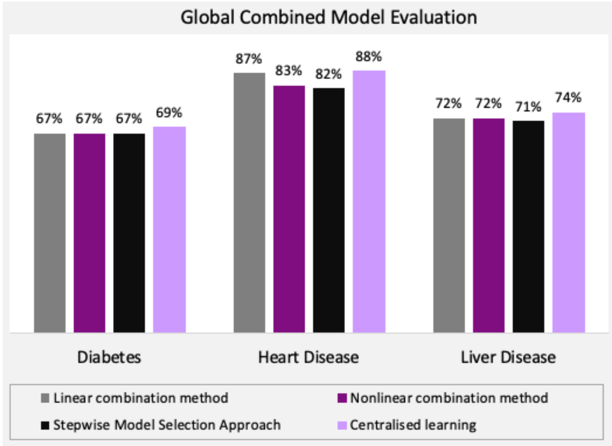


Figure 5.9. Global Combined Model Evaluation vs Centralised Learning for Non-random Partitioned Data

Table 5.17 shows the evaluation of our proposed methods and the research works [6, 48, 96, 166, 170, 176, 179, 181, 182]. It shows that the linear combination method is slightly better than other approaches in the heart disease dataset. In liver disease dataset, all methods got close and better results, and for diabetes dataset, [96] outperformed the other methods.

Table 5.17. Distributed Learning Methods Evaluation for Non-random Partitioned Data

Method	Diabetes	Heart Disease	Liver Disease
Linear Combination Method	67%	**87%**	**72%**
Nonlinear Combination Method	67%	83%	**72%**
Stepwise Model Selection Approach	67%	82%	71%
Tsoumakas et al. [6] - EV1	77%	84%	-
Tsoumakas et al. [6] - EV2	77%	83%	-
Tsoumakas et al. [6] - EV 3	77%	85%	-
Bashir et al. [48]	77%	84%	**72%**
Zhang et al. [96]	**80%**	-	-
Mandal et al. [166]	76%	-	-
Wang et al. [170]	**77%**	-	-
Gao et al. [176]	-	72%	-
Haque et al. [179]	78%	82%	-
Froelicher et al. [181]	78%	-	-
Ed-daoudy and Maalmi [182]	-	82%	-

ii. Local-level Modelling Results:

The following Tables 5.18 – 5.20 show the local combined model accuracy in each site for diabetes, heart disease, and liver disease datasets compared with the best local model and other model combination methods. Our method got better or similar results than the best local model except in diabetes dataset in site 1, heart disease dataset in site 2, and liver disease dataset in site 1.

Table 5.18. Local Combined Model Evaluation for Diabetes dataset

Methods	Combination method	Site1	Site2	Site3
Local-Level Modelling	Weighted Voting	72%	**63%**	58%
	Average Accuracy	73%	**63%**	64%
	Majority Voting	**74%**	**63%**	**65%**
The Best Local Model		73%	**63%**	52%

Table 5.19. Local Combined Model Evaluation for Heart disease dataset

Methods	Combination method	Site1	Site2	Site3
Local-Level Modelling	Weighted Voting	**80%**	85%	**86%**
	Average Accuracy	78%	83%	77%
	Majority Voting	**80%**	85%	**86%**
The Best Local Model		75%	**88%**	**86%**

Table 5.20. Local Combined Model Evaluation for Liver disease dataset

Methods	Combination method	Site1	Site2	Site3
Local-Level Modelling	Weighted Voting	40%	**76%**	**94%**
	Average Accuracy	45%	75%	78%
	Majority Voting	45%	**76%**	82%
The Best Local Model		**50%**	**76%**	71%

As shown in Table 5.21, the proposed methods got better or similar results in most datasets compared with the best local model in each site. Therefore, the sites are utilised from other sites models using the proposed methods.

Table 5.21. Local Combined Model Evaluation for Non-random Partitioned Data

Dataset	Linear combination method	Nonlinear combination method	Stepwise Model Selection Approach
Diabetes	Better in two sites	**The best in all sites**	Better in one site and similar in one site
Heart Disease	Better in one site and similar in one site	Better in one site and similar in two sites	Better in one site and similar in one site
Liver Disease	Better in one site	Better in one site and similar in one site	Better in one site and similar in one site

5.3.4. Discussion and Evaluation

The global combined model that developed using stepwise model selection and Gossip learning approaches showed improved prediction accuracy in most datasets compared to the centralised learning approach and the linear and nonlinear combination methods, especially in spine disease and liver disease datasets that are partitioned randomly. For the non-randomly partitioned datasets, our proposed decentralsied learning using stepwise model selection method got comparable results with the proposed linear and nonlinear combination methods and the centralised learning approach. Thus, the results showed that our proposed method could perform better or be comparable to the centralised learning approach without sharing distributed datasets to preserve data privacy. We saved the cost and time of data transformation from one site to another or data centralisation. Also, avoid using a server for controlling the learning process to avoid server issues and iterative communication and computation overheads. Therefore, we developed the global combined model and performed the models building and updating processes locally to preserve data privacy. Furthermore, the local combined model results showed that we could utilise other sites models to build an optimal and accurate combined model without sharing the data between the sites to preserve the data privacy for each site.

The impact of sites number on the combined model performance is not our focus in this thesis. However, increased computation and communication overheads, time-consuming, and scalability issues may arise if we deal with large sites number. It involves exchanging model rounds to compute the global average performance. These issues can be addressed by developing different selective decentralised learning strategies. For example, we could develop a search strategy that only considers the best sites contributions by examining its local models performance to include these sites in the decentralised learning process. This may reduce computation and communication overheads and time and enhance the proposed method scalability. In addition, there is a possibility for malicious attacks on the trained models to reveal meaningful information or retrieve training data. In future research, we will consider malicious attacks possibility on distributed sites and exchanged models.

5.4 PROPOSED METHOD FOR REGRESSION ALGORITHMS

Our method develops a combined model at global and local levels using a decentralised learning method. First, we apply the model selection strategy using stepwise model selection method, then update the selected models using mini-batch stochastic gradient descent. Finally, apply the linear combination method to develop the combined model for the final selected models.

In global and local level modelling approaches, we apply the methods used in classification algorithms in sections 5.3.1 and 5.3.2 and use RMSE and MAPE metrics for model performance evaluation instead of accuracy metric. We use RMSE for the model selection approach and apply the linear model combination method that assigns model weight by using simple weight average, error-based (RMSE), and performance-based (Accuracy) approaches.

5.4.1. Experimental Study

We conducted experiments to evaluate the prediction performance of the proposed method. We compared it with a centralised learning approach that moves all distributed data to a centralised database and with a distributed learning work [166].

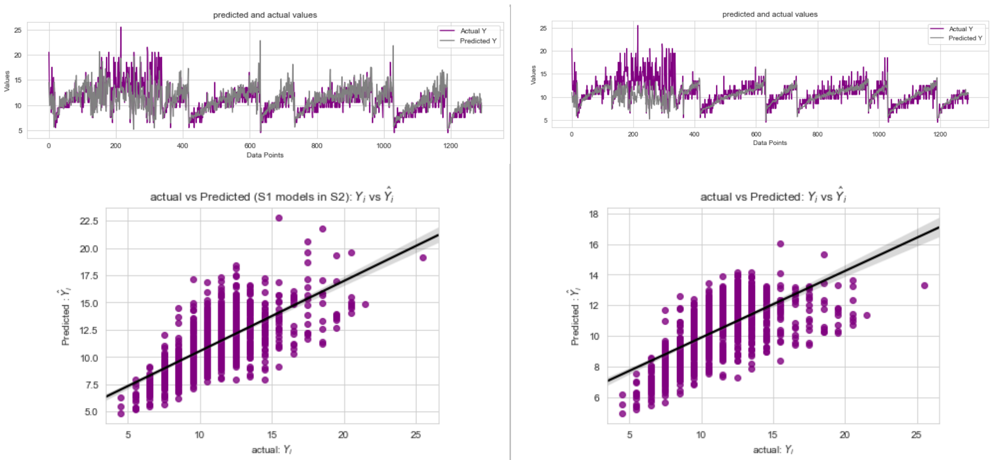
I. Distributed Data Simulation

We applied our methods to the datasets that used in chapter 3 (section 3.4.3): Parkinson disease, Boston housing, and Abalone datasets [29]. Before performing the experiments, the databases are first preprocessed to a suitable data format. We used data standardisation to adjust features to a common scale, such that the processed features have a mean of 0 and a standard deviation of 1. It is performed by subtracting the mean and then dividing by its standard deviation, and it is recommended to optimise the learning process in SGD [173-175]. Next, we applied data partitioning strategies that used in chapter 3 (section 3.4.3) to simulate distributed sites data: (1) random data partitioning and (2) non-random data partitioning. We assume that the distributed sites are homogeneous, and the data are independent and identically distributed in this study. We divided each dataset into different parts as distributed sites site1, site2, and site3. At each site, the dataset is divided into local and validation data. The local data partition is used to develop the local models, evaluate the received models, and update the selected models. The validation data is used to assess the final global and local combined model.

II. Models Building and Evaluation

We used different regression algorithms for model building and updating methods. The algorithms are Linear Regression (LR), Support Vector Regression (SVR), Decision Tree Regression (DTR), Neural Network Regressor (NNR), Random Forest Regressor (RFR), Least Absolute Shrinkage and Selection Operator regression (LASSO), Ridge, and ElasticNet. For LR, SVR, NNR, LASSO, Ridge, and ElasticNet, we used mini-batch Stochastic Gradient Descent, while in RFR and DT, we added model to the previous model. We applied **SGD Regressor** to implement Stochastic Gradient Descent (SGD) learning for LR, SVR, LASSO, Ridge, and ElasticNet algorithms [173-175]. Some Python methods have been used in LR, SVR, NNR, DTR, RFR, LASSO, Ridge, and ElasticNet algorithms to reuse a trained model in its previous state to perform the model update method [175]. Also, we updated the weights of an existing model using mini-batch SGD in LR, SVR, NNR, LASSO, Ridge, and ElasticNet algorithms [174, 175]. In DTR, we used Gradient Boosting Regressor to build the model in an additive fashion. In each step, a regression tree is fit on the negative gradient of the loss function. RFR is an ensemble method; we reused the previous model, trained different decision trees on data samples, and added it to the ensemble model [173-175]. The learning rate value for DTR model is 0.1, LR, SVR, LASSO, Ridge, and ElasticNet models is 0.01 with invscaling learning rate method, and in NNR model is 0.001 with constant learning rate method. For model evaluation and selection approaches, we used RMSE evaluation metric.

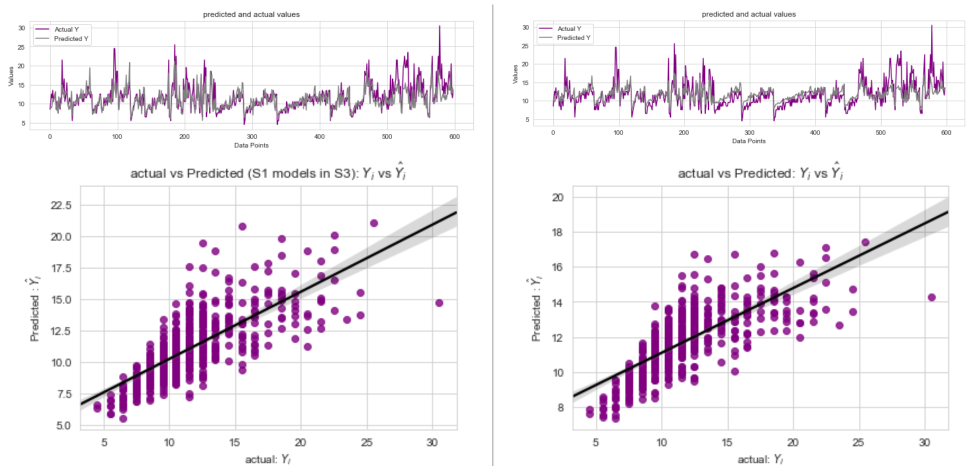
Figures 5.10 and 5.11 illustrate model updating method using mini-batch SGD for the best average RMSE model using other sites data. It shows LR model of site 1 for Abalone dataset that randomly partitioned before and after updating in site 2 and 3. In site 2, LR model RMSE = 1.99 before updating, and RMSE = 2.08 after updating process. While in site3, LR model RMSE = 2.40 before updating, and RMSE = 2.49 after updating process.



(a) Before updating the model

(b) After updating the model

Figure 5.10. Site 1 LR Model in site 2

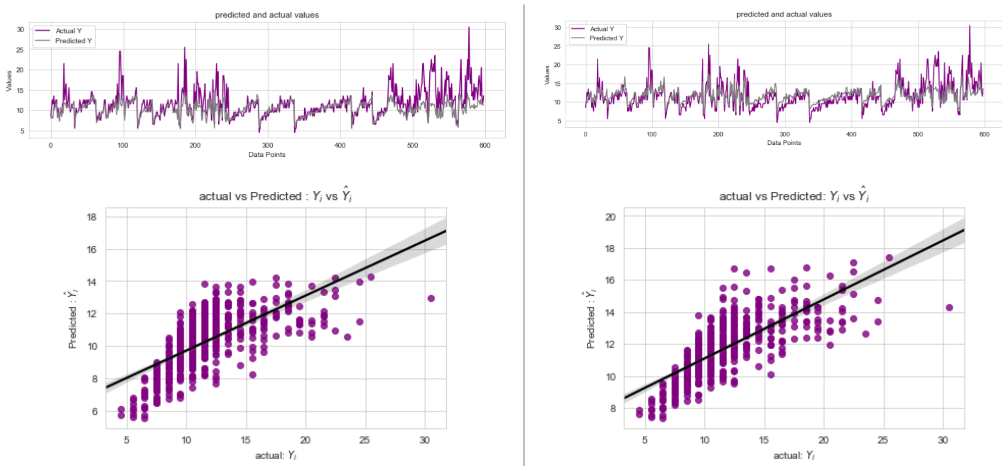


(a) Before updating the model

(b) After updating the model

Figure 5.11. Site 1 LR Model in site 3

Then, after evaluating the two updated models in all sites, we found that the LR model that updated in site 2 got better average RMSE than LR model that updated in site 3. Therefore, we select and send the best average RMSE model to the other sites that still not update this model (i.e. site 3). Figure 5.12 shows LR model before and after updating process in site 3. LR model RMSE = 2.97 before updating, and RMSE = 2.49 after updating process. LR model performance (RMSE) is improved after updating process.



(a) Before updating the model

(b) After updating the model

Figure 5.12. LR updated Model in Site 3

III. Experiments Results

We show the global and local level modelling results for the datasets partitioned in random and non-random scenarios. We compared the global combined model results with the best updated model (Single Best Model). In addition, for each site, we compared the local combined model with the best local model.

1) Random data partitioning:

i. Global-level modelling results:

The detailed results for the global-level modelling are illustrated in Appendix C. We evaluated the global combined models in each site instead of the server because the sites will not share their validation data with the server. Then, we sent the evaluation results to the server with the number of test data samples. Finally, the server receives the evaluation results from the sites, and then the global average RMSE and MAPE of the global combined models are calculated. Table 5.22 shows the global combined model evaluation results for Abalone, Parkinson (Total UPDRS), Parkinson (Motor UPDRS), and Boston Housing datasets that are partitioned randomly and compared with the best updated model (single best model). For Abalone dataset, all combination methods got similar and better results, while the linear model combination using the simple average method is slightly better than other methods in Parkinson (Motor UPDRS) and Boston Housing datasets. Besides, the best updated model in Parkinson (Total UPDRS) dataset performed better than the global combined models, and in the other datasets, the results are close.

Table 5.22. Global Combined Model Evaluation for Random Partitioned Data

Dataset	Global Combined Model						Single Best Model	
	Simple average		Error-based (RMSE)		Performance-based (Accuracy)			
	RMSE	MAPE	RMSE	MAPE	RMSE	MAPE	RMSE	MAPE
Abalone	2.4	13%	2.4	13%	2.4	13%	2.4 (LASSO – S231)	14%
Parkinson (Total UPDRS)	10.06	47%	10.06	47%	10.09	48%	**8.97** (RFR – S312)	**41%**
Parkinson (Motor UPDRS)	8.84	62%	8.88	63%	8.90	63%	**8.70** (RFR – S312)	**60%**
Boston Housing	**4.1**	**16%**	4.2	**16%**	4.2	**16%**	4.4 (RFR – S312)	**16%**

In Table 5.23, we compared the proposed method using stepwise model selection approach for the randomly partitioned datasets with our proposed linear and nonlinear combination and the centralised learning approach. The centralised learning approach got better RMSE results in Parkinson disease (Total UPDRS) and Parkinson disease (Motor UPDRS) datasets. Our proposed linear combination method outperformed the other methods and the centralised learning approach for Abalone and Boston Housing datasets.

Table 5.23. The Proposed Methods and Centralised Learning Approach Evaluation for Random Partitioned Data

Dataset	Linear combination method	Nonlinear combination method	Stepwise Model Selection Approach	Centralised learning
Parkinson disease (Total UPDRS)	8.97	10.16	10.06	**7.53**
Parkinson disease (Motor UPDRS)	7.30	8.22	8.84	**6.98**
Abalone	**2.36**	2.43	2.40	2.39
Boston Housing	**3.10**	3.55	4.10	3.20

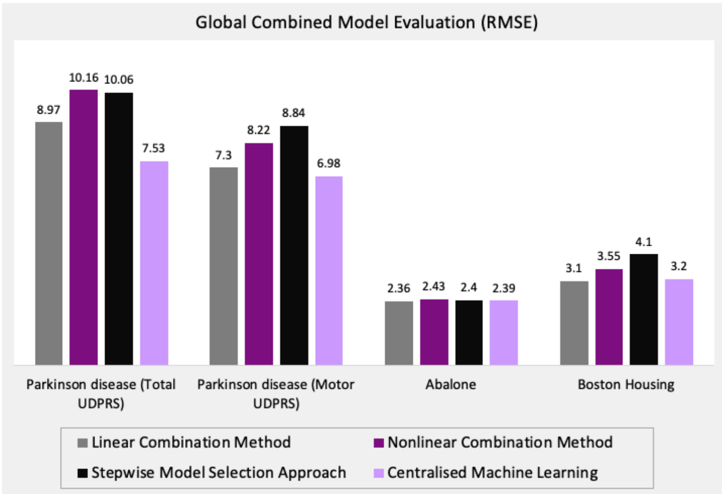


Figure 5.13. The Proposed Methods and Centralised Learning Approach Evaluation for Random Partitioned Data

Table 5.24 shows the RMSE results of our proposed method using stepwise model selection approach for Boston housing dataset compared with our proposed linear and nonlinear methods and the distributed learning method proposed in [166]. Our proposed linear combination method performed better than the other methods.

Table 5.24. Distributed Learning Methods Evaluation for Boston Housing Dataset

Method	Boston Housing
Linear Combination Method	**3.10**
Nonlinear Combination Method	3.55
Stepwise Model Selection Approach	4.10
Mandal et al. [166]	4.91

ii. Local-level Modelling Results:

Tables 5.25 - 5.28 show the local-level modelling results for Parkinson (Total UPDRS), Parkinson (Motor UPDRS), Boston Housing, and Abalone datasets that partitioned randomly and compared with the best local model. Our method is better than the best local model in Parkinson (Motor UPDRS) dataset in all sites and better in two sites in Parkinson (Total UPDRS) and Boston Housing datasets. Furthermore, in Abalone dataset, we got better result in one site and close results with the other two sites. Thus, we conclude that the distributed sites can utilise other sites models to improve the prediction performance.

Table 5.25. Local Combined Model Evaluation for Parkinson (Total UPDRS) dataset

Method	Combination Method	Site1		Site2		Site3	
		RMSE	MAPE	RMSE	MAPE	RMSE	MAPE
Local-Level Modelling	Simple average	**12.11**	69.08	**7.77**	**25.71**	**13.02**	**58.52**
	Error-based (RMSE)	12.29	70.17	8.09	26.03	13.24	59.40
	Performance-based (Accuracy)	12.34	70.52	7.89	25.97	**13.02**	**58.52**
The Best Local Model (Single Model)		**12.11**	**67.16**	9.29	28.09	14.78	64.78

Table 5.26. Local Combined Model Evaluation for Parkinson (Motor UPDRS) dataset

Method	Combination Method	Site1		Site2		Site3	
		RMSE	MAPE	RMSE	MAPE	RMSE	MAPE
Local-Level Modelling	Simple average	8.83	69.62	9.01	54.46	**9.78**	**59.74**
	Error-based (RMSE)	**8.80**	**68.83**	9.01	54.46	10.14	62.80
	Performance-based (Accuracy)	8.89	70.05	**9.00**	**54.44**	**9.78**	**59.74**
The Best Local Model (Single Model)		9.39	71.46	9.51	55.45	11.97	75.33

Table 5.27. Local Combined Model Evaluation for Boston Housing dataset

Method	Combination Method	Site1		Site2		Site3	
		RMSE	MAPE	RMSE	MAPE	RMSE	MAPE
Local-Level Modelling	Simple average	3.75	**17.07**	5.68	21.66	**4.08**	**17.63**
	Error-based (RMSE)	3.67	17.19	7.34	28.02	4.97	20.84
	Performance-based (Accuracy)	3.70	17.21	**4.14**	**17.22**	4.32	20.37
The Best Local Model (Single Model)		**3.60**	17.29	9.51	36.83	6.38	29.38

Table 5.28. Local Combined Model Evaluation Results for Abalone dataset

Method	Combination Method	Site1		Site2		Site3	
		RMSE	MAPE	RMSE	MAPE	RMSE	MAPE
Local-Level Modelling	Simple average	**2.91**	**14.91**	2.39	12.57	3.46	18.52
	Error-based (RMSE)	3.47	17.74	2.39	12.57	3.28	17.40
	Performance-based (Accuracy)	3.47	17.74	2.39	12.57	4.12	23.82
The Best Local Model (Single Model)		3.18	16.16	**2.35**	**12.51**	**2.57**	**14.72**

Table 5.29 shows the proposed local level modelling results using stepwise model selection strategy compared with the proposed linear and nonlinear combination methods for randomly partitioned datasets. The results are compared with the best local model in each site. The proposed method using stepwise model selection got improved results in Parkinson disease (Motor UPDRS) dataset for all sites and better results in one or two sites for the other datasets. The nonlinear combination method outperformed the best local model in Parkinson disease (Motor UPDRS), Parkinson disease (Motor UPDRS), and Boston Housing datasets in all sites. Therefore, the distributed sites can improve the prediction performance using other sites models and without sharing data to preserve data privacy.

Table 5.29. The Proposed Methods Evaluation for Random Partitioned Data

Dataset	Linear combination method	Nonlinear combination method	Stepwise Model Selection Approach
Parkinson disease (Total UPDRS)	Better in two sites	The best in all sites	Better in two sites and similar in one site
Parkinson disease (Motor UPDRS)	Better in one site	The best in all sites	The best in all sites
Abalone	The best in all sites	Better in two sites and similar in one site	Better in one site
Boston Housing	Better in two sites	The best in all sites	Better in two sites

2) Non-random data partitioning:

i. Global-level modelling results:

Table 5.30 shows the global combined model evaluation results for Parkinson (Total UPDRS), Parkinson (Total UPDRS), and Boston Housing datasets that partition non-randomly and compared with the best updated model (single best model). The global combined model is slightly better than the best updated model in Parkinson datasets and comparable performance in Boston housing data.

Table 5.30. Global Combined Model Evaluation for Non-random Partitioned Data

Dataset	Global Combined Model						Single Best Model	
	Simple average		Error-based (RMSE)		Performance-based (Accuracy)			
	RMSE	MAPE	RMSE	MAPE	RMSE	MAPE	RMSE	MAPE
Parkinson (Total UPDRS)	**10.1**	**34%**	**10.1**	**34%**	10.2	**34%**	10.5 (RFR-S312)	**34%**
Parkinson (Motor UPDRS)	7.8	**32%**	**7.6**	**32%**	7.7	**32%**	8.02 (Ridge – S123)	39%
Boston Housing	**7.1**	**42%**	7.2	43%	7.2	43%	7.2 (DTR- S231)	43%

For the datasets that are non-randomly divided, Table 5.31 shows the results for proposed method using stepwise model selection approach compared with the linear and nonlinear model combination methods and the centralised learning approach. The centralised learning approach outperformed the other methods in Boston Housing dataset. The nonlinear combination method got better RMSE result in Parkinson disease (Total UPDRS) dataset, and the linear combination method is the best in Parkinson disease (Motor UPDRS) dataset. On the other hand, our method using stepwise model selection approach got lower performance in Parkinson disease (Total UPDRS) and Boston Housing datasets and close result to the other methods in Parkinson disease (Motor UPDRS) dataset.

Table 5.31. The Proposed Methods and Centralised Learning Approach Evaluation for Non-random Partitioned Data

Dataset	Linear combination method	Nonlinear combination method	Stepwise Model Selection Approach	Centralised learning
Parkinson disease (Total UPDRS)	8.19	**6.94**	10.1	8.17
Parkinson disease (Motor UPDRS)	**7.1**	8.51	7.6	7.3
Boston Housing	3.67	4.01	7.1	**3.38**

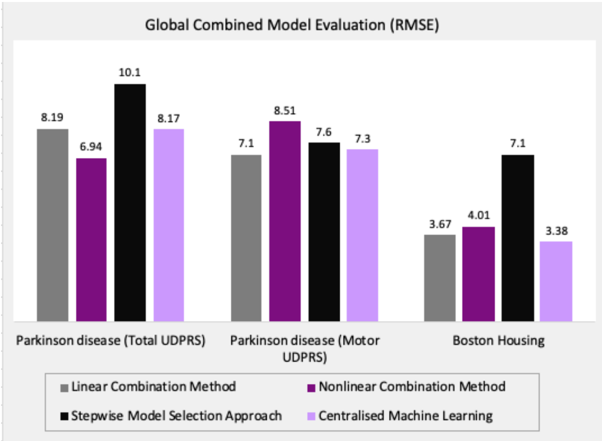


Figure 5.14. The Proposed Methods and Centralised Learning Approach Evaluation for Non-random Partitioned Data

Table 5.32 shows the RMSE results for Boston housing dataset that are non-random partitioned for all proposed methods and the proposed approach in [166]. Our proposed linear combination method outperformed the other approaches.

Table 5.32. Distributed Learning Methods Evaluation for Boston Housing Dataset

Method	Boston Housing
Linear Combination Method	**3.67**
Nonlinear Combination Method	4.01
Stepwise Model Selection Approach	7.1
Mandal et al. [166]	4.91

ii. Local-level modelling results:

Tables 5.33 - 5.35 show the local combined model results in each site for Parkinson (Total UPDRS), Parkinson (Total UPDRS), and Boston Housing dataset compared with the best local model. Our method outperformed the best local model in Parkinson (Total UPDRS) in site 2, and slightly better or close results to the best local in other datasets sites.

Table 5.33. Local Combined Model Evaluation for Parkinson (Total UPDRS) dataset

Method	Combination Method	Site1		Site2		Site3	
		RMSE	MAPE	RMSE	MAPE	RMSE	MAPE
Local-Level Modelling	Simple average	13.62	51.01	**5.25**	**20.20**	9.66	23.84
	Error-based (RMSE)	13.62	51.01	5.89	23.07	**9.62**	**23.78**
	Performance-based (Accuracy)	13.62	51.01	5.45	21.23	9.66	23.84
The Best Local Model (Single Model)		**13.21**	**48.11**	9.12	35.42	10.67	25.51

Table 5.34. Local Combined Model Evaluation for Parkinson (Motor UPDRS) Dataset

Method	Combination Method	Site1		Site2		Site3	
		RMSE	MAPE	RMSE	MAPE	RMSE	MAPE
Local-Level Modelling	Simple average	11.45	65.87	7.62	39.51	**8.18**	**25.13**
	Error-based (RMSE)	**11.36**	**65.36**	**6.91**	**36.43**	8.24	25.30
	Performance-based (Accuracy)	11.43	65.77	7.33	38.53	8.29	25.45
The Best Local Model (Single Model)		11.95	66.85	7.36	38.93	8.79	26.39

Table 5.35. Local Combined Model Evaluation for Boston Housing dataset

Method	Combination Method	Site1		Site2		Site3	
		RMSE	MAPE	RMSE	MAPE	RMSE	MAPE
Local-Level Modelling	Simple average	**2.38**	6.77	**5.09**	16.53	5.77	30.83
	Error-based (RMSE)	2.42	**6.74**	5.11	16.71	5.78	30.91
	Performance-based (Accuracy)	2.39	6.76	**5.09**	16.54	5.77	30.84
The Best Local Model (Single Model)		2.54	7.88	**5.09**	**15.99**	**5.75**	**30.72**

For the non-randomly partitioned datasets, as shown in Table 5.36, the proposed method using stepwise model selection approach got better RMSE results than the best local model in Parkinson disease (Motor UPDRS) dataset, and better results in one or two sites in Parkinson disease (Total UPDRS) and Boston

Housing datasets. The nonlinear combination method is the best in all datasets.

Table 5.36. The Proposed Methods Evaluation for Non-random Partitioned Data

Dataset	Linear combination method	Nonlinear combination method	Stepwise Model Selection Approach
Parkinson disease (Total UPDRS)	Better in two sites	The best in all sites	Better in two sites
Parkinson disease (Motor UPDRS)	Better in two sites	The best in all sites	The best in all sites
Boston Housing	Better in two sites	The best in all sites	Better in one site and similar in one site

5.4.2. Discussion and Evaluation

We compared the proposed method using stepwise model selection approach for the randomly partitioned datasets with our proposed linear and nonlinear combination and the centralised learning methods. All methods results are close in Abalone dataset, and the proposed linear combination method is slightly better than other methods and the centralised learning approach in Abalone and Boston Housing datasets. The centralised learning approach got better RMSE results in Parkinson disease (Total UPDRS) and Parkinson disease (Motor UPDRS) datasets. In non-randomly partitioned datasets, our method got lower performance in Parkinson disease (Total UPDRS) and Boston Housing datasets and a close result to the other methods in Parkinson disease (Motor UPDRS) dataset. The nonlinear combination method got better RMSE result in Parkinson disease (Total UPDRS) dataset, and the linear combination method is the best in Parkinson disease (Motor UPDRS) dataset. The centralised learning approach outperformed the other methods in Boston Housing dataset. We developed a global combined model without sharing data between distributed sites to preserve data privacy. We saved the cost and time of data transformation from one site to another or data centralisation. Also, avoid using a server for controlling the learning process to avoid server issues and iterative communication and computation overheads. We achieved comparable performance to the centralised learning approach in some datasets. We preserved data privacy and models updates for each site by performing model building and updating methods locally. From the local combined models results for the randomly and non-randomly partitioned datasets, we conclude that the distributed sites can utilise other sites models to improve the prediction performance without sharing the data between the sites to preserve the data privacy for each site. However, as discussed in section 5.3.4, several issues may arise if we deal with large sites number, such as increased computation and communication overheads, time-consuming, and scalability issues. In addition, there is a possibility of malicious attacks on the exchanged models. In future, we will suggest decentralised machine learning strategies to overcome these problems.

5.5 SUMMARY

This chapter presented our proposed decentralised learning methods using stepwise model selection approach for classification and regression algorithms. Our contribution is developing a decentralised learning approach to multiple, un-exchangeable, and distributed data resources without using a centralised learning method. At the same time, overcome individual local model limitations by utilising knowledge from other sites local models but keeping minimal communication in-between and preserving local data privacy. We aim to develop a global combined model for distributed sites without centralising data, share the data between the distributed sites, or use a central location to control the learning process. In addition, a site can build a local combined model by utilising other sites local models to improve the prediction performance. One of our objectives is to enhance the prediction performance as possible by examining different model selection and combination methods.

We investigated several model selection, weighting, and combination methods with different classification and regression algorithms. The experiment results showed that our proposed decentralised learning method using stepwise model selection approach to develop a global combined model performed better for classification algorithms than regression in most datasets. The global combined model using classification algorithms outperformed the centralised learning approach in most datasets and got close results for the remaining randomly partitioned datasets. In addition, for non-random partitioned data, the global combined model results and the centralised learning approach are close. In the global combined model using regression algorithms for randomly partitioned data, our method got close results with the centralised learning approach in Abalone and Boston housing datasets and lower prediction performance in Parkinson datasets. In Parkinson disease (Motor UPDRS), our method and the centralised learning approach results are close for the non-randomly partitioned data. We proved that we could maintain the learning performance at a similar level as centralised machine learning, and the distributed sites could utilise other sites models to improve the prediction performance.

The proposed model combination method using the stepwise model selection approach showed that we could improve the prediction performance at the global and local level for the distributed sites without using a central location to coordinate or control the learning process. We overcame the centralised learning control issues and preserved data privacy by avoiding data sharing between distributed sites or a server.

Chapter 6

Model Combination Method Using All Possible Sequence Combinations Approach

6.1 CHAPTER OVERVIEW

This chapter presents our proposed decentralised learning and model combination methods using all possible sequence combinations approach. Section 6.2 views our contribution and aims to develop a global combined model using all possible sequence combinations approach. Next, the proposed method for classification with its experiment results and discussion are shown in section 6.3 and for regression algorithms in section 6.4. Finally, section 6.5 presents the chapter summary.

6.2 INTRODUCTION

Proper model selection and combination strategies are fundamental for building an efficient combined model to improve prediction performance. Our contribution is developing a decentralised learning approach to un-exchangeable and distributed data without using a centralised learning method, exchanging lots of intermediate information, or using a central site for iterative communication or computation. Our focus in this chapter is on models evaluation and combination strategies using all possible sequence combinations approach to achieve the best-combined global model. We aim to build a global combined model with minimal exchanged information without sharing data between distributed sites to preserve data privacy and save the time and cost of data transformation between sites. Moreover, develop a learning approach for distributed sites without using complicated methods that need more communication between distributed sites or a central site to coordinate the learning process. The proposed decentralised learning method passes the trained models and evaluation results between the sites with minimal communication rounds than federated learning. We perform the model selection method using Gossip learning approach and the updating process using mini-batches stochastic gradient descent in each site locally and in a decentralised learning fashion. Finally, combine the final models using a linear combination approach.

6.3 PROPOSED METHOD FOR CLASSIFICATION ALGORITHMS

6.3.1. Global-level Modelling Approach:

We develop a global combined model using all possible sequence model combinations method. We aim to enhance the prediction performance by examining all possible sites sequences to build a global combined model. The proposed approach performs model selection and updating methods in each site locally in a decentralised learning fashion using Gossip learning method, then combines the best-updated models at the server using the linear combination method. First, we define all possible sequences for the participating sites; then, we apply the proposed method for each sequence. Figure 6.1 shows the proposed approach for one sequence. The first site in the sequence builds different local models using different classification algorithms, and then the site sends these models to the following site. Next, the following site evaluates the received models, selects the best models, and updates the selected models using mini-batch stochastic gradient descent. After updating, the site selects the best-updated models and sends the selected models to the following site in the sequence. Each site evaluates the models and only updates the best-updated models without losing the previous site data information.

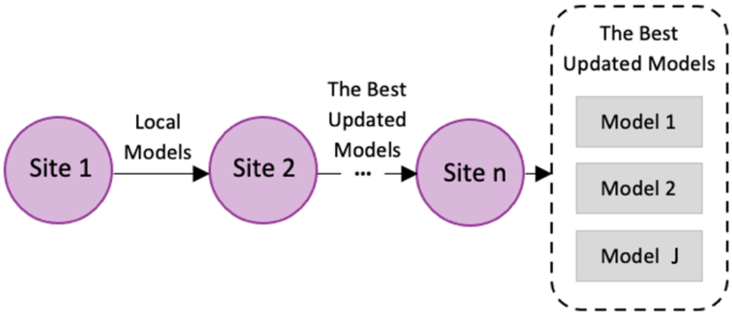


Figure 6.1. Sequence Learning Approach

As illustrated in Figure 6.2, after finishing model selection and updating processes using all sites data, we send the final models to all sites for final evaluation. Each site evaluates the received models based on their local data and then sends the models with their evaluation results and the data size to the server. Then, the server calculates the average models' accuracy., and combines the models using the linear combination method by assigning weight for each model based on its average accuracy to develop the global combined model. This approach does not expose the data resource and hence preserves data privacy. In addition, there is no computation or communication overhead between the server and distributed sites and minimum exchanged information between the sites and the server.

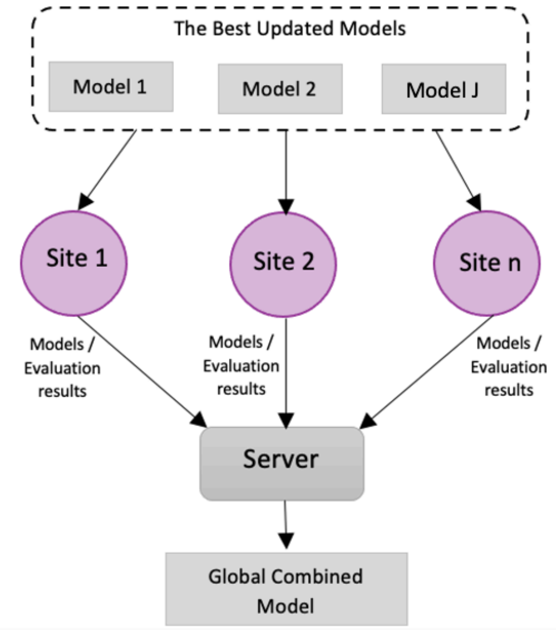


Figure 6.2. Models Combination Approach

After defining all possible sequences for the participated sites S_i, $i = 1,2, \dots, n$, where n is the sites number, and sequences number=n!, we apply the following steps for each sequence. Figure 6.3 illustrates the global-level modelling approach for one sequence:

1) The first site S_i in the sequence:
   a) Apply different j learning algorithms to build local models M_j, $j = 1,2, \dots, m$, where j is the algorithms number
   b) Use 10-fold cross-validation results to evaluate the models M_j based on the local data in site S_i and calculates the accuracy $Acc\left(M_j\right)$ using confusion matrix.
   c) Send the best models M_{j*} to the next site in the sequence
2) The next site receives the models then:
   a) Update the models based on its local date using mini-batch stochastic gradient descent method as follows:
      i. Divide the local data D_i into batches b
      ii. For each batch of p examples x_t with corresponding target value y_t, t = 1,2,3, …, p:
      iii. Compute the gradient estimate of the model loss function [143,161]:
      $$g = \frac{1}{p} \nabla_w \sum_{t=1}^{p} \ell \left((M_{j*} (x_t ; w), y_t\right) \qquad (6.1)$$
      iv. Update model parameter $w = w - \gamma g$, where γ is the learning rate, and w is the model parameter.

      For example, we calculate g of the loss function (prediction error) for SVM model, then we update the model with the gradient of the sum of the loss functions to minimise model error. In the model update approach for SVM model, we reuse a trained model in its previous state to start the updating process. In DT algorithm, the model update approach adds successive DT model to the previous model sequentially. Each DT model tries to improve on its predecessor by reducing model errors and fitting a new model to the residual errors made by the previous model. In each step, the models fit on the negative gradient of the loss function.
   b) Evaluates the updated models Acc (M_{j*}).

c) Select the best updated models M_{j*} using a defined performance threshold.

d) Send the selected model to the next site in the sequence.

3) Repeat the step 2 until finish the sites sequence and the selected models M_{j*} being updated using all n sites local data, then:

   a) Send the final updated models to all sites for final evaluation.

   b) Each site evaluates the received models based on its local data, then sends the evaluated models with the evaluation results and the data size to the server.

4) The server combines the received models using the linear combination method as follows:

   a) Calculates the average accuracy for each model $\mathrm{Acc}\left(M_{j*}\right)$.

$$\mathrm{Acc}\ (M_{j*}) = \sum_{i=1}^{n} \frac{D_i}{D} * \mathrm{Acc}\ (M_{j*})\ \text{in}\ S_i \qquad (6.2)$$

   , where i is site number, D_i is the number of samples of site i, and D is all sites' data samples number.

   b) Weighting each model based on its average accuracy $\mathrm{Acc}\left(M_{j*}\right)$.

$$w_{M_{j*}} = \frac{\mathrm{Acc}\ (M_{j*})}{\sum_{i=1}^{n} \mathrm{Acc}\ (M_{j*})}, \qquad (6.3)$$

   , where i=1, 2, …, n, $0 <= w_{M_{j*}} <= 1$ and $\sum_{i=1}^{n} w_{M_{j*}} = 1$

   c) Linearly combines the models to develop the global model M^{G*}:

$$M^{G*}\ (x) = \max \sum w_{M_{j*}}\ M_{j*}\ (x) \qquad (6.4)$$

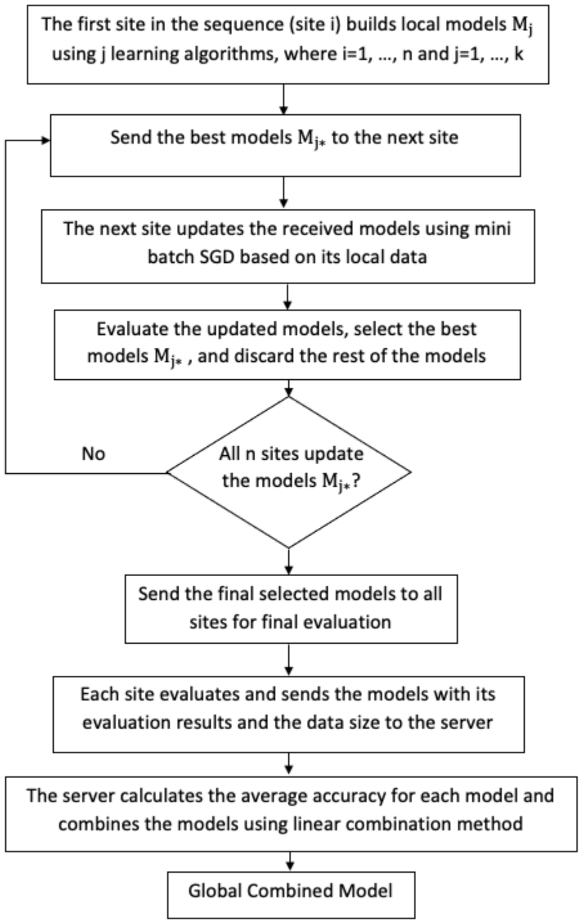


Figure 6.3. The Proposed Decentralised Learning Method

6.3.2. Experimental Study

The experiments are performed to evaluate the proposed global combined model performance, compared with different studies and the centralised learning approach that moves all distributed data to a centralised database.

I. Distributed Data Simulation

We applied the method to the datasets that used in chapter 3 (section 3.3.3). We used three-databases: blood transfusion, diabetes (non-randomly partitioned), and heart disease datasets [29]. Therefore, we divided each dataset into different parts as distributed sites site1, site2, and site3. At each site, the dataset is divided into local and validation data. The local data partition is used to develop the local models, evaluate the received models, and update the selected models. The validation data is used to assess the final global and local combined model.

II. Models Building and Evaluation

We applied the classification learning algorithms for model building and updating methods that used in chapter 5 (section 5.3.3). The algorithms are Support Vector Machine (linear and nonlinear SVM), Neural Network (NN), Naïve Bayes (NB), Random Forest (RF), Decision Tree (DT), and Logistic Regression (LR). We evaluated the global combined models in each site instead of the server because the sites will not share their validation data with the server. Each site sends the evaluation results to the server with the test data size. Then, the server receives the sites evaluation results and calculates the average accuracy of the global combined model.

III. Experiments Results

In Appendix D, we present the detailed results for the global-level modelling approach. Table 6.1 compares the proposed method with model combination using average accuracy, and the centralised learning approach for blood transfusion dataset that randomly partitioned. Our proposed method got the best result for the sequence Site1- Site2- Site3, and the two sequences that start from site 1 performed better than other sites sequences. Our proposed combination method is better than the combination method using average accuracy in most sites sequences.

Table 6.1. Global Combined Model Evaluation for All Sites Sequences for Blood Transfusion Dataset

Sequence	Combination method	Accuracy
Site1- Site2- Site3	The Proposed Method	**61%**
	Using Average accuracy	59%
Site1- Site3- Site2	The Proposed Method	55%
	Using Average accuracy	54%
Site2- Site1- Site3	The Proposed Method	52%
	Using Average accuracy	53%
Site2- Site3- Site1	The Proposed Method	52%
	Using Average accuracy	53%
Site3- Site1- Site2	The Proposed Method	52%
	Using Average accuracy	50%
Site3-Site2- Site1	Single model	53%
Centralised learning approach		60%

Table 6.2 shows the evaluation results for diabetes dataset that non-randomly partitioned. Our proposed method is slightly better than the

centralised learning method for the sequences Site2- Site3- Site1 and Site3- Site2- Site1. Furthermore, the proposed method is better than the combination method using average accuracy in all sites sequences except in Site2- Site1- Site3 sequence.

Table 6.2. Global Combined Model Evaluation for All Sites Sequences for Diabetes Dataset

Sequence	Combination method	Accuracy
Site1- Site2- Site3	The Proposed Method	62%
	Using Average accuracy	60%
Site1- Site3- Site2	The Proposed Method	68%
	Using Average accuracy	67%
Site2- Site1- Site3	The Proposed Method	52%
	Using Average accuracy	61%
Site2- Site3- Site1	The Proposed Method	**71%**
	Using Average accuracy	62%
Site3- Site1- Site2	The Proposed Method	67%
	Using Average accuracy	63%
Site3- Site2- Site1	The Proposed Method	**71%**
	Using Average accuracy	66%
Centralised learning approach		69%

For heart disease dataset that randomly partitioned, Table 6.3 shows that the central learning approach is better than our method but not far from our results for the sequences Site3-Site1-Site2 and Site3-Site2-Site1. Our proposed combination method performed better than the combination method using average accuracy in three sequences and a similar result in one sequence.

Table 6.3. Global Combined Model Evaluation for All Sites Sequences for Heart Disease Dataset

Sequence	Combination method	Accuracy
Site1- Site2- Site3	The Proposed Method	72%
	Using Average accuracy	83%
Site1- Site3- Site2	The Proposed Method	80%
	Using Average accuracy	79%
Site2- Site1- Site3	The Proposed Method	80%
	Using Average accuracy	81%
Site2- Site3- Site1	The Proposed Method	83%
	Using Average accuracy	83%
Site3- Site1- Site2	The Proposed Method	**90%**
	Using Average accuracy	86%
Site3- Site2- Site1	The Proposed Method	**90%**
	Using Average accuracy	81%
Centralised learning Approach		**92%**

Table 6.4 shows the results of the proposed method using all possible sites sequences approach compared with the proposed linear, nonlinear combination methods and models combination method using the stepwise model selection approach in chapters 3, 4, and 5, respectively, and with the centralised learning approach. In diabetes dataset, all possible sites sequences approach outperformed the other methods, and the result is slightly better than the centralised learning approach in blood transfusion dataset. Furthermore, the stepwise model selection approach is better than other methods in blood transfusion dataset and close to the proposed all possible sites sequences method. In heart disease dataset, the stepwise model selection approach and the proposed all possible sites sequences method results are similar and close to the centralised learning approach result. We conclude that we could improve the prediction performance and get better or comparable results compared to the centralised learning performance without using a server to coordinate the learning process or centralise the data with minimum information exchanged and communication rounds.

Table 6.4. Global Combined model and Centralised Learning Approach Evaluation

Method	Blood transfusion	Heart Disease	Diabetes (non-randomly partitioned)
Linear combination method	59%	86%	67%
Nonlinear combination method	57%	89%	67%
Stepwise Model Selection Approach	**63%**	90%	67%
All Possible Sequences Method	61%	90%	**71%**
Centralised Machine Learning	60%	**92%**	69%

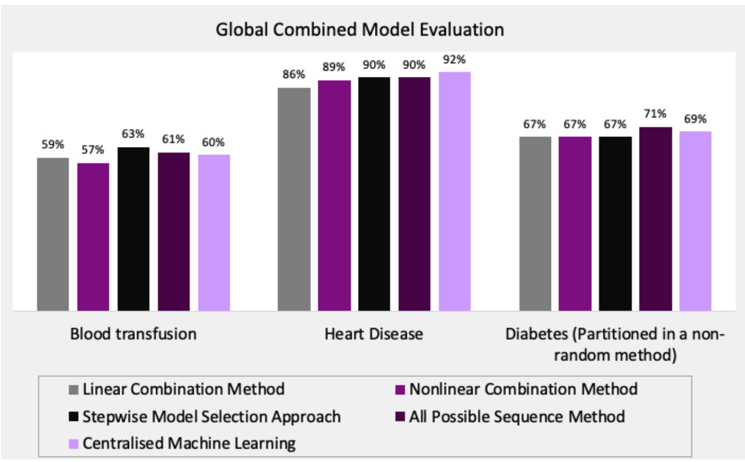


Figure 6.4.Global Level Modelling Methods and Centralised Learning Approach Evaluation

Table 6.5 presents our proposed method using all possible sequence combinations approach. We compared it to our proposed methods using linear and nonlinear model combination methods in chapters 3, 4, the proposed method using stepwise model selection approach in chapter 5, and with the research works [6, 48, 96, 166, 170 176,179, 181, 182]. There are no research works found for blood transfusion dataset, thus, we exclude it from the table. Our proposed all possible sequences and the stepwise models selection methods results are better than other methods in heart disease dataset. For diabetes dataset, the proposed method in [69] outperformed the other methods.

Table 6.5. Global Combined Models and related works Evaluation

Method	Heart Disease	Diabetes (Partitioned in a non-random method)
Linear Combination Method	86%	67%
Nonlinear Combination Method	89%	67%
Stepwise Model Selection Approach	**90%**	67%
All Possible Sequences Method	**90%**	71%
Tsoumakas et al. [6] - EV1	84%	77%
Tsoumakas et al. [6] - EV2	83%	77%
Tsoumakas et al. [6] - EV 3	85%	77%
Bashir et al. [48]	84%	77%
Zhang et al. [96]	-	**80%**
Mandal et al. [166]	-	76%
Wang et al. [170]	-	77%
Gao et al. [176]	72%	-
Haque et al. [179]	82%	78%
Froelicher et al. [181]	-	78%
Ed-daoudy and Maalmi [182]	82%	-

6.3.3. Discussion and Evaluation

The global combined model shows improved performance compared to the centralised learning approach. In diabetes dataset, the proposed method using all possible sites sequences method outperformed the linear and nonlinear combination and the centralised learning methods. For blood transfusion dataset, the proposed method is slightly better than the centralised learning approach and close to the stepwise model selection approach. In heart disease dataset, the stepwise model selection approach and the proposed approach using all possible sites sequences method results are similar and close to the centralised learning approach result. We improved the prediction performance and got better, or comparable results compared to the centralised learning performance without centralising the distributed datasets or using a server to coordinate the learning process to avoid the centralised learning control issues and overheads. We built the local models and updated the selected model locally with minimum information exchanged and communication rounds. The proposed approach exchanges the trained models and evaluation results, and this practical approach can substantially reduce privacy disclosure risks.

However, increased computation and communication overheads, time-consuming, and scalability issues may arise if we deal with large sites number. It involves large sites sequences number, and the developed global model from each sequence. These issues can be addressed by proposing a search strategy that examines and selects the sequences of the best global average performance before performing models combining phase to develop the global model. This may reduce computation and communication overheads and time and enhance the proposed method scalability. In addition, malicious attacks are possible on the trained models to retrieve training data or reveal meaningful information. In future, we will consider this issue and find a decentralised machine learning strategy to overcome this problem.

6.4 PROPOSED METHOD FOR REGRESSION ALGORITHMS

We develop a global combined model for regression algorithms using the similar approach that used in section 6.3.1, but we use RMSE and MAPE metrics for model performance evaluation instead of accuracy metric. We use RMSE for the model selection approach and then apply linear model combination methods by assigning a model weight using simple weight average, error-based (RMSE), and performance-based (Accuracy) approaches.

6.4.1. Experimental Study

We conducted experiments to evaluate the performance of the proposed method. We compared the global combined model with a centralised learning approach that moves all distributed data to a centralised database.

I. Distributed Data Simulation

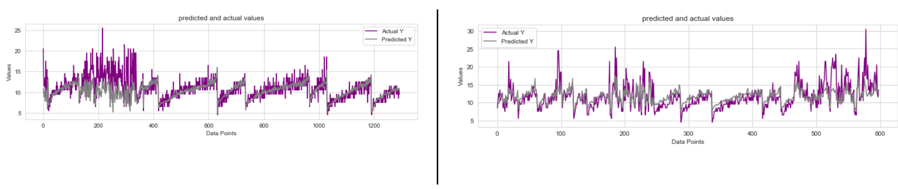
We applied our methods to the same preprocessed datasets that used in Chapter 3 (section 3.4.3): Parkinson disease and Abalone datasets [29]. We divided each dataset into different parts as distributed sites site1, site2, and site3. At each site, the dataset is divided into local and validation data. The local data partition is used to develop the local models, evaluate the received models, and update the selected models. The validation data is used to evaluate the final global and local combined model.

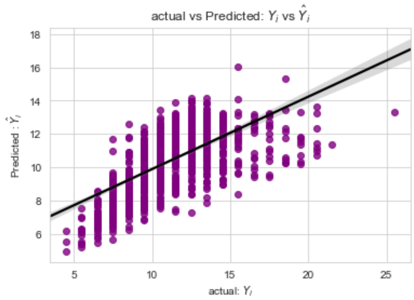
II. Models Building and Evaluation

We applied the same learning methods for model building and updating methods that used in chapter 5 (section 5.4.1). The algorithms are Linear Regression (LR), Support Vector Regression (SVR), Decision Tree Regression (DTR), Neural Network Regressor (NNR), Random Forest Regressor (RFR), Least Absolute Shrinkage and Selection Operator regression (LASSO), Ridge, and ElasticNet. We evaluated the global combined models in each site, and then each site sent the evaluation results to the server with the test data size. Finally, the server received the evaluation results and calculated the average RMSE of the global combined models.

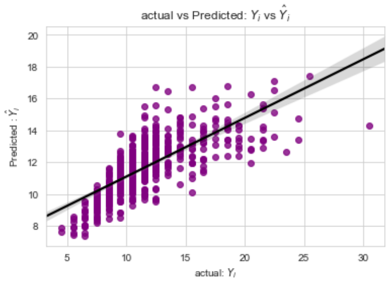
III. Experiments Results

The detailed results for the proposed method are illustrated in appendix D. Figure 6.5 shows LR model updating process for Abalone dataset during the sequence site1-site2-site3 using mini-batch SGD. Local RMSE for LR model in site 1 = 2.23. Then, in site 2, LR model RMSE after updating process = 2.08. While in site3, LR model RMSE after updating process = 2.49.





(a) using site 2 local data



(b) using site 3 local data

Figure 6.5. LR model updating method using mini-batch SGD

Table 6.6 shows the linear combination method using error-based model weighting is better than the other weighting methods in most sites sequences. In addition, we compare the global combined models for all sites sequences with the centralised learning approach. The central learning approach is slightly better than the proposed method in all sequences. We performed a statistical significance analysis to interpret the results and investigate if the centralised learning approach difference from our method is significant. We performed the evaluation experiment 100 times and computed the RMSE results difference between the centralised learning approach and our decentralised learning method as follows:

$$\mathrm{RMSE_{Difference}} = \mathrm{RMSE_{Decentralised}} - \mathrm{RMSE_{Centralised}} \quad (6.5)$$

If $\mathrm{RMSE_{Difference}} > 0$ in 95% of the experiments or more, then it implies that the differences between the centralised learning approach result and our method are statistically significant. We found that the centralised learning approach difference is significant in 98% of the experiments.

Table 6.6. Global Combined Model Evaluation (RMSE) for All Sites Sequences for Abalone Dataset

Sequence	Linear Model Combination Method	RMSE
Site 1- Site 2- Site 3	Simple average	2.52
	Error-based (RMSE)	2.52
	Performance-based (Accuracy)	2.52
Site 1- Site 3- Site 2	Simple average	2.85
	Error-based (RMSE)	2.79
	Performance-based (Accuracy)	2.85
Site 2- Site 1- Site 3	Simple average	2.53
	Error-based (RMSE)	2.51
	Performance-based (Accuracy)	2.53
Site 2- Site 3- Site 1	Simple average	2.68
	Error-based (RMSE)	2.64
	Performance-based (Accuracy)	2.64
Site 3- Site 1- Site 2	Simple average	2.85
	Error-based (RMSE)	2.80
	Performance-based (Accuracy)	2.79
Site 3- Site 2- Site 1	Simple average	2.68
	Error-based (RMSE)	2.64
	Performance-based (Accuracy)	2.68
Centralised learning approach		**2.39**

Table 6.7 compares the RMSE results of our global combined models for all sites sequences with the centralised learning approach for Parkinson disease (Total UPDRS) dataset that randomly partitioned. The proposed method using the error-based model weighting approach is slightly better than the central learning approach. In addition, the linear combination methods results in the sequence site1 – site 2 – site 3 are better than the other sequences.

Table 6.7. Global Combined Model Evaluation (RMSE) for All Sites Sequences Parkinson disease (Total UPDRS) dataset

Sequence	Linear Model Combination Method	RMSE
Site 1- Site 2- Site 3	Simple average	8.79
	Error-based (RMSE)	**7.46**
	Performance-based (Accuracy)	9.48
Site 1- Site 3- Site 2	Simple average	9.9
	Error-based (RMSE)	9.9
	Performance-based (Accuracy)	9.9
Site 2- Site 1- Site 3	Simple average	9.6
	Error-based (RMSE)	9.6
	Performance-based (Accuracy)	9.6

Site 2- Site 3- Site 1	Simple average	11.9
	Error-based (RMSE)	11.9
	Performance-based (Accuracy)	11.9
Site 3- Site 1- Site 2	Simple average	10.30
	Error-based (RMSE)	10.38
	Performance-based (Accuracy)	10.31
Site 3- Site 2- Site 1	Simple average	11.63
	Error-based (RMSE)	11.56
	Performance-based (Accuracy)	11.76
Centralised learning approach		7.53

Table 6.8 compares the global combined models for all sites sequences with the centralised learning approach for Parkinson disease (Motot UPDRS) dataset that partitioned non-randomly. Our proposed method in the sequence Site 1- Site 3- Site 2 outperformed all other sites sequences and centralised learning approach. In addition, the linear combination method RMSE results using the error-based model weighting approach got the best RMSE results in most sites sequences.

Table 6.8. Global Combined Model Evaluation (RMSE) for All Sites Sequences for Parkinson disease (Motor UPDRS) dataset

Sequence	Linear Model Combination Method	RMSE
Site 1- Site 2- Site 3	Simple average	10.79
	Error-based (RMSE)	10.79
	Performance-based (Accuracy)	10.79
Site 1- Site 3- Site 2	Simple average	7.23
	Error-based (RMSE)	**7.05**
	Performance-based (Accuracy)	7.22
Site 2- Site 1- Site 3	Simple average	12.01
	Error-based (RMSE)	11.98
	Performance-based (Accuracy)	11.99
Site 2- Site 3- Site 1	Simple average	10.51
	Error-based (RMSE)	10.60
	Performance-based (Accuracy)	10.61
Site 3- Site 1- Site 2	Simple average	7.98
	Error-based (RMSE)	7.93
	Performance-based (Accuracy)	7.97
Site 3- Site 2- Site 1	Simple average	10.47
	Error-based (RMSE)	10.47
	Performance-based (Accuracy)	10.47
Centralised learning approach		7.3

Table 6.9 shows the global combined model evaluation results using all possible sites sequences method compared with the proposed linear and nonlinear combination methods in chapters 3 and 4, respectively, and with the proposed model combination using the stepwise model selection approach in chapter 5. Our proposed method using all possible

sites sequences approach in Parkinson disease (Total UPDRS) and Parkinson disease (Motor UPDRS) datasets is better than the other methods and the centralised learning approach. Furthermore, in Abalone dataset, the linear combination method is slightly better than other methods. Our proposed methods achieved somewhat improved results than the centralised learning approach.

Table 6.9. Global Combined Models and Centralised Learning Approach Evaluation

Method	Parkinson disease (Total UPDRS)	Abalone	Parkinson disease (Motor UPDRS) (non-randomly partitioned)
Linear combination method	8.97	**2.36**	7.1
Nonlinear combination method	10.16	2.43	8.51
Stepwise Model Selection Approach	10.06	2.40	7.6
All Possible Sequences Method	**7.46**	2.51	**7.05**
Centralised learning	7.53	2.39	7.3

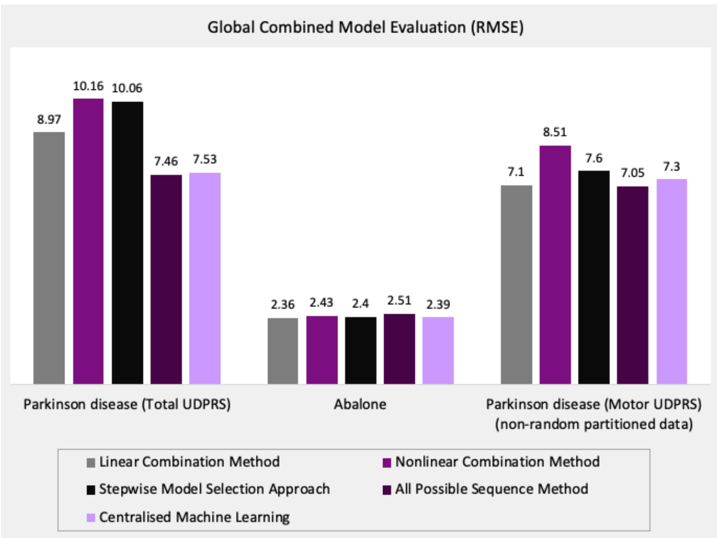


Figure 6.6. Global Level Modelling Methods and Centralised Learning Approach Evaluation

6.4.2. Discussion and Evaluation

We examined all possible sites sequences for the model combination approach to develop the global combined model and evaluated results with the previously proposed methods in chapters 3, 4, and 5 and with the centralised learning approach. The proposed method using all possible sites sequences approach is better than the other proposed methods and the centralised learning approach in Parkinson disease (Total UPDRS) and Parkinson disease (Motor UPDRS) datasets. In Abalone dataset, all methods results are close, and the linear combination method is slightly better than other methods. Furthermore, our proposed methods showed slightly improved results compared to the centralised learning approach using a decentralised learning approach. Moreover, the method performed the model building, selection, and updating processes locally to avoid data and models updates sharing between the distributed sites to preserve data privacy. Also, we developed the global model without using a server to coordinate the learning process to avoid server issues and communication and computation overheads. However, as discussed in section 6.3.3, there are several concerns, such as malicious attacks on the trained model or if we deal with a large sites number. We will consider these issues in future and propose solutions to overcome these concerns.

6.5 SUMMARY

In this chapter, we proposed decentralised learning and model combination methods using all possible sequence combinations for classification and regression algorithms. We considered all possible sites sequences during model learning, selection, weighting, and combination methods. Our contribution is developing a decentralised learning approach to private and distributed data resources without using a centralised learning method and preserving sites data privacy. Our aim is to develop a global combined model for distributed sites without centralising the data, share the data between the distributed sites, or use a central location to control the learning process. One of our objectives is to enhance the prediction performance as possible by examining all possible sites sequences to develop the global combined model. The experiment results showed that the proposed approach outperformed the centralised learning approach in most datasets. Thus, we could maintain the learning process at a similar level as centralised machine learning and improve the prediction performance at the global level for the distributed sites without using a centralised location for coordinating or controlling the learning process. Furthermore, we overcame the centralised learning control issues and preserved data privacy by avoiding sharing data between distributed sites or server.

Chapter 7

Conclusion and Future Work

7.1 OVERVIEW AND CONTRIBUTIONS OF THE THESIS

- We developed a decentralised learning approach for distributed data resources without using centralised machine learning method or relying on server for iterative communication or computation. We built combined prediction models derived from local learning outcomes at global and local levels using simple and efficient linear model combination models. Our privacy contribution is keeping data locally on each site and only the learning outcomes and minimum information (local models and model performance information) are exchangeable. We simulated distributed sites using different dataset partitioning scenarios and evaluated the performance of the proposed local and global level modelling methods on different classification and regression datasets. Our method showed its efficiency and applicability in the distributed environment. The global-level modelling for classification datasets got close or similar results with the centralised learning method in several datasets. In addition, the results of the global combined model for regression datasets are slightly better or close to the centralised learning approach. In the local-level modelling approach, all sites are utilised from other sites models and improved the prediction performance with minimal communication and information transfer to build its local combined model without requiring data sharing. There are no poor results obtained by models combining methods. We avoid using complicated methods that need more communication and computational cost and less information exchange than FL. We developed a global combined model without moving the local data to another site or a central server. The proposed method did not expose the data resource and hence preserved data privacy. We saved the cost and time of data transformation from one site to another and preserved data privacy.

- We developed global combined model using a decentralised version of model selection and nonlinear model combination strategies. The proposed method preserved data privacy by avoiding data sharing or centralisation with fewer communication rounds than FL. We developed a combined local model for each site by utilising the learning outcomes from local data resources from other sites. We conducted experiments on classification and regression datasets. The experiments showed similar or close results to the centralised learning approach for classification datasets, and comparable or better performance than the centralised learning approach for the regression datasets. The local combined model showed better prediction performance than the best local model in most datasets, proving that each site can utilise other sites models to improve the prediction performance without sharing the data between sites to preserve the data privacy. The proposed nonlinear combination approach provides an effective alternative with much less information sharing to preserve data privacy and reduced communication and computation costs.
- We developed a decentralised alternative to the federated learning approach without using a server or exchanging intermediate computing updates to overcome iterative learning process issues. We proposed model selection and updating strategies using stepwise model selection method and gossip learning approach that make the final combined model optimal and valuable for all sites. We minimised communication and computation overheads and preserved data privacy by only passing the models between the distributed sites and updating the models locally in each location without exchanging models updates information. We used a simple linear combination method to combine the best-updated models to develop combined models at global and local levels with less information sharing between the distributed sites. The experiment results showed that the proposed method performed better for classification datasets than regression. The global combined model using classification algorithms outperformed the centralised learning approach in most datasets and got close results for the remaining randomly partitioned datasets. In addition, for non-random partitioned data, the global combined model results and the centralised

learning approach are close. In the global combined model using regression algorithms, our method got close results with the centralised learning approach in some datasets. In addition, we showed that we could improve the prediction performance at the global and local level for the distributed sites without using a central location to coordinate or control the learning process. We overcame the centralised learning control issues and preserved data privacy by updating the models locally and avoiding data sharing between distributed sites or a server.

- We developed a decentralised learning approach that applies all possible sequence combinations approach to achieve the optimal global combined model without exchanging data or models updates information between the distributed sites. We preserved data privacy by performing the models selection and updating methods locally and minimised the communication and computation overheads. We improved the prediction performance and got better or comparable results compared to the centralised learning performance without sharing data or using a server to coordinate the learning process. The proposed method performed the model building, selection, and updating processes locally with minimum information exchanged and communication rounds, and preserved data and models updates. In addition, we developed the global model without using a server to coordinate the learning process to avoid server issues and communication and computation overhead. The proposed method showed improved performance compared to the centralised learning approach in most datasets. Thus, we could maintain the learning process at a similar level as centralised machine learning and improve the prediction performance at the global level for the distributed sites without using a centralised location for coordinating or controlling the learning process.

In general, the proposed decentralised learning methods contributed to preserved data privacy by avoided raw data sharing or moving to a centralised database to perform data resampling like the ensemble learning methods. Instead, these methods exchange the trained models and evaluation results, and this practical approach can substantially reduce privacy disclosure risks. Also, we avoided exchanging many intermediate information, using complicated privacy-preserving techniques, or relying on a central server for iterative computation and communication like federated learning. We focused on model evaluation, selection, and combination strategies to achieve the optimal combined global and local models that maximise the combined models performance. These methods lead toward a simpler and new direction for decentralised privacy-preserving machine learning by keeping data locally for each site and combining diverse and accurate models instead of complicated ways that increase communication and computational overheads. We used simple linear and nonlinear combination methods to combine the best models to develop the global and local combined models with less information sharing between the distributed sites. The proposed approaches achieved improved performance at a similar level as centralised machine learning in most datasets. In addition, the distributed sites are utilised form other sites data resources to improve its prediction performance. We show the main findings in section 7.2 and the limitations and obstacles we faced in section 7.3. Finally, the future works and the chapter summary are presented in sections 7.4 and 7.5, respectively.

7.2 MAIN FINDINGS

The evaluation results for the proposed decentralised learning methods showed that we could achieve better, or comparable prediction performance compared to the centralised learning approach. We developed the global model without pooling the distributed dataset to a central database, sharing data between the distributed sites, or using a server for the learning process.

For the randomly partitioned classification datasets, the global combined model for the proposed decentralised learning method using stepwise model selection approach performed better than the centralised learning approach in most datasets, while the proposed methods using linear and nonlinear combination methods got similar or close performance to the centralised learning approach. The decentralised learning approach using all possible sites sequence combinations strategy is better than or close to the centralised learning approach. Compared to the research works [6, 48, 96, 166, 170, 176-182], the proposed decentralised learning method using stepwise model selection approach achieved better results than the proposed methods using linear and nonlinear combination methods in most datasets. For non-randomly partitioned data, our proposed methods and the centralised learning approach performance got close performance. The decentralised learning approach using all possible sites sequence combinations strategy for a dataset is slightly better than the other proposed methods and the centralised learning approach. Furthermore, the proposed method using linear combination method is slightly better or close to the research works [6, 48, 96, 166, 170, 176-182], while the proposed learning method using nonlinear combination approach and stepwise model selection approach got similar or close performance to the research works in two datasets.

For local-level modelling approach evaluation, we compared the local combined model results with the best local model in all sites. Most of distributed sites improved their local performance by utilising knowledge from other sites and overcoming their individual model limitations without sharing data to preserve their local data privacy.

For the randomly partitioned regression datasets, the proposed method using the linear combination method is slightly better than our proposed method using linear nonlinear combination approach, the proposed method using stepwise model selection approach, and the centralised learning approach in some datasets and close performance in the other datasets. Furthermore, all our proposed methods results are better than the research work in [166]. In addition, the decentralised learning approach using all possible sites sequence combinations strategy is better than the other proposed methods and the centralised learning approach in two datasets. For non-randomly partitioned data, the decentralised learning approach using all possible sites sequence combinations approach is better than the other proposed methods and the centralised learning approach in a dataset. The decentralised learning approach using the nonlinear model combination performed better than the other methods and the centralised learning approach in a dataset. In contrast, the proposed method using the stepwise model selection strategy did not perform well in two datasets. In addition, the proposed method using the linear combination method performed better or close to the centralised learning approach. Compared to the research work in [166], the proposed methods using the linear and nonlinear combination approaches performed better than [166]. Besides, the distributed sites improved their local performance by utilising learning outcomes from other sites data resources to develop a global combined model without sharing, centralising, or disclosing the privacy of these data resources.

We avoided centralising the distributed data to a central location, sharing data between the distributed sites, or exchanging lots of intermediate information during the learning process. As a result, we minimised the communication costs for computation effectiveness and saved time of data transformation. Furthermore, only the learning outcomes and minimal information (learned models and model performance information) are exchangeable, and therefore we preserved local data privacy for each site. Moreover, we used simple model evaluation and selection strategies instead of complex privacy-preserving methods to avoid extensive communication between sites and minimise computation overhead. Moreover, we minimised communication overhead and single points of attacks or failure risks by avoiding using a server for iterative learning or controlling the learning process.

The proposed methods could be considered a promising aspect of privacy-preserving decentralised learning approach where data privacy concerns are significant and data privacy preservation is essential. Also, these methods could be applied to solve issues related to large data, such as memory limitation and huge data transformation cost and time. It could be partitioned into distributed data subsets and apply the decentralised learning for more efficient model learning and analysis and avoid server issues and overheads.

However, increased computation and communication overheads, time-consuming, and scalability issues may arise if we deal with large sites number. It involves exchanging model rounds to compute the global average performance, large sites sequences number, and the developed global model from each sequence. The impact of sites number on the combined model performance is not our focus, but these issues can be addressed by developing different selective decentralised learning strategies. These issues can be addressed by developing different selective decentralised learning strategies. For example, we could develop a search strategy that only considers the best sites contributions by examining its local models performance to include these sites in the decentralised learning process. Also, we could propose a search strategy that examines and selects the sequences of the best global average performance before performing models combining phase to develop the global model. This may reduce computation and communication overheads and time and enhance the proposed method scalability. In addition, the proposed decentralised machine learning approaches only exchange the trained models instead of data to preserve data privacy, and there is a possibility for malicious attacks on the models to reveal meaningful information or retrieve training data. We did not consider model attack cases in the distributed environment, and it is beyond the scope of our thesis. In future research, we will consider these issues to analyse the possible malicious attacks on distributed sites and exchanged models.

The proposed decentralised machine learning approaches are used to develop a global/local combined prediction model from multiple data resources that are distributed, private, and not exchangeable. In addition, these approaches avoided using a server to coordinate the learning process to overcome server issues and minimise communication and computation overheads. It showed its efficiency and applicability in the distributed environment and preserved data privacy. To apply these methods to distributed data resources, the distributed sites should first agree on the learning algorithms that will be used for learning models and have the same data attributes and target. Second, the sites must agree not to expose the data to another site, collude to retrieve the data from the model, or respond to or disclose their models to an external party. In addition, the distributed sites must share the models only with the agreed/trusted sites. The proposed approaches using stepwise model selection strategy and all possible sequence combinations approach are efficient and improved the prediction performance in distributed environment. However, it is not preferred if we deal with large sites number due to the increased computation and communication overheads, time-consuming, and scalability issues. The decentralised machine learning approaches using linear and nonlinear combination methods can be used to avoid large sites number issues.

7.3 OBSTACLES AND LIMITATIONS

In chapter 3, we faced an issue related to the model selection strategy in the local level modelling approach when the performance of the received models from other sites is lower than the best local model. As a result, there will be no selected models to develop the combined local model. We addressed this problem by proposing another model selection strategy (Method L2) to select the best model from each site even if the model did not perform better but not worse than the best local model performance. In addition, we could not find large and distributed datasets publicly available to apply our proposed methods, and we will investigate this issue in future research. However, from our experiments, even with small datasets, we improved the decentralised machine learning performance compared to the centralised machine learning approach and obtained reliable prediction results. In addition, we developed data partitioning scenarios to mimic the distributed data on the selected classification and regression datasets from [29] and [27]. The proposed decentralised learning methods focused on the best model selection. In chapters 3 and 4, in local level modelling, each site selected the best local model and compared the model with received models from other sites. An issue may arise if not all local models in a site perform well, and this will affect the combined model performance. We did not face this problem, but our solution is to find a different way to improve the learning algorithms to develop better local models performance or modify the model selection strategy.

In chapter 6, the proposed method selects the best models and then sends these models to other sites for updating process. We set a model performance threshold for model selection approach, but we faced an issue in some datasets when all models were lower than the predefined selection threshold. We addressed this problem by minimising the selection threshold to an acceptable performance level, and this solution did not adversely affect the global model performance.

The proposed decentralised learning methods developed the global model without data sharing or centralisation to achieve preserving data privacy objective. Therefore, when we evaluated the global model, we did not place shared/open data in the server to assess the global model. Instead, we evaluated the global model in each site using the local evaluation dataset and then took the weighted average performance.

Our aim is to design decentralised learning methods that preserve data privacy and avoid using a central location for the iterative learning process. We exchanged the models instead of data, but there is a possibility for malicious attacks on the trained models to retrieve training data or reveal meaningful information [26, 137, 164, 186, 190, 192]. Model privacy should be considered when developing approaches for distributed machine learning. In our proposed methods, we did not consider model attacks case in the distributed environment, and it is beyond the scope of our thesis. To minimise this risk in our proposed methods, we assumed the participated sites agreed not to expose the data to another site, collude to retrieve the data from the model, or respond to or disclose their models to an external party. In addition, the distributed sites only share the models with the agreed/trusted sites. Privacy-preserving federated learning approaches are proposed to preserve data privacy and secure models, such as Differential Privacy (DP), Cryptographic methods, or Secure Multiparty Computation protocol (SMC). Despite its efficiency in preserving the data and model privacy, it faces challenges and requirements and is discussed in chapter 2 (section 2.9). Also, privacy-preserving federated learning approaches do not always guarantee to preserve model privacy. For example, in [186], the differential privacy approach was ineffective against their model attack approach. In addition, DP may significantly reduce the model prediction performance [26]. Thus, future research will consider these issues to analyse the possible malicious attacks on distributed sites and exchanged models and propose a reliable and robust privacy-preserving decentralised learning approach that could prevent these attacks, preserve data privacy, and secure models without complex privacy-preserving techniques.

Our approaches are decentralised learning and depend on communication between the distributed sites to develop the global model. An arisen question is, "how these approaches will deal with incomplete information sent between the participated sites during learning or sent to the server for models combination?". This issue will be addressed as follows: the participating sites must first agree to remain active during model learning, computation, and combination processes. However, if a site dropped or did not send the required information, the decentralised learning will continue. For example, in chapters 3 and 4, suppose a site dropped or did not send the required information to the server for the model combination approach. In that case, the server will discard this site and complete the combination step with the active sites that sent its best global average model with the evaluation result and data size. We continue the learning and combination procedures because the global average model in each site and the global combined model in the server is a weighted model calculated based on the average evaluation results using all active sites data. In chapter 5, we updated a model in each site using the site local data, evaluated the model after each update step in all sites, and then took the weighted average performance to decide the next update step. If a site becomes offline, the decentralised learning approach will continue and recompute the average model performance with the active sites information. However, the inactive sites will not use the global model, and we should start the decentralised learning to include these sites and rebuild the global model.

In our thesis, we applied the proposed methods using three distributed sites data, and the impact of sites number on the combined model performance is not our focus. Our objective is to develop an optimal global model for distributed sites without sharing data, using a server for the iterative learning process, or using a centralised learning approach. In addition, to show that we could achieve global model prediction performance at a similar level as the centralised learning approach and if a distributed site can utilise other sites learning outcomes to improve its local prediction performance. However, increased computation and communication overheads, time-consuming, and scalability issues may arise if we deal with large sites number. It involves exchanging model rounds to compute the global average performance, large sites sequences number (the proposed method in chapter 6), and the developed global model from each sequence. As discussed in section 7.2, these issues can be addressed by developing different selective decentralised learning strategies.

7.4 FUTURE WORK

In the future, we will evaluate the reliability of the proposed methods by analysing the privacy of the combined models and measuring the computational, communication time, and costs. In addition, further research and experimental design will be considered for future studies that could address the problems and limitations discussed above. For example, analyse the potential model hacking and malicious attacks and examine the impact of collaborating sites number and local data size on the performance of local and global combined models. In future research, we will design decentralised learning approaches that preserve both data and model privacy without extensive communication between the associated sites or use complex privacy-preserving methods to avoid high computational requirements and overheads. Also, we will try to find large and distributed datasets from different resources to apply the proposed decentralised methods.

7.5 SUMMARY

This chapter showed an overview of the proposed decentralised learning methods developed for distributed data resources without using a server for iterative learning process or centralised learning approaches. In addition, we showed the main findings, the limitations and the obstacles we faced. We proposed several solutions for the limitations of the proposed methods. Then, we suggested future works to improve the proposed approaches in distributed learning environments.

Bibliography

[1] Huang, Z., Hu, R., Guo, Y., Chan-Tin, E. and Gong, Y., 2019. DP-ADMM: ADMM-based distributed learning with differential privacy. IEEE Transactions on Information Forensics and Security, 15, pp.1002-1012.

[2] Santos, M.F., Mathew, W. and Santos, H.D., 2011. Grid data mining by means of learning classifier systems and distributed model induction. In Proceedings of the 13th annual conference companion on Genetic and evolutionary computation (pp. 391-398). ACM.

[3] Czarnowski, I. and Jędrzejowicz, P., 2011. An agent-based framework for distributed learning. Engineering Applications of Artificial Intelligence, 24(1), pp.93-102.

[4] Liu, B., Cao, S.G. and He, W., 2011. Distributed data mining for e-business. Information Technology and Management, 12(2), pp.67-79.

[5] Zhou, L., Pan, S., Wang, J. and Vasilakos, A.V., 2017. Machine learning on big data: Opportunities and challenges. Neurocomputing, 237, pp.350-361.

[6] Tsoumakas, G., Katakis, I. and Vlahavas, I., 2004. Effective voting of heterogeneous classifiers. In European Conference on Machine Learning, pp. 465-476. Springer, Berlin, Heidelberg.

[7] Kokkinos, Y. and Margaritis, K.G. 2015. Confidence ratio affinity propagation in ensemble selection of neural network classifiers for distributed privacy-preserving data mining. Neurocomputing, 150, pp.513-528.

[8] Talia, D., Trunfio, P. and Marozzo, F. 2015. Data analysis in the cloud: models, techniques and applications. Elsevier.

[9] Peteiro-Barral, D. and Guijarro-Berdiñas, B., 2013. A survey of methods for distributed machine learning. Progress in Artificial Intelligence, 2(1), pp.1-11.

[10] Czarnowski, I., 2010. Prototype selection algorithms for distributed learning. Pattern recognition, 43(6), pp.2292-2300.

[11] Brisimi, T.S., Chen, R., Mela, T., Olshevsky, A., Paschalidis, I.C. and Shi, W., 2018. Federated learning of predictive models from federated Electronic Health Records. International journal of medical informatics, 112, pp.59-67.

[12] Papapetrou, O., Siberski, W. and Siersdorfer, S., 2015. Efficient model sharing for scalable collaborative classification. Peer-to-Peer Networking and Applications, 8(3), pp.384-398.

[13] Mendialdua, I., Arruti, A., Jauregi, E., Lazkano, E. and Sierra, B., 2015. Classifier Subset Selection to construct multi-classifiers by means of estimation of distribution algorithms. Neurocomputing, 157, pp.46-60.

[14] Thapa, C. and Camtepe, S., 2021. Precision health data: Requirements, challenges and existing techniques for data security and privacy. Computers in biology and medicine, 129, p.104130.

[15] Tsai, C.F., Hsu, Y.F. and Yen, D.C., 2014. A comparative study of classifier ensembles for bankruptcy prediction. Applied Soft Computing, 24, pp.977-984.

[16] Tsai, C.F., Lin, Y.C., Yen, D.C. and Chen, Y.M., 2011. Predicting stock returns by classifier ensembles. Applied Soft Computing, 11(2), pp.2452-2459.

[17] Chen, J., Li, K., Rong, H., Bilal, K., Yang, N. and Li, K., 2018. A disease diagnosis and treatment recommendation system based on big data mining and cloud computing. Information Sciences, 435, pp.124-149.

[18] Woloszynski, T., Kurzynski, M., Podsiadlo, P. and Stachowiak, G.W., 2012. A measure of competence based on random classification for dynamic ensemble selection. Information Fusion, 13(3), pp.207-213.

[19] Costantini, M. and Pappalardo, C., 2010. A hierarchical procedure for the combination of

forecasts. International journal of forecasting, 26(4), pp.725-743.

[20] Graefe, A., Armstrong, J.S., Jones Jr, R.J. and Cuzán, A.G., 2014. Combining forecasts: An application to elections. International Journal of Forecasting, 30(1), pp.43-54.

[21] Jeong, D.I. and Kim, Y.O., 2009. Combining single-value streamflow forecasts–A review and guidelines for selecting techniques. Journal of Hydrology, 377(3-4), pp.284-299.

[22] Alkhozae, M. and Zeng, X., 2021. Decentralised and Privacy Preserving Machine Learning for Multiple Distributed Data Resources. In UK Workshop on Computational Intelligence (pp. 235-250). Springer, Cham.

[23] Cano, I., Weimer, M., Mahajan, D., Curino, C. and Fumarola, G.M., 2016. Towards geo-distributed machine learning. arXiv preprint arXiv:1603.09035.

[24] Che, J., 2015. Optimal sub-models selection algorithm for combination forecasting model. Neurocomputing, 151, pp.364-375.

[25] West, D., Mangiameli, P., Rampal, R. and West, V., 2005. Ensemble strategies for a medical diagnostic decision support system: A breast cancer diagnosis application. European Journal of Operational Research, 162(2), pp.532-551.

[26] Shokri, R., Stronati, M., Song, C. and Shmatikov, V., 2017, May. Membership inference attacks against machine learning models. In 2017 IEEE symposium on security and privacy (SP), pp. 3-18. IEEE.

[27] Ulianova, S., 2019. Cardiovascular disease dataset. Kaggle.com. [https://www.kaggle.com/sulianova/cardiovascular-disease-dataset]

[28] Zeng, L., Li, L., Duan, L., Lu, K., Shi, Z., Wang, M., Wu, W. and Luo, P., 2012. Distributed data mining: a survey. Information Technology and Management, 13(4), pp.403-409.

[29] Dua, D. and Graff, C., 2019. UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.

[30] Opitz, D.W. and Shavlik, J.W., 1996. Generating accurate and diverse members of a neural-network ensemble. In Advances in neural information processing systems, pp. 535-541.

[31] Rathore, S.S. and Kumar, S., 2017. Linear and non-linear heterogeneous ensemble methods to predict the number of faults in software systems. Knowledge-Based Systems, 119, pp.232-256.

[32] Li, Y., Bai, C. and Reddy, C.K., 2016. A distributed ensemble approach for mining healthcare data under privacy constraints. Information sciences, 330, pp.245-259.

[33] Oza, N.C. and Tumer, K., 2008. Classifier ensembles: Select real-world applications. Information Fusion, 9(1), pp.4-20.

[34] Rokach, L., 2010. Ensemble-based classifiers. Artificial Intelligence Review, 33(1-2), pp.1-39.

[35] Ji, S., Pan, S., Long, G., Li, X., Jiang, J. and Huang, Z., 2019. Learning private neural language modeling with attentive aggregation. In 2019 International joint conference on neural networks (IJCNN), pp. 1-8. IEEE.

[36] Bashir, S., Qamar, U. and Khan, F.H., 2016. A multicriteria weighted vote based classifier ensemble for heart disease prediction. Computational Intelligence, 32(4), pp.615-645.

[37] Bashir, S., Qamar, U. and Khan, F.H., 2015. Heterogeneous classifiers fusion for dynamic breast cancer diagnosis using weighted vote based ensemble. Quality & Quantity, 49(5), pp.2061-2076.

[38] Mehdiyev, N., Enke, D., Fettke, P. and Loos, P., 2016. Evaluating forecasting methods by considering different accuracy measures. Procedia Computer Science, 95, pp.264-271.

[39] Li, Q., Wang, Q., Yang, Y. and Li, M., 2008. Reducing biases in individual software effort estimations: a combining approach. In Proceedings of the Second ACM-IEEE international symposium on Empirical software engineering and measurement, pp. 223-232. ACM.
[40] Zhang, C.X. and Duin, R.P., 2011. An experimental study of one-and two-level classifier fusion for different sample sizes. Pattern Recognition Letters, 32(14), pp.1756-1767.
[41] Jochems, A., Deist, T.M., Van Soest, J., Eble, M., Bulens, P., Coucke, P., Dries, W., Lambin, P. and Dekker, A., 2016. Distributed learning: developing a predictive model based on data from multiple hospitals without data leaving the hospital–a real life proof of concept. Radiotherapy and Oncology, 121(3), pp.459-467.
[42] Shipp, C.A. and Kuncheva, L.I., 2002. Relationships between combination methods and measures of diversity in combining classifiers. Information fusion, 3(2), pp.135-148.
[43] Adhikari, A., Jain, L.C. and Prasad, B., 2017. A State-of-the-Art Review of Knowledge Discovery in Multiple Databases. Journal of Intelligent Systems, 26(1), pp.23-34.
[44] Hsu, K.W. and Srivastava, J., 2011. Improving bagging performance through multi-algorithm ensembles. In Pacific-Asia Conference on Knowledge Discovery and Data Mining, pp. 471-482. Springer, Berlin, Heidelberg.
[45] Fumera, G. and Roli, F., 2005. A theoretical and experimental analysis of linear combiners for multiple classifier systems. IEEE transactions on pattern analysis and machine intelligence, 27(6), pp.942-956.
[46] Ponti Jr, M.P., 2011. Combining classifiers: from the creation of ensembles to the decision fusion. In 2011 24th SIBGRAPI Conference on Graphics, Patterns, and Images Tutorials, pp. 1-10. IEEE.
[47] Zarafshan, F., Latif-Shabgahi, G.R. and Karimi, A., 2010. A novel weighted voting algorithm based on neural networks for fault-tolerant systems. In Proceedings of the 3rd IEEE International Conference on Computer Science and Information Technology (ICCSIT'10), 9, pp. 135-139.
[48] Bashir, S., Qamar, U., Khan, F.H. and Naseem, L., 2016. HMV: a medical decision support framework using multi-layer classifiers for disease prediction. Journal of Computational Science, 13, pp.10-25.
[49] Bashir, S., Qamar, U. and Khan, F.H., 2015. BagMOOV: A novel ensemble for heart disease prediction bootstrap aggregation with multi-objective optimized voting. Australasian physical & engineering sciences in medicine, 38(2), pp.305-323.
[50] Damiani, A., Vallati, M., Gatta, R., Dinapoli, N., Jochems, A., Deist, T., Van Soest, J., Dekker, A. and Valentini, V., 2015. Distributed learning to protect privacy in multi-centric clinical studies. In Conference on artificial intelligence in medicine in Europe, pp. 65-75. Springer, Cham.
[51] Lu, C.L., Wang, S., Ji, Z., Wu, Y., Xiong, L., Jiang, X. and Ohno-Machado, L., 2015. WebDISCO: a web service for distributed cox model learning without patient-level data sharing. Journal of the American Medical Informatics Association, 22(6), pp.1212-1219.
[52] Cavalin, P.R., Sabourin, R. and Suen, C.Y., 2013. Dynamic selection approaches for multiple classifier systems. Neural Computing and Applications, 22(3-4), pp.673-688.
[53] Krawczyk, B., Cano, A. and Woźniak, M., 2018. Selecting local ensembles for multi-class imbalanced data classification. In 2018 International joint conference on neural networks (IJCNN), pp. 1-8. IEEE.
[54] Woods, K., Kegelmeyer, W.P. and Bowyer, K., 1997. Combination of multiple classifiers using local accuracy estimates. IEEE transactions on pattern analysis and machine intelligence, 19(4), pp.405-410.

[55] Smits, P.C., 2002. Multiple classifier systems for supervised remote sensing image classification based on dynamic classifier selection. IEEE Transactions on Geoscience and Remote Sensing, 40(4), pp.801-813.

[56] Sabourin, M., Mitiche, A., Thomas, D. and Nagy, G., 1993. Classifier combination for hand-printed digit recognition. In Proceedings of 2nd International Conference on Document Analysis and Recognition (ICDAR'93), pp. 163-166. IEEE.

[57] Giacinto, G. and Roli, F., 2001. Dynamic classifier selection based on multiple classifier behaviour. Pattern Recognition, 34(9), pp.1879-1881.

[58] Dankar, F.K., Madathil, N., Dankar, S.K. and Boughorbel, S., 2019. Privacy-Preserving Analysis of Distributed Biomedical Data: Designing Efficient and Secure Multiparty Computations Using Distributed Statistical Learning Theory. JMIR medical informatics, 7(2), p.e12702.

[59] Jia, Q., Guo, L., Jin, Z. and Fang, Y., 2016. Privacy-preserving data classification and similarity evaluation for distributed systems. In 2016 IEEE 36th International Conference on Distributed Computing Systems (ICDCS), pp. 690-699. IEEE.

[60] Rahulamathavan, Y., Phan, R.C.W., Veluru, S., Cumanan, K. and Rajarajan, M., 2013. Privacy-preserving multi-class support vector machine for outsourcing the data classification in cloud. IEEE Transactions on Dependable and Secure Computing, 11(5), pp.467-479.

[61] Jia, Q., Guo, L., Fang, Y. and Wang, G., 2018. Efficient Privacy-preserving Machine Learning in Hierarchical Distributed System. IEEE Transactions on Network Science and Engineering, 6(4), pp.599-612.

[62] Blanton, M., Kang, A.R., Karan, S. and Zola, J., 2018. Privacy Preserving Analytics on Distributed Medical Data. arXiv preprint arXiv:1806.06477.

[63] Cheng, H.P., Yu, P., Hu, H., Yan, F., Li, S., Li, H. and Chen, Y., 2018. LEASGD: an efficient and privacy-preserving decentralized algorithm for distributed learning. arXiv preprint arXiv:1811.11124.

[64] Krikunov, A.V. and Kovalchuk, S.V., 2015. Dynamic selection of ensemble members in multi-model hydrometeorological ensemble forecasting. Procedia Computer Science, 66, pp.220-227.

[65] Cruz, R.M., Sabourin, R., Cavalcanti, G.D. and Ren, T.I., 2015. META-DES: A dynamic ensemble selection framework using meta-learning. Pattern recognition, 48(5), pp.1925-1935.

[66] Ko, A.H., Sabourin, R. and Britto Jr, A.S., 2008. From dynamic classifier selection to dynamic ensemble selection. Pattern recognition, 41(5), pp.1718-1731.

[67] Cruz, R.M., Sabourin, R. and Cavalcanti, G.D., 2018. Dynamic classifier selection: Recent advances and perspectives. Information Fusion, 41, pp.195-216.

[68] Cruz, R.M., Sabourin, R. and Cavalcanti, G.D., 2014. On meta-learning for dynamic ensemble selection. In 2014 22nd International Conference on Pattern Recognition, pp. 1230-1235. IEEE.

[69] Santana, A., Soares, R.G., Canuto, A.M. and de Souto, M.C., 2006. A dynamic classifier selection method to build ensembles using accuracy and diversity. In 2006 Ninth Brazilian Symposium on Neural Networks (SBRN'06), pp. 36-41. IEEE.

[70] Roy, A., Cruz, R.M., Sabourin, R. and Cavalcanti, G.D., 2018. A study on combining dynamic selection and data preprocessing for imbalance learning. Neurocomputing, 286, pp.179-192.

[71] Kang, S. and Cho, S., 2015. Optimal construction of one-against-one classifier based on meta-learning. Neurocomputing, 167, pp.459-466.

[72] Wolpert, D.H., 1992. Stacked generalization. Neural Networks, 5(2) pp.241–260.
[73] Dzeroski, S. and Zenko, B., 2004. Is combining classifiers with stacking better than selecting the best one?. Machine learning, 54(3), pp.255-273.
[74] Liu, H., Xu, Y. and Chen, C., 2019. Improved pollution forecasting hybrid algorithms based on the ensemble method. Applied Mathematical Modelling, 73, pp.473-486.
[75] Wang, S.Q., Yang, J. and Chou, K.C., 2006. Using stacked generalization to predict membrane protein types based on pseudo-amino acid composition. Journal of theoretical biology, 242(4), pp.941-946.
[76] Peteiro-Barral, D. and Guijarro-Berdiñas, B., 2012. An Analysis of Clustering Approaches to Distributed Learning on Heterogeneously Distributed Datasets. In KES, pp. 69-78.
[77] Wang, Y., Wang, D., Geng, N., Wang, Y., Yin, Y. and Jin, Y., 2019. Stacking-based ensemble learning of decision trees for interpretable prostate cancer detection. Applied Soft Computing, 77, pp.188-204.
[78] Widodo, A. and Budi, I., 2011. Combination of time series forecasts using neural network. In Proceedings of the 2011 International Conference on Electrical Engineering and Informatics, pp. 1-6. IEEE.
[79] Lahiri, K. and Yang, L., 2016. A non-linear forecast combination procedure for binary outcomes. Studies in Nonlinear Dynamics & Econometrics, 20(4), pp.421-440.
[80] Cang, S., 2011. A non-linear tourism demand forecast combination model. Tourism Economics, 17(1), pp.5-20.
[81] Singh, N. and Singh, P., 2020. Stacking-based multi-objective evolutionary ensemble framework for prediction of diabetes mellitus. Biocybernetics and Biomedical Engineering, 40(1), pp.1-22.
[82] Dhanya, R., Paul, I.R., Akula, S.S., Sivakumar, M. and Nair, J.J., 2020. F-test feature selection in Stacking ensemble model for breast cancer prediction. Procedia Computer Science, 171, pp.1561-1570.
[83] Timmermann, A., 2006. Forecast combinations. Handbook of economic forecasting, 1, pp.135-196.
[84] Leng, H., Ao, P., Zhu, J.R. and Tang, H.G., 2017. Combination forecasting method and application based on the fractal dimension weight. Modern Physics Letters B, 31(19-21), p.1740053.
[85] Hibon, M. and Evgeniou, T., 2005. To combine or not to combine: selecting among forecasts and their combinations. International journal of forecasting, 21(1), pp.15-24.
[86] Adhikari, R. and Agrawal, R.K., 2014. Performance evaluation of weights selection schemes for linear combination of multiple forecasts. Artificial Intelligence Review, 42(4), pp.529-548.
[87] Chakour, I., El Mourabit, Y., Daoui, C. and Baslam, M., 2020. Multi-Agent System Based on Machine Learning for Early Diagnosis of Diabetes. In 2020 IEEE 6th International Conference on Optimization and Applications (ICOA), pp. 1-6. IEEE.
[88] Chamikara, M.A.P., Bertok, P., Khalil, I., Liu, D. and Camtepe, S., 2021. Privacy preserving distributed machine learning with federated learning. Computer Communications, 171, pp.112-125.
[89] Yang, Q., Liu, Y., Chen, T. and Tong, Y., 2019. Federated machine learning: Concept and applications. ACM Transactions on Intelligent Systems and Technology (TIST), 10(2), pp.1-19.
[90] Truex, S., Baracaldo, N., Anwar, A., Steinke, T., Ludwig, H., Zhang, R. and Zhou, Y., 2019. A hybrid approach to privacy-preserving federated learning. In Proceedings of the 12th ACM workshop on artificial intelligence and security, pp. 1-11.
[91] Wang, R., Ji, W., Liu, M., Wang, X., Weng, J., Deng, S., Gao, S. and Yuan, C.A., 2018. Review on mining data from multiple data sources. Pattern Recognition Letters, 109,

pp.120-128.
[92] Chen, S., 2018. A distributed algorithm for machine learning. In AIP Conference Proceedings,1955(1), pp. 040079). AIP Publishing LLC.
[93] Mohapatra, U.M., Majhi, B. and Satapathy, S.C., 2019. Financial time series prediction using distributed machine learning techniques. Neural Computing and Applications, 31(8), pp.3369-3384.
[94] Dong, J.R., 2002. A nonlinear combining forecast method based on fuzzy neural network. In Proceedings. International Conference on Machine Learning and Cybernetics, 4, pp. 2160-2164. IEEE.
[95] Asad, M., Moustafa, A. and Ito, T., 2020. FedOpt: Towards Communication Efficiency and Privacy Preservation in Federated Learning. Applied Sciences, 10(8), p.2864.
[96] Zhang, Z., Chung, F.L. and Wang, S., 2019. Collaborative classification mechanism for privacy-Preserving on horizontally partitioned data. Automatika, 60(1), pp.58-67.
[97] Kairouz, P., et al. 2021. Advances and open problems in federated learning. Foundations and Trends in Machine Learning:,14(1-2), pp.1-210.
[98] Tsai, C.F., Lin, W.C. and Ke, S.W., 2016. Big data mining with parallel computing: A comparison of distributed and MapReduce methodologies. Journal of Systems and Software, 122, pp.83-92.
[99] Taşcıkaraoğlu, A., 2017. Impacts of Accurate Renewable Power Forecasting on Optimum Operation of Power System. In Optimization in Renewable Energy Systems, pp. 159-175. Butterworth-Heinemann.
[100] Plis, S.M., Sarwate, A.D., Wood, D., Dieringer, C., Landis, D., Reed, C., Panta, S.R., Turner, J.A., Shoemaker, J.M., Carter, K.W. and Thompson, P., 2016. COINSTAC: a privacy enabled model and prototype for leveraging and processing decentralized brain imaging data. Frontiers in neuroscience, 10, p.365.
[101] Si, Y., Du, J., Li, Z., Jiang, X., Miller, T., Wang, F., Zheng, W.J. and Roberts, K., 2021. Deep representation learning of patient data from Electronic Health Records (EHR): A systematic review. Journal of Biomedical Informatics, 115, p.103671.
[102] Li, L., Fan, Y., Tse, M. and Lin, K.Y., 2020. A review of applications in federated learning. Computers & Industrial Engineering, 149, p.106854.
[103] Nilsson, A., Smith, S., Ulm, G., Gustavsson, E. and Jirstrand, M., 2018. A performance evaluation of federated learning algorithms. In Proceedings of the Second Workshop on Distributed Infrastructures for Deep Learning, pp. 1-8.
[104] Cruz, R.M.O., Sabourin, R., Cavalcanti, G.D.C., 2018. Prototype selection for dynamic classifier and ensemble selection. Neural Comput. Appl. 29(2), pp. 447–457.
[105] Hayden, E., Stomper, A. and Westerkamp, A., 2014. Selection versus averaging of logistic credit risk models. Journal of Risk, 16(5), pp.39-52.
[106] Wan, L., Chen, H., Ouyang, L. and Chen, Y., 2020. A new ensemble modeling approach for reliability-based design optimization of flexure-based bridge-type amplification mechanisms. The International Journal of Advanced Manufacturing Technology, 106(1), pp.47-63.
[107] Caruana, R., Niculescu-Mizil, A., Crew, G. and Ksikes, A., 2004. Ensemble selection from libraries of models. In Proceedings of the twenty-first international conference on Machine learning, pp.18-38.
[108] Allende-Cid, H., 2017. Distributed Machine Learning with Context-Awareness for the Regression Task. In Claudio Moraga: A Passion for Multi-Valued Logic and Soft Computing, pp. 305-322. Springer, Cham.
[109] Li, Z., Roberts, K., Jiang, X. and Long, Q., 2019. Distributed learning from multiple EHR databases: Contextual embedding models for medical events. Journal of biomedical informatics, 92, p.103138.
[110] Yin, S., Liu, L. and Hou, J., 2016. A multivariate statistical combination forecasting method for product quality evaluation. Information Sciences, 355, pp.229-236.
[111] Walkowiak, K., Sztajer, S. and Woźniak, M., 2011. Decentralized distributed computing system for privacy-preserving combined classifiers–modeling and optimization.

In International Conference on Computational Science and Its Applications, 6782(1), pp. 512-525. Springer, Berlin, Heidelberg.
[112] Kourou, K., Exarchos, T.P., Exarchos, K.P., Karamouzis, M.V. and Fotiadis, D.I., 2015. Machine learning applications in cancer prognosis and prediction. Computational and structural biotechnology journal, 13, pp.8-17.
[113] Tremblay, M., Kammer, M., Lange, H., Plattner, S., Baumgartner, C., Stegeman, J.A., Duda, J., Mansfeld, R. and Döpfer, D., 2019. Prediction model optimization using full model selection with regression trees demonstrated with FTIR data from bovine milk. Preventive veterinary medicine, 163, pp.14-23.
[114] Broomhead, D.S., Low, D., 1988. Radial basis functions, Multivariable functional interpolation and adaptive networks. Complex Syst. 2, pp. 321-355
[115] Song, C. and Fu, X., 2020. Research on different weight combination in air quality forecasting models. Journal of Cleaner Production, 261, p.121169.
[116] Kumar, R and Indrayan, A., 2011. Receiver operating characteristic (ROC) curve for medical researchers. Indian Pediatrics, 48(4), pp. 277–287.
[117] Hanley J.A. and McNeil, B.J., 1982. The meaning and use of the area under a receiver operating characteristic (ROC) curve. Radiology, 143(1), pp.29–36.
[118] Armstrong, J.S., 2001. Combining forecasts. In Principles of forecasting, pp. 417-439. Springer, Boston, MA.
[119] Rokach, L., 2010. Pattern classification using ensemble methods, 75. World Scientific.
[120] Kumar, L., Misra, S. and Rath, S.K., 2017. An empirical analysis of the effectiveness of software metrics and fault prediction model for identifying faulty classes. Computer Standards & Interfaces, 53, pp.1-32.
[121] Kittler, J., Hatef, M., Duin, R.P. and Matas, J., 1998. On combining classifiers. IEEE transactions on pattern analysis and machine intelligence, 20(3), pp.226-239.
[122] Tascikaraoglu, A. and Uzunoglu, M., 2014. A review of combined approaches for prediction of short-term wind speed and power. Renewable and Sustainable Energy Reviews, 34, pp.243-254.
[123] Zhang, L. and Zhou, W.D., 2011. Sparse ensembles using weighted combination methods based on linear programming. Pattern Recognition, 44(1), pp.97-106.
[124] Jackowski, K., 2018. New diversity measure for data stream classification ensembles. Engineering Applications of Artificial Intelligence, 74, pp.23-34.
[125] Barak, S., Arjmand, A. and Ortobelli, S., 2017. Fusion of multiple diverse predictors in stock market. Information Fusion, 36, pp.90-102.
[126] Pal, S.K. and Mitra, P., 2004. Pattern recognition algorithms for data mining. CRC press.
[127] Ngo, V., Wu, W., Yang, Y. and Zhang, B., 2017. Cooperative game-based method to determine the weights of load forecasting solution incorporated with various algorithms. The Journal of Engineering, 2017(13), pp.1312-1315.
[128] Zhong-Xiang, F., Shi-Sheng, L., Wei-Hua, Z. and Nan-Nan, Z., 2014. Combined prediction model of death toll for road traffic accidents based on independent and dependent variables. Computational intelligence and neuroscience, 2014.
[129] Apache, Hadoop, http://hadoop.apache.org , last accessed 2021/1/20.
[130] Apache, Spark, http://spark.apache.org/, last accessed 2020/6/27.
[131] Bates, J.M. and Granger, C.W.J., 1969. The combination of forecasts. Journal of the Operational Research Quarterly 20 (4), pp.451-468.
[132] Xing, E.P., Ho, Q., Xie, P. and Wei, D., 2016. Strategies and principles of distributed machine learning on big data. Engineering, 2(2), pp.179-195.
[133] Pompigna, A. and Rupi, F., 2018. Comparing practice-ready forecast models for weekly and monthly fluctuations of average daily traffic and enhancing accuracy by weighting methods. Journal of Traffic and Transportation Engineering, 5(4), pp.239-253.
[134] Rzychoń, M., Żogała, A. and Róg, L., 2021. SHAP-based interpretation of an XGBoost model in the prediction of grindability of coals and their blends. International Journal of Coal Preparation and Utilization, pp.1-21.
[135] Genre, V., Kenny, G., Meyler, A. and Timmermann, A., 2013. Combining expert

forecasts: Can anything beat the simple average?. International Journal of Forecasting, 29(1), pp.108-121.
[136] Shi, S.M., Da Xu, L. and Liu, B., 1999. Improving the accuracy of nonlinear combined forecasting using neural networks. Expert Systems with Applications, 16(1), pp.49-54.
[137] Fredrikson, M., Jha, S. and Ristenpart, T., 2015, October. Model inversion attacks that exploit confidence information and basic countermeasures. In Proceedings of the 22nd ACM SIGSAC conference on computer and communications security, pp. 1322-1333.
[138] Andrawis, R.R., Atiya, A.F. and El-Shishiny, H., 2011. Combination of long term and short term forecasts, with application to tourism demand forecasting. International Journal of Forecasting, 27(3), pp.870-886.
[139] Ormándi, R., Hegedűs, I. and Jelasity, M., 2013. Gossip learning with linear models on fully distributed data. Concurrency and Computation: Practice and Experience, 25(4), pp.556-571.
[140] Hegedűs, I., Danner, G. and Jelasity, M., 2021. Decentralized learning works: An empirical comparison of gossip learning and federated learning. Journal of Parallel and Distributed Computing, 148, pp.109-124.
[141] Hegedűs, I., Danner, G. and Jelasity, M., 2019. Gossip learning as a decentralized alternative to federated learning. In IFIP International Conference on Distributed Applications and Interoperable Systems, 11534, pp. 74-90. Springer, Cham.
[142] Bottou, L., 2012. Stochastic gradient descent tricks. In Neural networks: Tricks of the trade, 7700, pp. 421-436. Springer, Berlin, Heidelberg.
[143] Danner, G. and Jelasity, M., 2015. Fully distributed privacy preserving mini-batch gradient descent learning. In IFIP International conference on distributed applications and interoperable systems, 9038, pp. 30-44. Springer, Cham.
[144] Hegedűs, I., Danner, G. and Jelasity, M., 2019, Decentralized recommendation based on matrix factorization: A comparison of gossip and federated learning. In Joint European Conference on Machine Learning and Knowledge Discovery in Databases, 1167, pp. 317-332. Springer, Cham.
[145] Kotu, V. and Deshpande, B., 2018. Data science: concepts and practice. Morgan Kaufmann.
[146] Quinlan, J. R., 1986. Induction of decision trees. Machine Learning, 1(1), pp.81-106.
[147] Brieman, L. F., 1984. Classification and regression trees. Chapman and Hall.
[148] Cover, T. and Hart, P., 1967. Nearest neighbor pattern classification. IEEE transactions on information theory, 13(1), pp.21-27.
[149] Breiman, L., 2001. Random forests. Machine Learning, 45, pp.5-32.
[150] Ho, T.K., 1995. Random decision forests. In Proceedings of 3rd international conference on document analysis and recognition, 1, pp. 278-282. IEEE.
[151] Tibshirani, R., 1996. Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society: Series B (Methodological), 58(1), pp.267-288.
[152] Zou, H. and Hastie, T., 2005. Regularization and variable selection via the elastic net. Journal of the royal statistical society: series B (statistical methodology), 67(2), pp.301-320.
[153] Hoerl, A.E. and Kennard, R.W., 1970. Ridge regression: Biased estimation for nonorthogonal problems. Technometrics, 12(1), pp.55-67.
[154] Cuzzocrea, A.M., 2013. Models and algorithms for high-performance distributed data mining.
[155] Moon, S., Song, H.J., Sharma, V.D., Lyons, K.E., Pahwa, R., Akinwuntan, A.E., Devos, H.,2020. Classification of Parkinson's disease and essential tremor based on balance and gait characteristics from wearable motion sensors via machine learning techniques: a data-driven approach. Journal of NeuroEngineering and Rehabilitation 17(1), 1-8.
[156] Bottou, L., 2010. Large-scale machine learning with stochastic gradient descent. In Proceedings of COMPSTAT'2010, pp. 177-186. Physica-Verlag HD.
[157] Drucker, H., Burges, C.J., Kaufman, L., Smola, A. and Vapnik, V., 1996. Support vector regression machines. Advances in neural information processing systems, 9.

[158] Gardner, M.W. and Dorling, S.R., 1998. Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences. Atmospheric environment, 32(14-15), pp.2627-2636.
[159] Altman, N. S., 1992. An introduction to kernel and nearest-neighbor nonparametric regression. The American Statistician, 46(3), pp.175-185.
[160] Zhang, Z.L., Luo, X.G., García, S., Tang, J.F. and Herrera, F., 2017. Exploring the effectiveness of dynamic ensemble selection in the one-versus-one scheme. Knowledge-Based Systems, 125, pp.53-63.
[161] Danner, G., Berta, Á., Hegedűs, I. and Jelasity, M., 2018. Robust fully distributed minibatch gradient descent with privacy preservation. Security and Communication Networks, pp. 1-5.
[162] Terziyska, M., Todorov, Y. and Dobreva, M., 2018. Efficient error based metrics for fuzzy-neural network performance evaluation. In Advanced Computing in Industrial Mathematics, 728, pp. 185-201. Springer, Cham.
[163] Osamor, V.C. and Okezie, A.F., 2021. Enhancing the weighted voting ensemble algorithm for tuberculosis predictive diagnosis. Scientific Reports, 11(1), pp.1-11.
[164] Tramèr, F., Zhang, F., Juels, A., Reiter, M.K. and Ristenpart, T., 2016. Stealing machine learning models via prediction {APIs}. In 25th USENIX security symposium (USENIX Security 16), pp. 601-618.
[165] Hu, R., Guo, Y., Li, H., Pei, Q. and Gong, Y., 2020. Privacy-preserving personalized federated learning. In ICC 2020-2020 IEEE International Conference on Communications (ICC), pp. 1-6. IEEE.
[166] Mandal, K. and Gong, G., 2019. PrivFL: Practical privacy-preserving federated regressions on high-dimensional data over mobile networks. In Proceedings of the 2019 ACM SIGSAC Conference on Cloud Computing Security Workshop, pp. 57-68.
[167] Injadat, M., Moubayed, A., Nassif, A.B. and Shami, A., 2021. Machine learning towards intelligent systems: applications, challenges, and opportunities. Artificial Intelligence Review, 54(5), pp.3299-3348.
[168] Kotsiantis, S.B., Zaharakis, I.D. and Pintelas, P.E., 2006. Machine learning: a review of classification and combining techniques. Artificial Intelligence Review, 26(3), pp.159-190.
[169] Nguyen, T.T., Nguyen, M.P., Pham, X.C. and Liew, A.W.C., 2018. Linear and non-linear with fuzzy rule-based meta learner. Information Sciences, 422, pp.144-160.
[170] Wang, F., Zhu, H., Lu, R., Zheng, Y. and Li, H., 2021. A privacy-preserving and non-interactive federated learning scheme for regression training with gradient descent. Information Sciences, 552, pp.183-200.
[171] Abed, M., Imteaz, M.A., Ahmed, A.N. and Huang, Y.F., 2021. Application of long short-term memory neural network technique for predicting monthly pan evaporation. Scientific Reports, 11(1), pp.1-19.
[172] Mehta, S. and Patnaik, K.S., 2021. Improved prediction of software defects using ensemble machine learning techniques. Neural Computing and Applications, 33(16), pp.10551-10562.
[173] Rathore, S.S., 2021. An exploratory analysis of regression methods for predicting faults in software systems. Soft Computing, 25(23), pp.14841-14872.
[174] Géron, A., 2019. Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems. " O'Reilly Media, Inc.".
[175] Pedregosa, F. et al., 2011. Scikit-learn: Machine learning in Python. Journal of machine learning research, 12(Oct), pp.2825–2830.
[176] Gao, D., Liu, Y., Huang, A., Ju, C., Yu, H., Yang, Q.,2019. Privacy-preserving heterogeneous federated transfer learning. In: 2019 IEEE International Conference on Big Data (Big Data) , pp. 2552-2559, IEEE..
[177] Kasturi, A., Ellore, A.R., Hota, C., 2020. Fusion learning: A one shot federated learning. In: International Conference on Computational Science, pp. 424-436, Springer, Cham.
[178] Ma, H., Guo, X., Ping, Y., Wang, B., Yang, Y., Zhang, Z. and Zhou, J., 2019. PPCD: privacy-preserving clinical decision with cloud support. Plos one, 14(5), p.e0217349.

[179] Haque, R.U., Hasan, A.S.M., Jiang, Q. and Qu, Q., 2020. Privacy-Preserving K-Nearest Neighbors Training over Blockchain-Based Encrypted Health Data. Electronics, 9(12), p.2096.

[180] Sav, S., Pyrgelis, A., Troncoso-Pastoriza, J.R., Froelicher, D., Bossuat, J.P., Sousa, J.S. and Hubaux, J.P., 2020. POSEIDON: Privacy-Preserving Federated Neural Network Learning. arXiv preprint arXiv:2009.00349.

[181] Froelicher, D., Troncoso-Pastoriza, J.R., Pyrgelis, A., Sav, S., Sousa, J.S., Bossuat, J.P. and Hubaux, J.P., 2020. Scalable Privacy-Preserving Distributed Learning. arXiv preprint arXiv:2005.09532.

[182] Ed-daoudy, A. and Maalmi, K., 2020. Real-time heart disease detection and monitoring system based on fast machine learning using Spark. Health and Technology, 10(5), pp.1145-1154.

[183] Lundberg, S.M. and Lee, S.I., 2017, December. A unified approach to interpreting model predictions. In Proceedings of the 31st international conference on neural information processing systems (pp. 4768-4777).

[184] Cheng, F., Zhou, Y., Gao, J. and Zheng, S., 2016. Efficient Optimization of-Measure with Cost-Sensitive SVM. Mathematical Problems in Engineering, 2016.

[185] Visentini, I., Snidaro, L., Foresti, G.L.: Diversity-aware classifier ensemble selection via f-score. Info. Fusion 28, 24-43 (2016).

[186] Ateniese, G., Felici, G., Mancini, L.V., Spognardi, A., Villani, A. and Vitali, D., 2013. Hacking Smart Machines with Smarter Ones: How to Extract Meaningful Data from Machine Learning Classifiers. arXiv preprint arXiv:1306.4447.

[187] Jiang, C., Xu, C. and Zhang, Y., 2021. PFLM: Privacy-preserving federated learning with membership proof. Information Sciences, 576, pp.288-311.

[188] Vu, D.H., 2022. Privacy-preserving Naive Bayes classification in semi-fully distributed data model. Computers & Security, 115, p.102630.

[189] Ku, H., Susilo, W., Zhang, Y., Liu, W. and Zhang, M., 2022. Privacy-Preserving federated learning in medical diagnosis with homomorphic re-Encryption. Computer Standards & Interfaces, 80, p.103583.

[190] Lyu, L., Law, Y.W., Ng, K.S., Xue, S., Zhao, J., Yang, M. and Liu, L., 2020, October. Towards Distributed Privacy-Preserving Prediction. In 2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC) (pp. 4179-4184). IEEE.

[191] Francis, S., Tenison, I. and Rish, I., 2021. Towards causal federated learning for enhanced robustness and privacy. arXiv preprint arXiv:2104.06557.

[192] Xu, R., Baracaldo, N. and Joshi, J., 2021. Privacy-preserving machine learning: Methods, challenges and directions. arXiv preprint arXiv:2108.04417.

Appendix A

In this appendix, we present 5-fold and 10-fold cross validation results and the distributed datasets distributions for the classification and regression data that partitioned in random and non-random ways.

1) 5-fold and 10-fold Cross Validation Results

a) Classification

Dataset	Models		5-fold Cross-Validation	10-fold Cross-Validation
Breast Cancer Wisconsin (Diagnostic)	LR	Accuracy	93%	**94%**
		F-measure	93%	93%
	RF	Accuracy	93%	**94%**
		F-measure	**94%**	93%
	NB	Accuracy	**94%**	93%
		F-measure	93%	93%
	KNN	Accuracy	90%	**91%**
		F-measure	**90%**	90%
	DT	Accuracy	**9%**	90%
		F-measure	80%	**87%**
	SVM	Accuracy	52%	52%
		F-measure	36%	36%
	NN	Accuracy	52%	**57%**
		F-measure	35%	**39%**
Diabetes	LR	Accuracy	72%	72%
		F-measure	**61%**	57%
	RF	Accuracy	69%	69%
		F-measure	53%	55%
	NB	Accuracy	71%	72%
		F-measure	60%	**62%**
	KNN	Accuracy	**63%**	**62%**
		F-measure	43%	**49%**
	DT	Accuracy	**63%**	61%
		F-measure	51%	**55%**
	SVM	Accuracy	62%	62%
		F-measure	47%	47%

	NN	Accuracy	61%	**62%**
		F-measure	47%	47%
Heart Disease	LR	Accuracy	**84%**	82%
		F-measure	**84%**	83%
	RF	Accuracy	79%	83%
		F-measure	81%	**84%**
	NB	Accuracy	82%	**83%**
		F-measure	**87%**	80%
	KNN	Accuracy	63%	**66%**
		F-measure	61%	61%
	DT	Accuracy	64%	**76%**
		F-measure	73%	**76%**
	SVM	Accuracy	54%	**56%**
		F-measure	38%	**41%**
	NN	Accuracy	**59%**	58%
		F-measure	**46%**	37%
Cardiovascular Disease	LR	Accuracy	72%	72%
		F-measure	70%	**72%**
	RF	Accuracy	70%	**71%**
		F-measure	69%	**70%**
	NB	Accuracy	71%	71%
		F-measure	67%	**71%**
	KNN	Accuracy	68%	**69%**
		F-measure	67%	**68%**
	DT	Accuracy	62%	**63%**
		F-measure	61%	**63%**
	SVM	Accuracy	67%	67%
		F-measure	67%	67%
	NN	Accuracy	**59%**	53%
		F-measure	**41%**	39%
Diabetes (non-randomly partitioned)	LR	Accuracy	84%	84%
		F-measure	47%	**50%**
	RF	Accuracy	84%	84%
		F-measure	46%	**49%**
	NB	Accuracy	85%	85%
		F-measure	**57%**	55%
	KNN	Accuracy	81%	**83%**
		F-measure	**45%**	42%
	DT	Accuracy	**79%**	77%
		F-measure	40%	**48%**
	SVM	Accuracy	80%	80%
		F-measure	51%	**52%**
	NN	Accuracy	78%	**79%**
		F-measure	51%	51%
Liver Disease (non-randomly partitioned)	LR	Accuracy	**63%**	62%
		F-measure	59%	**60%**
	RF	Accuracy	**72%**	69%
		F-measure	**67%**	66%
	NB	Accuracy	59%	59%
		F-measure	**44%**	39%
	KNN	Accuracy	58%	58%
		F-measure	**59%**	54%
	DT	Accuracy	59%	**60%**

		F-measure	56%	**57%**
	SVM	Accuracy	52%	**54%**
		F-measure	69%	69%
	NN	Accuracy	**56%**	48%
		F-measure	**46%**	43%

b) Regression

Dataset	Models		5-fold Cross-Validation	10-fold Cross-Validation
Abalone	LR	RMSE	**2.20**	2.21
		MAPE	**14.08**	14.16
	RFR	RMSE	2.25	**2.21**
		MAPE	**13.79**	14.00
	RBFNN	RMSE	2.24	**2.21**
		MAPE	**27.37**	27.38
	KNNR	RMSE	2.29	**2.26**
		MAPE	13.69	**13.48**
	DTR	RMSE	2.78	**2.77**
		MAPE	**16.30**	17.05
	SVR	RMSE	2.54	**2.51**
		MAPE	13.64	**13.54**
	NNR	RMSE	**2.13**	2.14
		MAPE	**13.44**	13.49
	LASSO	RMSE	3.11	**3.10**
		MAPE	**21.74**	21.76
	Ridge	RMSE	2.28	**2.27**
		MAPE	14.42	**14.39**
	ElasticNet	RMSE	3.06	3.06
		MAPE	**21.23**	21.26
Parkinson Disease (total UPDRS)	LR	RMSE	10.32	**10.29**
		MAPE	28.49	**28.43**
	RFR	RMSE	9.11	**9.04**
		MAPE	23.39	**22.99**
	RBFNN	RMSE	11.88	**11.85**
		MAPE	35.01	**34.91**
	KNNR	RMSE	**11.37**	11.48
		MAPE	**31.18**	31.31
	DTR	RMSE	12.33	**11.93**
		MAPE	27.73	**25.98**
	SVR	RMSE	12.11	**12.03**
		MAPE	30.23	**29.96**
	NNR	RMSE	10.84	**10.81**
		MAPE	31.17	**31.12**
	LASSO	RMSE	11.90	**11.89**
		MAPE	34.65	**34.63**
	Ridge	RMSE	11.16	**11.14**

		MAPE	32.21	**32.15**
	ElasticNet	RMSE	11.89	11.89
		MAPE	34.64	34.64
Boston Housing	LR	RMSE	4.77	4.77
		MAPE	**17.82**	17.88
	RFR	RMSE	4.09	**3.63**
		MAPE	13.25	**12.93**
	RBFNN	RMSE	7.81	**7.76**
		MAPE	**28.66**	28.78
	KNNR	RMSE	6.69	**6.47**
		MAPE	21.75	**21.69**
	DTR	RMSE	5.57	**4.78**
		MAPE	18.96	**17.37**
	SVR	RMSE	4.62	**4.51**
		MAPE	16.05	**16.04**
	NNR	RMSE	9.82	**7.16**
		MAPE	40.51	**31.39**
	LASSO	RMSE	5.08	**4.88**
		MAPE	17.36	**16.91**
	Ridge	RMSE	**4.53**	4.55
		MAPE	**17.24**	17.39
	ElasticNet	RMSE	4.89	**4.72**
		MAPE	16.94	**16.79**

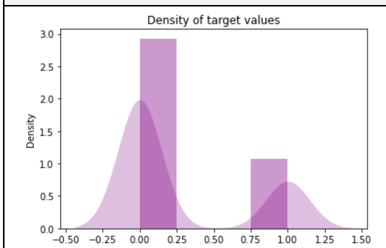
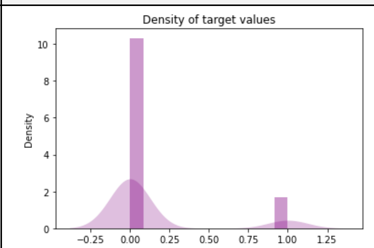
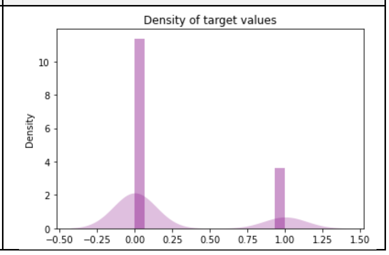
2) Datasets Distributions

a) Classification:

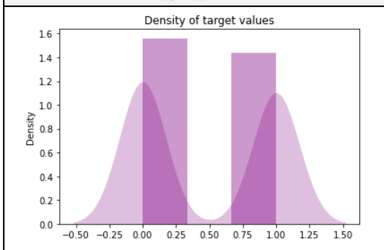
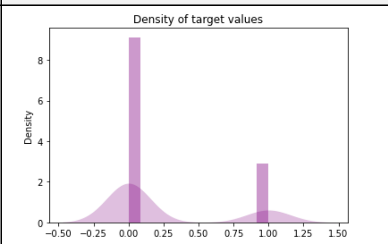
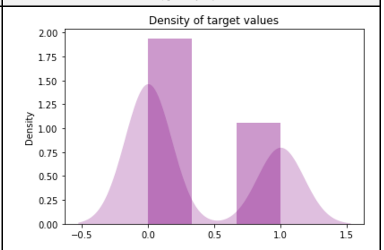
We used eight classification databases: blood transfusion, liver disease, diabetes, heart disease, lower back pain (spine disease), breast cancer Wisconsin (Diagnostic), breast cancer Wisconsin (Original) [29], and cardiovascular diseases [27]. The datasets are with a binary target value, 0 and 1. We applied the proposed methods using two dataset partitioning strategies to mimic a real-world scenario for distributed datasets for distributed sites. The strategies are (1) random data partitioning approach and (2) non-random data partitioning approach. For non-random data partitioning approach, we partitioned the data based on patients' age; we assumed each partition is from a hospital whose patients are in a specific age range to simulate the distributed data. The data are liver disease, diabetes, and heart disease datasets. Therefore, we divided each dataset into different parts as distributed sites site1, site2, and site3.

I. Randomly Partitioned data:

1. Blood Transfusion

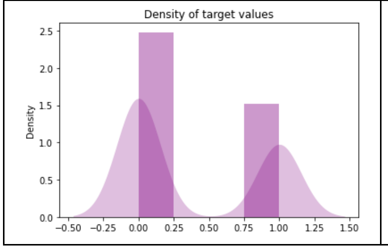
Site1	Site2	Site3
		

2. Breast Cancer Wisconsin (Diagnostic)

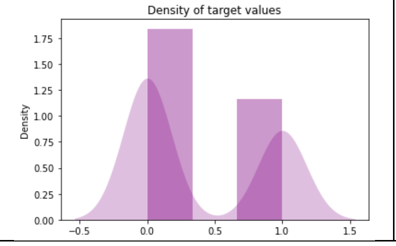
Site 1	Site 2	Site 3
		

3. Diabetes

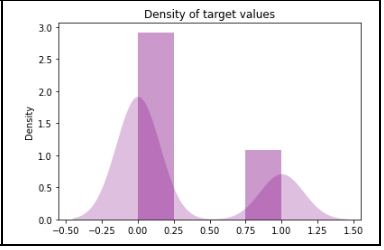
Site 1



Site 2

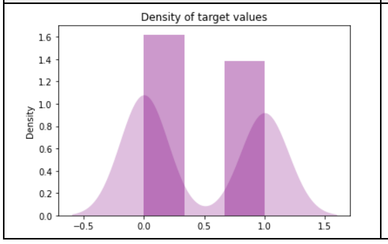


Site 3

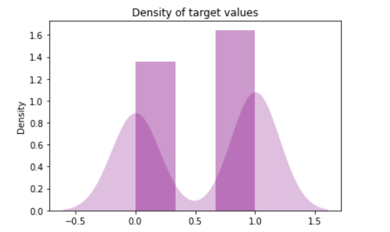


4. Heart Disease

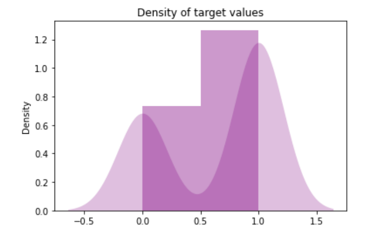
Site 1



Site 2

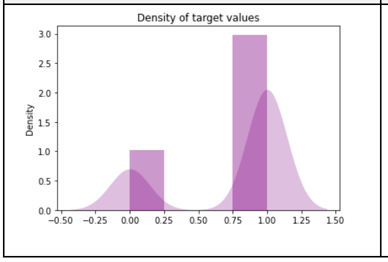


Site 3

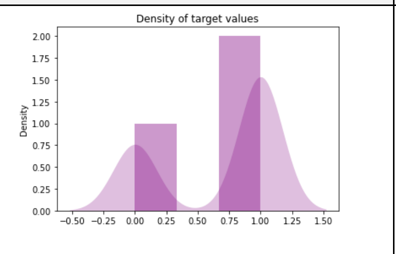


5. Liver Disease

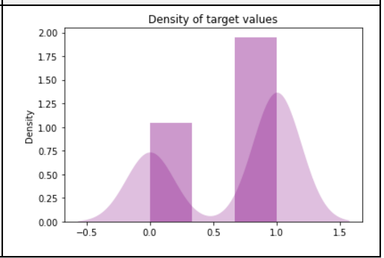
Site 1



Site 2

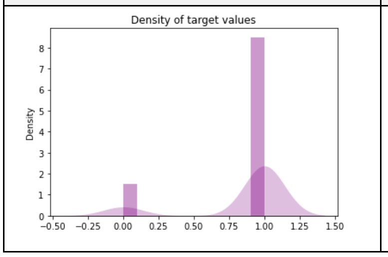


Site 3

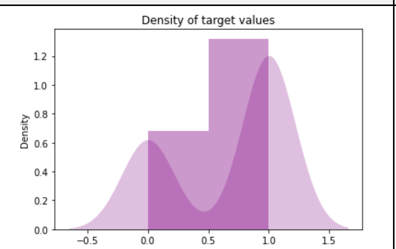


6. Spine Disease

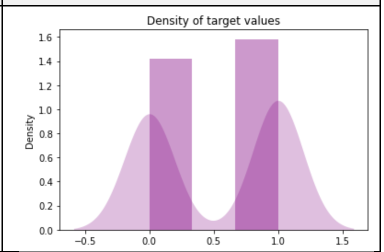
Site 1



Site 2



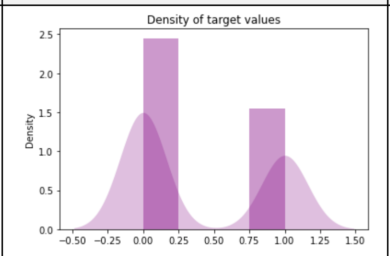
Site 3



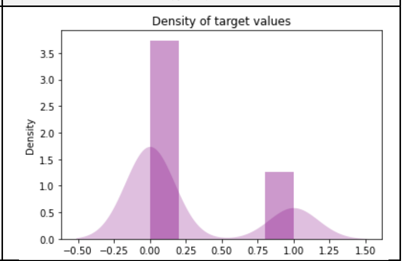
7. Breast Cancer Wisconsin (Original)

Site 1

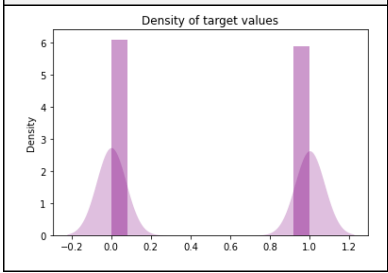
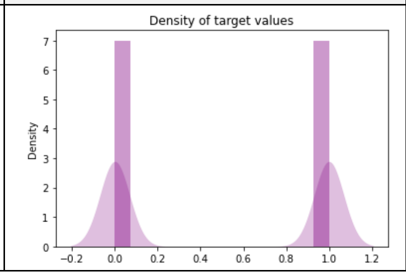
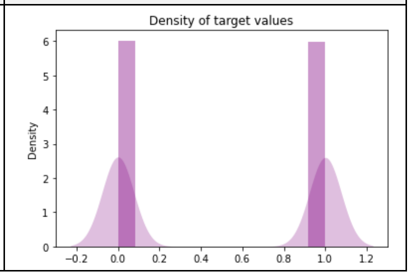
Site 2



Site 3

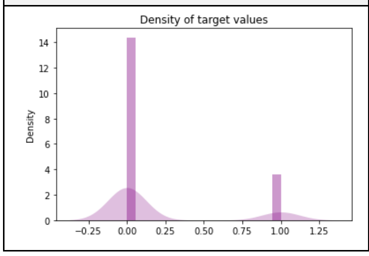
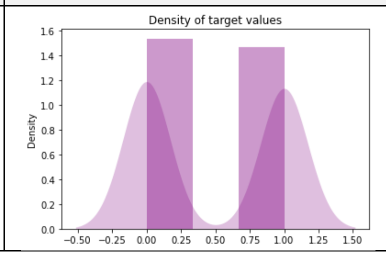
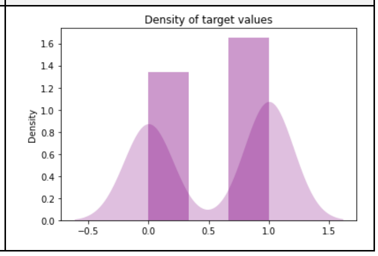


8. Cardiovascular diseases

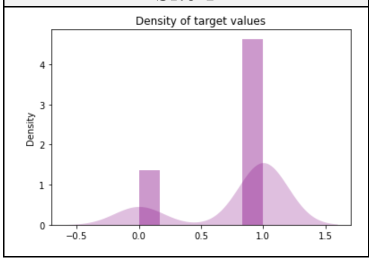
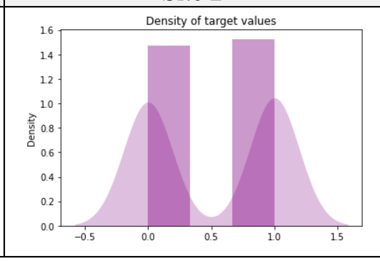
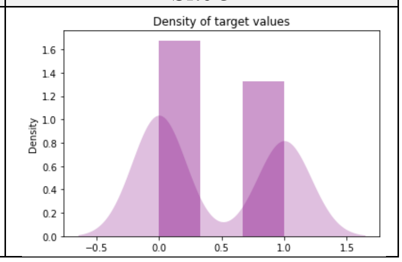
Site 1	Site 2	Site 3
		

II. Non-randomly Partitioned Data:

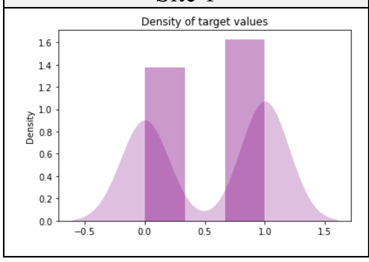
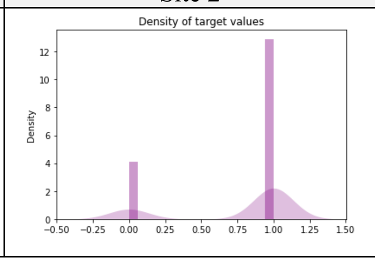
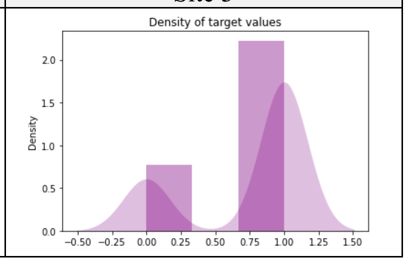
1. Diabetes

Site 1	Site 2	Site 3
		

2. Heart Disease

Site 1	Site 2	Site 3
		

3. Liver Disease

Site 1	Site 2	Site 3
		

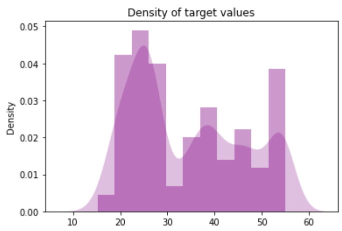
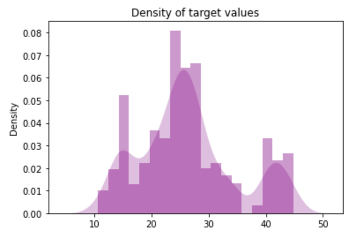
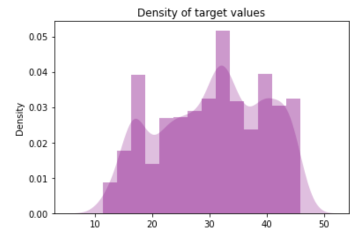
b) Regression

We used three regression databases: Parkinson disease, Boston housing, and Abalone datasets [29]. Parkinson disease data features are patient age and biomedical voice measurement with two target values, motor Unified Parkinson's Disease Rating Scale (UPDRS) and total Unified Parkinson's Disease Rating Scale

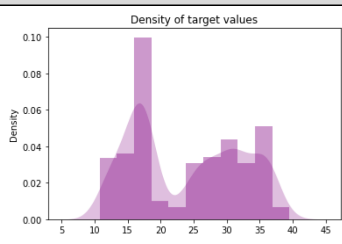
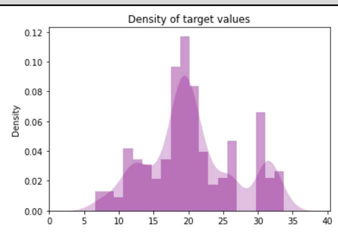
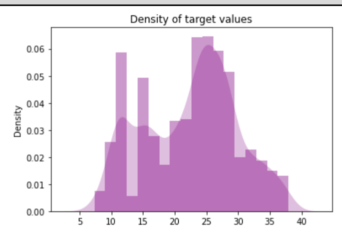
(UPDRS). The target values show the measurement of presence and severity of Parkinson disease. Total-UPDRS ranges between 0–176, 0 reflecting healthy status and 176 indicates total disability. Motor-UPDRS, which denotes to the motor section, the range is between 0–108, 0 indicates healthy status and 108 severe cases. Boston housing dataset are from several suburbs in Boston and includes economic, demographic, and land use features, and the median price of houses is the target value. Abalone dataset features are physical measurements that used to predict the age of Abalone. We applied the proposed methods using two dataset partitioning strategies: (1) random data partitioning approach and (2) non-random data partitioning approach. For the non-random data partitioning approach, we partitioned the Parkinson disease dataset by patient age and Boston housing dataset by per capita crime rate by town attribute to simulate that each data comes from different region. Therefore, we divided each dataset into different parts as distributed sites site1, site2, and site3.

I. Randomly Partitioned Data:

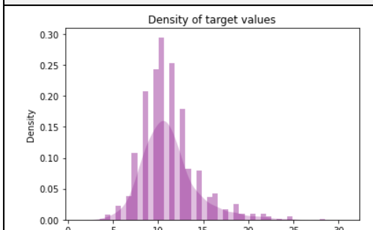
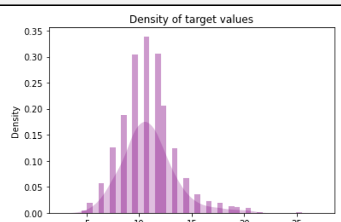
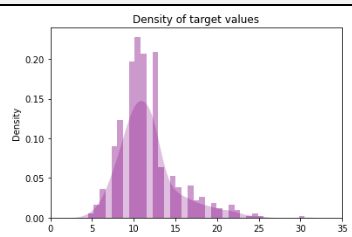
1. Parkinson Disease (Total UPDRS)

Site1	Site2	Site3
		

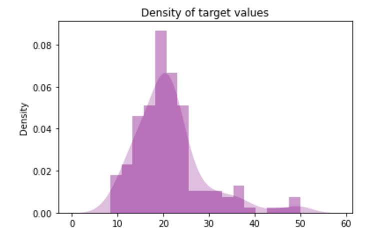
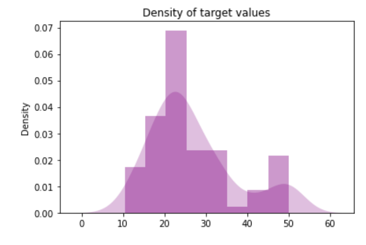
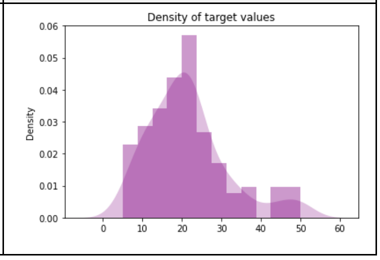
2. Parkinson Disease (Motor UPDRS)

Site1	Site2	Site3
		

3. Abalone

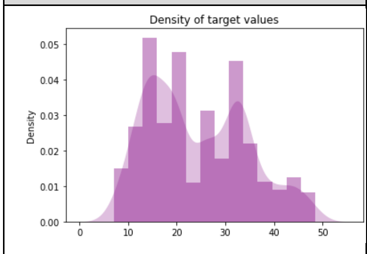
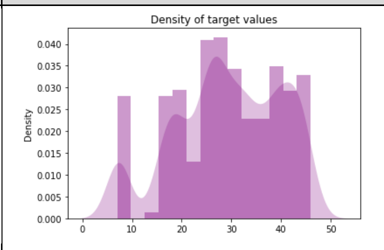
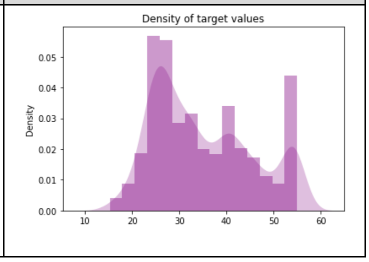
Site1	Site2	Site3
		

4. Boston Housing

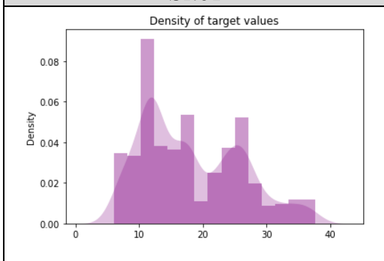
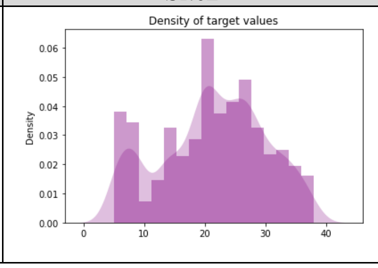
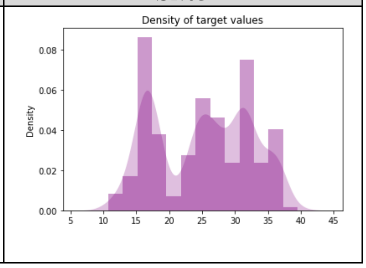
Site1	Site2	Site3
		

II. Non-randomly Partitioned Data:

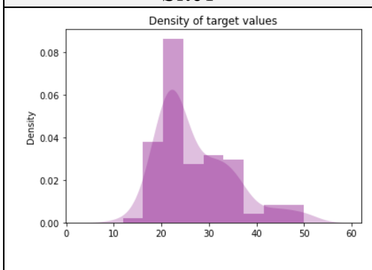
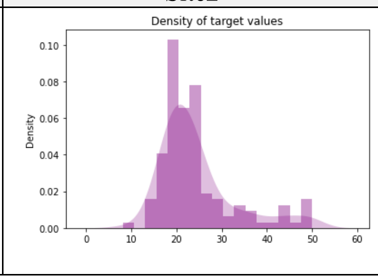
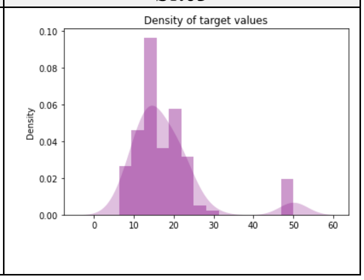
1. Parkinson Disease (Total UPDRS)

Site1	Site2	Site3
		

2. Parkinson Disease (Motor UPDRS)

Site1	Site2	Site3
		

3. Boston Housing

Site1	Site2	Site3
		

Appendix B

Detailed Results for the Proposed Method in Chapter 3

In this appendix we show the detailed results for global and local-level modelling approaches for the classification and regression datasets. It shows models evaluation results locally using 10-fold cross-validation results, on other distributed sites, and the average prediction performance based on sites data size.

a) Classification:

I. Randomly Partitioned Dataset:

1. Blood Transfusion Dataset

Table B. 1. Site 1 Models Evaluation Using Accuracy Metric

Models	Local accuracy (D1)	On Site2 (D2)	On Site3 (D3)	Average accuracy
LR (M_{11})	**74%**	**89%**	**78%**	**79%**
RF (M_{12})	68%	**91%**	75%	76%
NB (M_{13})	73%	**89%**	78%	78%
KNN (M_{14})	67%	85%	75%	74%
DT (M_{15})	70%	69%	67%	68%
SVM (M_{16})	73%	**91%**	75%	78%
NN (M_{17})	72%	86%	76%	77%

Table B. 2. Site 1 Models Evaluation Using F-Measure

Models	Local F-measure (D1)	On Site2 (D2)	On Site3 (D3)	Average F-measure
LR (M_{11})	**85%**	**94%**	**87%**	**88%**
RF (M_{12})	79%	**95%**	84%	85%
NB (M_{13})	83%	**94%**	**87%**	87%
KNN (M_{14})	80%	92%	85%	85%
DT (M_{15})	79%	80%	78%	79%
SVM (M_{16})	84%	**95%**	86%	87%
NN (M_{17})	65%	92%	86%	79%

Table B. 3. Site 2 Models Evaluation Using Accuracy Metric

Models	Local accuracy (D2)	On Site1 (D1)	On Site3 (D3)	Average accuracy
LR (M_{21})	**88%**	65%	74%	74%
RF (M_{22})	86%	53%	65%	65%
NB (M_{23})	85%	67%	75%	74%
KNN (M_{24})	85%	72%	**76%**	77%

DT (M_{25})	81%	55%	62%	64%
SVM (M_{26})	85%	**77%**	75%	**78%**
NN (M_{27})	81%	73%	76%	76%

Table B. 4. Site 2 Models Evaluation Using F-Measure

Models	Local F-measure (D2)	On Site1 (D1)	On Site3 (D3)	Average F-measure
LR (M_{21})	**93%**	74%	81%	81%
RF (M_{22})	92%	58%	72%	71%
NB (M_{23})	91%	79%	84%	84%
KNN (M_{24})	92%	83%	**86%**	86%
DT (M_{25})	90%	62%	70%	71%
SVM (M_{26})	92%	**85%**	86%	**87%**
NN (M_{27})	92%	**85%**	86%	87%

Table B. 5. Site 3 Models Evaluation Using Accuracy Metric

Models	Local accuracy (D3)	On Site1 (D1)	On Site2 (D2)	Average accuracy
LR (M_{31})	77%	**74%**	**89%**	**79%**
RF (M_{32})	**79%**	**74%**	83%	78%
NB (M_{33})	77%	70%	**89%**	77%
KNN (M_{34})	74%	71%	87%	76%
DT (M_{35})	77%	73%	85%	77%
SVM (M_{36})	77%	72%	85%	77%
NN (M_{37})	75%	73%	86%	77%

Table B. 6. Site 3 Models Evaluation Using F-Measure

Models	Local F-measure (D3)	On Site1 (D1)	On Site2 (D2)	Average F-measure
LR (M_{31})	**87%**	84%	**94%**	**87%**
RF (M_{32})	87%	84%	91%	86%
NB (M_{33})	86%	81%	**94%**	85%
KNN (M_{34})	85%	82%	**93%**	85%
DT (M_{35})	83%	83%	92%	85%
SVM (M_{36})	86%	84%	92%	86%
NN (M_{37})	86%	**85%**	92%	87%

2. Breast Cancer Wisconsin (Diagnostic):

Table B. 7. Site 1 models evaluation using accuracy metric

Models	Local accuracy (D1)	On Site2 (D2) (Best local model = 97%)	On Site3 (D3) (Best local model = 97%)	Average accuracy
LR (M_{11})	**94%**	**97%**	94%	**95%**
RF (M_{12})	94%	95%	**96%**	95%
NB (M_{13})	93%	95%	95%	94%
KNN (M_{14})	91%	95%	95%	93%
DT (M_{15})	90%	87%	79%	86%
SVM (M_{16})	52%	76%	65%	63%
NN (M_{17})	57%	76%	65%	65%

Table B. 8. Site 1 models evaluation using F-measure

Models	Local F-measure (D1)	On Site2 (D2) (Best local model = 92%)	On Site3 (D3) (Best local model = 96%)	Average F-measure
LR (M_{11})	**93%**	**95%**	92%	**93%**
RF (M_{12})	93%	90%	**95%**	92%
NB (M_{13})	93%	91%	94%	92%
KNN (M_{14})	90%	89%	92%	90%
DT (M_{15})	87%	79%	74%	80%
SVM (M_{16})	36%	66%	51%	49%
NN (M_{17})	39%	66%	51%	51%

Table B. 9. Site 2 models evaluation using accuracy metric

Models	Local accuracy (D2)	On Site1 (D1) (Best local model = 94%)	On Site3 (D3) (Best local model = 97%)	Average accuracy
LR (M_{21})	95%	90%	91%	92%
RF (M_{22})	95%	89%	**97%**	93%
NB (M_{23})	**97%**	**91%**	95%	**94%**
KNN (M_{24})	95%	85%	92%	90%
DT (M_{25})	96%	86%	87%	89%
SVM (M_{26})	76%	52%	65%	63%
NN (M_{27})	63%	52%	65%	59%

Table B. 10. Site 2 models evaluation using F-measure

Models	Local F-measure (D2)	On Site1 (D1) (Best local model = 93%)	On Site3 (D3) (Best local model = 96%)	Average F-measure
LR (M_{21})	89%	89%	87%	88%
RF (M_{22})	**92%**	87%	**95%**	90%
NB (M_{23})	90%	**90%**	94%	**91%**
KNN (M_{24})	83%	82%	88%	84%
DT (M_{25})	90%	84%	82%	85%
SVM (M_{26})	66%	36%	51%	52%
NN (M_{27})	57%	36%	51%	49%

Table B. 11. Site 3 models evaluation using accuracy metric

Models	Local accuracy (D3)	On Site1 (D1) (Best local model = 94%)	On Site2 (D2) (Best local model = 97%)	Average accuracy
LR (M_{31})	96%	**94%**	94%	**95%**
RF (M_{32})	**97%**	91%	96%	94%
NB (M_{33})	97%	90%	**97%**	94%
KNN (M_{34})	92%	89%	95%	92%
DT (M_{35})	95%	88%	95%	92%
SVM (M_{36})	65%	52%	76%	63%
NN (M_{37})	55%	52%	76%	60%

Table B. 12. Site 3 models evaluation using F-measure

Models	Local F-measure (D3)	On Site1 (D1) (Best local model = 93%)	On Site2 (D2) (Best local model = 92%)	Average F-measure
LR (M_{31})	92%	**93%**	87%	91%
RF (M_{32})	95%	90%	91%	92%
NB (M_{33})	**96%**	89%	**93%**	**93%**
KNN (M_{34})	89%	87%	90%	89%
DT (M_{35})	94%	87%	88%	90%
SVM (M_{36})	51%	36%	66%	51%
NN (M_{37})	37%	36%	66%	45%

3. Diabetes:

Table B. 13. Site 1 models evaluation using accuracy metric

Models	Local accuracy (D1)	On Site2 (D2) (Best local model = 73%)	On Site3 (D3) (Best local model = 81%)	Average accuracy
LR (M_{11})	**72%**	**75%**	**82%**	**76%**
RF (M_{12})	69%	**75%**	**82%**	75%
NB (M_{13})	72%	**75%**	78%	75%
KNN (M_{14})	62%	**75%**	78%	71%
DT (M_{15})	61%	67%	71%	66%
SVM (M_{16})	62%	61%	73%	65%
NN (M_{17})	62%	61%	73%	65%

Table B. 14. Site 1 models evaluation using F-measure

Models	Local F-measure (D1)	On Site2 (D2) (Best local model = 66%)	On Site3 (D3) (Best local model = 57%)	Average F-measure
LR (M_{11})	57%	62%	**61%**	59%
RF (M_{12})	55%	63%	**64%**	60%
NB (M_{13})	**62%**	65%	**62%**	**63%**
KNN (M_{14})	49%	**67%**	**61%**	57%
DT (M_{15})	55%	58%	52%	55%
SVM (M_{16})	47%	47%	53%	49%
NN (M_{17})	47%	47%	53%	49%

Table B. 15. Site 2 models evaluation using accuracy metric

Models	Local accuracy (D2)	On Site1 (D1) (Best local model = 72%)	On Site3 (D3) (Best local model = 81%)	Average accuracy
LR (M_{21})	72%	70%	78%	73%
RF (M_{22})	**73%**	**71%**	**79%**	**74%**
NB (M_{23})	71%	71%	78%	73%
KNN (M_{24})	70%	68%	71%	69%
DT (M_{25})	66%	67%	73%	69%
SVM (M_{26})	61%	62%	73%	65%
NN (M_{27})	60%	62%	72%	65%

Table B. 16. Site 2 models evaluation using F-measure

Models	Local F-measure (D2)	On Site1 (D1) (Best local model = 62%)	On Site3 (D3) (Best local model = 57%)	Average F-measure
LR (M_{21})	57%	58%	**60%**	58%
RF (M_{22})	60%	61%	**61%**	61%
NB (M_{23})	63%	**63%**	**65%**	**64%**
KNN (M_{24})	**66%**	59%	54%	59%
DT (M_{25})	55%	58%	**57%**	57%
SVM (M_{26})	47%	47%	52%	49%
NN (M_{27})	49%	46%	53%	49%

Table B. 17. Site 3 models evaluation using accuracy metric

Models	Local accuracy (D3)	On Site1 (D1) (Best local model = 72%)	On Site2 (D2) (Best local model = 73%)	Average accuracy
LR (M_{31})	78%	**72%**	71%	74%
RF (M_{32})	78%	**74%**	**75%**	**76%**
NB (M_{33})	**81%**	**72%**	**73%**	75%
KNN (M_{34})	80%	69%	66%	72%
DT (M_{35})	74%	68%	72%	71%
SVM (M_{36})	73%	62%	61%	65%
NN (M_{37})	70%	62%	62%	65%

Table B. 18. Site 3 models evaluation using F-measure

Models	Local F-measure (D3)	On Site1 (D1) (Best local model = 62%)	On Site2 (D2) (Best local model = 66%)	Average F-measure
LR (M_{31})	49%	56%	48%	52%
RF (M_{32})	50%	**61%**	**60%**	**57%**
NB (M_{33})	**57%**	57%	58%	57%
KNN (M_{34})	52%	50%	43%	49%
DT (M_{35})	51%	59%	60%	57%
SVM (M_{36})	48%	47%	47%	47%
NN (M_{37})	43%	47%	47%	46%

4. Heart disease:

Table B. 19. Site 1 models evaluation using accuracy metric

Models	Local accuracy (D1)	On Site2 (D2) (Best local model = 76%)	On Site3 (D3) (Best local model = 76%)	Average accuracy
LR (M_{11})	82%	75%	**85%**	**80%**
RF (M_{12})	**83%**	**76%**	**80%**	80%
NB (M_{13})	83%	74%	**83%**	80%
KNN (M_{14})	66%	56%	72%	64%
DT (M_{15})	76%	**77%**	**80%**	77%
SVM (M_{16})	56%	45%	37%	47%
NN (M_{17})	58%	45%	37%	48%

Table B. 20. Site 1 models evaluation using F-measure

Models	Local F-measure (D1)	On Site2 (D2) (Best local model = 75%)	On Site3 (D3) (Best local model = 84%)	Average F-measure
LR (M_{11})	**83%**	**75%**	**88%**	**81%**
RF (M_{12})	83%	**78%**	83%	81%
NB (M_{13})	80%	**75%**	**86%**	80%
KNN (M_{14})	61%	69%	81%	69%
DT (M_{15})	76%	**80%**	82%	79%
SVM (M_{16})	41%	28%	20%	31%
NN (M_{17})	37%	28%	29%	32%

Table B. 21. Site 2 models evaluation using accuracy metric

Models	Local accuracy (D2)	On Site1 (D1) (Best local model = 83%)	On Site3 (D3) (Best local model = 76%)	Average accuracy
LR (M_{21})	**76%**	**84%**	**83%**	**81%**
RF (M_{22})	73%	76%	**82%**	76%
NB (M_{23})	74%	78%	**80%**	77%
KNN (M_{24})	65%	73%	48%	64%
DT (M_{25})	68%	52%	75%	63%
SVM (M_{26})	58%	46%	63%	54%
NN (M_{27})	57%	54%	37%	51%

Table B. 22. Site 2 models evaluation using F-measure

Models	Local F-measure (D2)	On Site1 (D1) (Best local model = 83%)	On Site3 (D3) (Best local model = 84%)	Average F-measure
LR (M_{21})	74%	**84%**	**87%**	**81%**
RF (M_{22})	**75%**	74%	**85%**	77%
NB (M_{23})	73%	76%	**84%**	77%
KNN (M_{24})	72%	70%	54%	67%
DT (M_{25})	65%	63%	81%	68%
SVM (M_{26})	72%	63%	78%	70%
NN (M_{27})	37%	66%	51%	52%

Table B. 23. Site 3 models evaluation using accuracy metric

Models	Local accuracy (D3)	On Site1 (D1) (Best local model = 83%)	On Site2 (D2) (Best local model = 76%)	Average accuracy
LR (M_{31})	73%	68%	70%	70%
RF (M_{32})	75%	**82%**	75%	**78%**
NB (M_{33})	**76%**	62%	72%	69%
KNN (M_{34})	73%	64%	58%	64%
DT (M_{35})	68%	79%	**76%**	75%
SVM (M_{36})	63%	46%	55%	53%
NN (M_{37})	41%	43%	44%	43%

Table B. 24. Site 3 models evaluation using F-measure

Models	Local F-measure (D3)	On Site1 (D1) (Best local model = 83%)	On Site2 (D2) (Best local model = 75%)	Average F-measure
LR (M_{31})	**84%**	74%	**76%**	77%
RF (M_{32})	80%	**82%**	**79%**	**80%**
NB (M_{33})	82%	32%	71%	58%
KNN (M_{34})	76%	68%	69%	70%
DT (M_{35})	77%	77%	**80%**	78%
SVM (M_{36})	78%	63%	71%	69%
NN (M_{37})	31%	47%	54%	46%

5. Liver Disease:

Table B. 25. Site 1 models evaluation using accuracy metric

Models	Local accuracy (D1)	On Site2 (D2) (Best local model = 69%)	On Site3 (D3) (Best local model = 66%)	Average accuracy
LR (M_{11})	72%	66%	**66%**	69%
RF (M_{12})	75%	**67%**	60%	70%
NB (M_{13})	50%	45%	58%	50%
KNN (M_{14})	68%	66%	58%	65%
DT (M_{15})	69%	56%	59%	63%
SVM (M_{16})	**76%**	67%	65%	**71%**
NN (M_{17})	44%	36%	44%	42%

Table B. 26. Site 1 models evaluation using F-measure

Models	Local F-measure (D1)	On Site2 (D2) (Best local model = 80%)	On Site3 (D3) (Best local model = 73%)	Average F-measure
LR (M_{11})	83%	**80%**	**79%**	81%
RF (M_{12})	84%	78%	**74%**	80%
NB (M_{13})	51%	32%	53%	46%
KNN (M_{14})	81%	76%	70%	77%
DT (M_{15})	78%	65%	69%	72%
SVM (M_{16})	**87%**	**80%**	**79%**	**83%**
NN (M_{17})	20%	10%	26%	18%

Table B. 27. Site 2 models evaluation using accuracy metric

Models	Local accuracy (D2)	On Site1 (D1) (Best local model = 76%)	On Site3 (D3) (Best local model = 66%)	Average accuracy
LR (M_{21})	**69%**	71%	**68%**	70%
RF (M_{22})	67%	74%	63%	70%
NB (M_{23})	51%	63%	64%	60%
KNN (M_{24})	65%	73%	60%	68%
DT (M_{25})	60%	68%	60%	64%
SVM (M_{26})	67%	**75%**	65%	**71%**
NN (M_{27})	35%	33%	40%	35%

Table B. 28. Site 2 models evaluation using F-measure

Models	Local F-measure (D2)	On Site1 (D1) (Best local model = 87%)	On Site3 (D3) (Best local model = 73%)	Average F-measure
LR (M_{21})	77%	82%	**79%**	80%
RF (M_{22})	73%	84%	**74%**	79%
NB (M_{23})	49%	70%	66%	63%
KNN (M_{24})	76%	83%	**73%**	79%
DT (M_{25})	66%	78%	69%	73%
SVM (M_{26})	**80%**	**85%**	**79%**	**82%**
NN (M_{27})	25%	21%	16%	21%

Table B. 29. Site 3 models evaluation using accuracy metric

Models	Local accuracy (D3)	On Site1 (D1) (Best local model = 76%)	On Site2 (D2) (Best local model = 69%)	Average accuracy
LR (M_{31})	**66%**	71%	64%	68%
RF (M_{32})	64%	66%	61%	64%
NB (M_{33})	65%	62%	53%	60%
KNN (M_{34})	53%	64%	58%	60%
DT (M_{35})	59%	69%	68%	67%
SVM (M_{36})	65%	**75%**	**67%**	**71%**
NN (M_{37})	35%	25%	33%	29%

Table B. 30. Site 3 models evaluation using F-measure

Models	Local F-measure (D3)	On Site1 (D1) (Best local model = 87%)	On Site2 (D2) (Best local model = 80%)	Average F-measure
LR (M_{31})	**73%**	81%	69%	76%
RF (M_{32})	68%	78%	71%	74%
NB (M_{33})	64%	68%	49%	62%
KNN (M_{34})	65%	75%	67%	71%
DT (M_{35})	65%	80%	76%	76%
SVM (M_{36})	79%	**86%**	**80%**	**83%**
NN (M_{37})	25%	18%	16%	19%

6. Spine Disease:

Table B. 31. Site 1 models evaluation using accuracy metric

Models	Local accuracy (D1)	On Site2 (D2) (Best local model = 93%)	On Site3 (D3) (Best local model = 79%)	Average accuracy
LR (M_{11})	94%	88%	73%	84%
RF (M_{12})	94%	92%	76%	86%
NB (M_{13})	92%	90%	61%	79%
KNN (M_{14})	**95%**	**94%**	**78%**	**88%**
DT (M_{15})	91%	84%	71%	81%
SVM (M_{16})	85%	66%	53%	68%
NN (M_{17})	57%	82%	65%	65%

Table B. 32. Site 1 models evaluation using F-measure

Models	Local F-measure (D1)	On Site2 (D2) (Best local model = 93%)	On Site3 (D3) (Best local model = 82%)	Average F-measure
LR (M_{11})	97%	90%	65%	82%
RF (M_{12})	**98%**	**94%**	72%	86%
NB (M_{13})	95%	92%	73%	85%
KNN (M_{14})	97%	**95%**	**75%**	**87%**
DT (M_{15})	95%	88%	71%	84%
SVM (M_{16})	92%	80%	69%	80%
NN (M_{17})	75%	82%	61 %	70%

Table B. 33. Site 2 models evaluation using accuracy metric

Models	Local accuracy (D2)	On Site1 (D1) (Best local model = 95%)	On Site3 (D3) (Best local model = 79%)	Average accuracy
LR (M_{21})	86%	89%	72%	81%
RF (M_{22})	**93%**	90%	76%	**85%**
NB (M_{23})	92%	91%	75%	84%
KNN (M_{24})	86%	**92%**	**77%**	85%
DT (M_{25})	89%	84%	77%	82%
SVM (M_{26})	66%	85%	53%	68%
NN (M_{27})	66%	86%	68%	75%

Table B. 34. Site 2 models evaluation using F-measure

Models	Local F-measure (D2)	On Site1 (D1) (Best local model = 98%)	On Site3 (D3) (Best local model = 82%)	Average F-measure
LR (M_{21})	85%	94%	64%	80%
RF (M_{22})	92%	94%	**73%**	**85%**
NB (M_{23})	**93%**	**95%**	70%	84%
KNN (M_{24})	90%	95%	73%	85%
DT (M_{25})	91%	91%	73%	83%
SVM (M_{26})	80%	92%	36%	66%
NN (M_{27})	40%	83%	65%	67%

Table B. 35. Site 3 models evaluation using accuracy metric

Models	Local accuracy (D3)	On Site1 (D1) (Best local model = 95%)	On Site2 (D2) (Best local model = 93%)	Average accuracy
LR (M_{31})	75%	88%	90%	83%
RF (M_{32})	**79%**	90%	88%	84%
NB (M_{33})	79%	90%	90%	**85%**
KNN (M_{34})	76%	92%	**92%**	85%
DT (M_{35})	69%	**93%**	88%	82%
SVM (M_{36})	53%	85%	66%	68%
NN (M_{37})	55%	91%	88%	75%

Table B. 36. Site 3 models evaluation using F-measure

Models	Local F-measure (D3)	On Site1 (D1) (Best local model = 98%)	On Site2 (D2) (Best local model = 93%)	Average F-measure
LR (M_{31})	77%	93%	**93%**	86%
RF (M_{32})	**82%**	94%	92%	**89%**
NB (M_{33})	76%	94%	**93%**	86%
KNN (M_{34})	74%	95%	**94%**	86%
DT (M_{35})	74%	**96%**	91%	86%
SVM (M_{36})	69%	92%	80%	80%
NN (M_{37})	45%	95%	91%	73%

7. Breast Cancer Wisconsin (Original):

Table B. 37. Site 1 models evaluation using accuracy metric

Models	Local accuracy (D1)	On Site2 (D2) (Best local model =95%)	On Site3 (D3) (Best local model =99%)	Average accuracy
LR (M_{11})	95%	**95%**	**99%**	96%
RF (M_{12})	**98%**	**95%**	**99%**	**97%**
NB (M_{13})	96%	**95%**	98%	96%
KNN (M_{14})	97%	**95%**	98%	96%
DT (M_{15})	89%	92%	95%	92%
SVM (M_{16})	95%	93%	96%	94%
NN (M_{17})	93%	**95%**	98%	95%

Table B. 38. Site 1 models evaluation using F-measure

Models	Local F-measure (D1)	On Site2 (D2) (Best local model =96%)	On Site3 (D3) (Best local model =99%)	Average F-measure
LR (M_{11})	95%	**95%**	**99%**	96%
RF (M_{12})	**98%**	95%	**99%**	**97%**
NB (M_{13})	96%	95%	98%	96%
KNN (M_{14})	97%	95%	98%	96%
DT (M_{15})	91%	92%	95%	92%
SVM (M_{16})	95%	93%	96%	94%
NN (M_{17})	94%	95%	98%	95%

Table B. 39. Site 2 models evaluation using accuracy metric

Models	Local accuracy (D2)	On Site1 (D1) (Best local model = 98%)	On Site3 (D3) (Best local model =99%)	Average accuracy
LR (M_{21})	94%	92%	**99%**	94%
RF (M_{22})	**95%**	94%	**99%**	95%
NB (M_{23})	95%	**95%**	**99%**	**96%**
KNN (M_{24})	95%	92%	**99%**	95%
DT (M_{25})	93%	89%	**99%**	93%
SVM (M_{26})	95%	91%	98%	94%
NN (M_{27})	93%	89%	98%	93%

Table B. 40. Site 2 models evaluation using F-measure

Models	Local F-measure (D2)	On Site1 (D1) (Best local model =98%)	On Site3 (D3) (Best local model =99%)	Average F-measure
LR (M_{21})	**96%**	92%	**99%**	95%
RF (M_{22})	95%	94%	**99%**	95%
NB (M_{23})	95%	**95%**	**99%**	**96%**
KNN (M_{24})	95%	92%	**99%**	95%
DT (M_{25})	94%	89%	**99%**	93%
SVM (M_{26})	95%	91%	98%	94%
NN (M_{27})	95%	89%	98%	94%

Table B. 41. Site 3 models evaluation using accuracy metric

Models	Local accuracy (D3)	On Site1 (D1) (Best local model =98%)	On Site2 (D2) (Best local model =95%)	Average accuracy
LR (M_{31})	96%	83%	91%	89%
RF (M_{32})	98%	92%	94%	**94%**
NB (M_{33})	96%	**94%**	93%	94%
KNN (M_{34})	**99%**	88%	**95%**	94%
DT (M_{35})	96%	88%	91%	91%
SVM (M_{36})	96%	93%	93%	94%
NN (M_{37})	95%	83%	91%	89%

Table B. 42. Site 3 models evaluation using F-measure

Models	Local F-measure (D3)	On Site1 (D1) (Best local model =98%)	On Site2 (D2) (Best local model =96%)	Average F-measure
LR (M_{31})	96%	83%	91%	90%
RF (M_{32})	98%	92%	94%	**94%**
NB (M_{33})	97%	**94%**	93%	94%
KNN (M_{34})	**99%**	88%	**95%**	94%
DT (M_{35})	96%	88%	91%	91%
SVM (M_{36})	96%	93%	93%	94%
NN (M_{37})	93%	83%	91%	89%

8. Cardiovascular diseases

Table B. 43. Site 1 models evaluation using accuracy metric

Models	Local accuracy (D1)	On Site2 (D2) (Best local model =73%)	On Site3 (D3) (Best local model =73%)	Average accuracy
LR (M_{11})	**72%**	**73%**	**73%**	**73%**
RF (M_{12})	71%	71%	71%	71%
NB (M_{13})	71%	71%	71%	71%
KNN (M_{14})	69%	69%	68%	69%
DT (M_{15})	63%	63%	64%	63%
SVM (M_{16})	67%	67%	68%	67%
NN (M_{17})	53%	50%	50%	51%

Table B. 44. Site 1 models evaluation using F-measure

Models	Local F-measure (D1)	On Site2 (D2) (Best local model =73%)	On Site3 (D3) (Best local model =73%)	Average F-measure
LR (M_{11})	**72%**	**73%**	**72%**	**72%**
RF (M_{12})	70%	71%	71%	71%
NB (M_{13})	71%	71%	70%	71%
KNN (M_{14})	68%	69%	68%	68%
DT (M_{15})	63%	63%	64%	63%
SVM (M_{16})	67%	67%	68%	67%
NN (M_{17})	39%	33%	33%	35%

Table B. 45. Site 2 models evaluation using accuracy metric

Models	Local accuracy (D2)	On Site1 (D1) (Best local model =72%)	On Site3 (D3) (Best local model =73%)	Average accuracy
LR (M_{21})	**73%**	**72%**	**73%**	**73%**
RF (M_{22})	71%	71%	70%	71%
NB (M_{23})	71%	71%	71%	71%
KNN (M_{24})	69%	68%	68%	68%
DT (M_{25})	64%	64%	63%	64%
SVM (M_{26})	68%	68%	68%	68%
NN (M_{27})	53%	51%	50%	52%

Table B. 46. Site 2 models evaluation using F-measure

Models	Local F-measure (D2)	On Site1 (D1) (Best local model =72%)	On Site3 (D3) (Best local model =73%)	Average F-measure
LR (M_{21})	**73%**	**72%**	**73%**	**73%**
RF (M_{22})	71%	71%	70%	71%
NB (M_{23})	71%	71%	71%	71%
KNN (M_{24})	68%	68%	68%	68%
DT (M_{25})	63%	64%	63%	63%
SVM (M_{26})	68%	68%	68%	68%
NN (M_{27})	53%	34%	33%	42%

Table B. 47. Site 3 models evaluation using accuracy metric

Models	Local accuracy (D3)	On Site1 (D1) (Best local model =72%)	On Site2 (D2) (Best local model =73%)	Average accuracy
LR (M_{31})	**73%**	72%	73%	**73%**
RF (M_{32})	71%	71%	71%	71%
NB (M_{33})	71%	71%	71%	71%
KNN (M_{34})	69%	68%	69%	69%
DT (M_{35})	63%	62%	63%	63%
SVM (M_{36})	67%	67%	67%	67%
NN (M_{37})	59%	51%	50%	53%

Table B. 48. Site 3 models evaluation using F-measure

Models	Local F-measure (D3)	On Site1 (D1) (Best local model =72%)	On Site2 (D2) (Best local model =73%)	Average F-measure
LR (M_{31})	**73%**	**72%**	**73%**	**73%**
RF (M_{32})	71%	71%	71%	71%
NB (M_{33})	71%	71%	70%	70%
KNN (M_{34})	68%	68%	69%	68%
DT (M_{35})	64%	62%	63%	63%
SVM (M_{36})	67%	67%	67%	67%
NN (M_{37})	46%	34%	33%	37%

II. Non-randomly Partitioned Dataset:

1. Diabetes Dataset

Table B. 49. Site 1 Models Evaluation Using Accuracy Metric

Models	Local accuracy (D1)	On Site2 (D2)	On Site3 (D3)	Average accuracy
LR (M_{11})	84%	66%	62%	76%
RF (M_{12})	84%	**69%**	62%	76%
NB (M_{13})	**85%**	**68%**	**68%**	**77%**
KNN (M_{14})	83%	63%	56%	72%
DT (M_{15})	77%	63%	59%	70%
SVM (M_{16})	80%	51%	45%	66%
NN (M_{17})	79%	51%	45%	65%

Table B. 50. Site 1 Models Evaluation Using F-Measure

Models	Local F-measure (D1)	On Site2 (D2)	On Site3 (D3)	Average F-measure
LR (M_{11})	50%	55%	55%	52%
RF (M_{12})	49%	**65%**	65%	56%
NB(M_{13})	**55%**	**69%**	**73%**	**62%**
KNN (M_{14})	42%	49%	50%	45%
DT (M_{15})	48%	**65%**	67%	56%
SVM (M_{16})	52%	35%	28%	32%
NN (M_{17})	51%	35%	28%	43%

Table B. 51. Site 2 Models Evaluation Using Accuracy Metric

Models	Local accuracy (D2)	On Site1 (D1)	On Site3 (D3)	Average accuracy
LR (M_{21})	66%	**82%**	**71%**	**75%**
RF (M_{22})	64%	78%	70%	72%
NB (M_{23})	**68%**	81%	68%	75%
KNN (M_{24})	64%	71%	57%	67%
DT (M_{25})	62%	71%	64%	67%
SVM (M_{26})	50%	80%	45%	65%
NN (M_{27})	52%	80%	45%	66%

Table B. 52. Site 2 Models Evaluation Using F-Measure

Models	Local F-measure (D2)	On Site1 (D1)	On Site3 (D3)	Average F-measure
LR (M_{21})	62%	**60%**	**73%**	**62%**
RF (M_{22})	61%	**55%**	71%	59%
NB (M_{23})	**64%**	**57%**	68%	61%
KNN (M_{24})	62%	42%	61%	51%
DT (M_{25})	63%	43%	63%	52%
SVM (M_{26})	34%	53%	30%	44%
NN (M_{27})	40%	53%	28%	45%

Table B. 53. Site 3 Models Evaluation Using Accuracy Metric

Models	Local accuracy (D3)	On Site1 (D1)	On Site2 (D2)	Average accuracy
LR (M_{31})	61%	48%	62%	54%
RF (M_{32})	67%	70%	**64%**	68%
NB (M_{33})	65%	73%	63%	**69%**
KNN (M_{34})	**75%**	63%	62%	64%
DT (M_{35})	61%	60%	64%	61%
SVM (M_{36})	55%	20%	49%	34%
NN (M_{37})	51%	**80%**	51%	67%

Table B. 54. Site 3 Models Evaluation Using F-Measure

Models	Local F-measure (D3)	On Site1 (D1)	On Site2 (D2)	Average F-measure
LR (M_{31})	71%	40%	**65%**	52%
RF (M_{32})	73%	**49%**	**64%**	**57%**
NB (M_{33})	64%	42%	58%	50%
KNN (M_{34})	**79%**	43%	**64%**	55%
DT (M_{35})	64%	36%	61%	48%
SVM (M_{36})	71%	33%	**66%**	49%
NN (M_{37})	34%	48%	56%	44%

2. Heart disease:

Table B. 55. Site 1 models evaluation using accuracy metric

Models	Local accuracy (D1)	On Site2 (D2) (Best local model = 77%)	On Site3 (D3) (Best local model = 81%)	Average accuracy
LR (M_{11})	**89%**	**79%**	75%	**79%**
RF (M_{12})	89%	74%	72%	76%
NB (M_{13})	82%	76%	**76%**	77%
KNN (M_{14})	77%	64%	49%	62%
DT (M_{15})	79%	**77%**	65%	74%
SVM (M_{16})	77%	51%	44%	54%
NN (M_{17})	36%	49%	56%	49%

Table B. 56. Site 1 models evaluation using F-measure

Models	Local F-measure (D1)	On Site2 (D2) (Best local model = 78%)	On Site3 (D3) (Best local model = 79%)	Average F-measure
LR (M_{11})	**92%**	**81%**	69%	**80%**
RF (M_{12})	89%	**79%**	**74%**	79%
NB (M_{13})	86%	**78%**	74%	78%
KNN (M_{14})	85%	72%	56%	70%
DT (M_{15})	88%	**81%**	68%	78%
SVM (M_{16})	87%	67%	61%	69%
NN (M_{17})	31%	32%	40%	34%

Table B. 57. Site 2 models evaluation using accuracy metric

Models	Local accuracy (D2)	On Site1 (D1) (Best local model = 89%)	On Site3 (D3) (Best local model = 81%)	Average accuracy
LR (M_{21})	**77%**	**91%**	**84%**	**82%**
RF (M_{22})	73%	86%	76%	76%
NB (M_{23})	76%	**89%**	**84%**	81%
KNN (M_{24})	53%	66%	49%	54%
DT (M_{25})	73%	86%	60%	72%
SVM (M_{26})	51%	73%	46%	54%
NN (M_{27})	64%	86%	74%	71%

Table B. 58. Site 2 models evaluation using F-measure

Models	Local F-measure (D2)	On Site1 (D1) (Best local model = 92%)	On Site3 (D3) (Best local model = 79%)	Average F-measure
LR (M_{21})	**78%**	**94%**	**80%**	**82%**
RF (M_{22})	78%	91%	71%	78%
NB (M_{23})	74%	**93%**	**82%**	80%
KNN (M_{24})	62%	77%	44%	59%
DT (M_{25})	71%	91%	53%	69%
SVM (M_{26})	65%	84%	61%	67%
NN (M_{27})	42%	91%	67%	37%

Table B. 59. Site 3 models evaluation using accuracy metric

Models	Local accuracy (D3)	On Site1 (D1) (Best local model = 89%)	On Site2 (D2) (Best local model = 77%)	Average accuracy
LR (M_{31})	75%	**86%**	**78%**	79%
RF (M_{32})	71%	82%	**79%**	77%
NB (M_{33})	**81%**	86%	**78%**	**80%**
KNN (M_{34})	54%	59%	47%	51%
DT (M_{35})	58%	61%	76%	68%
SVM (M_{36})	56%	23%	49%	46%
NN (M_{37})	60%	23%	49%	47%

Table B. 60. Site 3 models evaluation using F-measure

Models	Local F-measure (D3)	On Site1 (D1) (Best local model = 92%)	On Site2 (D2) (Best local model = 78%)	Average F-measure
LR (M_{31})	75%	**91%**	**79%**	80%
RF (M_{32})	71%	87%	**80%**	79%
NB (M_{33})	**79%**	91%	**79%**	**81%**
KNN (M_{34})	49%	68%	39%	47%
DT (M_{35})	50%	71%	75%	67%
SVM (M_{36})	40%	8%	32%	30%
NN (M_{37})	46%	8%	32%	32%

3. Liver Disease:

Table B. 61. Site 1 models evaluation using accuracy metric

Models	Local accuracy (D1)	On Site2 (D2) (Best local model = 77%)	On Site3 (D3) (Best local model = 74%)	Average accuracy
LR (M_{11})	62%	63%	62%	62%
RF (M_{12})	**69%**	64%	66%	65%
NB (M_{13})	59%	58%	61%	58%
KNN (M_{14})	58%	60%	62%	60%
DT (M_{15})	60%	60%	62%	60%
SVM (M_{16})	54%	**76%**	**72%**	**71%**
NN (M_{17})	48%	24%	26%	28%

Table B. 62. Site 1 models evaluation using F-measure

Models	Local F-measure (D1)	On Site2 (D2) (Best local model = 87%)	On Site3 (D3) (Best local model = 85%)	Average F-measure
LR (M_{11})	60%	71%	69%	68%
RF (M_{12})	66%	72%	72%	71%
NB (M_{13})	39%	64%	65%	59%
KNN (M_{14})	54%	67%	68%	65%
DT (M_{15})	57%	69%	70%	67%
SVM (M_{16})	**69%**	**86%**	**84%**	**82%**
NN (M_{17})	43%	12%	11%	18%

Table B. 63. Site 2 models evaluation using accuracy metric

Models	Local accuracy (D2)	On Site1 (D1) (Best local model = 69%)	On Site3 (D3) (Best local model = 74%)	Average accuracy
LR (M_{21})	75%	54%	**74%**	71%
RF (M_{22})	71%	**64%**	69%	69%
NB (M_{23})	51%	61%	49%	52%
KNN (M_{24})	71%	52%	66%	66%
DT (M_{25})	60%	55%	63%	59%
SVM (M_{26})	**77%**	54%	**74%**	**72%**
NN (M_{27})	33%	50%	33%	36%

Table B. 64. Site 2 models evaluation using F-measure

Models	Local F-measure (D2)	On Site1 (D1) (Best local model = 69%)	On Site3 (D3) (Best local model = 85%)	Average F-measure
LR (M_{21})	86%	**70%**	**85%**	**83%**
RF (M_{22})	84%	**74%**	80%	81%
NB (M_{23})	55%	45%	48%	51%
KNN (M_{24})	82%	65%	78%	78%
DT (M_{25})	76%	62%	73%	73%
SVM (M_{26})	**87%**	**70%**	**85%**	83%
NN (M_{27})	18%	15%	19%	18%

Table B. 65. Site 3 models evaluation using accuracy metric

Models	Local accuracy (D3)	On Site1 (D1) (Best local model = 69%)	On Site2 (D2) (Best local model = 77%)	Average accuracy
LR (M_{31})	**74%**	55%	72%	69%
RF (M_{32})	68%	59%	73%	69%
NB (M_{33})	62%	**66%**	60%	61%
KNN (M_{34})	67%	62%	69%	67%
DT (M_{35})	70%	61%	66%	66%
SVM (M_{36})	73%	54%	76%	**71%**
NN (M_{37})	42%	45%	**49%**	47%

Table B. 66. Site 3 models evaluation using F-measure

Models	Local F-measure (D3)	On Site1 (D1) (Best local model = 69%)	On Site2 (D2) (Best local model = 87%)	Average F-measure
LR (M_{31})	79%	65%	83%	79%
RF (M_{32})	80%	68%	83%	79%
NB (M_{33})	66%	59%	67%	65%
KNN (M_{34})	80%	**71%**	81%	79%
DT (M_{35})	73%	65%	77%	74%
SVM (M_{36})	**85%**	**70%**	**86%**	**83%**
NN (M_{37})	55%	46%	60%	56%

b) Regression:

I. Randomly Partitioned Dataset:

1. Parkinson Disease (Total UPDRS):

Table B. 67. Site 1 Models Evaluation Using MAPE Metric

Models	Local MAPE (D1)	On Site2 (D2)	On Site3 (D3)	Average MAPE
LR (M_{11})	28%	59%	49%	45%
RFR (M_{12})	**23%**	52%	40%	37%
RBFNN (M_{13})	35%	51%	37%	39%
KNNR (M_{14})	31%	53%	38%	39%
DTR (M_{15})	26%	57%	46%	42%
SVR (M_{16})	30%	**46%**	**33%**	**35%**
NNR (M_{17})	31%	58%	44%	43%
Lasso (M_{18})	35%	49%	38%	39%
Ridge (M_{19})	32%	59%	38%	41%
ElasticNet (M_{110})	35%	51%	35%	39%

Table B. 68. Site 1 Models Evaluation Using RMSE Metric

Models	Local RMSE (D1)	On Site2 (D2)	On Site3 (D3)	Average RMSE
LR (M_{11})	10.30	14.55	18.60	14.91
RFR (M_{12})	**9.04**	13.34	11.96	11.32
RBFNN (M_{13})	11.85	11.92	10.28	11.18
KNNR (M_{14})	11.48	13.59	11.31	11.90

DTR (M_{15})	11.93	15.58	14.49	13.90
SVR (M_{16})	12.03	**10.94**	**9.64**	**10.74**
NNR (M_{17})	10.81	13.25	11.89	11.86
Lasso (M_{18})	11.89	11.32	10.51	11.16
Ridge (M_{19})	11.14	13.68	10.65	11.53
ElasticNet (M_{110})	11.89	11.43	9.70	10.83

Table B. 69. Site 2 Models Evaluation Using MAPE Metric

Models	Local MAPE (D2)	On Site1 (D1)	On Site3 (D3)	Average MAPE
LR (M_{21})	23%	37%	32%	31%
RFR (M_{22})	**17%**	39%	**30%**	30%
RBFNN (M_{23})	31%	**28%**	36%	32%
KNNR (M_{24})	26%	33%	32%	31%
DTR (M_{25})	20%	43%	32%	33%
SVR (M_{26})	24%	29%	31%	**29%**
NNR (M_{27})	22%	29%	33%	29%
Lasso (M_{28})	27%	30%	32%	30%
Ridge (M_{29})	22%	33%	31%	30%
ElasticNet (M_{210})	27%	**28%**	32%	29%

Table B. 70. Site 2 Models Evaluation Using RMSE Metric

Models	Local RMSE (D2)	On Site1 (D1)	On Site3 (D3)	Average RMSE
LR (M_{21})	7.24	15.31	13.82	12.76
RFR (M_{22})	**5.42**	16.23	13.74	12.60
RBFNN (M_{23})	8.28	14.19	18.41	14.63
KNNR (M_{24})	7.54	14.57	13.39	12.40
DTR (M_{25})	7.17	17.40	14.51	13.73
SVR (M_{26})	7.69	14.55	13.42	12.44
NNR (M_{27})	6.66	14.40	13.48	**12.17**
Lasso (M_{28})	8.29	**14.05**	13.16	12.30
Ridge (M_{29})	6.86	14.64	13.34	12.24
ElasticNet (M_{210})	8.28	14.16	**13.14**	12.33

Table B. 71. Site 3 Models Evaluation Using MAPE Metric

Models	Local MAPE (D3)	On Site1 (D1)	On Site2 (D2)	Average MAPE
LR (M_{31})	25%	35%	37%	31%
RFR (M_{32})	**17%**	37%	37%	**28%**
RBFNN (M_{33})	32%	32%	41%	34%
KNNR (M_{34})	25%	35%	45%	33%
DTR (M_{35})	21%	42%	39%	32%
SVR (M_{36})	29%	33%	40%	33%
NNR (M_{37})	27%	34%	**31%**	31%
Lasso (M_{38})	31%	35%	32%	33%
Ridge (M_{39})	26%	34%	36%	31%
ElasticNet (M_{310})	31%	**31%**	38%	33%

Table B. 72. Site 3 Models Evaluation Using RMSE Metric

Models	Local RMSE (D3)	On Site1 (D1)	On Site2 (D2)	Average RMSE
LR (M_{31})	8.04	13.07	9.05	9.94
RFR (M_{32})	**6.27**	14.47	9.49	**9.73**
RBFNN (M_{33})	9.09	12.46	9.68	10.34
KNNR (M_{34})	8.09	12.99	10.91	10.37
DTR (M_{35})	8.47	16.56	11.09	11.76
SVR (M_{36})	8.55	12.54	9.39	10.06
NNR (M_{37})	8.28	13.45	**7.84**	9.88
Lasso (M_{38})	9.03	13.42	8.02	10.24
Ridge (M_{39})	8.20	12.83	8.74	9.85
ElasticNet (M_{310})	9.04	**12.38**	9.37	10.22

2. Parkinson Disease (Motor UPDRS):

Table B. 73. Site 1 models evaluation using MAPE metric

Models	Local MAPE (D1)	On Site2 (D2)	On Site3 (D3)	Average MAPE
LR (M_{11})	28%	52%	45%	41%
RFR (M_{12})	23%	47%	41%	**36%**
RBFNN (M_{13})	37%	42%	38%	39%
KNNR (M_{14})	32%	47%	41%	39%
DTR (M_{15})	28%	56%	49%	44%
SVR (M_{16})	35%	47%	38%	39%
NNR (M_{17})	32%	45%	41%	38%
Lasso (M_{18})	37%	47%	44%	42%
Ridge (M_{19})	34%	51%	42%	41%
ElasticNet (M_{110})	37%	41%	37%	38%

Table B. 74. Site 1 models evaluation using RMSE

Models	Local RMSE (D1)	On Site2 (D2)	On Site3 (D3)	Average RMSE
LR (M_{11})	6.83	9.49	10.44	9.03
RFR (M_{12})	**5.97**	9.24	8.74	7.94
RBFNN (M_{13})	8.16	7.98	7.94	8.02
KNNR (M_{14})	7.76	9.39	9.02	8.69
DTR (M_{15})	8.44	11.29	11.05	10.25
SVR (M_{16})	7.86	8.46	7.82	7.98
NNR (M_{17})	7.34	7.46	8.20	7.74
Lasso (M_{18})	8.25	8.31	8.69	8.46
Ridge (M_{19})	7.61	9.31	8.54	8.42
ElasticNet (M_{110})	8.25	7.29	7.48	**7.69**

Table B. 75. Site 2 models evaluation using MAPE metric

Models	Local MAPE (D2)	On Site1 (D1)	On Site3 (D3)	Average MAPE
LR (M_{21})	23%	44%	31%	33%
RFR (M_{22})	17%	46%	29%	32%
RBFNN (M_{23})	32%	33%	36%	34%
KNNR (M_{24})	27%	38%	32%	33%
DTR (M_{25})	21%	50%	35%	36%
SVR (M_{26})	25%	34%	31%	31%
NNR (M_{27})	23%	35%	33%	32%
Lasso (M_{28})	28%	35%	34%	33%
Ridge (M_{29})	23%	38%	30%	31%
ElasticNet (M_{210})	28%	32%	32%	32%

Table B. 76. Site 2 models evaluation using RMSE

Models	Local RMSE (D2)	On Site1 (D1)	On Site3 (D3)	Average RMSE
LR (M_{21})	5.45	10.76	8.01	8.31
RFR (M_{22})	**4.0**	11.62	7.37	7.98
RBFNN (M_{23})	8.44	8.94	14.02	11.00
KNNR (M_{24})	5.72	9.75	7.62	7.87
DTR (M_{25})	5.46	12.67	8.61	9.20
SVR (M_{26})	5.74	9.36	7.42	7.66
NNR (M_{27})	5.07	9.25	7.73	7.60
Lasso (M_{28})	6.20	9.11	7.63	7.78
Ridge (M_{29})	5.19	9.83	**7.08**	**7.54**
ElasticNet (M_{210})	6.19	**8.86**	7.53	7.68

Table B. 77. Site 3 models evaluation using MAPE metric

Models	Local MAPE (D3)	On Site1 (D1)	On Site2 (D2)	Average MAPE
LR (M_{31})	28%	39%	33%	33%
RFR (M_{32})	18%	41%	37%	31%
RBFNN (M_{33})	35%	36%	37%	36%
KNNR (M_{34})	28%	39%	41%	35%
DTR (M_{35})	21%	46%	43%	35%
SVR (M_{36})	32%	37%	37%	35%
NNR (M_{37})	31%	42%	33%	35%
Lasso (M_{38})	35%	40%	27%	35%
Ridge (M_{39})	29%	39%	32%	33%
ElasticNet (M_{310})	35%	35%	34%	35%

Table B. 78. Site 3 models evaluation using RMSE

Models	Local RMSE (D3)	On Site1 (D1)	On Site2 (D2)	Average RMSE
LR (M_{31})	6.32	9.19	6.60	7.33
RFR (M_{32})	**4.96**	10.41	7.35	7.32
RBFNN (M_{33})	7.25	8.42	6.82	7.53
KNNR (M_{34})	6.48	9.24	7.75	7.69
DTR (M_{35})	6.55	12.08	9.11	8.98
SVR (M_{36})	6.86	8.59	6.82	7.42
NNR (M_{37})	6.57	9.59	6.25	7.49
Lasso (M_{38})	7.2	9.54	**5.70**	7.62
Ridge (M_{39})	6.47	9.08	6.17	**7.26**
ElasticNet (M_{310})	7.20	**8.36**	6.58	7.44

3. Abalone:

Table B. 79. Site 1 models evaluation using MAPE metric

Models	Local MAPE (D1)	On Site2 (D2)	On Site3 (D3)	Average MAPE
LR (M_{11})	14%	15%	14%	14%
RFR (M_{12})	14%	14%	14%	14%
RBFNN (M_{13})	27%	25%	28%	26%
KNNR (M_{14})	14%	13%	14%	14%
DTR (M_{15})	17%	18%	19%	18%
SVR (M_{16})	14%	12%	13%	13%
NNR (M_{17})	14%	14%	14%	14%
Lasso (M_{18})	22%	14%	14%	17%
Ridge (M_{19})	15%	14%	14%	15%
ElasticNet (M_{110})	21%	20%	21%	21%

Table B. 80. Site 1 models evaluation using RMSE

Models	Local RMSE (D1)	On Site2 (D2)	On Site3 (D3)	Average RMSE
LR (M_{11})	2.21	2.01	2.34	2.15
RFR (M_{12})	2.21	2.03	2.53	2.19
RBFNN (M_{13})	2.21	**1.91**	2.45	2.12
KNNR (M_{14})	2.26	1.91	2.62	2.18
DTR (M_{15})	2.77	2.88	3.32	2.93
SVR (M_{16})	2.51	1.91	2.63	2.27
NNR (M_{17})	**2.14**	1.91	**2.32**	**2.07**
Lasso (M_{18})	3.10	1.95	2.35	2.43
Ridge (M_{19})	2.27	1.94	2.36	2.14
ElasticNet (M_{110})	3.06	2.57	3.45	2.92

Table B. 81. Site 2 models evaluation using MAPE metric

Models	Local MAPE (D2)	On Site1 (D1)	On Site3 (D3)	Average MAPE
LR (M_{21})	12%	13%	13%	13%
RFR (M_{22})	13%	13%	14%	13%
RBFNN (M_{23})	24%	26%	27%	26%
KNNR (M_{24})	12%	13%	14%	13%
DTR (M_{25})	15%	17%	18%	17%
SVR (M_{26})	11%	14%	13%	13%
NNR (M_{27})	12%	13%	13%	12%
Lasso (M_{28})	20%	13%	12%	15%
Ridge (M_{29})	12%	13%	13%	13%
ElasticNet (M_{210})	18%	21%	21%	20%

Table B. 82. Site 2 models evaluation using RMSE

Models	Local RMSE (D2)	On Site1 (D1)	On Site3 (D3)	Average RMSE
LR (M_{21})	1.85	2.35	**2.40**	2.19
RFR (M_{22})	1.91	2.28	2.65	2.23
RBFNN (M_{23})	1.94	2.39	2.53	2.26
KNNR (M_{24})	1.87	**2.27**	2.70	2.22
DTR (M_{25})	2.44	2.88	3.37	2.83
SVR (M_{26})	1.99	2.76	2.85	2.51
NNR (M_{27})	**1.82**	2.30	2.41	**2.16**
Lasso (M_{28})	2.62	2.31	2.41	2.44
Ridge (M_{29})	1.87	2.45	2.51	2.26
ElasticNet (M_{210})	2.51	3.06	3.47	2.95

Table B. 83. Site 3 models evaluation using MAPE metric

Models	Local MAPE (D3)	On Site1 (D1)	On Site2 (D2)	Average MAPE
LR (M_{31})	14%	14%	14%	14%
RFR (M_{32})	15%	15%	15%	15%
RBFNN (M_{33})	30%	28%	26%	28%
KNNR (M_{34})	14%	14%	14%	14%
DTR (M_{35})	19%	19%	19%	19%
SVR (M_{36})	13%	14%	12%	13%
NNR (M_{37})	13%	14%	14%	14%
Lasso (M_{38})	23%	13%	14%	17%
Ridge (M_{39})	14%	14%	14%	14%
ElasticNet (M_{310})	21%	22%	21%	21%

Table B. 84. Site 3 models evaluation using RMSE

Models	Local RMSE (D3)	On Site1 (D1)	On Site2 (D2)	Average RMSE
LR (M_{31})	**2.27**	2.29	1.96	2.21
RFR (M_{32})	2.60	2.44	2.27	2.46
RBFNN (M_{33})	2.40	2.44	2.03	2.34
KNNR (M_{34})	2.49	2.40	2.04	2.36
DTR (M_{35})	3.35	3.24	3.20	3.27
SVR (M_{36})	2.61	2.60	1.94	2.47
NNR (M_{37})	2.29	**2.24**	1.94	**2.19**
Lasso (M_{38})	3.44	2.25	**1.92**	2.59
Ridge (M_{39})	2.36	2.37	1.96	2.28
ElasticNet (M_{310})	3.28	3.01	2.55	3.01

4. Boston housing:

Table B. 85. Site 1 models evaluation using MAPE metric

Models	Local MAPE (D1)	On Site2 (D2)	On Site3 (D3)	Average MAPE
LR (M_{11})	18%	15%	21%	18%
RFR (M_{12})	13%	12%	16%	14%
RBFNN (M_{13})	29%	27%	48%	35%
KNNR (M_{14})	22%	22%	27%	24%
DTR (M_{15})	17%	13%	19%	17%
SVR (M_{16})	16%	15%	21%	18%
NNR (M_{17})	31%	35%	32%	32%
Lasso (M_{18})	17%	16%	19%	17%
Ridge (M_{19})	17%	16%	21%	18%
ElasticNet (M_{110})	17%	17%	20%	18%

Table B. 86. Site 1 models evaluation using RMSE

Models	Local RMSE (D1)	On Site2 (D2)	On Site3 (D3)	Average RMSE
LR (M_{11})	4.77	7.52	5.00	5.51
RFR (M_{12})	**3.63**	6.01	**3.40**	**4.12**
RBFNN (M_{13})	7.76	11.70	10.15	9.55
KNNR (M_{14})	6.47	9.81	7.16	7.51
DTR (M_{15})	4.78	**5.72**	4.82	5.02
SVR (M_{16})	4.51	8.62	6.03	6.03
NNR (M_{17})	7.16	13.15	8.71	9.14
Lasso (M_{18})	4.88	7.76	3.79	5.18
Ridge (M_{19})	4.55	7.65	5.10	5.48
ElasticNet (M_{110})	4.72	7.77	5.54	5.74

Table B. 87. Site 2 models evaluation using MAPE metric

Models	Local MAPE (D2)	On Site1 (D1)	On Site3 (D3)	Average MAPE
LR (M_{21})	17%	63%	74%	56%
RFR (M_{22})	15%	30%	42%	31%
RBFNN (M_{23})	35%	49%	71%	53%
KNNR (M_{24})	27%	71%	93%	68%
DTR (M_{25})	19%	32%	32%	29%
SVR (M_{26})	14%	65%	75%	57%
NNR (M_{27})	40%	58%	62%	55%
Lasso (M_{28})	16%	49%	52%	42%
Ridge (M_{29})	15%	62%	72%	54%
ElasticNet (M_{210})	18%	44%	51%	40%

Table B. 88. Site 2 models evaluation using RMSE

Models	Local RMSE (D2)	On Site1 (D1)	On Site3 (D3)	Average RMSE
LR (M_{21})	4.94	14.60	13.81	12.01
RFR (M_{22})	5.003	**6.71**	**6.98**	**6.39**
RBFNN (M_{23})	10.19	9.66	11.61	10.48
KNNR (M_{24})	8.86	15.65	15.86	14.10
DTR (M_{25})	6.79	9.78	7.35	8.20
SVR (M_{26})	**4.81**	15.69	14.21	12.57
NNR (M_{27})	11.85	14.44	12.40	13.09
Lasso (M_{28})	5.16	11.77	10.09	9.59
Ridge (M_{29})	4.81	14.43	13.53	11.81
ElasticNet (M_{210})	5.35	10.02	9.45	8.70

Table B. 89. Site 3 models evaluation using MAPE metric

Models	Local MAPE (D3)	On Site1 (D1)	On Site2 (D2)	Average MAPE
LR (M_{31})	22%	20%	18%	20%
RFR (M_{32})	15%	15%	13%	15%
RBFNN (M_{33})	48%	28%	27%	35%
KNNR (M_{34})	25%	27%	28%	26%
DTR (M_{35})	17%	18%	17%	17%
SVR (M_{36})	21%	17%	16%	18%
NNR (M_{37})	74%	16%	15%	37%
Lasso (M_{38})	20%	84%	72%	58%
Ridge (M_{39})	21%	20%	18%	20%
ElasticNet (M_{310})	20%	19%	19%	19%

Table B. 90. Site 3 models evaluation using RMSE

Models	Local RMSE (D3)	On Site1 (D1)	On Site2 (D2)	Average RMSE
LR (M_{31})	4.48	5.26	8.00	5.64
RFR (M_{32})	**3.61**	**4.26**	7.78	**4.87**
RBFNN (M_{33})	9.87	7.81	11.70	9.47
KNNR (M_{34})	6.85	7.07	11.21	7.98
DTR (M_{35})	3.93	5.10	9.17	5.65
SVR (M_{36})	4.64	4.73	7.53	5.37
NNR (M_{37})	13.51	4.50	**6.04**	8.07
Lasso (M_{38})	4.93	17.26	17.04	12.83
Ridge (M_{39})	4.36	5.25	8.03	5.59
ElasticNet (M_{310})	5.09	5.00	7.72	5.68

II. Non-randomly Partitioned Dataset:

1. Parkinson Disease (Total UPDRS):

Table B. 91. Site 1 Models Evaluation Using MAPE Metric

Models	Local MAPE (D1)	On Site2 (D2)	On Site3 (D3)	Average MAPE
LR (M_{11})	35%	47%	31%	38%
RFR (M_{12})	**25%**	45%	34%	**35%**
RBFNN (M_{13})	48%	**44%**	28%	40%
KNNR (M_{14})	33%	**44%**	31%	36%
DTR (M_{15})	31%	54%	42%	42%
SVR (M_{16})	33%	45%	30%	36%
NNR (M_{17})	36%	**44%**	31%	37%
Lasso (M_{18})	41%	**44%**	29%	38%
Ridge (M_{19})	37%	**44%**	30%	37%
ElasticNet (M_{110})	40%	46%	**27%**	38%

Table B. 92. Site 1 Models Evaluation Using RMSE

Models	Local RMSE (D1)	On Site2 (D2)	On Site3 (D3)	Average RMSE
LR (M_{11})	8.96	12.63	14.94	12.15
RFR (M_{12})	**7.05**	12.71	16.78	12.14
RBFNN (M_{13})	11.03	**11.61**	14.49	12.36
KNNR (M_{14})	8.87	12.20	15.74	12.24
DTR (M_{15})	9.69	15.37	18.65	14.53
SVR (M_{16})	9.41	12.53	15.06	12.31
NNR (M_{17})	9.25	12.15	15.52	12.28
Lasso (M_{18})	9.69	11.93	15.16	12.23
Ridge (M_{19})	9.06	11.66	14.61	**11.75**
ElasticNet (M_{110})	9.68	11.69	**14.02**	11.78

Table B. 93. Site 2 Models Evaluation Using MAPE Metric

Models	Local MAPE (D2)	On Site1 (D1)	On Site3 (D3)	Average MAPE
LR (M_{21})	41%	56%	29%	42%
RFR (M_{22})	**33%**	**53%**	31%	**39%**
RBFNN (M_{23})	49%	59%	25%	44%
KNNR (M_{24})	43%	56%	29%	43%
DTR (M_{25})	39%	61%	40%	46%
SVR (M_{26})	45%	66%	27%	46%
NNR (M_{27})	44%	61%	**24%**	43%
Lasso (M_{28})	48%	62%	25%	45%
Ridge (M_{29})	43%	54%	29%	42%
ElasticNet (M_{210})	48%	59%	**24%**	44%

Table B. 94. Site 2 Models Evaluation Using RMSE

Models	Local RMSE (D2)	On Site1 (D1)	On Site3 (D3)	Average RMSE
LR (M_{21})	9.37	12.19	13.38	11.63
RFR (M_{22})	**8.40**	11.60	14.23	11.38
RBFNN (M_{23})	10.55	11.44	12.22	11.40
KNNR (M_{24})	10.24	11.92	12.98	11.70
DTR (M_{25})	11.45	14.12	17.38	14.29
SVR (M_{26})	10.37	12.88	13.13	12.12
NNR (M_{27})	9.88	11.87	12.41	11.38
Lasso (M_{28})	10.59	11.97	12.26	11.60
Ridge (M_{29})	9.76	**11.13**	13.29	11.37
ElasticNet (M_{210})	10.59	11.32	**12.08**	**11.32**

Table B. 95. Site 3 Models Evaluation Using MAPE Metric

Models	Local MAPE (D3)	On Site1 (D1)	On Site2 (D2)	Average MAPE
LR (M_{31})	25%	89%	85%	67%
RFR (M_{32})	**19%**	82%	62%	55%
RBFNN (M_{33})	28%	78%	61%	56%
KNNR (M_{34})	26%	76%	61%	54%
DTR (M_{35})	22%	86%	67%	59%
SVR (M_{36})	25%	**68%**	**56%**	**51%**
NNR (M_{37})	28%	85%	65%	60%
Lasso (M_{38})	28%	85%	65%	60%
Ridge (M_{39})	27%	83%	64%	58%
ElasticNet (M_{310})	28%	78%	61%	56%

Table B. 96. Site 3 Models Evaluation Using RMSE

Models	Local RMSE (D3)	On Site1 (D1)	On Site2 (D2)	Average RMSE
LR (M_{31})	9.58	19.41	26.13	18.46
RFR (M_{32})	**7.77**	16.06	13.17	12.38
RBFNN (M_{33})	10.53	14.52	12.09	12.40
KNNR (M_{34})	10.16	14.96	12.77	12.65
DTR (M_{35})	10.48	17.99	15.15	14.58
SVR (M_{36})	10.69	**12.93**	**11.49**	**11.71**
NNR (M_{37})	10.32	15.84	13.15	13.13
Lasso (M_{38})	10.57	15.80	13.21	13.22
Ridge (M_{39})	10.22	15.36	12.92	12.86
ElasticNet (M_{310})	10.56	14.42	12.15	12.39

2. Parkinson Disease (Motor UPDRS):

Table B. 97. Site 1 models evaluation using MAPE metric

Models	Local MAPE (D1)	On Site2 (D2)	On Site3 (D3)	Average MAPE
LR (M_{11})	36%	54%	34%	41%
RFR (M_{12})	27%	51%	36%	38%
RBFNN (M_{13})	48%	50%	30%	42%
KNNR (M_{14})	35%	51%	34%	40%
DTR (M_{15})	31%	58%	44%	44%
SVR (M_{16})	34%	50%	32%	39%
NNR (M_{17})	37%	50%	31%	40%
Lasso (M_{18})	41%	49%	32%	41%
Ridge (M_{19})	38%	50%	33%	41%
ElasticNet (M_{110})	41%	51%	29%	41%

Table B. 98. Site 1 models evaluation using RMSE

Models	Local RMSE (D1)	On Site2 (D2)	On Site3 (D3)	Average RMSE
LR (M_{11})	6.98	10.15	10.16	9.09
RFR (M_{12})	**5.51**	10.05	11.52	9.00
RBFNN (M_{13})	8.54	9.24	9.56	9.11
KNNR (M_{14})	6.89	9.78	10.65	9.09
DTR (M_{15})	7.26	11.91	13.20	10.77
SVR (M_{16})	7.21	9.70	10.16	9.01
NNR (M_{17})	7.17	9.43	10.04	8.87
Lasso (M_{18})	7.44	9.54	10.28	9.08
Ridge (M_{19})	7.02	9.29	9.86	8.71
ElasticNet (M_{110})	7.44	**9.22**	**9.22**	**8.62**

Table B. 99. Site 2 models evaluation using MAPE metric

Models	Local MAPE (D2)	On Site1 (D1)	On Site3 (D3)	Average MAPE
LR (M_{21})	45%	58%	32%	45%
RFR (M_{22})	34%	55%	35%	41%
RBFNN (M_{23})	54%	58%	28%	47%
KNNR (M_{24})	48%	58%	33%	46%
DTR (M_{25})	38%	71%	40%	49%
SVR (M_{26})	49%	63%	31%	48%
NNR (M_{27})	48%	63%	28%	47%
Lasso (M_{28})	53%	61%	29%	48%
Ridge (M_{29})	47%	54%	33%	45%
ElasticNet (M_{210})	53%	58%	29%	46%

Table B. 100. Site 2 models evaluation using RMSE

Models	Local RMSE (D2)	On Site1 (D1)	On Site3 (D3)	Average RMSE
LR (M_{21})	7.44	10.01	9.37	8.93
RFR (M_{22})	**6.36**	9.29	9.85	8.49
RBFNN (M_{23})	8.53	8.62	8.33	8.49
KNNR (M_{24})	8.49	9.41	9.77	9.22
DTR (M_{25})	8.66	12.46	11.39	10.83
SVR (M_{26})	8.30	9.49	9.16	8.98
NNR (M_{27})	7.94	9.25	**8.06**	8.42
Lasso (M_{28})	8.59	8.99	8.27	8.62
Ridge (M_{29})	7.91	8.56	9.60	8.68
ElasticNet (M_{210})	8.59	**8.45**	8.09	**8.38**

Table B. 101. Site 3 models evaluation using MAPE metric

Models	Local MAPE (D3)	On Site1 (D1)	On Site2 (D2)	Average MAPE
LR (M_{31})	23%	87%	86%	66%
RFR (M_{32})	19%	81%	72%	57%
RBFNN (M_{33})	29%	74%	65%	56%
KNNR (M_{34})	25%	73%	69%	56%
DTR (M_{35})	23%	82%	75%	61%
SVR (M_{36})	27%	77%	68%	58%

NNR (M_{37})	26%	77%	74%	59%
Lasso (M_{38})	29%	76%	73%	60%
Ridge (M_{39})	25%	83%	72%	61%
ElasticNet (M_{310})	29%	72%	64%	55%

Table B. 102. Site 3 models evaluation using RMSE

Models	Local RMSE (D3)	On Site1 (D1)	On Site2 (D2)	Average RMSE
LR (M_{31})	6.07	13.05	16.55	11.95
RFR (M_{32})	**5.39**	11.94	10.97	9.47
RBFNN (M_{33})	7.22	10.45	9.57	9.10
KNNR (M_{34})	6.61	10.93	10.46	9.36
DTR (M_{35})	7.38	13.10	12.06	10.88
SVR (M_{36})	6.68	10.92	10.06	9.24
NNR (M_{37})	6.40	11.19	11.75	9.81
Lasso (M_{38})	7.20	11.04	11.42	9.91
Ridge (M_{39})	6.37	11.81	10.61	9.63
ElasticNet (M_{310})	7.19	**10.12**	**9.39**	**8.92**

3. Boston housing:

Table B. 103. Site 1 models evaluation using MAPE metric

Models	Local MAPE (D1)	On Site2 (D2)	On Site3 (D3)	Average MAPE
LR (M_{11})	11%	14%	43%	23%
RFR (M_{12})	9%	11%	50%	24%
RBFNN (M_{13})	25%	34%	87%	50%
KNNR (M_{14})	21%	22%	45%	30%
DTR (M_{15})	12%	16%	51%	27%
SVR (M_{16})	11%	14%	34%	21%
NNR (M_{17})	38%	42%	85%	56%
Lasso (M_{18})	14%	44%	46%	36%
Ridge (M_{19})	11%	14%	39%	22%
ElasticNet (M_{110})	15%	26%	89%	45%

Table B. 104. Site 1 models evaluation using RMSE

Models	Local RMSE (D1)	On Site2 (D2)	On Site3 (D3)	Average RMSE
LR (M_{11})	3.33	3.92	11.00	6.25
RFR (M_{12})	**3.13**	**3.23**	**9.00**	**5.25**
RBFNN (M_{13})	7.67	8.91	13.10	10.02
KNNR (M_{14})	6.96	8.33	9.73	8.42
DTR (M_{15})	3.92	4.42	9.76	6.16
SVR (M_{16})	3.59	3.86	9.11	5.64
NNR (M_{17})	11.37	10.09	15.39	12.35
Lasso (M_{18})	4.37	15.23	13.41	11.34
Ridge (M_{19})	3.21	3.85	10.38	5.97
ElasticNet (M_{110})	4.72	7.57	16.78	9.98

Table B. 105. Site 2 models evaluation using MAPE metric

Models	Local MAPE (D2)	On Site1 (D1)	On Site3 (D3)	Average MAPE
LR (M_{21})	11%	9%	59%	27%
RFR (M_{22})	9%	12%	35%	19%
RBFNN (M_{23})	27%	21%	72%	41%
KNNR (M_{24})	21%	23%	34%	26%
DTR (M_{25})	14%	15%	31%	21%
SVR (M_{26})	11%	11%	58%	28%
NNR (M_{27})	59%	19%	73%	52%
Lasso (M_{28})	11%	20%	85%	41%
Ridge (M_{29})	10%	10%	59%	27%
ElasticNet (M_{210})	14%	15%	51%	27%

Table B. 106. Site 2 models evaluation using RMSE

Models	Local RMSE (D2)	On Site1 (D1)	On Site3 (D3)	Average RMSE
LR (M_{21})	2.85	**3.47**	10.78	5.85
RFR (M_{22})	2.84	4.03	**7.63**	**4.89**
RBFNN (M_{23})	8.19	8.23	11.56	9.39
KNNR (M_{24})	6.71	8.10	9.33	8.05
DTR (M_{25})	3.93	5.19	8.08	5.78
SVR (M_{26})	3.10	4.08	10.33	5.96
NNR (M_{27})	14.45	6.16	12.96	11.44
Lasso (M_{28})	3.67	6.27	16.16	8.88
Ridge (M_{29})	**2.81**	3.51	10.68	5.81
ElasticNet (M_{210})	4.40	4.85	8.56	6.01

Table B. 107. Site 3 models evaluation using MAPE metric

Models	Local MAPE (D3)	On Site1 (D1)	On Site2 (D2)	Average MAPE
LR (M_{31})	29%	31%	30%	30%
RFR (M_{32})	19%	15%	14%	16%
RBFNN (M_{33})	39%	30%	24%	31%
KNNR (M_{34})	27%	36%	34%	32%
DTR (M_{35})	24%	20%	20%	22%
SVR (M_{36})	21%	27%	29%	26%
NNR (M_{37})	61%	21%	22%	36%
Lasso (M_{38})	26%	31%	29%	29%
Ridge (M_{39})	29%	59%	59%	48%
ElasticNet (M_{310})	27%	19%	18%	22%

Table B. 108. Site 3 models evaluation using RMSE

Models	Local RMSE (D3)	On Site1 (D1)	On Site2 (D2)	Average RMSE
LR (M_{31})	6.22	8.92	8.46	7.80
RFR (M_{32})	**4.37**	**5.33**	**4.90**	**4.84**
RBFNN (M_{33})	9.07	12.05	10.63	10.50
KNNR (M_{34})	6.19	12.73	12.13	10.20
DTR (M_{35})	5.09	8.85	8.51	7.40
SVR (M_{36})	6.69	10.73	11.34	9.51
NNR (M_{37})	11.52	9.02	9.37	10.03
Lasso (M_{38})	6.36	8.92	8.46	7.85
Ridge (M_{39})	7.19	18.09	16.49	13.67
ElasticNet (M_{310})	6.36	7.92	7.08	7.07

Appendix C

Detailed Results for the Proposed Method in Chapter 5

This appendix presents the global and local level modelling detailed results using stepwise model selection and gossip learning methods for classification and regression datasets.

Global-level Modelling:

a) Classification

I. Randomly Partitioned Data:

1. Heart Disease Dataset:

The following tables show the detailed steps results for heart disease dataset that partitioned randomly. Tables C.1 – C.3 show the site1, site2, and site3 local models accuracy, the models evaluation results in other sites, and the weighted average accuracy based on the data size that used for evaluation.

Table C. 1. Site 1 Local Models Evaluation Results for heart disease dataset

S1		In S2	In S3	Weighted average accuracy
Models	Accuracy	Accuracy	Accuracy	
LR	75%	60%	62%	66%
RF	79%	71%	77%	75%
NB	85%	73%	78%	79%
SVM (nonlinear)	78%	63%	67%	70%
SVM (linear)	80%	70%	77%	75%
NN	81%	73%	78%	77%
DT (GD)	82%	70%	87%	79%

Table C. 2. Site 2 Local Models Evaluation Results for heart disease dataset

S2		In S1	In S3	Weighted average accuracy
Models	Accuracy	Accuracy	Accuracy	
LR	69%	83%	73%	75%
RF	70%	78%	80%	75%
NB	75%	84%	82%	80%
SVM (nonlinear)	66%	80%	75%	74%
SVM (linear)	67%	75%	70%	71%
NN	74%	83%	80%	79%
DT (GD)	72%	78%	85%	77%

Table C. 3. Site 3 Local Models Evaluation Results for heart disease dataset

S3		In S1	In S2	Weighted average accuracy
Models	Accuracy	Accuracy	Accuracy	
LR	75%	75%	60%	69%
RF	82%	87%	75%	81%
NB	77%	81%	75%	78%
SVM (nonlinear)	73%	79%	73%	75%
SVM (linear)	79%	76%	68%	74%
NN	82%	80%	72%	77%
DT (GD)	80%	82%	74%	78%

For each learning model, we select the best average accuracy model.

Table C. 4. Local Models Average Accuracy for All sites

Models	S1 model average accuracy	S2 model average accuracy	S3 model average accuracy	The best model
LR	66%	**75%**	69%	LR-S2
RF	75%	75%	**81%**	RF-S3
NB	79%	**80%**	78%	NB-S2
SVM (nonlinear)	70%	74%	**75%**	SVM (nonlinear)-S3
SVM (linear)	**75%**	71%	74%	SVM (linear)-S1
NN	77%	**79%**	77%	NN-S2
DT	**79%**	77%	78%	DT-S1

Then, we sent the selected models to other sites and applied mini-batch stochastic gradient descent to update the models. Table C.5 shows the selected LR model of site 2 after sending it to site 1 and site 3 for updating method.

Table C. 5. Logistic Regression (LR) Model Evaluation Results Before and After Updating Method

The best model	In S1 (LR-S21)		In S3 (LR-S23)	
	Before update	After update	Before update	After update
LR-S2	83%	81%	73%	77%

Table C.6 shows the updated models evaluation results in all sites and the weighted average accuracy. LR model of site 2 that is updated in site 3 is the best. Therefore, we sent this model to site 1 to update it, as shown in Table C.7.

Table C. 6. Logistic Regression (LR) Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Weighted Average accuracy
LR-S21	81%	63%	67%	71%
LR-S23	75%	68%	77%	**73%**

Table C. 7. Logistic Regression (LR) Updated Model Evaluation Results Before and After Updating Method

The best model	In S1 (LR-S231)	
	Before update	After update
LR-S23	75%	81%

Tables C.8-C.25 show the same model selection and updating strategies that are applied to LR model for all learning models RF, NB, SVM, NN, and DT algorithms.

Table C. 8. Random Forest (RF) Model Evaluation Results Before and After Updating Method

The best model	In S1 (RF-S31)		In S2 (RF-S32)	
	Before update	After update	Before update	After update
RF-S3	87%	90%	75%	89%

Table C. 9. Random Forest (RF) Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Weighted average accuracy
RF-S31	90%	72%	77%	80%
RF-S32	80%	89%	82%	**83%**

Table C. 10. Random Forest (RF) Updated Model Evaluation Results Before and After Updating Method

The best model	In S1 (RF-S321)	
	Before update	After update
RF-S32	80%	91%

Table C. 11. Naïve Bayes (NB) Model Evaluation Results Before and After Updating Method

The best model	In S1 (NB-S21)		In S3 (NB-S23)	
	Before update	After update	Before update	After update
NB-S2	84%	83%	82%	87%

Table C. 12. Naïve Bayes (NB) Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Weighted average accuracy
NB-S21	83%	73%	73%	77%
NB-S21	81%	75%	87%	**80%**

Table C. 13. Naïve Bayes (NB) Updated Model Evaluation Results Before and After Updating Method

The best model	In S3 (NB-S213)	
	Before update	After update
NB-S21	87%	85%

Table C. 14. Nonlinear Support Vector Machine (SVM) Model Evaluation Results Before and After Updating Method

The best model	In S1(SVM (nonlinear)-S31)		In S2 (SVM nonlinear-S32)	
	Before update	After update	Before update	After update
SVM nonlinear -S3	79%	85%	73%	73%

Table C. 15. Nonlinear Support Vector Machine (SVM) Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Weighted average accuracy
SVM nonlinear-S31	85%	65%	80%	76%
SVM nonlinear-S32	78%	73%	82%	**77%**

Table C. 16. Nonlinear Support Vector Machine (SVM) Updated Model Evaluation Results Before and After Updating Method

The best model	In S1 (SVM nonlinear-S321)	
	Before update	After update
SVM nonlinear-S32	78%	81%

Table C. 17. Linear Support Vector Machine (SVM) Model Evaluation Results Before and After Updating Method

The best model	In S2 (SVM linear)-S12)		In S3 (SVM linear)-S13)	
	Before update	After update	Before update	After update
SVM linear-S1	70%	73%	77%	83%

Table C. 18. Linear Support Vector Machine (SVM) Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Weighted average accuracy
SVM linear-S12	83%	73%	75%	**77%**
SVM linear-S13	76%	72%	83%	76%

Table C. 19. Linear Support Vector Machine (SVM) Updated Model Evaluation Results Before and After Updating Method

The best model	In S3 (SVM linear-S123)	
	Before update	After update
SVM linear-S12	75%	87%

Table C. 20. Neural Network (NN) Model Evaluation Results Before and After Updating Method

The best model	In S1 (NN-S21)		In S3 (NN-S23)	
	Before update	After update	Before update	After update
NN-S2	83%	83%	80%	68%

Table C. 21. Neural Network (NN) Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Weighted average accuracy
NN-S21	83%	90%	80%	**85%**
NN-S23	62%	67%	68%	65%

Table C. 22. Neural Network (NN) Updated Model Evaluation Results Before and After Updating Method

The best model	In S3 (NN-S213)	
	Before update	After update
NN-S21	80%	80%

Table C. 23. Decision Tree (DT) Model Evaluation Results Before and After Updating Method

The best model	In S2 (DT-S12)		In S3 (DT-S13)	
	Before update	After update	Before update	After update
DT-S1	70%	92%	87%	90%

Table C. 24. Decision Tree (DT) Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Weighted average accuracy
DT-S12	73%	92%	78%	**81%**
DT-S13	82%	72%	90%	80%

Table C. 25. Decision Tree (DT) Updated Model Evaluation Results Before and After Updating Method

The best model	In S3 (DT-S123)	
	Before update	After update
DT-S12	78%	92%

Finally, we evaluated the final updated models in all sites and then sent the models with their evaluation results and the data size used for models evaluation to the server. The server calculates the average accuracy for each model and then selects the best average accuracy models for the linear combination method. As illustrated in Table C.26, the best three models are selected and combined using the linear combination method.

Table C. 26. Updated Models Evaluation Results in All sites

The best model	In S1	In S2	In S3	Weighted average accuracy
LR-S231	81%	63%	67%	71%
RF-S321	91%	72%	77%	**80%**
NB-S213	80%	71%	85%	78%
SVM nonlinear-S321	81%	59%	67%	70%
SVM linear-S123	77%	71%	87%	77%
NN-S213	83%	90%	80%	**84%**
DT-S123	82%	72%	92%	**81%**

We applied the same methodology on the rest of classification datasets.

2. Blood Transfusion Dataset:

Table C. 27. Site 1 local models evaluation

S1		In S2	In S3	Weighted average
Models	Accuracy	Accuracy	Accuracy	
LR	68%	35%	41%	50%
RF	62%	63%	75%	67%
NB	73%	86%	78%	78%
SVM (nonlinear)	70%	85%	74%	75%
SVM (linear)	70%	87%	77%	77%
NN	74%	89%	77%	79%
DT (GD)	73%	89%	78%	79%

Table C. 28. Site 2 local models evaluation

S2		In S1	In S3	Weighted average
Models	Accuracy	Accuracy	Accuracy	
LR	87%	53%	54%	61%
RF	76%	72%	77%	75%
NB	85%	71%	75%	76%
SVM (nonlinear)	84%	72%	78%	77%
SVM (linear)	75%	76%	80%	77%
NN	87%	74%	77%	78%
DT (GD)	87%	68%	75%	75%

Table C. 29. Site 3 local models evaluation

S3		In S1	In S2	Weighted average
Models	Accuracy	Accuracy	Accuracy	
LR	72%	57%	55%	62%
RF	65%	76%	87%	74%
NB	74%	70%	85%	75%
SVM (nonlinear)	71%	64%	79%	70%
SVM (linear)	71%	56%	61%	63%
NN	76%	74%	89%	78%
DT (GD)	76%	74%	88%	78%

Table C. 30. Local Models Average Accuracy for All sites

Models	S1 model accuracy	S2 model accuracy	S3 model accuracy	The best model
LR	50%	61%	**62%**	LR-S3
RF	67%	**75%**	74%	RF-S2
NB	**78%**	76%	75%	NB-S1
SVM (nonlinear)	75%	**77%**	70%	SVM (nonlinear)-S2
SVM (linear)	**77%**	**77%**	63%	SVM (linear)-S1 and S2
NN	**79%**	78%	78%	NN-S1
DT	**79%**	75%	78%	DT-S1

Table C. 31. Logistic Regression (LR) Model Evaluation Results After Updating Method

The best model	In S1 (LR-S31)	In S2 (LR-S32)
	After update the model	After update the model
LR-S3	62%	79%

Table C. 32. Logistic Regression (LR) Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average accuracy
LR-S31	62%	65%	66%	64%
LR-S32	71%	79%	74%	**74%**

Table C. 33. Logistic Regression (LR) Updated Model Evaluation Results After Updating Method

The best model	In S1 (LR-S321)
	After update the model
LR-S32	72%

Table C. 34. Random Forest (RF) Model Evaluation Results After Updating Method

The best model	In S1 (RF-S21)	In S3 (RF-S23)
	After update the model	After update the model
RF-S2	92%	93%

Table C. 35. Random Forest (RF) Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average accuracy
RF-S21	92%	77%	74%	82%
RF-S23	75%	80%	93%	**83%**

Table C. 36. Random Forest (RF) Updated Model Evaluation Results After Updating Method

The best model	In S1 (RF-S231)
	After update the model
RF-S23	92%

Table C. 37. Naïve Bayes (NB) Model Evaluation Results After Updating Method

The best model	In S2 (NB-S12)	In S3 (NB-S13)
	After update the model	After update the model
NB-S1	65%	77%

Table C. 38. Naïve Bayes (NB) Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average accuracy
NB-S12	61%	65%	58%	61%
NB-S13	70%	85%	77%	**76%**

Table C. 39. Naïve Bayes (NB) Updated Model Evaluation Results After Updating Method

The best model	In S2 (NB-S132)
	After update the model
NB-S13	85%

Table C. 40. Nonlinear Support Vector Machine (SVM) Model Evaluation Results After Updating Method

The best model	In S1(SVM (nonlinear)-S21)	In S3 (SVM nonlinear-S23)
	After update the model	After update the model
SVM nonlinear -S2	73%	73%

Table C. 41. Nonlinear Support Vector Machine (SVM) Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average accuracy
SVM nonlinear-S21	73%	86%	75%	**77%**
SVM nonlinear-S23	68%	83%	73%	73%

Table C. 42. Nonlinear Support Vector Machine (SVM) Updated Model Evaluation Results After Updating Method

The best model	In S3 (SVM nonlinear-S213)
	After update the model
SVM nonlinear-S21	79%

Table C. 43. Linear Support Vector Machine (SVM) Model Evaluation Results After Updating Method

The best model	In S2 (SVM linear)-S12)	In S3 (SVM linear)-S13)
	After update the model	After update the model
SVM (linear)-S1	87%	73%

Table C. 44. Linear Support Vector Machine (SVM) Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average accuracy
SVM linear-S12	74%	87%	78%	**79%**
SVM linear-S13	67%	86%	73%	74%

Table C. 45. Linear Support Vector Machine (SVM) Updated Model Evaluation Results After Updating Method

The best model	In S3 (SVM linear-S123)
	After update the model
SVM linear-S12	73%

Table C. 46. Linear Support Vector Machine (SVM) Model Evaluation Results After Updating Method

The best model	In S1(SVM linear-S21)	In S3 (SVM linear-S23)
	After update the model	After update the model
SVM linear-S2	73%	77%

Table C. 47. Linear Support Vector Machine (SVM) Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average accuracy
SVM linear-S21	73%	87%	77%	78%
SVM linear-S23	74%	89%	77%	**79%**

Table C. 48. Linear Support Vector Machine (SVM) Updated Model Evaluation Results After Updating Method

The best model	In S1 (SVM linear-S231)
	After update the model
SVM linear-S23	73%

Table C. 49. Neural Network (NN) Model Evaluation Results After Updating Method

The best model	In S2 (NN-S12)	In S3 (NN-S13)
	After update the model	After update the model
NN-S1	80%	76%

Table C. 50. Neural Network (NN) Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average accuracy
NN-S12	65%	80%	73%	71%
NN-S13	69%	87%	76%	**76%**

Table C. 51. Neural Network (NN) Updated Model Evaluation Results After Updating Method

The best model	In S2 (NN-S132)
	After update the model
NN-S13	86%

Table C. 52. Decision Tree (DT) Model Evaluation Results After Updating Method

The best model	In S2 (DT-S12)	In S3 (DT-S13)
	After update the model	After update the model
DT-S1	96%	92%

Table C. 53. Decision Tree (DT) Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average accuracy
DT-S12	72%	96%	74%	**78%**
DT-S13	64%	78%	92%	77%

Table C. 54. Decision Tree (DT) Updated Model Evaluation Results After Updating Method

The best model	In S3 (DT-S123)
	After update the model
DT-S12	92%

Table C. 55. Updated Models Evaluation Results in All sites

The best model	In S1	In S2	In S3	Average accuracy
LR-S321	72%	87%	77%	77%
RF-S231	92%	77%	74%	**82%**
NB-S132	70%	85%	76%	76%
SVM nonlinear-S213	71%	89%	79%	**78%**
SVM linear-S123	72%	82%	73%	75%
SVM linear-S231	73%	87%	77%	**78%**
NN-S132	73%	86%	76%	77%
DT-S123	64%	77%	92%	77%

3. Breast Cancer Wisconsin (Diagnostic):

Table C. 56. Site 1 Local Models Evaluation

S1		In S2	In S3	Weighted average
Models	Accuracy	Accuracy	Accuracy	
LR	94%	85%	95%	92%
RF	94%	85%	97%	92%
NB	92%	83%	96%	90%
SVM (nonlinear)	94%	83%	97%	92%
SVM (linear)	92%	83%	95%	90%
NN	96%	87%	97%	94%
DT (GD)	93%	85%	96%	91%

Table C. 57. Site 2 Local Models Evaluation

S2		In S1	In S3	Weighted average
Models	Accuracy	Accuracy	Accuracy	
LR	96%	88%	94%	92%
RF	96%	81%	93%	89%
NB	96%	84%	97%	91%
SVM (nonlinear)	95%	85%	93%	90%
SVM (linear)	95%	87%	96%	92%
NN	97%	85%	94%	91%
DT (GD)	95%	82%	93%	89%

Table C. 58. Site 3 Local Models Evaluation

S3		In S1	In S2	Weighted average
Models	Accuracy	Accuracy	Accuracy	
LR	99%	92%	93%	94%
RF	96%	89%	92%	92%
NB	97%	90%	89%	92%
SVM (nonlinear)	99%	92%	95%	95%
SVM (linear)	98%	92%	97%	95%
NN	98%	90%	93%	93%
DT (GD)	95%	89%	90%	91%

Table C. 59. Local Models Average Accuracy for All sites

Models	S1 model accuracy	S2 model accuracy	S3 model accuracy	The best model
LR	92%	92%	**94%**	LR-S3
RF	**92%**	89%	**92%**	RF-S1 - RF-S3
NB	90%	91%	**92%**	NB-S3

SVM (nonlinear)	92%	90%	**95%**	SVM (nonlinear)-S3
SVM (linear)	90%	92%	**95%**	SVM (linear)-S3
NN	**94%**	91%	93%	NN-S1
DT	**91%**	89%	**91%**	DT-S1 - DT-S3

Table C. 60. Logistic Regression (LR) Model Evaluation Results After Updating Method

The best model	In S1 (LR-S31)	In S2 (LR-S32)
	After update the model	After update the model
LR-S3	95%	95%

Table C. 61. Logistic Regression (LR) Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average accuracy
LR-S31	95%	79%	93%	90%
LR-S32	86%	95%	96%	**92%**

Table C. 62. Logistic Regression (LR) Updated Model Evaluation Results After Updating Method

The best model	In S1 (LR-S321)
	After update the model
LR-S32	89%

Table C. 63. Random Forest (RF) Model Evaluation Results After Updating Method

The best model	In S2 (RF-S12)	In S3 (RF-S13)
	After update the model	After update the model
RF-S1	98%	98%

Table C. 64. Random Forest (RF) Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average accuracy
RF-S12	81%	98%	92%	89%
RF-S13	89%	92%	98%	**93%**

Table C. 65. Random Forest (RF) Updated Model Evaluation Results After Updating Method

The best model	In S2 (RF-S132)
	After update the model
RF-S13	99%

Table C. 66. Random Forest (RF) Model Evaluation Results After Updating Method

The best model	In S1 (RF-S31)	In S2 (RF-S32)
	After update the model	After update the model
RF-S3	98%	98%

Table C. 67. Random Forest (RF) Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average accuracy
RF-S31	98%	87%	97%	**94%**
RF-S32	80%	98%	93%	89%

Table C. 68. Random Forest (RF) Updated Model Evaluation Results After Updating Method

The best model	In S2 (RF-S312)
	After update the model
RF-S31	94%

Table C. 69. Naïve Bayes (NB) Model Evaluation Results After Updating Method

The best model	In S1 (NB-S31)	In S2 (NB-S32)
	After update the model	After update the model
NB-S3	89%	97%

Table C. 70. Naïve Bayes (NB) Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average accuracy
NB-S31	89%	86%	95%	**90%**
NB-S32	81%	97%	91%	89%

Table C. 71. Naïve Bayes (NB) Updated Model Evaluation Results After Updating Method

The best model	In S2 (NB-S312)
	After update the model
NB-S31	97%

Table C. 72. Nonlinear Support Vector Machine (SVM) Model Evaluation Results After Updating Method

The best model	In S1(SVM (nonlinear)-S31)	In S2 (SVM nonlinear-S32)
	After update the model	After update the model
SVM nonlinear -S3	94%	81%

Table C. 73. Nonlinear Support Vector Machine (SVM) Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average accuracy
SVM nonlinear-S31	94%	81%	90%	**89%**
SVM nonlinear-S32	88%	81%	81%	84%

Table C. 74. Nonlinear Support Vector Machine (SVM) Updated Model Evaluation Results After Updating Method

The best model	In S2 (SVM nonlinear-S312)
	After update the model
SVM nonlinear-S31	90%

Table C. 75. Linear Support Vector Machine (SVM) Model Evaluation Results After Updating Method

The best model	In S1 (SVM linear)-S31)	In S2 (SVM linear)-S32)
	After update the model	After update the model
SVM (linear)-S3	92%	93%

Table C. 76. Linear Support Vector Machine (SVM) Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average accuracy
SVM linear-S31	92%	74%	85%	84%
SVM linear-S32	80%	93%	88%	**86%**

Table C. 77. Linear Support Vector Machine (SVM) Updated Model Evaluation Results After Updating Method

The best model	In S1 (SVM linear-S321)
	After update the model
SVM linear-S32	96%

Table C. 78. Neural Network (NN) Model Evaluation Results After Updating Method

The best model	In S2 (NN-S12)	In S3 (NN-S13)
	After update the model	After update the model
NN-S1	91%	83%

Table C. 79. Neural Network (NN) Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average accuracy
NN-S12	74%	91%	81%	81%
NN-S13	82%	79%	83%	**82%**

Table C. 80. Neural Network (NN) Updated Model Evaluation Results After Updating Method

The best model	In S2 (NN-S132)
	After update the model
NN-S13	93%

Table C. 81. Decision Tree (DT) Model Evaluation Results After Updating Method

The best model	In S2 (DT-S12)	In S3 (DT-S13)
	After update the model	After update the model
DT-S1	98%	98%

Table C. 82. Decision Tree (DT) Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average accuracy
DT-S12	82%	98%	91%	89%
DT-S13	90%	91%	98%	**93%**

Table C. 83. Decision Tree (DT) Updated Model Evaluation Results After Updating Method

The best model	In S2 (DT-S132)
	After update the model
DT-S13	97%

Table C. 84. Decision Tree (DT) Model Evaluation Results After Updating Method

The best model	In S1 (DT-S31)	In S2 (DT-S32)
	After update the model	After update the model
DT-S3	98%	98%

Table C. 85. Decision Tree (DT) Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average accuracy
DT-S31	98%	86%	94%	**93%**
DT-S32	82%	98%	91%	89%

Table C. 86. Decision Tree (DT) Updated Model Evaluation Results After Updating Method

The best model	In S2 (DT-S312)
	After update the model
DT-S31	93%

Table C. 87. Updated Models Evaluation Results in All sites

The best model	In S1	In S2	In S3	Average accuracy
LR-S321	89%	83%	93%	88%
RF-S132	81%	99%	93%	**90%**
RF-S312	80%	94%	91%	88%
NB-S312	85%	97%	97%	**92%**
SVM nonlinear-S312	90%	90%	86%	89%
SVM linear-S321	96%	88%	97%	**94%**
NN-S132	81%	93%	91%	88%
DT-S132	82%	97%	91%	89%
DT-S312	82%	93%	91%	88%

4. Diabetes:

Table C. 88. Site 1 Local Models Evaluation

S1		In S2	In S3	Weighted average
Models	Accuracy	Accuracy	Accuracy	
LR	68%	64%	72%	68%
RF	68%	70%	79%	72%
NB	70%	75%	80%	74%
SVM (nonlinear)	67%	67%	70%	68%
SVM (linear)	66%	71%	72%	69%
NN	73%	76%	82%	77%
DT (GD)	67%	73%	81%	73%

Table C. 89. Site 2 Local Models Evaluation

S2		In S1	In S3	Weighted average
Models	Accuracy	Accuracy	Accuracy	
LR	70%	68%	72%	70%
RF	70%	71%	78%	73%
NB	71%	73%	80%	75%
SVM (nonlinear)	66%	72%	78%	73%
SVM (linear)	68%	68%	74%	70%
NN	75%	71%	80%	75%
DT (GD)	72%	71%	77%	73%

Table C. 90. Site 3 Local Models Evaluation

S3		In S1	In S2	Weighted average
Models	Accuracy	Accuracy	Accuracy	
LR	76%	70%	75%	73%
RF	79%	70%	71%	73%
NB	80%	72%	72%	75%
SVM (nonlinear)	73%	68%	66%	69%
SVM (linear)	73%	68%	67%	69%
NN	81%	73%	75%	76%
DT (GD)	78%	71%	71%	73%

Table C. 91. Local Models Average Accuracy for All sites

Models	S1 model accuracy	S2 model accuracy	S3 model accuracy	The best model
LR	68%	70%	**73%**	LR-S3
RF	72%	**73%**	**73%**	RF-S2 - RF-S3
NB	74%	**75%**	**75%**	NB-S2 - NB-S3
SVM (nonlinear)	68%	**73%**	69%	SVM (nonlinear)-S2
SVM (linear)	69%	**70%**	69%	SVM (linear)-S2
NN	**77%**	75%	76%	NN-S1
DT	**73%**	**73%**	**73%**	DT-S1 - DT-S2- DT-S3

Table C. 92. Logistic Regression (LR) Model Evaluation Results After Updating Method

The best model	In S1 (LR-S31)	In S2 (LR-S32)
	After update the model	After update the model
LR-S3	70%	75%

Table C. 93. Logistic Regression (LR) Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average accuracy
LR-S31	70%	73%	78%	**73%**
LR-S32	68%	75%	65%	68%

Table C. 94. Logistic Regression (LR) Updated Model Evaluation Results After Updating Method

The best model	In S2 (LR-S312)
	After update the model
LR-S31	75%

Table C. 95. Random Forest (RF) Model Evaluation Results After Updating Method

The best model	In S1 (RF-S21)	In S3 (RF-S23)
	After update the model	After update the model
RF-S2	97%	97%

Table C. 96. Random Forest (RF) Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average accuracy
RF-S21	97%	77%	78%	**86%**
RF-S23	70%	74%	97%	80%

Table C. 97. Random Forest (RF) Updated Model Evaluation Results After Updating Method

The best model	In S3 (RF-S)
	After update the model
RF-S21	91%

Table C. 98. Random Forest (RF) Model Evaluation Results After Updating Method

The best model	In S1 (RF-S31)	In S2 (RF-S32)
	After update the model	After update the model
RF-S3	96%	61%

Table C. 99. Random Forest (RF) Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average accuracy
RF-S31	96%	75%	77%	**85%**
RF-S32	62%	61%	73%	65%

Table C. 100. Random Forest (RF) Updated Model Evaluation Results After Updating Method

The best model	In S2 (RF-S312)
	After update the model
RF-S31	95%

Table C. 101. Naïve Bayes (NB) Model Evaluation Results After Updating Method

The best model	In S1 (NB-S21)	In S3 (NB-S23)
	After update the model	After update the model
NB-S2	73%	82%

Table C. 102. Naïve Bayes (NB) Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average accuracy
NB-S21	73%	71%	77%	74%
NB-S23	72%	72%	82%	**75%**

Table C. 103. Naïve Bayes (NB) Updated Model Evaluation Results After Updating Method

The best model	In S1 (NB-S231)
	After update the model
NB-S23	72%

Table C. 104. Naïve Bayes (NB) Model Evaluation Results After Updating Method

The best model	In S1 (NB-S31)	In S2 (NB-S32)
	After update the model	After update the model
NB-S3	73%	75%

Table C. 105. Naïve Bayes (NB) Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average accuracy
NB-S31	73%	71%	77%	74%
NB-S32	76%	75%	77%	**76%**

Table C. 106. Naïve Bayes (NB) Updated Model Evaluation Results After Updating Method

The best model	In S1 (NB-S321)
	After update the model
NB-S32	72%

Table C. 107. Nonlinear Support Vector Machine (SVM) Model Evaluation Results After Updating Method

The best model	In S1(SVM (nonlinear)-S21)	In S3 (SVM nonlinear-S23)
	After update the model	After update the model
SVM nonlinear -S2	72%	72%

Table C. 108. Nonlinear Support Vector Machine (SVM) Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average accuracy
SVM nonlinear-S21	72%	75%	78%	**75%**
SVM nonlinear-S23	69%	71%	72%	70%

Table C. 109. Nonlinear Support Vector Machine (SVM) Updated Model Evaluation Results After Updating Method

The best model	In S3 (SVM nonlinear-S213)
	After update the model
SVM nonlinear-S21	75%

Table C. 110. Linear Support Vector Machine (SVM) Model Evaluation Results After Updating Method

The best model	In S1 (SVM linear)-S21)	In S3 (SVM linear)-S23)
	After update the model	After update the model
SVM (linear)-S2	70%	74%

Table C. 111. Linear Support Vector Machine (SVM) Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average accuracy
SVM linear-S21	70%	73%	75%	**72%**
SVM linear-S23	69%	69%	74%	71%

Table C. 112. Linear Support Vector Machine (SVM) Updated Model Evaluation Results After Updating Method

The best model	In S3 (SVM linear-S213)
	After update the model
SVM linear-S21	74%

Table C. 113. Neural Network (NN) Model Evaluation Results After Updating Method

The best model	In S2 (NN-S12)	In S3 (NN-S13)
	After update the model	After update the model
NN-S1	65%	72%

Table C. 114. Neural Network (NN) Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average accuracy
NN-S12	65%	65%	70%	**67%**
NN-S13	63%	61%	72%	66%

Table C. 115. Neural Network (NN) Updated Model Evaluation Results After Updating Method

The best model	In S3 (NN-S123)
	After update the model
NN-S12	70%

Table C. 116. Decision Tree (DT) Model Evaluation Results After Updating Method

The best model	In S2 (DT-S12)	In S3 (DT-S13)
	After update the model	After update the model
DT-S1	96%	92%

Table C. 117. Decision Tree (DT) Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average accuracy
DT-S12	71%	96%	75%	78%
DT-S13	72%	75%	92%	**79%**

Table C. 118. Decision Tree (DT) Updated Model Evaluation Results After Updating Method

The best model	In S2 (DT-S132)
	After update the model
DT-S13	93%

Table C. 119. Decision Tree (DT) Model Evaluation Results After Updating Method

The best model	In S1 (DT-S21)	In S3 (DT-S23)
	After update the model	After update the model
DT-S2	82%	90%

Table C. 120. Decision Tree (DT) Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average accuracy
DT-S21	82%	77%	79%	**80%**
DT-S23	72%	75%	90%	79%

Table C. 121. Decision Tree (DT) Updated Model Evaluation Results After Updating Method

The best model	In S3 (DT-S213)
	After update the model
DT-S21	95%

Table C. 122. Decision Tree (DT) Model Evaluation Results After Updating Method

The best model	In S1 (DT-S31)	In S2 (DT-S32)
	After update the model	After update the model
DT-S3	95%	97%

Table C. 123. Decision Tree (DT) Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average accuracy
DT-S31	95%	77%	79%	**86%**
DT-S32	71%	97%	76%	78%

Table C. 124. Decision Tree (DT) Updated Model Evaluation Results After Updating Method

The best model	In S2 (DT-S312)
	After update the model
DT-S31	93%

Table C. 125. Updated Models Evaluation Results in All sites

The best model	In S1	In S2	In S3	Average accuracy
LR-S312	70%	75%	75%	73%
RF-S213	70%	74%	91%	**78%**
RF-S312	72%	95%	75%	**78%**
NB-S231	72%	72%	79%	74%
NB-S321	72%	72%	79%	74%

SVM nonlinear-S213	72%	70%	75%	73%
SVM linear-S213	69%	69%	74%	71%
NN-S123	65%	65%	70%	67%
DT-S132	71%	93%	75%	77%
DT-S213	72%	75%	95%	**80%**
DT-S312	71%	93%	76%	77%

5. Liver Disease:

Table C. 126. Site 1 Local Models Evaluation

S1		In S2	In S3	Weighted average
Models	Accuracy	Accuracy	Accuracy	
LR	68%	68%	65%	67%
RF	67%	63%	55%	63%
NB	50%	50%	63%	53%
SVM (nonlinear)	64%	70%	67%	66%
SVM (linear)	71%	62%	64%	67%
NN	73%	65%	65%	69%
DT (GD)	74%	64%	65%	69%

Table C. 127. Site 2 Local Models Evaluation

S2		In S1	In S3	Weighted average
Models	Accuracy	Accuracy	Accuracy	
LR	64%	60%	60%	61%
RF	59%	68%	62%	64%
NB	52%	52%	62%	54%
SVM (nonlinear)	66%	73%	61%	68%
SVM (linear)	58%	56%	51%	56%
NN	67%	70%	69%	69%
DT (GD)	64%	71%	65%	68%

Table C. 128. Site 3 Local Models Evaluation

S3		In S1	In S2	Weighted average
Models	Accuracy	Accuracy	Accuracy	
LR	64%	54%	53%	56%
RF	62%	60%	65%	62%
NB	63%	54%	55%	56%
SVM (nonlinear)	68%	68%	64%	67%
SVM (linear)	60%	54%	55%	55%
NN	56%	72%	64%	66%
DT (GD)	65%	74%	66%	70%

Table C. 129. Local Models Average Accuracy for All sites

Models	S1 model accuracy	S2 model accuracy	S3 model accuracy	The best model
LR	**67%**	61%	56%	LR-S1
RF	63%	**64%**	62%	RF-S2
NB	53%	54%	**56%**	NB-S3
SVM (nonlinear)	66%	**68%**	67%	SVM (nonlinear)-S2
SVM (linear)	**67%**	56%	55%	SVM (linear)-S1
NN	**69%**	68%	66%	NN-S1
DT	69%	68%	**70%**	DT-S3

Table C. 130. Logistic Regression (LR) Model Evaluation Results After Updating Method

The best model	In S2 (LR-S12)	In S3 (LR-S13)
	After update the model	After update the model
LR-S1	72%	65%

Table C. 131. Logistic Regression (LR) Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average accuracy
LR-S12	66%	72%	64%	**67%**
LR-S13	66%	64%	65%	65%

Table C. 132. Logistic Regression (LR) Updated Model Evaluation Results After Updating Method

The best model	In S3 (LR-S123)
	After update the model
LR-S12	65%

Table C. 133. Random Forest (RF) Model Evaluation Results After Updating Method

The best model	In S1 (RF-S21)	In S3 (RF-S23)
	After update the model	After update the model
RF-S2	84%	83%

Table C. 134. Random Forest (RF) Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average accuracy
RF-S21	84%	66%	62%	**74%**
RF-S23	64%	58%	83%	66%

Table C. 135. Random Forest (RF) Updated Model Evaluation Results After Updating Method

The best model	In S3 (RF-S213)
	After update the model
RF-S21	87%

Table C. 136. Naïve Bayes (NB) Model Evaluation Results After Updating Method

The best model	In S1 (NB-S31)	In S2 (NB-S32)
	After update the model	After update the model
NB-S3	50%	53%

Table C. 137. Naïve Bayes (NB) Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average accuracy
NB-S31	50%	50%	63%	53%
NB-S32	52%	53%	62%	**54%**

Table C. 138. Naïve Bayes (NB) Updated Model Evaluation Results After Updating Method

The best model	In S1 (NB-S321)
	After update the model
NB-S32	50%

Table C. 139. Nonlinear Support Vector Machine (SVM) Model Evaluation Results After Updating Method

The best model	In S1(SVM (nonlinear)-S21)	In S3 (SVM nonlinear-S23)
	After update the model	After update the model
SVM nonlinear -S2	73%	69%

Table C. 140. Nonlinear Support Vector Machine (SVM) Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average accuracy
SVM nonlinear-S21	73%	66%	61%	**68%**
SVM nonlinear-S23	61%	63%	69%	63%

Table C. 141. Nonlinear Support Vector Machine (SVM) Updated Model Evaluation Results After Updating Method

The best model	In S3 (SVM nonlinear-S213)
	After update the model
SVM nonlinear-S21	68%

Table C. 142. Linear Support Vector Machine (SVM) Model Evaluation Results After Updating Method

The best model	In S2 (SVM linear)-S12)	In S3 (SVM linear)-S13)
	After update the model	After update the model
SVM (linear)-S1	59%	56%

Table C. 143. Linear Support Vector Machine (SVM) Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average accuracy
SVM linear-S12	58%	59%	64%	59%
SVM linear-S13	63%	58%	56%	**60%**

Table C. 144. Linear Support Vector Machine (SVM) Updated Model Evaluation Results After Updating Method

The best model	In S2 (SVM linear-S132)
	After update the model
SVM linear-S13	70%

Table C. 145. Neural Network (NN) Model Evaluation Results After Updating Method

The best model	In S2 (NN-S12)	In S3 (NN-S13)
	After update the model	After update the model
NN-S1	59%	70%

Table C. 146. Neural Network (NN) Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average accuracy
NN-S12	67%	59%	57%	63%
NN-S13	67%	70%	70%	**68%**

Table C. 147. Neural Network (NN) Updated Model Evaluation Results After Updating Method

The best model	In S2 (NN-S132)
	After update the model
NN-S13	68%

Table C. 148. Decision Tree (DT) Model Evaluation Results After Updating Method

The best model	In S1 (DT-S31)	In S2 (DT-S32)
	After update the model	After update the model
DT-S3	86%	85%

Table C. 149. Decision Tree (DT) Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average accuracy
DT-S31	86%	64%	62%	**75%**
DT-S32	69%	85%	66%	73%

Table C. 150. Decision Tree (DT) Updated Model Evaluation Results After Updating Method

The best model	In S2 (DT-S312)
	After update the model
DT-S31	92%

Table C. 151. Updated Models Evaluation Results in All sites

The best model	In S1	In S2	In S3	Average accuracy
LR-S123	66%	64%	65%	65%
RF-S213	64%	58%	87%	**68%**
NB-S321	50%	50%	63%	53%
SVM nonlinear-S213	65%	64%	68%	65%
SVM linear-S132	65%	70%	61%	66%
NN-S132	68%	68%	67%	**68%**
DT-S312	69%	92%	66%	**75%**

6. Spine Disease:

Table C. 152. Site 1 Local Models Evaluation

S1		In S2	In S3	Weighted average
Models	Accuracy	Accuracy	Accuracy	
LR	88%	90%	72%	82%
RF	97%	90%	54%	78%
NB	92%	78%	60%	76%
SVM (nonlinear)	89%	94%	64%	80%
SVM (linear)	90%	90%	69%	82%
NN	87%	76%	55%	72%
DT (GD)	92%	84%	54%	75%

Table C. 153. Site 2 Local Models Evaluation

S2		In S1	In S3	Weighted average
Models	Accuracy	Accuracy	Accuracy	
LR	84%	88%	71%	80%
RF	79%	90%	75%	82%
NB	87%	83%	78%	82%
SVM (nonlinear)	79%	82%	76%	79%
SVM (linear)	77%	86%	65%	76%
NN	84%	86%	72%	80%
DT (GD)	78%	82%	75%	78%

Table C. 154. Site 3 Local Models Evaluation

S3		In S1	In S2	Weighted average
Models	Accuracy	Accuracy	Accuracy	
LR	75%	87%	86%	82%
RF	76%	77%	92%	80%
NB	77%	62%	70%	70%
SVM (nonlinear)	75%	77%	92%	79%
SVM (linear)	74%	75%	90%	78%
NN	76%	68%	82%	74%
DT (GD)	80%	70%	92%	78%

Table C. 155. Local Models Average Accuracy for All sites

Models	S1 model accuracy	S2 model accuracy	S3 model accuracy	The best model
LR	**82%**	80%	**82%**	LR-S1 - LR-S3
RF	78%	**82%**	80%	RF-S2
NB	76%	**82%**	70%	NB-S2
SVM (nonlinear)	**80%**	79%	79%	SVM (nonlinear)-S1
SVM (linear)	**82%**	76%	78%	SVM (linear)-S1
NN	72%	**80%**	74%	NN-S2
DT	75%	**78%**	**78%**	DT-S2 - DT-S3

Table C. 156. Logistic Regression (LR) Model Evaluation Results After Updating Method

The best model	In S2 (LR-S12)	In S3 (LR-S13)
	After update the model	After update the model
LR-S1	78%	69%

Table C. 157. Logistic Regression (LR) Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average accuracy
LR-S12	77%	78%	63%	**72%**
LR-S13	63%	76%	69%	68%

Table C. 158. Logistic Regression (LR) Updated Model Evaluation Results After Updating Method

The best model	In S3 (LR-S123)
	After update the model
LR-S12	69%

Table C. 159. Logistic Regression (LR) Model Evaluation Results After Updating Method

The best model	In S1 (LR-S31)	In S2 (LR-S32)
	After update the model	After update the model
LR-S3	81%	98%

Table C. 160. Logistic Regression (LR) Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average accuracy
LR-S31	81%	82%	66%	75%
LR-S32	91%	98%	75%	**86%**

Table C. 161. Logistic Regression (LR) Updated Model Evaluation Results After Updating Method

The best model	In S1 (LR-S321)
	After update the model
LR-S32	81%

Table C. 162. Random Forest (RF) Model Evaluation Results After Updating Method

The best model	In S1 (RF-S21)	In S3 (RF-S23)
	After update the model	After update the model
RF-S2	99%	97%

Table C. 163. Random Forest (RF) Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average accuracy
RF-S21	99%	90%	54%	79%
RF-S23	72%	84%	97%	**84%**

Table C. 164. Random Forest (RF) Updated Model Evaluation Results After Updating Method

The best model	In S1 (RF-S231)
	After update the model
RF-S23	95%

Table C. 165. Naïve Bayes (NB) Model Evaluation Results After Updating Method

The best model	In S1 (NB-S21)	In S3 (NB-S23)
	After update the model	After update the model
NB-S2	89%	80%

Table C. 166. Naïve Bayes (NB) Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average accuracy
NB-S21	89%	66%	53%	70%
NB-S23	62%	70%	80%	**71%**

Table C. 167. Naïve Bayes (NB) Updated Model Evaluation Results After Updating Method

The best model	In S1 (NB-S231)
	After update the model
NB-S23	89%

Table C. 168. Nonlinear Support Vector Machine (SVM) Model Evaluation Results After Updating Method

The best model	In S2 (SVM (nonlinear)-S12)	In S3 (SVM nonlinear-S13)
	After update the model	After update the model
SVM nonlinear -S1	76%	69%

Table C. 169. Nonlinear Support Vector Machine (SVM) Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average accuracy
SVM nonlinear-S12	83%	76%	59%	**72%**
SVM nonlinear-S13	54%	80%	69%	65%

Table C. 170. Nonlinear Support Vector Machine (SVM) Updated Model Evaluation Results After Updating Method

The best model	In S3 (SVM nonlinear-S123)
	After update the model
SVM nonlinear-S12	69%

Table C. 171. Linear Support Vector Machine (SVM) Model Evaluation Results After Updating Method

The best model	In S2 (SVM linear)-S12)	In S3 (SVM linear)-S13)
	After update the model	After update the model
SVM (linear)-S1	88%	73%

Table C. 172. Linear Support Vector Machine (SVM) Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average accuracy
SVM linear-S12	86%	88%	69%	**80%**
SVM linear-S13	73%	80%	73%	74%

Table C. 173. Linear Support Vector Machine (SVM) Updated Model Evaluation Results After Updating Method

The best model	In S3 (SVM linear-S123)
	After update the model
SVM linear-S12	76%

Table C. 174. Neural Network (NN) Model Evaluation Results After Updating Method

The best model	In S1 (NN-S21)	In S3 (NN-S23)
	After update the model	After update the model
NN-S2	86%	73%

Table C. 175. Neural Network (NN) Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average accuracy
NN-S21	86%	98%	72%	**83%**
NN-S23	57%	76%	73%	67%

Table C. 176. Neural Network (NN) Updated Model Evaluation Results After Updating Method

The best model	In S3 (NN-S213)
	After update the model
NN-S21	72%

Table C. 177. Decision Tree (DT) Model Evaluation Results After Updating Method

The best model	In S1 (DT-S21)	In S3 (DT-S23)
	After update the model	After update the model
DT-S2	97%	92%

Table C. 178. Decision Tree (DT) Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average accuracy
DT-S21	97%	82%	54%	77%
DT-S23	72%	86%	92%	**83%**

Table C. 179. Decision Tree (DT) Updated Model Evaluation Results After Updating Method

The best model	In S1 (DT-S231)
	After update the model
DT-S23	92%

Table C. 180. Decision Tree (DT) Model Evaluation Results After Updating Method

The best model	In S1 (DT-S31)	In S2 (DT-S32)
	After update the model	After update the model
DT-S3	89%	99%

Table C. 181. Decision Tree (DT) Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average accuracy
DT-S31	89%	82%	54%	74%
DT-S32	81%	99%	75%	**82%**

Table C. 182. Decision Tree (DT) Updated Model Evaluation Results After Updating Method

The best model	In S1 (DT-S321)
	After update the model
DT-S32	92%

Table C. 183. Updated Models Evaluation Results in All sites

The best model	In S1	In S2	In S3	Average accuracy
LR-S123	63%	76%	69%	68%
LR-S321	81%	82%	66%	75%
RF-S231	99%	90%	54%	**79%**
NB-S231	89%	66%	53%	70%
SVM nonlinear-S123	80%	80%	69%	**76%**
SVM linear-S123	58%	78%	76%	69%
NN-S213	58%	78%	76%	69%
DT-S231	86%	98%	72%	**83%**
DT-S321	92%	82%	54%	75%

7. Breast Cancer Wisconsin (Original):

Table C. 184. Site 1 Local Models Evaluation

S1		In S2	In S3	Weighted average
Models	Accuracy	Accuracy	Accuracy	
LR	96%	95%	93%	95%
RF	95%	93%	97%	94%
NB	96%	95%	96%	95%
SVM (nonlinear)	96%	96%	96%	96%
SVM (linear)	93%	95%	96%	95%
NN	96%	95%	98%	96%
DT (GD)	96%	95%	96%	95%

Table C. 185. Site 2 Local Models Evaluation

S2		In S1	In S3	Weighted average
Models	Accuracy	Accuracy	Accuracy	
LR	93%	97%	98%	95%
RF	96%	92%	96%	95%
NB	96%	95%	97%	96%
SVM (nonlinear)	94%	97%	92%	94%
SVM (linear)	93%	93%	97%	94%
NN	95%	97%	99%	96%
DT (GD)	95%	91%	98%	94%

Table C. 186. Site 3 Local Models Evaluation

S3		In S1	In S2	Weighted average
Models	Accuracy	Accuracy	Accuracy	
LR	99%	90%	93%	93%
RF	97%	86%	87%	89%
NB	96%	92%	95%	94%

SVM (nonlinear)	99%	88%	96%	94%
SVM (linear)	99%	90%	95%	94%
NN	99%	89%	94%	93%
DT (GD)	97%	78%	84%	85%

Table C. 187. Local Models Average Accuracy for All sites

Models	S1 model accuracy	S2 model accuracy	S3 model accuracy	The best model
LR	**95%**	**95%**	93%	LR-S1 - LR-S2
RF	94%	**95%**	89%	RF-S2
NB	95%	**96%**	94%	NB-S2
SVM (nonlinear)	**96%**	94%	94%	SVM (nonlinear)-S1
SVM (linear)	**95%**	94%	94%	SVM (linear)-S1
NN	**96%**	**96%**	93%	NN-S1 - NN-S2
DT	**95%**	94%	85%	DT-S1

Table C. 188. Logistic Regression (LR) Model Evaluation Results After Updating Method

The best model	In S2 (LR-S12)	In S3 (LR-S13)
	After update the model	After update the model
LR-S1	88%	99%

Table C. 189. Logistic Regression (LR) Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average accuracy
LR-S12	79%	88%	90%	86%
LR-S13	94%	95%	99%	**95%**

Table C. 190. Logistic Regression (LR) Updated Model Evaluation Results After Updating Method

The best model	In S2 (LR-S132)
	After update the model
LR-S13	88%

Table C. 191. Logistic Regression (LR) Model Evaluation Results After Updating Method

The best model	In S1 (LR-S21)	In S3 (LR-S23)
	After update the model	After update the model
LR-S2	97%	97%

Table C. 192. Logistic Regression (LR) Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average accuracy
LR-S21	97%	96%	93%	95%
LR-S23	96%	96%	97%	**96%**

Table C. 193. Logistic Regression (LR) Updated Model Evaluation Results After Updating Method

The best model	In S1 (LR-S231)
	After update the model
LR-S23	97%

Table C. 194. Random Forest (RF) Model Evaluation Results After Updating Method

The best model	In S1 (RF-S21)	In S3 (RF-S23)
	After update the model	After update the model
RF-S2	99%	99%

Table C. 195. Random Forest (RF) Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average accuracy
RF-S21	99%	94%	69%	**90%**
RF-S23	86%	86%	99%	89%

Table C. 196. Random Forest (RF) Updated Model Evaluation Results After Updating Method

The best model	In S3 (RF-S213)
	After update the model
RF-S21	94%

Table C. 197. Naïve Bayes (NB) Model Evaluation Results After Updating Method

The best model	In S1 (NB-S21)	In S3 (NB-S23)
	After update the model	After update the model
NB-S2	97%	97%

Table C. 198. Naïve Bayes (NB) Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average accuracy
NB-S21	97%	95%	96%	**96%**
NB-S23	92%	95%	97%	94%

Table C. 199. Naïve Bayes (NB) Updated Model Evaluation Results After Updating Method

The best model	In S3 (NB-S213)
	After update the model
NB-S21	97%

Table C. 200. Nonlinear Support Vector Machine (SVM) Model Evaluation Results After Updating Method

The best model	In S2 (SVM (nonlinear)-S12)	In S3 (SVM nonlinear-S13)
	After update the model	After update the model
SVM nonlinear -S1	95%	98%

Table C. 201. Nonlinear Support Vector Machine (SVM) Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average accuracy
SVM nonlinear-S12	94%	95%	94%	95%
SVM nonlinear-S13	96%	96%	98%	**96%**

Table C. 202. Nonlinear Support Vector Machine (SVM) Updated Model Evaluation Results After Updating Method

The best model	In S2 (SVM nonlinear-S132)
	After update the model
SVM nonlinear-S13	97%

Table C. 203. Linear Support Vector Machine (SVM) Model Evaluation Results After Updating Method

The best model	In S2 (SVM linear)-S12)	In S3 (SVM linear)-S13)
	After update the model	After update the model
SVM (linear)-S1	95%	99%

Table C. 204. Linear Support Vector Machine (SVM) Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average accuracy
SVM linear-S12	94%	95%	98%	**95%**
SVM linear-S13	93%	95%	99%	94%

Table C. 205. Linear Support Vector Machine (SVM) Updated Model Evaluation Results After Updating Method

The best model	In S3 (SVM linear-S123)
	After update the model
SVM linear-S12	99%

Table C. 206. Neural Network (NN) Model Evaluation Results After Updating Method

The best model	In S2 (NN-S12)	In S3 (NN-S13)
	After update the model	After update the model
NN-S1	76%	98%

Table C. 207. Neural Network (NN) Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average accuracy
NN-S12	74%	76%	87%	**78%**
NN-S13	56%	60%	98%	66%

Table C. 208. Neural Network (NN) Updated Model Evaluation Results After Updating Method

The best model	In S3 (NN-S123)
	After update the model
NN-S12	88%

Table C. 209. Neural Network (NN) Model Evaluation Results After Updating Method

The best model	In S1 (NN-S21)	In S3 (NN-S23)
	After update the model	After update the model
NN-S2	97%	94%

Table C. 210. Neural Network (NN) Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average accuracy
NN-S21	97%	95%	99%	**96%**
NN-S23	75%	78%	94%	80%

Table C. 211. Neural Network (NN) Updated Model Evaluation Results After Updating Method

The best model	In S3 (NN-S213)
	After update the model
NN-S21	95%

Table C. 212. Decision Tree (DT) Model Evaluation Results After Updating Method

The best model	In S2 (DT-S12)	In S3 (DT-S13)
	After update the model	After update the model
DT-S1	99%	99%

Table C. 213. Decision Tree (DT) Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average accuracy
DT-S12	94%	99%	97%	**97%**
DT-S13	85%	77%	99%	84%

Table C. 214. Decision Tree (DT) Updated Model Evaluation Results After Updating Method

The best model	In S3 (DT-S123)
	After update the model
DT-S12	99%

Table C. 215. Updated Models Evaluation Results in All sites

The best model	In S1	In S2	In S3	Average accuracy
LR-S132	79%	88%	90%	86%
LR-S231	97%	96%	93%	**96%**
RF-S213	86%	84%	94%	87%
NB-S213	92%	95%	97%	94%
SVM nonlinear-S132	95%	97%	89%	**95%**
SVM linear-S123	93%	93%	99%	94%
NN-S123	57%	62%	75%	63%
NN-S213	95%	94%	95%	**95%**
DT-S123	85%	77%	99%	84%

8. Cardiovascular Disease:

Table C. 216. Site 1 Local Models Evaluation

S1		In S2	In S3	Weighted average
Models	Accuracy	Accuracy	Accuracy	
LR	71%	72%	71%	71%
RF	68%	69%	68%	68%
NB	71%	71%	73%	72%
SVM (nonlinear)	72%	72%	71%	71%
SVM (linear)	72%	72%	72%	72%
NN	73%	73%	73%	73%
DT (GD)	72%	73%	72%	72%

Table C. 217. Site 2 Local Models Evaluation

S2		In S1	In S3	Weighted average
Models	Accuracy	Accuracy	Accuracy	
LR	72%	72%	72%	72%
RF	70%	69%	68%	69%
NB	71%	71%	71%	71%
SVM (nonlinear)	72%	72%	72%	72%
SVM (linear)	72%	72%	71%	71%
NN	73%	72%	73%	72%
DT (GD)	73%	73%	73%	73%

Table C. 218. Site 3 Local Models Evaluation

S3		In S1	In S2	Weighted average
Models	Accuracy	Accuracy	Accuracy	
LR	72%	71%	71%	71%
RF	69%	69%	69%	69%
NB	71%	71%	71%	71%
SVM (nonlinear)	72%	69%	69%	69%
SVM (linear)	71%	71%	72%	71%
NN	73%	72%	73%	72%
DT (GD)	72%	72%	72%	72%

Table C. 219. Local Models Average Accuracy for All sites

Models	S1 model accuracy	S2 model accuracy	S3 model accuracy	The best model
LR	71%	**72%**	71%	LR-S2
RF	68%	**69%**	**69%**	RF-S2 - RF-S3
NB	**72%**	71%	71%	NB-S1
SVM (nonlinear)	71%	**72%**	69%	SVM (nonlinear)-S2
SVM (linear)	**72%**	71%	71%	SVM (linear)-S1
NN	**73%**	72%	72%	NN-S1
DT	72%	**73%**	72%	DT-S2

Table C. 220. Logistic Regression (LR) Model Evaluation Results After Updating Method

The best model	In S1 (LR-S21)	In S3 (LR-S23)
	After update the model	After update the model
LR-S2	70%	69%

Table C. 221. Logistic Regression (LR) Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average accuracy
LR-S21	70%	66%	66%	**67%**
LR-S23	65%	65%	69%	66%

Table C. 222. Logistic Regression (LR) Updated Model Evaluation Results After Updating Method

The best model	In S3 (LR-S213)
	After update the model
LR-S21	69%

Table C. 223. Random Forest (RF) Model Evaluation Results After Updating Method

The best model	In S1 (RF-S21)	In S3 (RF-S23)
	After update the model	After update the model
RF-S2	81%	79%

Table C. 224. Random Forest (RF) Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average accuracy
RF-S21	81%	69%	69%	**73%**
RF-S23	69%	69%	79%	72%

Table C. 225. Random Forest (RF) Updated Model Evaluation Results After Updating Method

The best model	In S3 (RF-S231)
	After update the model
RF-S21	81%

Table C. 226. Random Forest (RF) Model Evaluation Results After Updating Method

The best model	In S1 (RF-S31)	In S2 (RF-S32)
	After update the model	After update the model
RF-S3	81%	81%

Table C. 227. Random Forest (RF) Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average accuracy
RF-S31	81%	69%	69%	73%
RF-S32	69%	81%	69%	**74%**

Table C. 228. Random Forest (RF) Updated Model Evaluation Results After Updating Method

The best model	In S1 (RF-S321)
	After update the model
RF-S32	80%

Table C. 229. Naïve Bayes (NB) Model Evaluation Results After Updating Method

The best model	In S2 (NB-S12)	In S3 (NB-S13)
	After update the model	After update the model
NB-S1	71%	71%

Table C. 230. Naïve Bayes (NB) Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average accuracy
NB-S12	71%	71%	73%	**72%**
NB-S13	71%	71%	71%	71%

Table C. 231. Naïve Bayes (NB) Updated Model Evaluation Results After Updating Method

The best model	In S3 (NB-S123)
	After update the model
NB-S12	71%

Table C. 232. Nonlinear Support Vector Machine (SVM) Model Evaluation Results After Updating Method

The best model	In S1(SVM (nonlinear)-S21)	In S3 (SVM nonlinear-S23)
	After update the model	After update the model
SVM nonlinear -S2	66%	70%

Table C. 233. Nonlinear Support Vector Machine (SVM) Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average accuracy
SVM nonlinear-S21	66%	66%	66%	66%
SVM nonlinear-S23	67%	68%	70%	**68%**

Table C. 234. Nonlinear Support Vector Machine (SVM) Updated Model Evaluation Results After Updating Method

The best model	In S1 (SVM nonlinear-S231)
	After update the model
SVM nonlinear-S23	70%

Table C. 235. Linear Support Vector Machine (SVM) Model Evaluation Results After Updating Method

The best model	In S2 (SVM linear)-S12)	In S3 (SVM linear)-S13)
	After update the model	After update the model
SVM (linear)-S1	69%	70%

Table C. 236. Linear Support Vector Machine (SVM) Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average accuracy
SVM linear-S12	68%	69%	68%	68%
SVM linear-S13	70%	70%	70%	**70%**

Table C. 237. Linear Support Vector Machine (SVM) Updated Model Evaluation Results After Updating Method

The best model	In S2 (SVM linear-S132)
	After update the model
SVM linear-S13	69%

Table C. 238. Neural Network (NN) Model Evaluation Results After Updating Method

The best model	In S2 (NN-S12)	In S3 (NN-S13)
	After update the model	After update the model
NN-S1	72%	70%

Table C. 239. Neural Network (NN) Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average accuracy
NN-S12	72%	72%	73%	**72%**
NN-S13	70%	70%	70%	70%

Table C. 240. Neural Network (NN) Updated Model Evaluation Results After Updating Method

The best model	In S3 (NN-S123)
	After update the model
NN-S12	73%

Table C. 241. Decision Tree (DT) Model Evaluation Results After Updating Method

The best model	In S1 (DT-S21)	In S3 (DT-S21)
	After update the model	After update the model
DT-S2	74%	74%

Table C. 242. Decision Tree (DT) Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average accuracy
DT-S21	74%	74%	73%	**74%**
DT-S23	73%	73%	74%	73%

Table C. 243. Decision Tree (DT) Updated Model Evaluation Results After Updating Method

The best model	In S3 (DT-S213)
	After update the model
DT-S213	74%

Table C. 244. Updated Models Evaluation Results in All sites

The best model	In S1	In S2	In S3	Average accuracy
LR-S213	69%	69%	69%	69%
RF-S213	69%	69%	81%	**73%**
RF-S321	80%	69%	69%	**72%**
NB-S123	71%	71%	71%	71%
SVM nonlinear-S231	70%	70%	70%	70%
SVM linear-S132	69%	69%	68%	69%
NN-S123	70%	72%	73%	71%
DT-S213	73%	73%	74%	**73%**

II. Non-Randomly Partitioned Data:

1. Diabetes

Table C. 245. Site 1 Local Models Evaluation

S1		In S2	In S3	Weighted average
Models	Accuracy	Accuracy	Accuracy	
LR	83%	58%	48%	68%
RF	83%	60%	53%	70%
NB	84%	62%	60%	73%
SVM (nonlinear)	82%	66%	60%	73%
SVM (linear)	78%	59%	47%	66%
NN	84%	62%	53%	71%
DT (GD)	84%	62%	55%	72%

Table C. 246. Site 2 Local Models Evaluation

S2		In S1	In S3	Weighted average
Models	Accuracy	Accuracy	Accuracy	
LR	64%	70%	72%	69%
RF	66%	73%	60%	68%
NB	65%	73%	69%	70%
SVM (nonlinear)	59%	53%	61%	56%
SVM (linear)	61%	65%	69%	65%
NN	65%	70%	74%	69%
DT (GD)	67%	72%	72%	70%

Table C. 247. Site 3 Local Models Evaluation

S3		In S1	In S2	Weighted average
Models	Accuracy	Accuracy	Accuracy	
LR	52%	42%	59%	49%
RF	65%	61%	61%	62%
NB	65%	61%	63%	62%
SVM (nonlinear)	68%	50%	56%	55%
SVM (linear)	57%	60%	64%	61%
NN	64%	61%	65%	63%
DT (GD)	62%	56%	64%	60%

Table C. 248. Local Models Average Accuracy for All sites

Models	S1 model accuracy	S2 model accuracy	S3 model accuracy	The best model
LR	68%	**69%**	49%	LR-S2
RF	**70%**	68%	62%	RF-S1
NB	**73%**	70%	62%	NB-S1
SVM (nonlinear)	**73%**	56%	55%	SVM (nonlinear)-S1
SVM (linear)	**66%**	65%	61%	SVM (linear)-S1
NN	**71%**	69%	63%	NN-S1
DT	**72%**	70%	60%	DT-S1

Table C. 249. Logistic Regression (LR) Model Evaluation Results After Updating Method

The best model	In S1 (LR-S21)	In S3 (LR-S23)
	After update the model	After update the model
LR-S2	86%	62%

Table C. 250. Logistic Regression (LR) Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average accuracy
LR-S21	86%	64%	59%	**74%**
LR-S23	52%	57%	62%	55%

Table C. 251. Logistic Regression (LR) Updated Model Evaluation Results After Updating Method

The best model	In S3 (LR-S213)
	After update the model
LR-S21	60%

Table C. 252. Random Forest (RF) Model Evaluation Results After Updating Method

The best model	In S2 (RF-S12)	In S3 (RF-S13)
	After update the model	After update the model
RF-S1	80%	68%

Table C. 253. Random Forest (RF) Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average accuracy
RF-S12	70%	80%	66%	**72%**
RF-S13	60%	59%	68%	61%

Table C. 254. Random Forest (RF) Updated Model Evaluation Results After Updating Method

The best model	In S3 (RF-S123)
	After update the model
RF-S12	74%

Table C. 255. Naïve Bayes (NB) Model Evaluation Results After Updating Method

The best model	In S2 (NB-S12)	In S3 (NB-S13)
	After update the model	After update the model
NB-S1	70%	68%

Table C. 256. Naïve Bayes (NB) Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average accuracy
NB-S12	73%	70%	69%	**71%**
NB-S13	61%	63%	68%	63%

Table C. 257. Naïve Bayes (NB) Updated Model Evaluation Results After Updating Method

The best model	In S3 (NB-S123)
	After update the model
NB-S12	68%

Table C. 258. Nonlinear Support Vector Machine (SVM) Model Evaluation Results After Updating Method

The best model	In S2 (SVM (nonlinear)-S12)	In S3 (SVM nonlinear-S13)
	After update the model	After update the model
SVM nonlinear -S1	66%	53%

Table C. 259. Nonlinear Support Vector Machine (SVM) Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average accuracy
SVM nonlinear-S12	51%	66%	75%	**60%**
SVM nonlinear-S13	58%	55%	53%	56%

Table C. 260. Nonlinear Support Vector Machine (SVM) Updated Model Evaluation Results After Updating Method

The best model	In S3 (SVM nonlinear-S123)
	After update the model
SVM nonlinear-S12	71%

Table C. 261. Linear Support Vector Machine (SVM) Model Evaluation Results After Updating Method

The best model	In S2 (SVM linear)-S12)	In S3 (SVM linear)-S13)
	After update the model	After update the model
SVM (linear)-S1	66%	69%

Table C. 262. Linear Support Vector Machine (SVM) Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average accuracy
SVM linear-S12	67%	66%	76%	**68%**
SVM linear-S13	54%	63%	69%	60%

Table C. 263. Linear Support Vector Machine (SVM) Updated Model Evaluation Results After Updating Method

The best model	In S3 (SVM linear-S123)
	After update the model
SVM linear-S12	67%

Table C. 264. Neural Network (NN) Model Evaluation Results After Updating Method

The best model	In S2 (NN-S12)	In S3 (NN-S13)
	After update the model	After update the model
NN-S1	65%	59%

Table C. 265. Neural Network (NN) Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average accuracy
NN-S12	74%	65%	59%	**68%**
NN-S13	52%	64%	59%	57%

Table C. 266. Neural Network (NN) Updated Model Evaluation Results After Updating Method

The best model	In S3 (NN-S123)
	After update the model
NN-S12	59%

Table C. 267. Decision Tree (DT) Model Evaluation Results After Updating Method

The best model	In S2 (DT-S12)	In S3 (DT-S13)
	After update the model	After update the model
DT-S1	82%	68%

Table C. 268. Decision Tree (DT) Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average accuracy
DT-S12	66%	82%	64%	**70%**
DT-S13	56%	64%	68%	61%

Table C. 269. Decision Tree (DT) Updated Model Evaluation Results After Updating Method

The best model	In S3 (DT-S123)
	After update the model
DT-S12	74%

Table C. 270. Updated Models Evaluation Results in All sites

The best model	In S1	In S2	In S3	Average accuracy
LR-S213	50%	54%	60%	53%
RF-S123	57%	60%	74%	61%
NB-S123	61%	63%	68%	**63%**
SVM nonlinear-S123	55%	65%	71%	61%
SVM linear-S123	53%	66%	67%	60%
NN-S123	74%	65%	59%	**68%**
DT-S123	57%	64%	74%	**62%**

2. Heart Disease:

Table C. 271. Site 1 Local Models Evaluation

S1		In S2	In S3	Weighted average
Models	Accuracy	Accuracy	Accuracy	
LR	90%	67%	62%	70%
RF	83%	78%	72%	77%
NB	76%	51%	44%	54%
SVM (nonlinear)	83%	72%	66%	72%
SVM (linear)	90%	66%	66%	71%
NN	85%	72%	65%	72%
DT (GD)	86%	72%	57%	70%

Table C. 272. Site 2 Local Models Evaluation

S2		In S1	In S3	Weighted average
Models	Accuracy	Accuracy	Accuracy	
LR	72%	70%	76%	73%
RF	73%	80%	79%	76%
NB	76%	80%	84%	79%
SVM (nonlinear)	71%	75%	81%	75%
SVM (linear)	77%	68%	75%	75%
NN	83%	73%	81%	80%
DT (GD)	78%	89%	84%	82%

Table C. 273. Site 3 Local Models Evaluation

S3		In S1	In S2	Weighted average
Models	Accuracy	Accuracy	Accuracy	
LR	74%	61%	72%	70%
RF	77%	73%	71%	73%
NB	81%	73%	78%	78%
SVM (nonlinear)	78%	66%	71%	72%
SVM (linear)	77%	73%	75%	75%
NN	77%	68%	75%	74%
DT (GD)	78%	80%	77%	78%

Table C. 274. Local Models Average Accuracy for All sites

Models	S1 model accuracy	S2 model accuracy	S3 model accuracy	The best model
LR	70%	**73%**	70%	LR-S2
RF	**77%**	76%	73%	RF-S1
NB	54%	**79%**	78%	NB-S2
SVM (nonlinear)	72%	**75%**	72%	SVM (nonlinear)-S2
SVM (linear)	71%	**75%**	**75%**	SVM (linear)-S2 and S3
NN	72%	**80%**	74%	NN-S2
DT	70%	**82%**	78%	DT-S2

Table C. 275. Logistic Regression (LR) Model Evaluation Results After Updating Method

The best model	In S1 (LR-S21)	In S3 (LR-S23)
	After update the model	After update the model
LR-S2	91%	81%

Table C. 276. Logistic Regression (LR) Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average accuracy
LR-S21	91%	71%	79%	**77%**
LR-S23	64%	75%	81%	75%

Table C. 277. Logistic Regression (LR) Updated Model Evaluation Results After Updating Method

The best model	In S3 (LR-S213)
	After update the model
LR-S21	81%

Table C. 278. Random Forest (RF) Model Evaluation Results After Updating Method

The best model	In S2 (RF-S12)	In S3 (RF-S13)
	After update the model	After update the model
RF-S1	92%	91%

Table C. 279. Random Forest (RF) Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average accuracy
RF-S12	70%	92%	69%	**81%**
RF-S13	68%	78%	91%	80%

Table C. 280. Random Forest (RF) Updated Model Evaluation Results After Updating Method

The best model	In S3 (RF-S123)
	After update the model
RF-S12	64%

Table C. 281. Naïve Bayes (NB) Model Evaluation Results After Updating Method

The best model	In S1 (NB-S21)	In S3 (NB-S23)
	After update the model	After update the model
NB-S2	86%	87%

Table C. 282. Naïve Bayes (NB) Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average accuracy
NB-S21	86%	51%	44%	56%
NB-S23	73%	78%	87%	**80%**

Table C. 283. Naïve Bayes (NB) Updated Model Evaluation Results After Updating Method

The best model	In S1 (NB-S231)
	After update the model
NB-S23	86%

Table C. 284. Nonlinear Support Vector Machine (SVM) Model Evaluation Results After Updating Method

The best model	In S1(SVM (nonlinear)-S21)	In S3 (SVM nonlinear-S23)
	After update the model	After update the model
SVM nonlinear -S2	91%	84%

Table C. 285. Nonlinear Support Vector Machine (SVM) Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average accuracy
SVM nonlinear-S21	91%	72%	68%	75%
SVM nonlinear-S23	68%	79%	84%	**78%**

Table C. 286. Nonlinear Support Vector Machine (SVM) Updated Model Evaluation Results After Updating Method

The best model	In S1 (SVM nonlinear-S231)
	After update the model
SVM nonlinear-S23	70%

Table C. 287. Linear Support Vector Machine (SVM) Model Evaluation Results After Updating Method

The best model	In S1 (SVM linear)-S21)	In S3 (SVM linear)-S23)
	After update the model	After update the model
SVM (linear)-S2	85%	72%

Table C. 288. Linear Support Vector Machine (SVM) Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average accuracy
SVM linear-S21	85%	74%	63%	**73%**
SVM linear-S23	64%	71%	72%	70%

Table C. 289. Linear Support Vector Machine (SVM) Updated Model Evaluation Results After Updating Method

The best model	In S3 (SVM linear-S213)
	After update the model
SVM linear-S21	72%

Table C. 290. Linear Support Vector Machine (SVM) Model Evaluation Results After Updating Method

The best model	In S1(SVM linear-S31)	In S2 (SVM linear-S32)
	After update the model	After update the model
SVM linear-S3	91%	77%

Table C. 291. Linear Support Vector Machine (SVM) Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average accuracy
SVM linear-S31	91%	73%	75%	77%
SVM linear-S32	91%	77%	75%	**79%**

Table C. 292. Linear Support Vector Machine (SVM) Updated Model Evaluation Results After Updating Method

The best model	In S1 (SVM linear-S321)
	After update the model
SVM linear-S32	86%

Table C. 293. Neural Network (NN) Model Evaluation Results After Updating Method

The best model	In S1 (NN-S21)	In S3 (NN-S23)
	After update the model	After update the model
NN-S2	73%	72%

Table C. 294. Neural Network (NN) Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average accuracy
NN-S21	73%	90%	81%	**84%**
NN-S23	61%	68%	72%	68%

Table C. 295. Neural Network (NN) Updated Model Evaluation Results After Updating Method

The best model	In S3 (NN-S213)
	After update the model
NN-S21	81%

Table C. 296. Decision Tree (DT) Model Evaluation Results After Updating Method

The best model	In S1 (DT-S21)	In S3 (DT-S23)
	After update the model	After update the model
DT-S2	97%	94%

Table C. 297. Decision Tree (DT) Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average accuracy
DT-S21	97%	71%	59%	73%
DT-S23	64%	68%	94%	**75%**

Table C. 298. Decision Tree (DT) Updated Model Evaluation Results After Updating Method

The best model	In S1 (DT-S231)
	After update the model
DT-S23	73%

Table C. 299. Updated Models Evaluation Results in All sites

The best model	In S1	In S2	In S3	Average accuracy
LR-S213	64%	75%	81%	**75%**
RF-S123	64%	58%	64%	61%
NB-S231	86%	51%	44%	56%
SVM nonlinear-S231	70%	59%	53%	59%
SVM linear-S213	64%	71%	72%	70%
SVM linear-S321	86%	70%	75%	**75%**
NN-S213	73%	90%	81%	**84%**
DT-S231	73%	71%	59%	68%

3. Liver Disease:

Table C. 300. Site 1 Local Models Evaluation

S1		In S2	In S3	Weighted average
Models	Accuracy	Accuracy	Accuracy	
LR	64%	58%	57%	59%
RF	66%	59%	66%	62%
NB	57%	50%	56%	53%
SVM (nonlinear)	59%	59%	59%	59%
SVM (linear)	65%	68%	71%	68%
NN	52%	62%	62%	60%
DT (GD)	63%	55%	64%	58%

Table C. 301. Site 2 Local Models Evaluation

S2		In S1	In S3	Weighted average
Models	Accuracy	Accuracy	Accuracy	
LR	65%	54%	73%	64%
RF	67%	56%	66%	65%
NB	51%	65%	52%	54%
SVM (nonlinear)	68%	57%	75%	67%
SVM (linear)	66%	54%	71%	65%
NN	76%	54%	74%	71%
DT (GD)	75%	54%	74%	71%

Table C. 302. Site 3 Local Models Evaluation

S3		In S1	In S2	Weighted average
Models	Accuracy	Accuracy	Accuracy	
LR	71%	56%	68%	66%
RF	69%	67%	68%	68%
NB	62%	57%	62%	61%
SVM (nonlinear)	69%	55%	69%	66%
SVM (linear)	66%	53%	69%	65%
NN	73%	54%	76%	71%
DT (GD)	73%	69%	61%	65%

Table C. 303. Local Models Average Accuracy for All sites

Models	S1 model accuracy	S2 model accuracy	S3 model accuracy	The best model
LR	59%	64%	**66%**	LR-S3
RF	62%	65%	**68%**	RF-S3
NB	53%	54%	**61%**	NB-S3
SVM (nonlinear)	59%	**67%**	66%	SVM (nonlinear)-S2
SVM (linear)	**68%**	65%	65%	SVM (linear)-S1
NN	60%	**71%**	**71%**	NN-S2 - NN-S3
DT	58%	**71%**	65%	DT-S2

Table C. 304. Logistic Regression (LR) Model Evaluation Results After Updating Method

The best model	In S1 (LR-S31)	In S2 (LR-S32)
	After update the model	After update the model
LR-S3	63%	73%

Table C. 305. Logistic Regression (LR) Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average accuracy
LR-S31	63%	65%	68%	65%
LR-S32	56%	73%	75%	**70%**

Table C. 306. Logistic Regression (LR) Updated Model Evaluation Results After Updating Method

The best model	In S1 (LR-S321)
	After update the model
LR-S32	63%

Table C. 307. Random Forest (RF) Model Evaluation Results After Updating Method

The best model	In S1 (RF-S31)	In S2 (RF-S32)
	After update the model	After update the model
RF-S3	75%	72%

Table C. 308. Random Forest (RF) Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average accuracy
RF-S31	75%	59%	62%	63%
RF-S32	55%	72%	74%	**69%**

Table C. 309. Random Forest (RF) Updated Model Evaluation Results After Updating Method

The best model	In S1 (RF-S321)
	After update the model
RF-S32	69%

Table C. 310. Naïve Bayes (NB) Model Evaluation Results After Updating Method

The best model	In S1 (NB-S31)	In S2 (NB-S32)
	After update the model	After update the model
NB-S3	59%	52%

Table C. 311. Naïve Bayes (NB) Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average accuracy
NB-S31	59%	50%	56%	53%
NB-S32	65%	52%	52%	**55%**

Table C. 312. Naïve Bayes (NB) Updated Model Evaluation Results After Updating Method

The best model	In S1(NB-S321)
	After update the model
NB-S32	59%

Table C. 313. Nonlinear Support Vector Machine (SVM) Model Evaluation Results After Updating Method

The best model	In S1(SVM (nonlinear)-S21)	In S3 (SVM nonlinear-S23)
	After update the model	After update the model
SVM nonlinear -S2	63%	75%

Table C. 314. Nonlinear Support Vector Machine (SVM) Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average accuracy
SVM nonlinear-S21	63%	59%	56%	59%
SVM nonlinear-S23	57%	66%	75%	**66%**

Table C. 315. Nonlinear Support Vector Machine (SVM) Updated Model Evaluation Results After Updating Method

The best model	In S1 (SVM nonlinear-S231)
	After update the model
SVM nonlinear-S23	62%

Table C. 316. Linear Support Vector Machine (SVM) Model Evaluation Results After Updating Method

The best model	In S2 (SVM linear)-S12)	In S3 (SVM linear)-S13)
	After update the model	After update the model
SVM (linear)-S1	54%	72%

Table C. 317. Linear Support Vector Machine (SVM) Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average accuracy
SVM linear-S12	55%	54%	53%	54%
SVM linear-S13	59%	71%	72%	**69%**

Table C. 318. Linear Support Vector Machine (SVM) Updated Model Evaluation Results After Updating Method

The best model	In S2 (SVM linear-S132)
	After update the model
SVM linear-S13	71%

Table C. 319. Neural Network (NN) Model Evaluation Results After Updating Method

The best model	In S1 (NN-S21)	In S3 (NN-S13)
	After update the model	After update the model
NN-S2	55%	74%

Table C. 320. Neural Network (NN) Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average accuracy
NN-S21	55%	67%	73%	66%
NN-S23	57%	76%	74%	**72%**

Table C. 321. Neural Network (NN) Updated Model Evaluation Results After Updating Method

The best model	In S1 (NN-S231)
	After update the model
NN-S23	57%

Table C. 322. Neural Network (NN) Model Evaluation Results After Updating Method

The best model	In S1 (NN-S31)	In S2 (NN-S32)
	After update the model	After update the model
NN-S3	54%	71%

Table C. 323. Neural Network (NN) Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average accuracy
NN-S31	54%	67%	63%	**64%**
NN-S32	53%	71%	51%	63%

Table C. 324. Neural Network (NN) Updated Model Evaluation Results After Updating Method

The best model	In S2 (NN-S312)
	After update the model
NN-S31	76%

Table C. 325. Decision Tree (DT) Model Evaluation Results After Updating Method

The best model	In S1 (DT-S21)	In S3 (DT-S23)
	After update the model	After update the model
DT-S2	71%	79%

Table C. 326. Decision Tree (DT) Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average accuracy
DT-S21	71%	55%	62%	60%
DT-S23	62%	64%	79%	**67%**

Table C. 327. Decision Tree (DT) Updated Model Evaluation Results After Updating Method

The best model	In S1 (DT-S231)
	After update the model
DT-S23	72%

Table C. 328. Updated Models Evaluation Results in All sites

The best model	In S1	In S2	In S3	Average accuracy
LR-S321	63%	65%	68%	65%
RF-S321	69%	59%	62%	62%
NB-S321	59%	50%	56%	53%
SVM nonlinear-S231	62%	59%	56%	59%
SVM linear-S132	59%	71%	72%	**69%**

NN-S231	57%	76%	74%	**72%**
NN-S312	54%	76%	74%	**71%**
DT-S231	72%	55%	62%	60%

b) Regression:

I. Randomly Partitioned Data:

1. Abalone:

The following tables show the detailed steps results for Abalone dataset that partitioned randomly. Tables C.329 – C.331 show the site1, site2, and site3 local models RMSE and MAPE, the models evaluation results in other sites, and the weighted average RMSE and MAPE based on the data size that used for evaluation.

Table C. 329. Site 1 Local Models Evaluation Results for Abalone dataset

S1		In S2	In S3	Weighted average RMSE (MAPE)
Models	RMSE (MAPE)	RMSE (MAPE)	RMSE (MAPE)	
LR	2.23 (14.37)	1.99 (13.39)	2.40 (13.08)	2.16 (13.66)
SVR	2.31 (12.99)	1.91 (12.25)	2.61 (13.28)	2.19 (12.72)
DTR	2.20 (13.44)	1.94 (13.14)	2.59 (13.64)	2.16 (13.35)
NNR	2.03 (12.59)	1.82 (12.53)	2.49 (13.37)	2.03 (12.73)
RFR	2.19 (14.02)	2.05 (14.01)	2.64 (14.72)	2.22 (14.16)
Lasso	2.23 (14.53)	2.09 (13.92)	2.38 (13.02)	2.19 (13.94)
Ridge	2.21 (14.31)	2.16 (14.61)	2.38 (13.23)	2.22 (14.22)
ElasticNet	2.25 (14.64)	2.09 (15.56)	2.71 (16.69)	2.27 (15.48)

Table C. 330. Site 2 Local Models Evaluation Results for Abalone dataset

S2		In S1	In S3	Weighted average RMSE (MAPE)
Models	RMSE (MAPE)	RMSE (MAPE)	RMSE (MAPE)	
LR	1.91 (12.25)	2.30 (13.12)	2.56 (13.01)	2.18 (12.70)
SVR	1.96 (11.64)	2.47 (13.25)	2.72 (12.87)	2.29 (12.44)
DTR	1.81 (11.77)	2.26 (13.20)	2.66 (13.05)	2.14 (12.52)
NNR	1.74 (11.41)	2.10 (12.31)	2.49 (12.66)	2.02 (11.98)
RFR	1.88 (12.35)	2.39 (14.45)	2.70 (13.67)	2.22 (13.34)
Lasso	1.87 (12.38)	2.35 (13.36)	2.47 (12.44)	2.16 (12.72)
Ridge	1.90 (12.37)	2.27 (13.55)	2.46 (12.86)	2.14 (12.87)
ElasticNet	1.89 (12.67)	2.35 (13.36)	2.47 (12.44)	2.17 (12.86)

Table C. 331. Site 3 Local Models Evaluation Results for Abalone dataset

S3		In S1	In S2	Weighted average RMSE (MAPE)
Models	RMSE (MAPE)	RMSE (MAPE)	RMSE (MAPE)	
LR	2.28 (13.60)	2.45 (16.19)	2.19 (15.35)	2.29 (15.27)
SVR	2.34 (12.77)	2.48 (15.58)	2.15 (14.48)	2.30 (14.49)
DTR	2.49 (14.25)	2.61 (17.01)	2.30 (15.78)	2.44 (15.88)
NNR	2.26 (13.40)	2.31 (15.64)	2.07 (14.82)	2.19 (14.80)
RFR	2.54 (14.97)	2.67 (17.52)	2.35 (16.02)	2.49 (16.31)
Lasso	2.32 (14.09)	2.38 (16.07)	2.11 (15.25)	2.24 (15.28)
Ridge	2.30 (13.79)	2.37 (16.62)	2.14 (15.96)	2.25 (15.73)
ElasticNet	2.29 (13.91)	2.42 (15.74)	2.14 (14.87)	2.27 (14.96)

For each algorithm, we selected the best average RMSE model. Table C.332 shows that most of the selected models are from site 2.

Table C. 332. Local Models Average RMSE and MAPE for All Sites

Models	S1 model RMSE (MAPE)	S2 model RMSE (MAPE)	S3 model RMSE (MAPE)	The best model
LR	**2.16 (13.66)**	2.18 (12.70)	2.29 (15.27)	LR-S1
SVR	**2.19 (12.72)**	2.29 (12.44)	2.30 (14.49)	SVR-S1
DTR	2.16 (13.35)	**2.14 (12.52)**	2.44 (15.88)	DTR-S2
NNR	2.03 (12.73)	**2.02 (11.98)**	2.19 (14.80)	NNR-S2
RFR	**2.22 (14.16)**	**2.22 (13.34)**	2.49 (16.31)	RFR-S1 – RFR-S2
Lasso	2.19 (13.94)	**2.16 (12.72)**	2.24 (15.28)	LASSO-S2
Ridge	2.22 (14.22)	**2.14 (12.87)**	2.25 (15.73)	RIDGE-S2
ElasticNet	2.27 (15.48)	**2.17 (12.86)**	2.27 (14.96)	ElasticNet-S2

Then, we sent the selected models to other sites and applied mini-batch stochastic gradient descent to update these models. Table C.333 shows the selected LR model of site 1 after sending it to site 2 and site 3 for the updating process.

Table C. 333. Linear Regression (LR) Model Evaluation Results Before and After Updating Method

The best model	In S2 (LR-S12)		In S3 (LR-S13)	
	Before update RMSE (MAPE)	After update RMSE (MAPE)	Before update RMSE (MAPE)	After update RMSE (MAPE)
LR-S1	1.99 (13.39)	2.08 (11.36)	2.40 (13.08)	2.49 (15.07)

Table C.334 shows the updated LR models evaluation results after sending them to all sites and the weighted average RMSE and MAPE. LR model of site 1 that updated in site 2 is the best. So, we sent this model to site 3 to update it using site 3 local data.

Table C. 334. Linear Regression (LR) Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average RMSE (MAPE)
LR-S12	2.07 (13.54)	2.08 (11.36)	2.97 (13.51)	**2.26 (12.55)**
LR-S13	2.43 (17.41)	2.06 (16.11)	2.49 (15.07)	2.28 (16.33)

As shown in Table C.335, the model performance (RMSE) is improved after updating process.

Table C. 335. Linear Regression (LR) Updated Model Evaluation Results Before and After Updating Method

The best model	In S3 (LR-S123)	
	Before update the model	After update the model
LR-S12	2.97 (13.51)	2.49 (15.07)

Tables C.336- C.359 show the model selection and updating methods that applied for the models SVR, DTR, NNR, RFR, Lasso, Ridge, and ElasticNet.

Table C. 336. Support Vector Regressor (SVR) Model Evaluation Results Before and After Updating Method

The best model	In S2 (SVR -S12)		In S3 (SVR -S13)	
	Before update	After update	Before update	After update
SVR-S1	1.91 (12.25)	2.08 (11.42)	2.61 (13.28)	2.34 (12.49)

Table C. 337. Support Vector Regressor (SVR) Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average RMSE (MAPE)
SVR -S12	2.70 (13.61)	2.08 (11.42)	3.01 (13.69)	2.49 (12.64)
SVR -S13	2.39 (14.79)	2.04 (13.67)	2.34 (12.49)	**2.22 (13.80)**

Table C. 338. Support Vector Regressor (SVR) Updated Model Evaluation Results Before and After Updating Method

The best model	In S2 (SVR -S132)	
	Before update	After update
SVR -S13	2.04 (13.67)	2.15 (11.52)

Table C. 339. Decision Tree Regressor (DTR) Model Evaluation Results Before and After Updating Method

The best model	In S1 (DTR-S21)		In S3 (DTR -S23)	
	Before update	After update	Before update	After update
DTR -S2	2.26 (13.20)	1.51 (9.85)	2.66 (13.05)	1.48 (9.34)

Table C. 340. Decision Tree Regressor (DTR) Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average RMSE (MAPE)
DTR -S21	1.51 (9.85)	1.94 (13.17)	2.58 (13.65)	**1.93 (12.14)**
DTR -S23	2.63 (16.99)	2.30 (15.72)	1.48 (9.34)	2.24 (14.81)

Table C. 341. Decision Tree Regressor (DTR) Updated Model Evaluation Results Before and After Updating Method

The best model	In S3 (DTR -S213)	
	Before update	After update
DTR -S21	2.58 (13.65)	1.48 (9.34)

Table C. 342. Neural Network Regressor (NNR) Model Evaluation Results Before and After Updating Method

The best model	In S1 (NNR -S21)		In S3 (NNR-S23)	
	Before update	After update	Before update	After update
NNR-S2	2.10 (12.31)	1.95 (12.09)	2.49 (12.66)	2.18 (12.84)

Table C. 343. Neural Network Regressor (NNR) Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average RMSE (MAPE)
NNR -S21	1.95 (12.09)	1.83 (12.45)	2.47 (13.18)	**2.01 (12.48)**
NNR -S23	2.33 (15.73)	2.08 (14.90)	2.18 (12.84)	2.19 (14.75)

Table C. 344. Neural Network Regressor (NNR) Updated Model Evaluation Results Before and After Updating Method

The best model	In S3 (NNR -S213)	
	Before update	After update
NNR -S21	2.47 (13.18)	2.17 (12.78)

Table C. 345. Random Forest Regressor (RFR) Model (1) Evaluation Results Before and After Updating Method

The best model	In S2 (RFR-S12)		In S3 (RFR-S13)	
	Before update	After update	Before update	After update
RFR-S1	2.05 (14.01)	0.80 (4.98)	2.64 (14.72)	1.06 (5.88)

Table C. 346. Random Forest Regressor (RFR) Updated Model (1) Evaluation Results

The best model	In S1	In S2	In S3	Average RMSE (MAPE)
RFR-S12	2.32 (13.73)	0.80 (4.98)	2.65 (13.75)	**1.70 (9.80)**
RFR-S13	2.78 (17.62)	2.47 (16.32)	1.06 (5.88)	2.28 (14.57)

Table C. 347. Random Forest Regressor (RFR) Updated Model (1) Evaluation Results Before and After Updating Method

The best model	In S3 (RFR-S123)	
	Before update	After update
RFR-S12	2.65 (13.75)	1.08 (5.72)

Table C. 348. Random Forest Regressor (RFR) Model (2) Evaluation Results Before and After Updating Method

The best model	In S1 (RFR-S21)		In S3 (RFR-S23)	
	Before update	After update	Before update	After update
RFR-S2	2.39 (14.45)	0.94 (5.62)	2.70 (13.67)	1.08 (5.88)

Table C. 349. Random Forest Regressor (RFR) Updated Model (2) Evaluation Results

The best model	In S1	In S2	In S3	Average RMSE (MAPE)
RFR-S21	0.94 (5.62)	2.07 (13.99)	2.69 (14.67)	**1.81 (11.29)**
RFR-S23	2.76 (17.88)	2.47 (16.63)	1.08 (5.88)	2.28 (14.79)

Table C. 350. Random Forest Regressor (RFR) Updated Model (2) Evaluation Results Before and After Updating Method

The best model	In S3 (RFR-S213)	
	Before update	After update
RFR-S21	2.69 (14.67)	1.08 (5.88)

Table C. 351. LASSO Model Evaluation Results Before and After Updating Method

The best model	In S1(Lasso-S21)		In S3 (Lasso-S23)	
	Before update	After update	Before update	After update
Lasso-S2	2.35 (13.36)	2.71 (13.34)	2.47 (12.44)	2.50 (14.39)

Table C. 352. LASSO Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average RMSE (MAPE)
Lasso-S21	2.71 (13.34)	2.11 (11.44)	3.05 (13.86)	2.51 (12.59)
Lasso-S23	2.38 (16.26)	1.99 (15.02)	2.50 (14.39)	**2.22 (15.31)**

Table C. 353. LASSO Updated Model Evaluation Results Before and After Updating Method

The best model	In S1 (Lasso-S231)	
	Before update	After update
Lasso-S23	2.38 (16.26)	2.30 (14.16)

Table C. 354. Ridge Model Evaluation Results Before and After Updating Method

The best model	In S1 (Ridge-S21)		In S3 (Ridge-S23)	
	Before update	After update	Before update	After update
Ridge-S2	2.27 (13.55)	2.72 (13.64)	2.46 (12.86)	2.50 (15.01)

Table C. 355. Ridge Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average RMSE (MAPE)
Ridge -S21	2.72 (13.64)	2.12 (11.60)	3.08 (14.12)	2.52 (12.82)
Ridge -S23	2.44 (17.39)	2.06 (16.01)	2.50 (15.01)	**2.28 (16.27)**

Table C. 356. Ridge Updated Model Evaluation Results Before and After Updating Method

The best model	In S1 (Ridge -S231)	
	Before update	After update
Ridge -S23	2.44 (17.39)	2.31 (13.30)

Table C. 357. ElasticNet Model Evaluation Results Before and After Updating Method

The best model	In S1 (ElasticNet -S21)		In S3 (ElasticNet-S23)	
	Before update	After update	Before update	After update
ElasticNet-S2	2.35 (13.36)	2.68 (13.18)	2.47 (12.44)	2.48 (14.06)

Table C. 358. ElasticNet Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average RMSE (MAPE)
ElasticNet -S21	2.68 (13.18)	2.12 (11.53)	3.08 (14.03)	2.51 (12.62)
ElasticNet-S23	2.41 (16.29)	1.98 (14.74)	2.48 (14.06)	**2.23 (15.12)**

Table C. 359. ElasticNet Updated Model Evaluation Results Before and After Updating Method

The best model	In S1 (ElasticNet-S231)	
	Before update	After update
ElasticNet -S23	2.41 (16.29)	2.28 (15.23)

Finally, we evaluated the final updated models in all sites, then sent the models with the evaluation results and the data size that used for evaluation to the sever. The server calculates the average RMSE for each model and then selects the best average RMSE models for the linear combination method. As shown in Table C.360, we selected the best three models and combined these models using linear combination methods, Simple weight average, Error-based (RMSE), and Performance-based (Accuracy) methods.

Table C. 360. Updated Model Evaluation Results in All Sites

The best model	In S1	In S2	In S3	Average RMSE (MAPE)
LR-S123	2.43 (17.41)	2.05 (16.11)	2.49 (15.07)	2.27 (16.33)
SVR-S132	2.81 (14.04)	2.15 (11.52)	3.09 (13.99)	2.57 (12.89)
DTR-S213	2.61 (16.98)	2.29 (15.72)	1.48 (9.34)	2.23 (14.81)
NNR-S213	2.31 (15.60)	2.07 (14.82)	2.17 (12.78)	**2.17 (14.66)**
RFR-S123	2.78 (17.62)	2.47 (16.32)	1.08 (5.72)	2.28 (14.54)
RFR-S213	2.76 (17.88)	2.47 (16.63)	1.08 (5.88)	2.28 (14.79)
Lasso-S231	2.30 (14.16)	1.95 (13.35)	2.46 (12.87)	**2.18 (13.52)**
Ridge-S231	2.31 (13.30)	1.91 (12.51)	2.64 (13.41)	**2.19 (12.97)**
ElasticNet-S231	2.28 (15.23)	1.98 (14.75)	2.64 (14.65)	2.22 (14.89)

We applied the same methodology on the rest of regression datasets.

2. Parkinson Disease (Total UPDRS):

Table C. 361. Site 1 Local Models Evaluation

S1		In S2	In S3	Weighted average
Models	RMSE (MAPE)	RMSE (MAPE)	RMSE (MAPE)	
LR	11.02 (30.4)	13.58 (56.17)	12.61 (42.68)	12.32 (41.86)
SVR	10.82 (25.93)	12.29 (50.25)	11.26 (37.91)	11.36 (36.92)
DTR	8.92 (24.06)	12.50 (51.31)	11.99 (41.56)	11.10 (38.12)
NNR	7.66 (20.01)	12.18 (48.94)	11.64 (39.89)	10.46 (35.50)
RFR	9.1 (23.7)	13.12 (52.75)	12.16 (41.16)	11.38 (38.18)
Lasso	11.17 (30.27)	13.21 (55.43)	11.81 (40.68)	11.93 (40.78)
Ridge	11.15 (30.53)	12.98 (52.24)	11.50 (38.90)	11.74 (39.34)
ElasticNet	10.74 (28.73)	13.87 (57.34)	12.79 (42.82)	12.37 (41.65)

Table C. 362. Site 2 Local Models Evaluation

S2		In S1	In S3	Weighted average
Models	RMSE (MAPE)	RMSE (MAPE)	RMSE (MAPE)	
LR	11.65 (32.59)	11.14 (32.23)	12.61 (42.68)	11.89 (36.81)
SVR	6.83 (20.80)	16.84 (36.61)	10.31 (25.93)	11.63 (28.22)
DTR	5.15 (16.88)	17.06 (39.79)	8.96 (23.14)	10.72 (27.13)
NNR	5.06 (16.66)	17.66 (45.19)	9.64 (25.91)	11.19 (30.05)
RFR	5.39 (16.74)	16.39 (36.73)	9.25 (24.27)	10.68 (26.57)
Lasso	15.74 (51.63)	36.34 (88.59)	35.32 (76.54)	30.96 (74.54)
Ridge	28.34 (83.33)	56.64 (98.94)	47.39 (84.63)	45.87 (89.04)
ElasticNet	16.63 (52.09)	16.37 (35.40)	10.15 (26.61)	13.76 (35.62)

Table C. 363. Site 3 Local Models Evaluation

	S3	In S1	In S2	Weighted average
Models	RMSE (MAPE)	RMSE (MAPE)	RMSE (MAPE)	
LR	8.99 (27.33)	13.67 (32.54)	10.07 (38.97)	10.79 (31.84)
SVR	8.20 (24.86)	14.03 (33.04)	8.96 (35.20)	10.31 (30.04)
DTR	6.37 (18.06)	15.66 (38.91)	9.35 (37.69)	10.15 (29.65)
NNR	5.85 (16.49)	16.38 (43.56)	8.51 (32.19)	9.96 (29.19)
RFR	6.33 (16.79)	16.27 (37.55)	8.96 (34.44)	10.24 (27.88)
Lasso	9.48 (28.36)	14.05 (32.34)	8.81 (34.43)	10.83 (31.13)
Ridge	8.85 (27.37)	14.61 (35.14)	8.77 (34.77)	10.73 (31.71)
ElasticNet	9.66 (28.37)	14.74 (37.85)	10.78 (38.98)	11.60 (34.04)

Table C. 364. Local Models Average RMSE (MAPE) for All sites

Models	S1 model RMSE (MAPE)	S2 model RMSE (MAPE)	S3 model RMSE (MAPE)	The best model
LR	12.32 (41.86)	11.89 (36.81)	**10.79 (31.84)**	LR-S3
SVR	11.36 (36.92)	11.63 (28.22)	**10.31 (30.04)**	SVR-S3
DTR	11.10 (38.12)	10.72 (27.13)	**10.15 (29.65)**	DTR-S3
NNR	10.46 (35.50)	11.19 (30.05)	**9.96 (29.19)**	NNR-S3
RFR	11.38 (38.18)	10.68 (26.57)	**10.24 (27.88)**	RFR-S3
Lasso	11.93 (40.78)	30.96 (74.54)	**10.83 (31.13)**	LASSO-S3
Ridge	11.74 (39.34)	45.87 (89.04)	**10.73 (31.71)**	RIDGE-S33
ElasticNet	12.37 (41.65)	13.76 (35.62)	**11.60 (34.04)**	ElasticNet-S3

Table C. 365. Linear Regression (LR) Model Evaluation Results After Updating Method

The best model	In S1 (LR-S31)	In S2 (LR-S32)
	After update the model	After update the model
LR-S3	10.84 (29.11)	7.16 (20.54)

Table C. 366. Linear Regression (LR) Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average RMSE (MAPE)
LR-S31	10.84 (29.11)	11.43 (48.28)	10.21 (36.16)	**10.71 (36.74)**
LR-S32	17.14 (35.57)	7.16 (20.54)	11.04 (26.88)	12.12 (28.23)

Table C. 367. Linear Regression (LR) Updated Model Evaluation Results After Updating Method

The best model	In S2 (LR-S312)
	After update the model
LR-S31	6.71 (21.50)

Table C. 368. Support Vector Regressor (SVR) Model Evaluation Results After Updating Method

The best model	In S1 (SVR -S31)	In S2 (SVR -S32)
	After update the model	After update the model
SVR-S3	10.83 (29.22)	7.18 (20.53)

Table C. 369. Support Vector Regressor (SVR) Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average RMSE (MAPE)
SVR -S31	10.83 (29.22)	11.64 (49.20)	10.50 (37.04)	**10.88 (37.38)**
SVR -S32	17.03 (35.27)	7.18 (20.53)	10.95 (26.54)	12.05 (27.98)

Table C. 370. Support Vector Regressor (SVR) Updated Model Evaluation Results After Updating Method

The best model	In S2 (SVR -S312)
	After update the model
SVR -S31	18.70 (57.32)

Table C. 371. Decision Tree Regressor (DTR) Model Evaluation Results After Updating Method

The best model	In S1 (DTR-S31)	In S2 (DTR -S32)
	After update the model	After update the model
DTR -S3	6.94 (18.63)	3.59 (11.77)

Table C. 372. Decision Tree Regressor (DTR) Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average RMSE (MAPE)
DTR -S31	6.94 (18.63)	12.50 (51.33)	11.98 (41.47)	10.44 (36.29)
DTR -S32	17.06 (39.77)	3.59 (11.77)	8.95 (23.13)	**10.34 (25.89)**

Table C. 373. Decision Tree Regressor (DTR) Updated Model Evaluation Results After Updating Method

The best model	In S1 (DTR -S321)
	After update the model
DTR -S32	6.94 (18.63)

Table C. 374. Neural Network Regressor (NNR) Model Evaluation Results After Updating Method

The best model	In S1(NNR -S31)	In S2 (NNR-S32)
	After update the model	After update the model
NNR-S3	27.43 (66.38)	26.95 (96.49)

Table C. 375. Neural Network Regressor (NNR) Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average RMSE (MAPE)
NNR -S31	27.43 (66.38)	19.73 (60.35)	23.17 (64.27)	**23.75 (64.02)**
NNR -S32	35.54 (97.30)	26.95 (96.49)	31.13 (97.14)	31.58 (97.04)

Table C. 376. Neural Network Regressor (NNR) Updated Model Evaluation Results After Updating Method

The best model	In S2 (NNR -S312)
	After update the model
NNR -S31	23.83 (81.91)

Table C. 377. Random Forest Regressor (RFR) Model Evaluation Results After Updating Method

The best model	In S1 (RFR-S31)	In S2 (RFR-S32)
	After update the model	After update the model
RFR-S3	3.84 (9.13)	2.19 (6.25)

Table C. 378. Random Forest Regressor (RFR) Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average RMSE (MAPE)
RFR-S31	3.84 (9.13)	12.94 (52.10)	12.63 (42.50)	**9.80 (33.79)**
RFR-S32	16.39 (36.72)	2.19 (6.25)	9.21 (24.72)	9.89 (24.25)

Table C. 379. Random Forest Regressor (RFR) Updated Model Evaluation Results After Updating Method

The best model	In S2 (RFR-S312)
	After update the model
RFR-S31	2.30 (6.54)

Table C. 380. LASSO Model Evaluation Results Before and After Updating Method

The best model	In S1(Lasso-S31)	In S2 (Lasso-S32)
	After update the model	After update the model
Lasso-S3	10.80 (28.83)	7.14 (20.56)

Table C. 381. LASSO Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average RMSE (MAPE)
Lasso-S31	10.80 (28.83)	11.61 (48.85)	10.41 (36.75)	**10.83 (37.04)**
Lasso-S32	17.24 (35.89)	7.14 (20.56)	10.99 (26.70)	12.13 (28.26)

Table C. 382. LASSO Updated Model Evaluation Results After Updating Method

The best model	In S2 (Lasso-S312)
	After update the model
Lasso-S31	9.95 (27.75)

Table C. 383. Ridge Model Evaluation Results After Updating Method

The best model	In S1 (Ridge-S31)	In S2 (Ridge-S32)
	After update the model	After update the model
Ridge-S3	10.95 (28.40)	7.17 (20.36)

Table C. 384. Ridge Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average RMSE (MAPE)
Ridge-S31	10.95 (28.40)	11.35 (46.80)	10.32 (36.06)	**10.77 (36.11)**
Ridge-S32	17.12 (35.43)	7.17 (20.36)	10.95 (26.54)	12.08 (27.99)

Table C. 385. Ridge Updated Model Evaluation Results After Updating Method

The best model	In S2 (Ridge -S312)
	After update the model
Ridge -S31	6.85 (22.73)

Table C. 386. ElasticNet Model Evaluation Results After Updating Method

The best model	In S1 (ElasticNet-S31)	In S2 (ElasticNet-S32)
	After update the model	After update the model
ElasticNet-S3	10.96 (29.30)	7.29 (20.54)

Table C. 387. ElasticNet Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average RMSE (MAPE)
ElasticNet-S31	10.96 (29.30)	11.89 (49.96)	10.80 (37.79)	**11.11 (37.91)**
ElasticNet-S32	17.14 (35.53)	7.29 (20.54)	10.97 (26.67)	12.12 (28.12)

Table C. 388. ElasticNet Updated Model Evaluation Results After Updating Method

The best model	In S2 (ElasticNet-S312)
	After update the model
ElasticNet -S31	8.69 (27.50)

Table C. 389. Updated Model Evaluation Results in All Sites

The best model	In S1	In S2	In S3	Average RMSE (MAPE)
LR312	16.58 (36.71)	6.71 (21.50)	9.90 (25.68)	11.34 (28.32)
SVR312	27.07 (66.95)	18.70 (57.32)	22.69 (62.91)	23.18 (62.91)
DTR321	6.94 (18.63)	12.50 (51.31)	11.99 (41.48)	**10.44 (36.30)**
NNR312	32.32 (85.03)	23.83 (81.91)	27.97 (84.99)	28.41 (84.26)
RFR312	16.42 (37.19)	2.30 (6.54)	9.25 (24.34)	**9.95 (24.31)**
Lasso312	18.89 (43.48)	9.95 (27.75)	12.10 (28.64)	13.82 (33.32)
Ridge312	16.18 (36.67)	6.85 (22.73)	9.82 (26.28)	**11.21 (28.86)**
ElasticNet312	19.43 (44.68)	8.69 (27.50)	11.68 (31.44)	13.52 (34.86)

3. Parkinson Disease (Motor UPDRS):

Table C. 390. Site 1 Local Models Evaluation

S1		In S2	In S3	Weighted average
Models	RMSE (MAPE)	RMSE (MAPE)	RMSE (MAPE)	
LR	7.31 (29.77)	9.17 (49.91)	8.76 (41.98)	8.38 (39.85)
SVR	7.06 (27.67)	9.35 (48.48)	8.70 (40.19)	8.31 (38.05)
DTR	5.97 (23.65)	8.57 (44.07)	8.68 (41.50)	7.76 (36.23)
NNR	5.28 (19.98)	9.06 (45.42)	8.36 (40.02)	7.51 (34.70)
RFR	6.01 (22.88)	9.02 (45.19)	9.09 (43.04)	8.06 (36.90)
Lasso	7.39 (30.40)	8.65 (45.56)	8.45 (40.46)	8.15 (38.36)
Ridge	7.09 (28.87)	8.68 (46.05)	8.26 (39.86)	7.97 (37.72)
ElasticNet	7.16 (29.80)	8.07 (39.74)	8.05 (37.48)	7.76 (35.49)

Table C. 391. Site 2 Local Models Evaluation

S2		In S1	In S3	Weighted average
Models	RMSE (MAPE)	RMSE (MAPE)	RMSE (MAPE)	
LR	8.25 (31.37)	13.92 (50.51)	9.50 (33.28)	10.66 (38.51)
SVR	5.08 (21.66)	11.50 (41.31)	7.84 (29.42)	8.38 (31.48)
DTR	3.89 (17.51)	12.22 (45.90)	6.99 (27.10)	7.97 (31.01)
NNR	3.86 (17.25)	12.65 (49.80)	7.56 (28.79)	8.35 (32.95)
RFR	4.13 (17.80)	11.33 (41.11)	7.22 (28.99)	7.83 (30.30)
Lasso	10.81 (39.09)	12.64 (46.53)	8.11 (30.66)	10.25 (37.92)
Ridge	15.91 (65.14)	18.20 (67.06)	15.79 (45.27)	16.61 (57.23)
ElasticNet	10.64 (44.33)	12.76 (47.08)	8.16 (30.73)	10.27 (39.39)

Table C. 392. Site 3 Local Models Evaluation

S3		In S1	In S2	Weighted average
Models	RMSE (MAPE)	RMSE (MAPE)	RMSE (MAPE)	
LR	7.37 (30.09)	9.88 (38.37)	7.12 (36.44)	8.21 (34.71)
SVR	6.55 (27.42)	9.78 (36.79)	6.99 (34.72)	7.79 (32.61)
DTR	5.15 (20.43)	11.49 (43.73)	7.05 (36.23)	7.77 (32.27)

NNR	4.71 (18.39)	11.44 (45.73)	6.71 (32.40)	7.48 (31.10)
RFR	4.96 (18.32)	11.14 (41.19)	7.24 (36.13)	7.62 (30.50)
Lasso	7.60 (30.48)	9.72 (36.84)	7.65 (38.40)	8.39 (34.86)
Ridge	9.01 (34.50)	9.94 (38.99)	6.79 (35.08)	8.85 (36.47)
ElasticNet	6.75 (28.95)	9.42 (37.70)	7.89 (40.08)	7.98 (34.91)

Table C. 393. Local Models Average RMSE (MAPE) for All sites

Models	S1 model RMSE (MAPE)	S2 model RMSE (MAPE)	S3 model RMSE (MAPE)	The best model
LR	8.38 (39.85)	10.66 (38.51)	**8.21 (34.71)**	LR-S3
SVR	8.31 (38.05)	8.38 (31.48)	**7.79 (32.61)**	SVR-S3
DTR	**7.76 (36.23)**	7.97 (31.01)	7.77 (32.27)	DTR-S1
NNR	7.51 (34.70)	8.35 (32.95)	**7.48 (31.10)**	NNR-S3
RFR	8.06 (36.90)	7.83 (30.30)	**7.62 (30.50)**	RFR-S3
Lasso	**8.15 (38.36)**	10.25 (37.92)	8.39 (34.86)	LASSO-S1
Ridge	**7.97 (37.72)**	16.61 (57.23)	8.85 (36.47)	RIDGE-S1
ElasticNet	**7.76 (35.49)**	10.27 (39.39)	7.98 (34.91)	ElasticNet-S1

Table C. 394. Linear Regression (LR) Model Evaluation Results After Updating Method

The best model	In S1 (LR-S31)	In S2 (LR-S32)
	After update the model	After update the model
LR-S3	7.30 (29.94)	5.40 (21.47)

Table C. 395. Linear Regression (LR) Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average RMSE (MAPE)
LR-S31	7.30 (29.94)	7.84 (40.63)	8.08 (38.58)	**7.76 (36.22)**
LR-S32	11.27 (36.67)	5.40 (21.47)	8.07 (28.44)	8.48 (29.48)

Table C. 396. Linear Regression (LR) Updated Model Evaluation Results After Updating Method

The best model	In S2 (LR-S312)
	After update the model
LR-S31	6.11 (27.11)

Table C. 397. Support Vector Regressor (SVR) Model Evaluation Results After Updating Method

The best model	In S1 (SVR -S31)	In S2 (SVR -S32)
	After update the model	After update the model
SVR-S3	7.35 (30.26)	5.42 (21.55)

Table C. 398. Support Vector Regressor (SVR) Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average RMSE (MAPE)
SVR -S31	7.35 (30.26)	8.30 (43.58)	8.49 (40.56)	**8.07 (37.88)**
SVR -S32	11.14 (36.29)	5.42 (21.55)	7.99 (28.22)	8.41 (29.28)

Table C. 399. Support Vector Regressor (SVR) Updated Model Evaluation Results After Updating Method

The best model	In S2 (SVR -S312)
	After update the model
SVR -S31	12.34 (48.01)

Table C. 400. Decision Tree Regressor (DTR) Model Evaluation Results After Updating Method

The best model	In S2 (DTR-S12)	In S3 (DTR -S13)
	After update the model	After update the model
DTR -S1	2.74 (12.28)	4.16 (16.35)

Table C. 401. Decision Tree Regressor (DTR) Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average RMSE (MAPE)
DTR -S12	12.24 (46.01)	2.74 (12.28)	6.99 (27.09)	7.70 (29.78)
DTR -S13	11.50 (43.73)	7.08 (36.38)	4.16 (16.35)	**7.28 (30.19)**

Table C. 402. Decision Tree Regressor (DTR) Updated Model Evaluation Results After Updating Method

The best model	In S2 (DTR -S132)
	After update the model
DTR -S13	2.74 (12.28)

Table C. 403. Neural Network Regressor (NNR) Model Evaluation Results After Updating Method

The best model	In S1 (NNR -S31)	In S2 (NNR-S32)
	After update the model	After update the model
NNR -S3	20.36 (74.43)	20.68 (95.39)

Table C. 404. Neural Network Regressor (NNR) Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average RMSE (MAPE)
NNR -S31	20.36 (74.43)	16.61 (69.51)	18.58 (71.75)	**18.69 (72.09)**
NNR -S32	24.56 (96.10)	20.68 (95.39)	22.88 (96.07)	22.90 (95.92)

Table C. 405. Neural Network Regressor (NNR) Updated Model Evaluation Results After Updating Method

The best model	In S2 (NNR -S312)
	After update the model
NNR -S31	18.58 (83.19)

Table C. 406. Random Forest Regressor (RFR) Model Evaluation Results After Updating Method

The best model	In S1 (RFR-S31)	In S2 (RFR-S32)
	After update the model	After update the model
RFR-S3	2.63 (9.16)	1.65 (6.33)

Table C. 407. Random Forest Regressor (RFR) Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average RMSE (MAPE)
RFR-S31	2.63 (9.16)	8.86 (44.40)	9.08 (42.92)	**6.89 (32.13)**
RFR-S32	11.45 (40.98)	1.65 (6.33)	7.33 (29.29)	7.32 (27.64)

Table C. 408. Random Forest Regressor (RFR) Updated Model Evaluation Results After Updating Method

The best model	In S2 (RFR-S312)
	After update the model
RFR-S31	1.74 (6.82)

Table C. 409. LASSO Model Evaluation Results Before and After Updating Method

The best model	In S2 (Lasso-S12)	In S3 (Lasso-S13)
	After update the model	After update the model
Lasso-S1	5.40 (21.60)	6.51 (28.61)

Table C. 410. LASSO Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average RMSE (MAPE)
Lasso-S12	11.39 (37.36)	5.40 (21.60)	8.08 (28.54)	8.53 (29.78)
Lasso-S13	9.25 (35.83)	6.34 (32.13)	6.51 (28.61)	**7.37 (31.84)**

Table C. 411. LASSO Updated Model Evaluation Results After Updating Method

The best model	In S2 (Lasso-S132)
	After update the model
Lasso-S13	5.35 (24.48)

Table C. 412. Ridge Model Evaluation Results After Updating Method

The best model	In S2 (Ridge-S12)	In S3 (Ridge-S13)
	After update the model	After update the model
Ridge-S1	5.41 (21.43)	6.76 (28.91)

Table C. 413. Ridge Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average RMSE (MAPE)
Ridge -S12	11.22 (36.59)	5.41 (21.43)	7.99 (28.12)	8.44 (29.31)
Ridge -S13	9.59 (36.10)	6.53 (30.97)	6.76 (28.91)	**7.64 (31.77)**

Table C. 414. Ridge Updated Model Evaluation Results After Updating Method

The best model	In S2 (Ridge -S132)
	After update the model
Ridge -S13	5.19 (23.21)

Table C. 415. ElasticNet Model Evaluation Results After Updating Method

The best model	In S2 (ElasticNet -S12)	In S3 (ElasticNet-S13)
	After update the model	After update the model
ElasticNet-S1	5.49 (21.72)	6.41 (27.74)

Table C. 416. ElasticNet Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average RMSE (MAPE)
ElasticNet -S12	11.19 (36.83)	5.49 (21.72)	8.01 (28.43)	8.45 (29.59)
ElasticNet-S13	9.25 (35.29)	6.52 (32.98)	6.41 (27.74)	**7.37 (31.49)**

Table C. 417. ElasticNet Updated Model Evaluation Results After Updating Method

The best model	In S2 (ElasticNet-S132)
	After update the model
ElasticNet -S13	5.33 (23.27)

Table C. 418. Updated Model Evaluation Results in All Sites

The best model	In S1	In S2	In S3	Average RMSE (MAPE)
LR-S312	11.74 (44.67)	6.11 (27.11)	8.10 (34.14)	8.82 (35.93)
SVR-S312	16.33 (53.25)	12.34 (48.01)	14.66 (50.60)	14.65 (50.85)
DTR-S132	12.24 (45.99)	2.74 (12.28)	6.99 (27.11)	**7.70 (29.78)**
NNR-S312	22.46 (84.88)	18.58 (83.19)	20.82 (85.10)	20.82 (84.57)
RFR-S312	11.52 (42.33)	1.74 (6.82)	7.27 (29.16)	**7.34 (28.14)**
Lasso-S132	11.39 (41.69)	5.35 (24.48)	7.46 (30.10)	**8.25 (32.57)**
Ridge-S132	11.19 (39.68)	5.19 (23.21)	7.77 (30.30)	8.28 (31.69)
ElasticNet-S132	11.43 (40.45)	5.33 (23.27)	7.63 (29.30)	8.33 (31.53)

4. Boston Housing:

Table C. 419. Site 1 Local Models Evaluation

S1		In S2	In S3	Weighted average
Models	RMSE (MAPE)	RMSE (MAPE)	RMSE (MAPE)	
LR	4.66 (17.52)	9.37 (21.04)	5.10 (19.97)	5.78 (19.20)
SVR	4.91 (16.08)	10.27 (21.64)	6.17 (21.54)	6.49 (19.38)
DTR	3.46 (12.12)	8.57 (17.76)	4.01 (18.33)	4.70 (15.73)
NNR	3.65 (12.73)	7.74 (20.24)	4.25 (18.86)	4.71 (16.68)
RFR	4.20 (14.44)	8.19 (16.80)	3.98 (17.77)	4.91 (16.24)
Lasso	4.63 (18.12)	9.36 (21.45)	5.37 (21.05)	5.87 (19.96)
Ridge	4.85 (18.68)	9.05 (20.90)	5.25 (21.75)	5.85 (20.35)
ElasticNet	4.76 (17.49)	8.67 (19.72)	5.74 (24.89)	5.93 (20.90)

Table C. 420. Site 2 Local Models Evaluation

S2		In S1	In S3	Weighted average
Models	RMSE (MAPE)	RMSE (MAPE)	RMSE (MAPE)	
LR	4.62 (15.33)	9.46 (41.93)	9.46 (53.27)	8.49 (41.15)
SVR	5.50 (17.51)	9.52 (42.42)	9.91 (57.56)	8.87 (43.49)
DTR	4.23 (13.18)	7.49 (31.72)	6.46 (34.76)	6.43 (29.23)
NNR	4.18 (12.22)	7.39 (30.99)	6.50 (35.69)	6.39 (29.12)
RFR	5.49 (14.79)	7.99 (36.20)	7.53 (41.30)	7.31 (33.96)
Lasso	4.76 (16.26)	9.63 (42.11)	9.90 (55.37)	8.76 (42.24)
Ridge	5.08 (16.68)	10.47 (45.88)	10.66 (59.81)	9.47 (45.61)
ElasticNet	4.96 (16.47)	9.63 (42.11)	9.89 (55.37)	8.8 (42.29)

Table C. 421. Site 3 Local Models Evaluation

S3		In S1	In S2	Weighted average
Models	RMSE (MAPE)	RMSE (MAPE)	RMSE (MAPE)	
LR	4.45 (20.62)	5.71 (21.99)	9.41 (21.50)	5.95 (21.34)
SVR	4.50 (19.71)	5.22 (19.18)	9.47 (21.77)	5.78 (19.91)
DTR	3.22 (13.32)	4.15 (15.14)	8.45 (15.92)	4.64 (14.57)
NNR	2.71 (11.80)	4.81 (15.20)	7.44 (19.01)	4.49 (14.60)
RFR	3.22 (14.71)	4.12 (14.65)	7.89 (16.15)	4.51 (14.97)
Lasso	4.34 (20.24)	5.56 (21.32)	9.77 (21.44)	5.91 (20.91)
Ridge	4.32 (20.42)	5.83 (22.45)	9.98 (22.12)	6.06 (21.57)
ElasticNet	4.29 (20.47)	5.73 (21.90)	9.68 (21.76)	5.94 (21.30)

Table C. 422. Local Models Average RMSE (MAPE) for All sites

Models	S1 model RMSE (MAPE)	S2 model RMSE (MAPE)	S3 model RMSE (MAPE)	The best model
LR	**5.78 (19.20)**	8.49 (41.15)	5.95 (21.34)	LR-S1
SVR	6.49 (19.38)	8.87 (43.49)	**5.78 (19.91)**	SVR-S3
DTR	4.70 (15.73)	6.43 (29.23)	**4.64 (14.57)**	DTR-S3
NNR	4.71 (16.68)	6.39 (29.12)	**4.49 (14.60)**	NNR-S3
RFR	4.91 (16.24)	7.31 (33.96)	**4.51 (14.97)**	RFR-S3
Lasso	**5.87 (19.96)**	8.76 (42.24)	5.91 (20.91)	LASSO-S1
Ridge	**5.85 (20.35)**	9.47 (45.61)	6.06 (21.57)	RIDGE-S1
ElasticNet	**5.93 (20.90)**	8.8 (42.29)	5.94 (21.30)	ElasticNet-S1

Table C. 423. Linear Regression (LR) Model Evaluation Results After Updating Method

The best model	In S2 (LR-S12)	In S3 (LR-S13)
	After update the model	After update the model
LR-S1	11.01 (36.87)	6.56 (24.94)

Table C. 424. Linear Regression (LR) Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average RMSE (MAPE)
LR-S12	7.96 (32.99)	11.01 (36.87)	8.35 (37.23)	8.73 (35.46)
LR-S13	7.04 (25.02)	12.94 (37.05)	6.56 (24.94)	**8.03 (27.39)**

Table C. 425. Linear Regression (LR) Updated Model Evaluation Results After Updating Method

The best model	In S2 (LR-S132)
	After update the model
LR-S13	11.29 (37.63)

Table C. 426. Support Vector Regressor (SVR) Model Evaluation Results After Updating Method

The best model	In S1 (SVR -S31)	In S2 (SVR -S32)
	After update the model	After update the model
SVR-S3	12.63 (55.14)	27.87 (95.99)

Table C. 427. Support Vector Regressor (SVR) Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average RMSE (MAPE)
SVR -S31	12.63 (55.14)	19.09 (63.57)	13.09 (55.58)	**14.11 (57.01)**
SVR -S32	21.81 (95.04)	27.87 (95.99)	22.79 (94.41)	23.41 (94.98)

Table C. 428. Support Vector Regressor (SVR) Updated Model Evaluation Results After Updating Method

The best model	In S2 (SVR -S312)
	After update the model
SVR -S31	27.87 (95.99)

Table C. 429. Decision Tree Regressor (DTR) Model Evaluation Results After Updating Method

The best model	In S1 (DTR-S31)	In S2 (DTR -S32)
	After update the model	After update the model
DTR -S3	0.63 (2.62)	0.43 (1.55)

Table C. 430. Decision Tree Regressor (DTR) Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average RMSE (MAPE)
DTR -S31	0.63 (2.62)	8.49 (17.65)	3.99 (18.39)	**3.55 (11.93)**
DTR -S32	7.17 (30.53)	0.43 (1.55)	6.46 (34.96)	5.54 (26.51)

Table C. 431. Decision Tree Regressor (DTR) Updated Model Evaluation Results After Updating Method

The best model	In S2 (DTR -S312)
	After update the model
DTR -S31	0.44 (1.55)

Table C. 432. Neural Network Regressor (NNR) Model Evaluation Results After Updating Method

The best model	In S1 (NNR -S31)	In S2 (NNR-S32)
	After update the model	After update the model
NNR -S3	22.43 (98.42)	28.60 (99.61)

Table C. 433. Neural Network Regressor (NNR) Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average RMSE (MAPE)
NNR -S31	22.43 (98.42)	28.48 (98.84)	23.41 (98.10)	**24.03 (98.38)**
NNR -S32	22.46 (99.36)	28.60 (99.61)	23.39 (98.03)	24.06 (98.88)

Table C. 434. Neural Network Regressor (NNR) Updated Model Evaluation Results After Updating Method

The best model	In S2 (NNR -S312)
	After update the model
NNR-S31	28.19 (97.03)

Table C. 435. Random Forest Regressor (RFR) Model Evaluation Results After Updating Method

The best model	In S1 (RFR-S31)	In S2 (RFR-S32)
	After update the model	After update the model
RFR-S3	2.21 (5.35)	2.08 (5.09)

Table C. 436. Random Forest Regressor (RFR) Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average RMSE (MAPE)
RFR-S31	2.21 (5.35)	7.45 (15.65)	3.75 (18.43)	**3.87 (12.64)**
RFR-S32	7.59 (34.04)	2.08 (5.09)	7.09 (40.73)	6.29 (30.93)

Table C. 437. Random Forest Regressor (RFR) Updated Model Evaluation Results After Updating Method

The best model	In S2 (RFR-S312)
	After update the model
RFR-S31	2.49 (6.67)

Table C. 438. LASSO Model Evaluation Results Before and After Updating Method

The best model	In S2 (Lasso-S12)	In S3 (Lasso-S13)
	After update the model	After update the model
Lasso-S1	11.31 (36.59)	6.67 (26.58)

Table C. 439. LASSO Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average RMSE (MAPE)
Lasso-S12	8.48 (36.34)	11.31 (36.59)	9.23 (41.87)	9.35 (38.60)
Lasso-S13	7.25 (25.89)	13.36 (38.55)	6.67 (26.58)	**8.24 (28.70)**

Table C. 440. LASSO Updated Model Evaluation Results After Updating Method

The best model	In S2 (Lasso-S132)
	After update the model
Lasso-S13	11.65 (38.32)

Table C. 441. Ridge Model Evaluation Results After Updating Method

The best model	In S2 (Ridge-S12)	In S3 (Ridge-S13)
	After update the model	After update the model
Ridge-S1	11.13 (37.44)	6.64 (27.15)

Table C. 442. Ridge Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average RMSE (MAPE)
Ridge -S12	7.29 (28.84)	11.13 (37.44)	7.47 (31.07)	**8.13 (31.45)**
Ridge -S13	7.29 (26.38)	13.37 (38.75)	6.64 (27.15)	8.24 (29.16)

Table C. 443. Ridge Updated Model Evaluation Results After Updating Method

The best model	In S3 (Ridge -S132)
	After update the model
Ridge -S12	6.56 (24.94)

Table C. 444. ElasticNet Model Evaluation Results After Updating Method

The best model	In S2 (ElasticNet -S12)	In S3 (ElasticNet-S13)
	After update the model	After update the model
ElasticNet-S1	11.53 (39.46)	6.32 (26.97)

Table C. 445. ElasticNet Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average RMSE (MAPE)
ElasticNet -S12	7.64 (30.36)	11.53 (39.46)	7.58 (31.70)	8.39 (32.72)
ElasticNet-S13	7.29 (26.63)	13.19 (38.24)	6.32 (26.97)	**8.08 (29.09)**

Table C. 446. ElasticNet Updated Model Evaluation Results After Updating Method

The best model	In S2 (ElasticNet-S132)
	After update the model
ElasticNet -S13	11.13 (39.13)

Table C. 447. Updated Model Evaluation Results in All Sites

The best model	In S1	In S2	In S3	Average RMSE (MAPE)
LR-S132	8.96 (39.02)	11.29 (37.63)	10.03 (48.77)	9.85 (42.64)
SVR-S312	21.81 (95.04)	27.87 (95.99)	22.79 (94.41)	23.41 (94.97)

DTR-S312	7.04 (29.92)	0.44 (1.55)	6.36 (34.35)	**5.45 (26.02)**
NNR-S312	22.06 (96.66)	28.19 (97.03)	22.98 (96.34)	23.65 (96.61)
RFR-S312	7.49 (34.77)	2.49 (6.67)	7.65 (45.86)	**6.55 (33.59)**
Lasso-S132	9.29 (40.29)	11.65 (38.32)	10.44 (49.11)	10.22 (43.42)
Ridge-S123	7.04 (25.03)	12.94 (37.05)	6.56 (24.94)	**8.03 (27.39)**
ElasticNet-S132	8.84 (38.37)	11.13 (39.13)	9.45 (44.79)	9.54 (41.09)

5. Abalone:

Table C. 448. Site 1 Local Models Evaluation

S1		In S2	In S3	Weighted average
Models	RMSE (MAPE)	RMSE (MAPE)	RMSE (MAPE)	
LR	2.23 (14.37)	1.99 (13.39)	2.40 (13.08)	2.16 (13.66)
SVR	2.31 (12.99)	1.91 (12.25)	2.61 (13.28)	2.19 (12.72)
DTR	2.20 (13.44)	1.94 (13.14)	2.59 (13.64)	2.16 (13.35)
NNR	2.03 (12.59)	1.82 (12.53)	2.49 (13.37)	2.03 (12.73)
RFR	2.19 (14.02)	2.05 (14.01)	2.64 (14.72)	2.22 (14.16)
Lasso	2.23 (14.53)	2.09 (13.92)	2.38 (13.02)	2.19 (13.94)
Ridge	2.21 (14.31)	2.16 (14.61)	2.38 (13.23)	2.22 (14.22)
ElasticNet	2.25 (14.64)	2.09 (15.56)	2.71 (16.69)	2.27 (15.48)

Table C. 449. Site 2 Local Models Evaluation

S2		In S1	In S3	Weighted average
Models	RMSE (MAPE)	RMSE (MAPE)	RMSE (MAPE)	
LR	1.91 (12.25)	2.30 (13.12)	2.56 (13.01)	2.18 (12.70)
SVR	1.96 (11.64)	2.47 (13.25)	2.72 (12.87)	2.29 (12.44)
DTR	1.81 (11.77)	2.26 (13.20)	2.66 (13.05)	2.14 (12.52)
NNR	1.74 (11.41)	2.10 (12.31)	2.49 (12.66)	2.02 (11.98)
RFR	1.88 (12.35)	2.39 (14.45)	2.70 (13.67)	2.22 (13.34)
Lasso	1.87 (12.38)	2.35 (13.36)	2.47 (12.44)	2.16 (12.72)
Ridge	1.90 (12.37)	2.27 (13.55)	2.46 (12.86)	2.14 (12.87)
ElasticNet	1.89 (12.67)	2.35 (13.36)	2.47 (12.44)	2.17 (12.86)

Table C. 450. Site 3 Local Models Evaluation

S3		In S1	In S2	Weighted average
Models	RMSE (MAPE)	RMSE (MAPE)	RMSE (MAPE)	
LR	2.28 (13.60)	2.45 (16.19)	2.19 (15.35)	2.29 (15.27)
SVR	2.34 (12.77)	2.48 (15.58)	2.15 (14.48)	2.30 (14.49)
DTR	2.49 (14.25)	2.61 (17.01)	2.30 (15.78)	2.44 (15.88)
NNR	2.26 (13.40)	2.31 (15.64)	2.07 (14.82)	2.19 (14.80)
RFR	2.54 (14.97)	2.67 (17.52)	2.35 (16.02)	2.49 (16.31)
Lasso	2.32 (14.09)	2.38 (16.07)	2.11 (15.25)	2.24 (15.28)
Ridge	2.30 (13.79)	2.37 (16.62)	2.14 (15.96)	2.25 (15.73)
ElasticNet	2.29 (13.91)	2.42 (15.74)	2.14 (14.87)	2.27 (14.96)

Table C. 451. Local Models Average RMSE (MAPE) for All sites

Models	S1 model RMSE (MAPE)	S2 model RMSE (MAPE)	S3 model RMSE (MAPE)	The best model
LR	**2.16 (13.66)**	2.18 (12.70)	2.29 (15.27)	LR-S1
SVR	**2.19 (12.72)**	2.29 (12.44)	2.30 (14.49)	SVR-S1
DTR	2.16 (13.35)	**2.14 (12.52)**	2.44 (15.88)	DTR-S2
NNR	2.03 (12.73)	**2.02 (11.98)**	2.19 (14.80)	NNR-S2
RFR	**2.22 (14.16)**	**2.22 (13.34)**	2.49 (16.31)	RFR-S1 – RFR-S2
Lasso	2.19 (13.94)	**2.16 (12.72)**	2.24 (15.28)	LASSO-S2

Ridge	2.22 (14.22)	**2.14 (12.87)**	2.25 (15.73)	RIDGE-S2
ElasticNet	2.27 (15.48)	**2.17 (12.86)**	2.27 (14.96)	ElasticNet-S2

Table C. 452. Linear Regression (LR) Model Evaluation Results After Updating Method

The best model	In S2 (LR-S12)	In S3 (LR-S13)
	After update the model	After update the model
LR-S1	2.08 (11.36)	2.49 (15.07)

Table C. 453. Linear Regression (LR) Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average RMSE (MAPE)
LR-S12	2.07 (13.54)	2.08 (11.36)	2.97 (13.51)	**2.26 (12.55)**
LR-S13	2.43 (17.41)	2.06 (16.11)	2.49 (15.07)	2.28 (16.33)

Table C. 454. Linear Regression (LR) Updated Model Evaluation Results After Updating Method

The best model	In S3 (LR-S123)
	After update the model
LR-S12	2.49 (15.07)

Table C. 455. Support Vector Regressor (SVR) Model Evaluation Results After Updating Method

The best model	In S2 (SVR -S12)	Table C. 456. In S3 (SVR -S13)
	After update the model	After update the model
SVR-S1	2.08 (11.42)	2.34 (12.49)

Table C. 457. Support Vector Regressor (SVR) Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average RMSE (MAPE)
SVR -S12	2.70 (13.61)	2.08 (11.42)	3.01 (13.69)	2.49 (12.64)
SVR -S13	2.39 (14.79)	2.04 (13.67)	2.34 (12.49)	**2.22 (13.80)**

Table C. 458. Support Vector Regressor (SVR) Updated Model Evaluation Results After Updating Method

The best model	In S2 (SVR -S132)
	After update the model
SVR -S13	2.15 (11.52)

Table C. 459. Decision Tree Regressor (DTR) Model Evaluation Results After Updating Method

The best model	In S1 (DTR-S21)	In S3 (DTR -S23)
	After update the model	After update the model
DTR -S2	1.51 (9.85)	1.48 (9.34)

Table C. 460. Decision Tree Regressor (DTR) Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average RMSE (MAPE)
DTR -S21	1.51 (9.85)	1.94 (13.17)	2.58 (13.65)	**1.93 (12.14)**
DTR -S23	2.63 (16.99)	2.30 (15.72)	1.48 (9.34)	2.24 (14.81)

Table C. 461. Decision Tree Regressor (DTR) Updated Model Evaluation Results After Updating Method

The best model	In S3 (DTR -S213)
	After update the model
DTR -S21	1.48 (9.34)

Table C. 462. Neural Network Regressor (NNR) Model Evaluation Results After Updating Method

The best model	In S1 (NNR -S21)	In S3 (NNR-S23)
	After update the model	After update the model
NNR-S2	1.95 (12.09)	2.18 (12.84)

Table C. 463. Neural Network Regressor (NNR) Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average RMSE (MAPE)
NNR -S21	1.95 (12.09)	1.83 (12.45)	2.47 (13.18)	**2.01 (12.48)**
NNR -S23	2.33 (15.73)	2.08 (14.90)	2.18 (12.84)	2.19 (14.75)

Table C. 464. Neural Network Regressor (NNR) Updated Model Evaluation Results After Updating Method

The best model	In S3 (NNR -S213)
	After update the model
NNR -S21	2.17 (12.78)

Table C. 465. Random Forest Regressor (RFR) Model Evaluation Results After Updating Method

The best model	In S2 (RFR-S12)	In S3 (RFR-S13)
	After update the model	After update the model
RFR-S1	0.80 (4.98)	1.06 (5.88)

Table C. 466. Random Forest Regressor (RFR) Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average RMSE (MAPE)
RFR-S12	2.32 (13.73)	0.80 (4.98)	2.65 (13.75)	**1.70 (9.80)**
RFR-S13	2.78 (17.62)	2.47 (16.32)	1.06 (5.88)	2.28 (14.57)

Table C. 467. Random Forest Regressor (RFR) Updated Model Evaluation Results After Updating Method

The best model	In S3 (RFR-S123)
	After update the model
RFR-S12	1.08 (5.72)

Table C. 468. Random Forest Regressor (RFR) Model Evaluation Results After Updating Method

The best model	In S1 (RFR-S21)	In S3 (RFR-S23)
	After update the model	After update the model
RFR-S2	0.94 (5.62)	1.08 (5.88)

Table C. 469. Random Forest Regressor (RFR) Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average RMSE (MAPE)
RFR-S21	0.94 (5.62)	2.07 (13.99)	2.69 (14.67)	**1.81 (11.29)**
RFR-S23	2.76 (17.88)	2.47 (16.63)	1.08 (5.88)	2.28 (14.79)

Table C. 470. Random Forest Regressor (RFR) Updated Model Evaluation Results After Updating Method

The best model	In S3 (RFR-S213)
	After update the model
RFR-S21	1.08 (5.88)

Table C. 471. LASSO Model Evaluation Results Before and After Updating Method

The best model	In S1(Lasso-S21)	In S3 (Lasso-S23)
	After update the model	After update the model
Lasso-S2	2.71 (13.34)	2.50 (14.39)

Table C. 472. LASSO Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average RMSE (MAPE)
Lasso-S21	2.71 (13.34)	2.11 (11.44)	3.05 (13.86)	2.51 (12.59)
Lasso-S23	2.38 (16.26)	1.99 (15.02)	2.50 (14.39)	**2.22 (15.31)**

Table C. 473. LASSO Updated Model Evaluation Results After Updating Method

The best model	In S1 (Lasso-S231)
	After update the model
Lasso-S23	2.30 (14.16)

Table C. 474. Ridge Model Evaluation Results After Updating Method

The best model	In S1 (Ridge-S21)	In S3 (Ridge-S23)
	After update the model	After update the model
Ridge-S2	2.72 (13.64)	2.50 (15.01)

Table C. 475. Ridge Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average RMSE (MAPE)
Ridge -S21	2.72 (13.64)	2.12 (11.60)	3.08 (14.12)	2.52 (12.82)
Ridge -S23	2.44 (17.39)	2.06 (16.01)	2.50 (15.01)	**2.28 (16.27)**

Table C. 476. Ridge Updated Model Evaluation Results After Updating Method

The best model	In S1 (Ridge -S231)
	After update the model
Ridge -S23	2.31 (13.30)

Table C. 477. ElasticNet Model Evaluation Results After Updating Method

The best model	In S1 (ElasticNet -S21)	In S3 (ElasticNet-S23)
	After update the model	After update the model
ElasticNet-S2	2.68 (13.18)	2.48 (14.06)

Table C. 478. ElasticNet Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average RMSE (MAPE)
ElasticNet -S21	2.68 (13.18)	2.12 (11.53)	3.08 (14.03)	2.51 (12.62)
ElasticNet-S23	2.41 (16.29)	1.98 (14.74)	2.48 (14.06)	**2.23 (15.12)**

Table C. 479. ElasticNet Updated Model Evaluation Results After Updating Method

The best model	In S1 (ElasticNet-S231)
	After update the model
ElasticNet -S23	2.28 (15.23)

Table C. 480. Updated Model Evaluation Results in All Sites

The best model	In S1	In S2	In S3	Average RMSE (MAPE)
LR-S123	2.43 (17.41)	2.05 (16.11)	2.49 (15.07)	2.27 (16.33)
SVR-S132	2.81 (14.04)	2.15 (11.52)	3.09 (13.99)	2.57 (12.89)
DTR-S213	2.61 (16.98)	2.29 (15.72)	1.48 (9.34)	2.23 (14.81)
NNR-S213	2.31 (15.60)	2.07 (14.82)	2.17 (12.78)	**2.17 (14.66)**
RFR-S123	2.78 (17.62)	2.47 (16.32)	1.08 (5.72)	2.28 (14.54)
RFR-S213	2.76 (17.88)	2.47 (16.63)	1.08 (5.88)	2.28 (14.79)
Lasso-S231	2.30 (14.16)	1.95 (13.35)	2.46 (12.87)	**2.18 (13.52)**
Ridge-S231	2.31 (13.30)	1.91 (12.51)	2.64 (13.41)	**2.19 (12.97)**
ElasticNet-S231	2.28 (15.23)	1.98 (14.75)	2.64 (14.65)	2.22 (14.89)

II. Non-randomly Partitioned Data:

1. Parkinson Disease (Total UPDRS):

Table C. 481. Site 1 Local Models Evaluation

S1		In S2	In S3	Weighted average
Models	RMSE (MAPE)	RMSE (MAPE)	RMSE (MAPE)	
LR	9.80 (24.61)	14.28 (66.59)	9.52 (23.37)	11.23 (38.49)
SVR	10.52 (21.29)	12.47 (58.63)	10.74 (21.44)	11.25 (34.03)
DTR	7.91 (19.34)	14.24 (65.68)	6.90 (16.58)	9.74 (34.21)
NNR	6.85 (16.72)	12.84 (59.61)	6.44 (14.93)	8.75 (30.72)
RFR	7.68 (17.83)	14.09 (64.12)	5.11 (10.90)	9.04 (31.35)
Lasso	10.12 (25.05)	14.35 (65.41)	10.16 (22.65)	11.57 (38.01)
Ridge	12.18 (29.83)	15.08 (68.88)	10.15 (24.09)	12.52 (41.27)
ElasticNet	10.44 (26.43)	14.45 (67.15)	9.58 (23.57)	11.53 (39.36)

Table C. 482. Site 2 Local Models Evaluation

S2		In S1	In S3	Weighted average
Models	RMSE (MAPE)	RMSE (MAPE)	RMSE (MAPE)	
LR	10.32 (42.88)	13.68 (26.85)	13.92 (28.04)	12.61 (32.68)
SVR	9.71 (42.28)	13.69 (29.93)	13.77 (30.41)	12.36 (34.28)
DTR	8.04 (32.45)	13.01 (30.98)	12.89 (30.42)	11.28 (31.30)
NNR	8.02 (33.06)	13.27 (32.75)	13.60 (33.49)	11.59 (33.09)
RFR	8.36 (32.59)	12.96 (28.80)	13.08 (29.08)	11.43 (30.18)
Lasso	9.92 (42.67)	14.09 (29.30)	14.26 (30.20)	12.73 (34.13)
Ridge	10.58 (42.15)	14.69 (32.49)	14.87 (33.18)	13.35 (35.99)
ElasticNet	9.87 (42.43)	14.09 (29.31)	14.26 (30.21)	12.71 (34.06)

Table C. 483. Site 3 Local Models Evaluation

S3		In S1	In S2	Weighted average
Models	RMSE (MAPE)	RMSE (MAPE)	RMSE (MAPE)	
LR	9.85 (24.99)	10.82 (25.16)	13.83 (64.79)	11.53 (38.58)
SVR	9.80 (23.03)	9.81 (23.49)	15.16 (68.56)	11.62 (38.67)

DTR	7.85 (19.97)	6.78 (17.52)	14.31 (66.70)	9.68 (35.02)
NNR	6.93 (17.33)	5.39 (13.83)	14.82 (68.83)	9.09 (33.65)
RFR	7.79 (18.90)	4.48 (10.41)	14.10 (64.50)	8.81 (31.51)
Lasso	11.73 (29.65)	9.89 (25.19)	14.25 (66.78)	11.96 (40.75)
Ridge	10.39 (25.78)	10.44 (26.22)	14.28 (66.79)	11.73 (39.87)
ElasticNet	9.99 (24.97)	10.51 (27.36)	15.64 (70.62)	12.08 (41.30)

Table C. 484. Local Models Average RMSE (MAPE) for All sites

Models	S1 model RMSE (MAPE)	S2 model RMSE (MAPE)	S3 model RMSE (MAPE)	The best model
LR	**11.23 (38.49)**	12.61 (32.68)	11.53 (38.58)	LR-S1
SVR	**11.25 (34.03)**	12.36 (34.28)	11.62 (38.67)	SVR-S1
DTR	9.74 (34.21)	11.28 (31.30)	**9.68 (35.02)**	DTR-S3
NNR	**8.75 (30.72)**	11.59 (33.09)	9.09 (33.65)	NNR-S1
RFR	9.04 (31.35)	11.43 (30.18)	**8.81 (31.51)**	RFR-S3
Lasso	**11.57 (38.01)**	12.73 (34.13)	11.96 (40.75)	LASSO-S1
Ridge	12.52 (41.27)	13.35 (35.99)	**11.73 (39.87)**	RIDGE-S3
ElasticNet	**11.53 (39.36)**	12.71 (34.06)	12.08 (41.30)	ElasticNet-S1

Table C. 485. Linear Regression (LR) Model Evaluation Results After Updating Method

The best model	In S2 (LR-S12)	In S3 (LR-S13)
	After update the model	After update the model
LR-S1	9.75 (41.06)	11.05 (29.53)

Table C. 486. Linear Regression (LR) Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average RMSE (MAPE)
LR-S12	13.60 (28.19)	9.75 (41.06)	13.67 (28.70)	12.31 (32.72)
LR-S13	11.35 (31.35)	11.54 (32.24)	11.05 (29.53)	**11.32 (31.07)**

Table C. 487. Linear Regression (LR) Updated Model Evaluation Results After Updating Method

The best model	In S2 (LR-S132)
	After update the model
LR-S13	9.85 (42.12)

Table C. 488. Support Vector Regressor (SVR) Model Evaluation Results After Updating Method

The best model	In S2 (SVR -S12)	In S3 (SVR -S13)
	After update the model	After update the model
SVR-S1	9.75 (40.70)	9.73 (22.61)

Table C. 489. Support Vector Regressor (SVR) Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average RMSE (MAPE)
SVR -S12	13.51 (27.17)	9.75 (40.70)	13.61 (27.86)	12.26 (31.99)
SVR -S13	9.88 (23.45)	14.98 (68.06)	9.73 (22.61)	**11.57 (38.35)**

Table C. 490. Support Vector Regressor (SVR) Updated Model Evaluation Results After Updating Method

The best model	In S2 (SVR -S132)
	After update the model
SVR -S13	18.11 (46.75)

Table C. 491. Decision Tree Regressor (DTR) Model Evaluation Results After Updating Method

The best model	In S1 (DTR-S31)	In S2 (DTR -S32)
	After update the model	After update the model
DTR -S3	5.95 (14.82)	6.25 (24.65)

Table C. 492. Decision Tree Regressor (DTR) Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average RMSE (MAPE)
DTR -S31	5.94 (14.82)	14.24 (65.68)	6.89 (16.58)	**9.07 (32.67)**
DTR -S32	13.03 (31.11)	6.25 (24.65)	12.92 (30.52)	10.69 (28.72)

Table C. 493. Decision Tree Regressor (DTR) Updated Model Evaluation Results After Updating Method

The best model	In S2 (DTR -S312)
	After update the model
DTR -S31	6.25 (24.65)

Table C. 494. Neural Network Regressor (NNR) Model Evaluation Results After Updating Method

The best model	In S2 (NNR -S12)	In S3 (NNR-S13)
	After update the model	After update the model
NNR -S1	9.09 (38.82)	5.50 (13.62)

Table C. 495. Neural Network Regressor (NNR) Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average RMSE (MAPE)
NNR -S12	12.79 (27.68)	9.09 (38.82)	12.95 (28.32)	11.58 (31.67)
NNR -S13	5.57 (14.21)	14.28 (66.43)	5.50 (13.62)	**8.51 (31.78)**

Table C. 496. Neural Network Regressor (NNR) Updated Model Evaluation Results After Updating Method

The best model	In S2 (NNR -S132)
	After update the model
NNR -S13	22.36 (65.89)

Table C. 497. Random Forest Regressor (RFR) Model Evaluation Results After Updating Method

The best model	In S1 (RFR-S31)	In S2 (RFR-S32)
	After update the model	After update the model
RFR-S3	3.31 (6.94)	3.61 (13.15)

Table C. 498. Random Forest Regressor (RFR) Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average RMSE (MAPE)
RFR-S31	3.31 (6.94)	14.07 (63.73)	5.22 (10.68)	**7.58 (27.44)**
RFR-S32	13.17 (29.72)	3.61 (13.15)	13.44 (29.95)	10.01 (24.16)

Table C. 499. Random Forest Regressor (RFR) Updated Model Evaluation Results After Updating Method

The best model	In S2 (RFR-S312)
	After update the model
RFR-S31	3.55 (12.03)

Table C. 500. LASSO Model Evaluation Results Before and After Updating Method

The best model	In S2 (Lasso-S12)	In S3 (Lasso-S13)
	After update the model	After update the model
Lasso-S1	9.76 (41.10)	9.83 (25.40)

Table C. 501. LASSO Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average RMSE (MAPE)
Lasso-S12	13.49 (27.26)	9.76 (41.10)	13.60 (27.99)	12.26 (32.20)
Lasso-S13	10.07 (26.57)	13.84 (65.95)	9.83 (25.40)	**11.27 (39.58)**

Table C. 502. LASSO Updated Model Evaluation Results After Updating Method

The best model	In S2 (Lasso-S132)
	After update the model
Lasso-S13	11.13 (48.08)

Table C. 503. Ridge Model Evaluation Results After Updating Method

The best model	In S1 (Ridge-S31)	In S2 (Ridge-S32)
	After update the model	After update the model
Ridge-S3	10.46 (24.71)	9.77 (40.32)

Table C. 504. Ridge Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average RMSE (MAPE)
Ridge -S31	10.46 (24.71)	12.74 (62.24)	10.36 (24.31)	**11.20 (37.34)**
Ridge -S32	13.77 (28.08)	9.77 (40.32)	13.86 (28.70)	12.44 (32.44)

Table C. 505. Ridge Updated Model Evaluation Results After Updating Method

The best model	In S2 (Ridge -S312)
	After update the model
Ridge -S31	9.55 (40.98)

Table C. 506. ElasticNet Model Evaluation Results After Updating Method

The best model	In S2 (ElasticNet -S12)	In S3 (ElasticNet-S13)
	After update the model	After update the model
ElasticNet-S1	9.79 (41.01)	9.92 (26.51)

Table C. 507. ElasticNet Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average RMSE (MAPE)
ElasticNet -S12	13.47 (26.99)	9.79 (41.01)	13.57 (27.68)	12.25 (31.98)
ElasticNet-S13	10.27 (28.24)	13.65 (66.48)	9.92 (26.51)	**11.31 (40.69)**

Table C. 508. ElasticNet Updated Model Evaluation Results After Updating Method

The best model	In S2 (ElasticNet-S132)
	After update the model
ElasticNet -S132	9.53 (39.91)

Table C. 509. Updated Model Evaluation Results in All Sites

The best model	In S1	In S2	In S3	Average RMSE (MAPE)
LR-S132	13.38 (31.91)	9.85 (42.12)	14.37 (31.85)	12.49 (35.36)
SVR-S132	22.45 (52.73)	18.11 (46.75)	22.98 (53.88)	21.14 (51.06)
DTR-S312	12.99 (31.22)	6.25 (24.65)	12.88 (30.62)	**10.66 (28.79)**
NNR-S132	27.18 (67.81)	22.36 (65.89)	27.77 (68.65)	25.73 (67.43)
RFR-S312	12.31 (28.58)	3.55 (12.03)	12.54 (28.67)	**9.40 (22.98)**
Lasso-S132	13.67 (31.30)	11.13 (48.08)	13.66 (31.02)	12.80 (36.91)
Ridge-S312	13.46 (28.50)	9.55 (40.98)	13.54 (28.97)	**12.16 (32.89)**
ElasticNet-132	13.95 (29.36)	9.53 (39.91)	14.01 (29.71)	12.47 (33.06)

2. Parkinson Disease (Motor UPDRS):

Table C. 510. Site 1 Local Models Evaluation

S1		In S2	In S3	Weighted average
Models	RMSE (MAPE)	RMSE (MAPE)	RMSE (MAPE)	
LR	6.04 (22.33)	11.17 (72.17)	6.54 (23.33)	7.94 (39.59)
SVR	6.09 (21.20)	11.26 (72.66)	6.49 (21.63)	7.97 (38.83)
DTR	5.02 (18.44)	11.06 (71.39)	4.96 (16.73)	7.05 (35.89)
NNR	4.49 (15.69)	11.07 (71.23)	4.72 (14.95)	6.80 (34.34)
RFR	5.06 (17.82)	11.12 (70.86)	3.99 (11.40)	6.78 (33.80)
Lasso	6.19 (22.91)	10.84 (71.48)	6.31 (23.35)	7.81 (39.56)
Ridge	6.05 (22.68)	10.61 (69.25)	6.25 (21.85)	7.66 (38.25)
ElasticNet	6.15 (23.09)	10.58 (69.16)	6.44 (23.34)	7.75 (38.83)

Table C. 511. Site 2 Local Models Evaluation

S2		In S1	In S3	Weighted average
Models	RMSE (MAPE)	RMSE (MAPE)	RMSE (MAPE)	
LR	9.25 (50.90)	10.45 (36.98)	10.71 (37.33)	10.12 (41.82)
SVR	7.73 (46.84)	9.23 (31.75)	9.52 (32.38)	8.81 (37.08)
DTR	6.17 (33.83)	8.84 (33.87)	9.07 (33.86)	8.01 (33.85)
NNR	5.95 (32.14)	11.27 (39.72)	12.05 (41.15)	9.71 (37.60)
RFR	6.40 (33.45)	8.87 (33.97)	8.99 (33.28)	8.07 (33.57)
Lasso	7.88 (47.01)	10.45 (34.82)	10.81 (35.76)	9.69 (39.26)
Ridge	7.81 (46.12)	10.51 (36.13)	10.89 (36.98)	9.71 (39.80)
ElasticNet	7.99 (47.05)	10.45 (34.82)	10.81 (35.76)	9.73 (39.28)

Table C. 512. Site 3 Local Models Evaluation

S3		In S1	In S2	Weighted average
Models	RMSE (MAPE)	RMSE (MAPE)	RMSE (MAPE)	
LR	6.38 (23.61)	6.39 (24.59)	11.58 (74.66)	8.15 (41.30)
SVR	6.23 (22.93)	6.05 (23.12)	12.11 (78.09)	8.17 (41.75)
DTR	5.36 (19.95)	4.39 (16.86)	11.43 (74.82)	7.09 (37.55)
NNR	5.04 (17.72)	3.93 (14.44)	11.82 (76.64)	6.97 (36.64)
RFR	5.42 (19.26)	2.86 (9.94)	11.48 (74.21)	6.61 (34.77)
Lasso	7.33 (26.97)	5.97 (23.14)	11.19 (74.02)	8.18 (41.66)
Ridge	6.52 (24.99)	6.76 (27.14)	12.92 (83.43)	8.78 (45.59)
ElasticNet	6.60 (24.76)	6.09 (24.09)	11.76 (77.46)	8.18 (42.45)

Table C. 513. Local Models Average RMSE (MAPE) for All sites

Models	S1 model RMSE (MAPE)	S2 model RMSE (MAPE)	S3 model RMSE (MAPE)	The best model
LR	**7.94 (39.59)**	10.12 (41.82)	8.15 (41.30)	LR-S1
SVR	**7.97 (38.83)**	8.81 (37.08)	8.17 (41.75)	SVR-S1
DTR	**7.05 (35.89)**	8.01 (33.85)	7.09 (37.55)	DTR-S1
NNR	**6.80 (34.34)**	9.71 (37.60)	6.97 (36.64)	NNR-S1
RFR	6.78 (33.80)	8.07 (33.57)	**6.61 (34.77)**	RFR-S3
Lasso	**7.81 (39.56)**	9.69 (39.26)	8.18 (41.66)	LASSO-S1
Ridge	**7.66 (38.25)**	9.71 (39.80)	8.78 (45.59)	RIDGE-S1
ElasticNet	**7.75 (38.83)**	9.73 (39.28)	8.18 (42.45)	ElasticNet-S1

Table C. 514. Linear Regression (LR) Model Evaluation Results After Updating Method

The best model	In S2 (LR-S12)	In S3 (LR-S13)
	After update the model	After update the model
LR-S1	7.87 (44.44)	6.92 (26.76)

Table C. 515. Linear Regression (LR) Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average RMSE (MAPE)
LR-S12	9.41 (31.77)	7.87 (44.44)	9.69 (32.35)	8.98 (36.26)
LR-S13	6.83 (27.27)	11.67 (77.61)	6.92 (26.76)	**8.50 (44.22)**

Table C. 516. Linear Regression (LR) Updated Model Evaluation Results After Updating Method

The best model	In S2 (LR-S132)
	After update the model
LR-S13	7.87 (46.82)

Table C. 517. Support Vector Regressor (SVR) Model Evaluation Results After Updating Method

The best model	In S2 (SVR -S12)	In S3 (SVR -S13)
	After update the model	After update the model
SVR-S1	7.43 (44.84)	6.19 (22.59)

Table C. 518. Support Vector Regressor (SVR) Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average RMSE (MAPE)
SVR -S12	9.68 (32.87)	7.43 (44.84)	9.99 (33.56)	9.01 (37.16)
SVR -S13	6.11 (23.12)	12.81 (80.79)	6.19 (22.59)	**8.41 (42.56)**

Table C. 519. Support Vector Regressor (SVR) Updated Model Evaluation Results After Updating Method

The best model	In S2 (SVR -S132)
	After update the model
SVR -S13	7.64 (46.31)

Table C. 520. Decision Tree Regressor (DTR) Model Evaluation Results After Updating Method

The best model	In S2 (DTR-S12)	In S3 (DTR -S13)
	After update the model	After update the model
DTR -S1	4.76 (25.82)	4.26 (15.78)

Table C. 521. Decision Tree Regressor (DTR) Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average RMSE (MAPE)
DTR -S12	8.84 (33.87)	4.76 (25.82)	9.07 (33.85)	7.53 (31.13)
DTR -S13	4.39 (16.87)	11.42 (74.81)	4.26 (15.78)	**6.74 (36.22)**

Table C. 522. Decision Tree Regressor (DTR) Updated Model Evaluation Results After Updating Method

The best model	In S2 (DTR -S132)
	After update the model
DTR -S13	4.76 (25.82)

Table C. 523. Neural Network Regressor (NNR) Model Evaluation Results After Updating Method

The best model	In S2 (NNR -S12)	In S3 (NNR-S13)
	After update the model	After update the model
NNR -S1	6.84 (39.92)	3.82 (13.38)

Table C. 524. Neural Network Regressor (NNR) Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average RMSE (MAPE)
NNR -S12	8.98 (31.18)	6.84 (39.92)	9.55 (32.35)	8.43 (34.53)
NNR -S13	3.76 (13.79)	12.09 (78.49)	3.82 (13.38)	**6.61 (35.66)**

Table C. 525. Neural Network Regressor (NNR) Updated Model Evaluation Results After Updating Method

The best model	In S2 (NNR -S132)
	After update the model
NNR -S13	5.63 (31.13)

Table C. 526. Random Forest Regressor (RFR) Model Evaluation Results After Updating Method

The best model	In S1 (RFR-S31)	In S2 (RFR-S32)
	After update the model	After update the model
RFR-S3	2.19 (6.94)	2.72 (13.24)

Table C. 527. Random Forest Regressor (RFR) Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average RMSE (MAPE)
RFR-S31	2.19 (6.94)	11.39 (72.32)	4.08 (11.37)	**5.92 (30.59)**
RFR-S32	8.49 (32.70)	2.72 (13.24)	8.74 (32.69)	6.61 (26.08)

Table C. 528. Random Forest Regressor (RFR) Updated Model Evaluation Results After Updating Method

The best model	In S2 (RFR-S312)
	After update the model
RFR-S31	2.62 (12.35)

Table C. 529. LASSO Model Evaluation Results Before and After Updating Method

The best model	In S2 (Lasso-S12)	In S3 (Lasso-S13)
	After update the model	After update the model
Lasso-S1	7.86 (44.22)	6.26 (23.97)

Table C. 530. LASSO Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average RMSE (MAPE)
Lasso-S12	9.66 (32.57)	7.86 (44.22)	9.92 (33.09)	9.13 (36.69)
Lasso-S13	6.08 (24.02)	10.19 (68.92)	6.26 (23.97)	**7.53 (39.27)**

Table C. 531. LASSO Updated Model Evaluation Results After Updating Method

The best model	In S2 (Lasso-S132)
	After update the model
Lasso-S13	8.01 (46.65)

Table C. 532. Ridge Model Evaluation Results After Updating Method

The best model	In S2 (Ridge-S12)	In S3 (Ridge-S13)
	After update the model	After update the model
Ridge-S1	7.97 (46.66)	7.85 (29.35)

Table C. 533. Ridge Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average RMSE (MAPE)
Ridge -S12	9.60 (32.69)	7.97 (46.66)	9.82 (33.05)	**9.11 (37.55)**
Ridge -S13	7.82 (30.27)	12.59 (79.71)	7.85 (29.35)	9.45 (46.78)

Table C. 534. Ridge Updated Model Evaluation Results After Updating Method

The best model	In S3 (Ridge -S123)
	After update the model
Ridge -S12	6.92 (26.76)

Table C. 535. ElasticNet Model Evaluation Results After Updating Method

The best model	In S2 (ElasticNet -S12)	In S3 (ElasticNet-S13)
	After update the model	After update the model
ElasticNet-S1	7.88 (45.15)	7.25 (26.53)

Table C. 536. ElasticNet Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average RMSE (MAPE)
ElasticNet -S12	9.55 (32.46)	7.88 (45.15)	9.78 (32.85)	9.05 (36.89)
ElasticNet-S13	7.04 (26.52)	11.16 (72.31)	7.25 (26.53)	**8.51 (42.09)**

Table C. 537. ElasticNet Updated Model Evaluation Results After Updating Method

The best model	In S2 (ElasticNet-S132)
	After update the model
ElasticNet -S13	7.62 (45.95)

Table C. 538. Updated Model Evaluation Results in All Sites

The best model	In S1	In S2	In S3	Average RMSE (MAPE)
LR-S132	10.43 (36.11)	7.87 (46.82)	10.57 (36.01)	9.60 (39.72)
SVR-S132	9.23 (32.23)	7.64 (46.31)	9.49 (32.69)	8.77 (37.16)
DTR-S132	8.84 (33.89)	4.76 (25.82)	9.08 (33.88)	**7.53 (31.14)**

NNR-S132	11.68 (41.46)	5.63 (31.13)	12.46 (42.89)	9.87 (38.40)
RFR-S312	8.85 (34.74)	2.62 (12.35)	8.98 (34.22)	**6.77 (26.96)**
Lasso-S132	9.30 (31.55)	8.01 (46.65)	9.64 (32.43)	8.97 (36.96)
Ridge-S123	6.83 (27.27)	11.67 (77.61)	6.92 (26.76)	**8.50 (44.22)**
ElasticNet-S132	9.79 (33.51)	7.62 (45.95)	10.06 (34.04)	9.14 (37.91)

3. Boston Housing:

Table C. 539. Site 1 Local Models Evaluation

S1		In S2	In S3	Weighted average
Models	RMSE (MAPE)	RMSE (MAPE)	RMSE (MAPE)	
LR	3.22 (10.38)	3.98 (15.84)	12.82 (81.04)	7.29 (40.28)
SVR	3.35 (10.51)	3.86 (15.07)	12.36 (78.24)	7.11 (38.97)
DTR	2.95 (8.49)	4.12 (15.25)	12.07 (78.33)	6.95 (38.45)
NNR	2.76 (9.04)	5.57 (18.60)	13.27 (72.93)	7.81 (37.46)
RFR	3.15 (9.53)	3.90 (14.52)	12.15 (77.16)	6.97 (38.08)
Lasso	3.18 (10.65)	4.01 (16.03)	12.89 (81.50)	7.31 (40.60)
Ridge	3.25 (10.59)	3.94 (15.67)	12.82 (80.89)	7.28 (40.23)
ElasticNet	3.35 (11.02)	4.03 (16.23)	12.86 (81.43)	7.36 (40.75)

Table C. 540. Site 2 Local Models Evaluation

S2		In S1	In S3	Weighted average
Models	RMSE (MAPE)	RMSE (MAPE)	RMSE (MAPE)	
LR	3.01 (10.46)	4.10 (11.89)	11.18 (65.99)	6.60 (33.10)
SVR	3.16 (10.76)	4.62 (13.58)	10.77 (62.58)	6.64 (32.33)
DTR	2.69 (8.84)	4.35 (12.22)	10.19 (59.45)	6.19 (30.09)
NNR	3.32 (11.17)	4.13 (11.82)	9.51 (56.46)	6.04 (29.48)
RFR	2.80 (8.97)	4.56 (11.91)	10.63 (61.87)	6.46 (31.01)
Lasso	2.85 (9.78)	4.22 (12.26)	11.10 (65.03)	6.56 (32.62)
Ridge	2.95 (10.18)	4.20 (12.48)	10.99 (64.55)	6.54 (32.62)
ElasticNet	2.98 (10.26)	4.22 (12.26)	11.10 (65.03)	6.6 (32.77)

Table C. 541. Site 3 Local Models Evaluation

S3		In S1	In S2	Weighted average
Models	RMSE (MAPE)	RMSE (MAPE)	RMSE (MAPE)	
LR	6.29 (27.94)	10.70 (34.16)	9.37 (27.55)	8.54 (29.69)
SVR	6.36 (20.10)	12.14 (38.39)	10.44 (30.39)	9.32 (28.67)
DTR	3.93 (16.89)	10.47 (32.66)	8.86 (27.78)	7.37 (24.89)
NNR	4.17 (18.15)	11.43 (38.85)	9.83 (36.59)	8.05 (29.89)
RFR	4.22 (17.75)	10.14 (32.20)	8.68 (27.26)	7.33 (24.94)
Lasso	6.33 (27.99)	10.65 (33.79)	9.28 (26.17)	8.51 (29.18)
Ridge	6.30 (27.72)	10.61 (35.01)	9.08 (28.19)	8.43 (30.05)
ElasticNet	5.92 (26.81)	10.74 (34.60)	9.26 (27.12)	8.37 (29.24)

Table C. 542. Local Models Average RMSE (MAPE) for All sites

Models	S1 model RMSE (MAPE)	S2 model RMSE (MAPE)	S3 model RMSE (MAPE)	The best model
LR	7.29 (40.28)	**6.60 (33.10)**	8.54 (29.69)	LR-S2
SVR	7.11 (38.97)	**6.64 (32.33)**	9.32 (28.67)	SVR-S2
DTR	6.95 (38.45)	**6.19 (30.09)**	7.37 (24.89)	DTR-S2
NNR	7.81 (37.46)	**6.04 (29.48)**	8.05 (29.89)	NNR-S2
RFR	6.97 (38.08)	**6.46 (31.01)**	7.33 (24.94)	RFR-S2
Lasso	7.31 (40.60)	**6.56 (32.62)**	8.51 (29.18)	LASSO-S2
Ridge	7.28 (40.23)	**6.54 (32.62)**	8.43 (30.05)	RIDGE-S2
ElasticNet	7.36 (40.75)	**6.6 (32.77)**	8.37 (29.24)	ElasticNet-S2

Table C. 543. Linear Regression (LR) Model Evaluation Results After Updating Method

The best model	In S1 (LR-S21)	In S3 (LR-S23)
	After update the model	After update the model
LR-S2	17.60 (63.94)	7.35 (26.78)

Table C. 544. Linear Regression (LR) Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average RMSE (MAPE)
LR-S21	17.60 (63.94)	15.23 (60.11)	11.03 (40.22)	14.26 (53.30)
LR-S23	14.34 (50.33)	12.23 (44.43)	7.35 (26.78)	**10.91 (39.14)**

Table C. 545. Linear Regression (LR) Updated Model Evaluation Results After Updating Method

The best model	In S1 (LR-S231)
	After update the model
LR-S23	9.17 (31.91)

Table C. 546. Support Vector Regressor (SVR) Model Evaluation Results After Updating Method

The best model	In S1 (SVR -S21)	In S3 (SVR -S23)
	After update the model	After update the model
SVR-S2	17.78 (65.22)	6.62 (17.87)

Table C. 547. Support Vector Regressor (SVR) Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average RMSE (MAPE)
SVR -S21	17.78 (65.22)	15.23 (61.68)	11.16 (41.16)	14.36 (54.53)
SVR -S23	12.08 (38.30)	10.37 (30.28)	6.62 (17.87)	**9.38 (27.72)**

Table C. 548. Support Vector Regressor (SVR) Updated Model Evaluation Results After Updating Method

The best model	In S1 (SVR -S231)
	After update the model
SVR -S23	3.16 (8.79)

Table C. 549. Decision Tree Regressor (DTR) Model Evaluation Results After Updating Method

The best model	In S1 (DTR-S21)	In S3 (DTR -S23)
	After update the model	After update the model
DTR -S2	0.31 (1.01)	0.71 (3.78)

Table C. 550. Decision Tree Regressor (DTR) Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average RMSE (MAPE)
DTR -S21	0.31 (1.01)	4.16 (15.50)	12.08 (78.40)	6.17 (36.31)
DTR -S23	10.46 (32.63)	8.85 (27.78)	0.71 (3.78)	**6.08 (19.63)**

Table C. 551. Decision Tree Regressor (DTR) Updated Model Evaluation Results After Updating Method

The best model	In S1 (DTR -S231)
	After update the model
DTR -S23	0.31 (1.01)

Table C. 552. Neural Network Regressor (NNR) Model Evaluation Results After Updating Method

The best model	In S1(NNR -S21)	In S3 (NNR-S23)
	After update the model	After update the model
NNR -S2	27.69 (97.40)	2.20 (9.93)

Table C. 553. Neural Network Regressor (NNR) Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average RMSE (MAPE)
NNR -S21	27.69 (97.40)	25.35 (96.90)	20.01 (96.20)	23.92 (96.77)
NNR -S23	10.64 (34.95)	9.58 (33.76)	2.20 (9.93)	**6.95 (24.58)**

Table C. 554. Neural Network Regressor (NNR) Updated Model Evaluation Results After Updating Method

The best model	In S1 (NNR -S231)
	After update the model
NNR -S23	1.05 (2.82)

Table C. 555. Random Forest Regressor (RFR) Model Evaluation Results After Updating Method

The best model	In S1 (RFR-S21)	In S3 (RFR-S23)
	After update the model	After update the model
RFR-S2	1.28 (3.47)	1.85 (7.55)

Table C. 556. Random Forest Regressor (RFR) Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average RMSE (MAPE)
RFR-S21	1.28 (3.47)	4.10 (15.48)	12.31 (79.01)	6.53 (37.29)
RFR-S23	10.27 (32.74)	8.43 (26.66)	1.85 (7.55)	**6.35 (20.84)**

Table C. 557. Random Forest Regressor (RFR) Updated Model Evaluation Results After Updating Method

The best model	In S1 (RFR-S231)
	After update the model
RFR-S23	1.51 (3.56)

Table C. 558. LASSO Model Evaluation Results Before and After Updating Method

The best model	In S1(Lasso-S21)	In S3 (Lasso-S23)
	After update the model	After update the model
Lasso-S2	17.58 (63.70)	7.64 (27.82)

Table C. 559. LASSO Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average RMSE (MAPE)
Lasso-S21	17.58 (63.70)	15.10 (59.90)	11.01 (39.32)	14.21 (52.81)
Lasso-S23	14.52 (51.46)	12.55 (45.68)	7.64 (27.82)	**11.18 (40.27)**

Table C. 560. LASSO Updated Model Evaluation Results After Updating Method

The best model	In S1 (Lasso-S231)
	After update the model
Lasso-S23	8.93 (30.77)

Table C. 561. Ridge Model Evaluation Results After Updating Method

The best model	In S1 (Ridge-S21)	In S3 (Ridge-S23)
	After update the model	After update the model
Ridge-S2	17.62 (63.89)	7.52 (26.11)

Table C. 562. Ridge Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average RMSE (MAPE)
Ridge -S21	17.62 (63.89)	15.18 (60.01)	11.09 (39.49)	14.28 (52.97)
Ridge -S23	14.48 (50.75)	12.51 (44.90)	7.52 (26.11)	**11.10 (39.14)**

Table C. 563. Ridge Updated Model Evaluation Results After Updating Method

The best model	In S1 (Ridge -S132)
	After update the model
Ridge -S23	9.16 (31.46)

Table C. 564. ElasticNet Model Evaluation Results After Updating Method

The best model	In S1 (ElasticNet -S21)	In S3 (ElasticNet-S23
	After update the model	After update the model
ElasticNet-S2	17.51 (64.17)	7.48 (29.02)

Table C. 565. ElasticNet Updated Model Evaluation Results

The best model	In S1	In S2	In S3	Average RMSE (MAPE)
ElasticNet -S21	17.51 (64.17)	15.11 (60.40)	11.13 (39.78)	14.24 (53.28)
ElasticNet-S23	14.28 (50.94)	12.25 (45.23)	7.48 (29.02)	**10.95 (40.46)**

Table C. 566. ElasticNet Updated Model Evaluation Results After Updating Method

The best model	In S1 (ElasticNet-S231)
	After update the model
ElasticNet -S23	9.39 (32.83)

Table C. 567. Updated Model Evaluation Results in All Sites

The best model	In S1	In S2	In S3	Average RMSE (MAPE)
LR-S231	9.17 (31.91)	6.96 (25.20)	7.56 (33.17)	7.86 (30.40)
SVR-S231	3.16 (8.79)	9.76 (34.65)	19.32 (92.51)	11.60 (50.03)
DTR-S231	0.31 (1.01)	4.10 (15.12)	12.09 (78.48)	**6.16 (36.23)**
NNR-S231	1.05 (2.82)	6.62 (28.46)	19.63 (96.86)	10.15 (48.13)
RFR-S231	1.51 (3.56)	3.99 (15.55)	12.28 (77.52)	**6.56 (36.74)**
Lasso-S231	8.93 (30.77)	6.68 (23.77)	7.80 (35.13)	**7.80 (30.41)**
Ridge-S231	9.16 (31.46)	6.90 (23.57)	7.84 (36.25)	7.95 (31.01)
ElasticNet-S231	9.39 (32.83)	7.39 (26.88)	7.49 (31.73)	8.03 (30.60)

Local-level Modelling:

a) Classification

I. Randomly Partitioned Data:

1. Blood Transfusion:

Tables C.568-C.570 show the local-level modelling steps results in each site for blood transfusion dataset. It shows the accuracy of the selected models from other sites, the models accuracy after updating process using mini-batch SGD, and the selected best updated models. Finally, the selected models are combined with the best local model using linear combination methods.

Table C. 568. Site 1 Local-level Modelling Evaluation Results for Blood Transfusion Dataset

Selected models in S1	Accuracy	Updated Model Accuracy	Final models in S1
NN1 (Best Local Model)	74%		NN1 (Best Local Model)
SVM2 (linear)	76%	73%	SVM2 (linear)
NN2	74%	73%	NN2
RF3	76%	92%	RF3
NN3	74%	74%	NN3
DT3	74%	88%	DT3

Table C. 569. Site 2 Local-level Modelling Evaluation Results for Blood Transfusion Dataset

Selected models in S2	Accuracy	Updated Model Accuracy	Final models in S2
LR2 (Best Local Model)	87%		LR2 (Best Local Model)
SVM1 (linear)	87%	85%	SVM1 (linear)
NN1	89%	77%	NN1
DT1	89%	93%	DT1
RF3	87%	96%	RF3
NN3	89%	89%	NN3
DT3	88%	95%	DT3

Table C. 570. Site 3 Local-level Modelling Evaluation Results for Blood Transfusion Dataset

Selected models in S3	Accuracy	Updated Model Accuracy	Final models in S3
NN3 (Best Local Model)	76%		NN3 (Best Local Model)
NB1	78%	77%	NB1
SVM1 linear	77%	75%	SVM1 linear
NN1	77%	76%	NN1
DT1	78%	92%	DT1
RF2	77%	93%	RF2
SVM2 non	78%	76%	SVM2 non
SVM2 linear	80%	68%	
NN2	77%	65%	

We applied the local level modelling approach on the rest of regression datasets.

2. Breast Cancer Wisconsin (Diagnostic):

Table C. 571. Local-level Modelling in site 1

Selected models in S1	Accuracy	Accuracy after update the models	Final models in S1
NN1	96%		NN1
LR2	88%	96%	LR2
LR3	92%	95%	LR3

Table C. 572. Local-level Modelling in site 2

Selected models in S2	Accuracy	Accuracy after update the models	Final models in S2
NN2	97%		NN2
NN1	87%	74%	NN1
SVM linear 3	97%	95%	SVM linear 3

Table C. 573. Local-level Modelling in site 3

Selected models in S3	Accuracy	Accuracy after update the models	Final models in S3
LR3	99%		LR3
RF1	97%	99%	RF1
NB2	97%	98%	NB2

3. Diabetes:

Table C. 574. Local-level Modelling in site 1

Selected models in S1	Accuracy	Accuracy after update the models	Final models in S1
NN1	73%		NN1
NB2	73%	73%	NB2
NN3	73%	64%	

Table C. 575. Local-level Modelling in site 2

Selected models in S2	Accuracy	Accuracy after update the models	Final models in S2
NN2	75%		NN2
NB1	75%	73%	NB1
NN1	76%	63%	
LR3	75%	75%	LR3
NN3	75%	75%	NN3

Table C. 576. Local-level Modelling in site 3

Selected models in S3	Accuracy	Accuracy after update the models	Final models in S3
NN3	81%		NN3
NN1	82%	72%	NN1
DT1	81%	99%	DT1
NB2	80%	82%	NB2

4. Heart Disease:

Table C. 577. Local-level Modelling in site 1

Selected models in S1	Accuracy	Accuracy after update the models	Final models in S1
NB1	85%		NB1
NB2	84%	83%	NB2
RF3	87%	99%	RF3

Table C. 578. Local-level Modelling in site 2

Selected models in S2	Accuracy	Accuracy after update the models	Final models in S2
NB2	75%		NB2
NB1	73%	81%	NB1
RF3	75%	99%	RF3
NB3	75%	81%	NB3

Table C. 579. Local-level Modelling in site 3

Selected models in S3	Accuracy	Accuracy after update the models	Final models in S3
RF3	82%		RF3
DT1	87%	96%	DT1
NB2	82%	85%	NB2
DT2	85%	96%	DT2

5. Spine Disease:

Table C. 580. Local-level Modelling in site 1

Selected models in S1	Accuracy	Accuracy after update the models	Final models in S1
RF1	97%		RF1
RF2	90%	98%	RF2
LR3	87%	84%	LR3

Table C. 581. Local-level Modelling in site 2

Selected models in S2	Accuracy	Accuracy after update the models	Final models in S2
NB2	87%		NB2
LR1	90%	84%	LR1
RF1	90%	97%	RF1
SVM1 (nonlinear)	94%	78%	SVM1 (nonlinear)
SVM1 (linear)	90%	86%	SVM1 (linear)
RF3	92%	98%	RF3
SVM3 (nonlinear)	92%	86%	SVM3 (nonlinear)
SVM3 (linear)	90%	78%	SVM3 (linear)
DT3	92%	76%	DT3

Table C. 582. Local-level Modelling in site 3

Selected models in S3	Accuracy	Accuracy after update the models	Final models in S3
DT3	80%		DT3
LR1	72%	75%	LR1
NB2	78%	77%	NB2

6. Breast Cancer Wisconsin (Original):

Table C. 583. Local-level Modelling in site 1

Selected models in S1	Accuracy	Accuracy after update the models	Final models in S1
LR1	96%		LR1
LR2	97%	95%	LR2
SVM2 (nonlinear)	97%	97%	SVM2 (nonlinear)
NN2	97%	81%	NN2
NB3	92%	97%	NB3

Table C. 584. Local-level Modelling in site 2

Selected models in S2	Accuracy	Accuracy after update the models	Final models in S2
RF2	96%		RF2
SVM1 (nonlinear)	96%	97%	SVM1 (nonlinear)
SVM3 (nonlinear)	96%	96%	SVM3 (nonlinear)

Table C. 585. Local-level Modelling in site 3

Selected models in S3	Accuracy	Accuracy after update the models	Final models in S3
LR3	99%		LR3
NN1	98%	97%	NN1
NN2	99%	93%	NN2

7. Liver Disease:

Table C. 586. Local-level Modelling in site 1

Selected models in S1	Accuracy	Accuracy after update the models	Final models in S1
DT1	74%		DT1
SVM2 (nonlinear)	73%	74%	SVM2 (nonlinear)
DT3	74%	95%	DT3

Table C. 587. Local-level Modelling in site 2

Selected models in S2	Accuracy	Accuracy after update the models	Final models in S2
NN2	67%		NN2
LR1	68%	64%	
SVM1 (nonlinear)	70%	71%	SVM1 (nonlinear)
DT3	66%	94%	DT3

Table C. 588. Local-level Modelling in site 3

Selected models in S3	Accuracy	Accuracy after update the models	Final models in S3
SVM 3 (nonlinear)	68%		SVM 3 (nonlinear)
SVM1 (nonlinear)	67%	68%	SVM1 (nonlinear)
NN2	69%	66%	NN2

8. Cardiovascular Disease:

Table C. 589. Local-level Modelling in site 1

Selected models in S1	Accuracy	Accuracy after update the models	Final models in S1
NN1	73%		NN1
DT2	73%	74%	DT2
NN3	72%	61%	

Table C. 590. Local-level Modelling in site 2

Selected models in S2	Accuracy	Accuracy after update the models	Final models in S2
NN2	73%		NN2
NN1	73%	67%	
DT1	73%	74%	DT1
NN3	73%	73%	NN3

Table C. 591. Local-level Modelling in site 3

Selected models in S3	Accuracy	Accuracy after update the models	Final models in S3
NN3	73%		NN3
NB1	73%	71%	NB1
NN1	73%	61%	
NN2	73%	66%	
DT2	73%	74%	DT2

II. Non-Randomly Partitioned Data:

1. Diabetes:

Table C. 592. Local-level Modelling in site 1

Selected models in S1	Accuracy	Accuracy after update the models	Final models in S1
NB1	84%		NB1
RF2	73%	92%	RF2
RF3	61%	91%	RF3

Table C. 593. Local-level Modelling in site 2

Selected models in S2	Accuracy	Accuracy after update the models	Final models in S2
DT2	67%		DT2
SVM1 (nonlinear)	66%	69%	SVM1 (nonlinear)
NN3	65%	62%	NN3

Table C. 594. Local-level Modelling in site 3

Selected models in S3	Accuracy	Accuracy after update the models	Final models in S3
SVM3 (nonlinear)	68%		SVM3 (nonlinear)
NB1	60%	68%	
LR2	72%	60%	
NB2	69%	70%	NB2
SVM2 linear	69%	71%	SVM2 linear
NN2	74%	63%	
DT2	72%	97%	DT2

2. Heart Disease:

Table C. 595. Local-level Modelling in site 1

Selected models in S1	Accuracy	Accuracy after update the models	Final models in S1
LR1	90%		LR1
DT2	89%	96%	DT2
DT3	80%	96%	DT3

Table C. 596. Local-level Modelling in site 2

Selected models in S2	Accuracy	Accuracy after update the models	Final models in S2
NN2	83%		NN2
RF1	78%	95%	RF1
NB3	78%	78%	NB3

Table C. 597. Local-level Modelling in site 3

Selected models in S3	Accuracy	Accuracy after update the models	Final models in S3
NB3	81%		NB3
RF1	72%	95%	RF1
NB2	84%	85%	NB2
SVM 2 (nonlinear)	81%	75%	SVM 2 (nonlinear)
NN2	81%	71%	NN2
DT2	84%	98%	DT2

3. Liver Disease:

Table C. 598. Local-level Modelling in site 1

Selected models in S1	Accuracy	Accuracy after update the models	Final models in S1
RF1	66%		RF1
NB2	65%	65%	
RF3	67%	95%	RF3
DT3	69%	95%	DT3

Table C. 599. Local-level Modelling in site 2

Selected models in S2	Accuracy	Accuracy after update the models	Final models in S2
NN2	76%		NN2
SVM1 linear	68%	71%	SVM1 linear
NN3	76%	70%	NN3

Table C. 600. Local-level Modelling in site 3

Selected models in S3	Accuracy	Accuracy after update the models	Final models in S3
NN3	73%		NN3
SVM1 linear	71%	72%	SVM1 linear
LR2	73%	73%	LR2
SVM 2 (nonlinear)	75%	75%	SVM 2 (nonlinear)
NN2	74%	70%	NN2
DT2	74%	94%	DT2

b) Regression

I. Randomly Partitioned Data:

1. Parkinson Disease (Total UPDRS):

Table C. 601. Local-level Modelling in site 1

Selected models in S1	RMSE	RMSE (MAPE) after update the models	Final models in S1
NNR1	7.66		NNR1
LR2	11.14	10.84 (29.10)	LR2
LR3	13.67	10.83 (29.22)	LR3

Table C. 602. Local-level Modelling in site 2

Selected models in S2	RMSE	RMSE (MAPE) after update the models	Final models in S2
NNR2	5.06		NNR2
NNR1	12.18	8.22 (25.95)	NNR1
NNR3	8.51	6.76 (21.33)	NNR3

Table C. 603. Local-level Modelling in site 3

Selected models in S3	RMSE	RMSE (MAPE) after update the models	Final models in S3
NNR3	5.85		NNR3
SVR1	11.26	8.75 (27.90)	SVR1
DTR2	8.96	10.19 (33.19)	

2. Parkinson Disease (Motor UPDRS):

Table C. 604. Local-level Modelling in site 1

Selected models in S1	RMSE	RMSE (MAPE) after update the models	Final models in S1
NNR1	5.28		NNR1
RFR2	11.33	2.48 (8.52)	RFR2
ElasticNet3	9.42	7.37 (29.67)	ElasticNet3

Table C. 605. Local-level Modelling in site 2

Selected models in S2	RMSE	RMSE (MAPE) after update the models	Final models in S2
NNR2	3.86		NNR2
ElasticNet1	8.07	5.47 (21.73)	ElasticNet1
NNR3	6.71	18.82 (84.64)	

Table C. 606. Local-level Modelling in site 3

Selected models in S3	RMSE	RMSE (MAPE) after update the models	Final models in S3
NNR3	4.71		NNR3
ElasticNet1	8.05	6.91 (31.87)	
RFR2	7.22	6.53 (15.68)	RFR2

3. Boston Housing:

Table C. 607. Local-level Modelling in site 1

Selected models in S1	RMSE	RMSE (MAPE) after update the models	Final models in S1
DTR1	3.46		DTR1
NNR2	7.39	4.79 (13.52)	NNR2
RFR3	4.12	4.32 (15.21)	RFR3

Table C. 608. Local-level Modelling in site 2

Selected models in S2	RMSE	RMSE (MAPE) after update the models	Final models in S2
NNR2	4.18		NNR2
RFR1	8.19	8.55 (12.85)	RFR1
NNR3	7.44	4.01 (12.15)	NNR3

Table C. 609. Local-level Modelling in site 3

Selected models in S3	RMSE	RMSE (MAPE) after update the models	Final models in S3
NNR3	2.71		NNR3
RFR1	3.98	6.42 (11.63)	RFR1
DTR2	6.46	3.63 (19.90)	DTR2

4. Abalone:

Table C. 610. Local-level Modelling in site 1

Selected models in S1	RMSE	RMSE (MAPE) after update the models	Final models in S1
RFR1	2.19		RFR1
NNR2	2.10	2.07 (14.33)	NNR2
NNR3	2.31	2.08 (14.37)	NNR3

Table C. 611. Local-level Modelling in site 2

Selected models in S2	RMSE	RMSE (MAPE) after update the models	Final models in S2
NNR2	1.74		NNR2
NNR1	1.82	1.69 (11.11)	NNR1
NNR3	2.07	1.69 (11.06)	NNR3

Table C. 612. Local-level Modelling in site 3

Selected models in S3	RMSE	RMSE (MAPE) after update the models	Final models in S3
NNR3	2.26		NNR3
Lasso1	2.38	3.50 (18.90)	Lasso1
Ridge2	2.46	3.45 (18.90)	Ridge2

II. Non-randomly Partitioned Data:

1. Parkinson Disease (Total UPDRS):

Table C. 613. Local-level Modelling in site 1

Selected models in S1	RMSE	RMSE (MAPE) after update the models	Final models in S1
NNR1	6.85		NNR1
RFR2	12.96	10.27 (18.53)	RFR2
DTR3	6.78	10.38 (19.69)	DTR3
NNR3	5.39	12.20 (23.93)	NNR3
RFR3	4.48	11.36 (23.23)	RFR3

Table C. 614. Local-level Modelling in site 2

Selected models in S2	RMSE	RMSE (MAPE) after update the models	Final models in S2
NNR2	8.02		NNR2
SVR1	12.47	12.92 (45.68)	
LR3	13.83	13.53 (45.24)	

Table C. 615. Local-level Modelling in site 3

Selected models in S3	RMSE	RMSE (MAPE) after update the models	Final models in S3
NNR3	6.93		NNR3
DTR1	6.90	6.15 (15.65)	DTR1
NNR1	6.44	5.30 (13.01)	NNR1
RFR1	5.11	3.30 (7.34)	RFR1
DTR2	12.89	6.15 (15.65)	DTR2

2. Parkinson Disease (Motor UPDRS):

Table C. 616. Local-level Modelling in site 1

Selected models in S1	RMSE	RMSE (MAPE) after update the models	Final models in S1
NNR1	4.49		NNR1
DTR2	8.84	3.88 (14.22)	DTR2
DTR3	4.39	3.88 (14.22)	DTR3
NNR3	3.93	19.96 (72.32)	
RFR3	2.86	2.19 (7.08)	RFR3

Table C. 617. Local-level Modelling in site 2

Selected models in S2	RMSE	RMSE (MAPE) after update the models	Final models in S2
NNR2	5.95		NNR2
ElasticNet1	10.58	11.30 (47.92)	
Lasso3	11.19	11.43 (48.14)	

Table C. 618. Local-level Modelling in site 3

Selected models in S3	RMSE	RMSE (MAPE) after update the models	Final models in S3
NNR3	5.04		NNR3
DTR1	4.96	6.32 (23.28)	DTR1
NNR1	4.72	23.28 (89.48)	

RFR1	3.99	6.11 (21.11)	RFR1
RFR2	8.99	5.94 (16.09)	RFR2

3. Boston Housing:

Table C. 619. Local-level Modelling in site 1

Selected models in S1	RMSE	RMSE (MAPE) after update the models	Final models in S1
NNR1	2.76		NNR1
LR2	4.10	17.47 (63.96)	
RFR3	10.14	1.37 (3.69)	RFR3

Table C. 620. Local-level Modelling in site 2

Selected models in S2	RMSE	RMSE (MAPE) after update the models	Final models in S2
DTR2	2.69		DTR2
SVR1	3.86	15.16 (61.37)	
RFR3	8.68	1.33 (3.81)	RFR3

Table C. 621. Local-level Modelling in site 3

Selected models in S3	RMSE	RMSE (MAPE) after update the models	Final models in S3
DTR3	3.93		DTR3
DTR1	12.07	0.70 (3.77)	DTR1
NNR2	9.51	19.95 (97.78)	

Appendix D

Detailed Results for the Proposed Method in Chapter 6

In this appendix we present the proposed method detailed results using all possible sites sequence combinations approach for classification and regression datasets.

a) Classification

Table D.1 shows the models evaluation results for the first sequence (site 1 – site 2 – site 3) for the randomly partitioned blood transfusion dataset. It shows the evaluation of the local models in site1 and the updated models in sites 2 and 3. In site 1, LR and RF models got lower performance, so we discarded these models and sent the rest to the following site in the sequence (site 2), then to site 3. NB and DT models are the best updated models, so we selected and sent these models to all sites for evaluation.

Table D. 1. Models Evaluation in site 1, site2, and site 3

Models	Accuracy (in site 1)	Accuracy (in site 2)	Accuracy (in site 3)
LR	68%		
RF	62%		
NB	**73%**	**85%**	**76%**
SVM (nonlinear)	**70%**	**88%**	68%
SVM (linear)	**70%**	**87%**	62%
NN	**74%**	**83%**	44%
DT	**73%**	**98%**	**92%**

Table D.2 presents the updated models evaluation results in all sites. These results are sent to the server with the models and the data size to calculate the average accuracy. The server calculated the average accuracy and then combined the models using the linear combination method to develop the global combined model by assigning a weight for each model based on its average accuracy.

Table D. 2. Best Updated Models Average Accuracy

Final Model	Accuracy in S1	Accuracy in S2	Accuracy in S3	Average accuracy
NB	73%	86%	76%	77 %
DT	64%	78%	92%	79 %

Table D.3 shows the global combined model result compared with the model combining approach using average accuracy. Our proposed method is slightly better than the other combined model.

Table D. 3. Global Combined Model Evaluation

Global Combined Model	Model accuracy
Our proposed method	**61%**
Using average accuracy combination method	59%

Tables D.4 – D.18 show the proposed method for the rest of the sites sequences to build the global combined model. The tables show that the NB, SVM (nonlinear), NN, DT, and FR models are the best updated models in the sequences that got the best average accuracy and used to build the global combined model.

Table D. 4. Models Evaluation in site 1, site3, and site 2

Models	Accuracy (in site 1)	Accuracy (in site 3)	Accuracy (in site 2)
LR	68%		
RF	62%		
NB	**73%**	**76%**	**87%**
SVM (nonlinear)	**70%**	**79%**	**83%**
SVM (linear)	**70%**	**77%**	69%
NN	**74%**	**78%**	68%
DT	**73%**	**92%**	**89%**

Table D. 5. Best Updated Models Average Accuracy

Final Model	Accuracy in S1	Accuracy in S2	Accuracy in S3	Average accuracy
NB	74%	87%	77%	78%
SVM (nonlinear)	68%	83%	76%	75%
DT	72%	89%	73%	76%

Table D. 6. Global Combined Model Evaluation

Global Combined Model	Model accuracy
Our proposed method	**55%**
Using average accuracy combination method	54%

Table D. 7. Models Evaluation in site 2, site1, and site 3

Models	Accuracy (in site 2)	Accuracy (in site 1)	Accuracy (in site 3)
LR	**83%**	68%	
RF	**76%**	**90%**	**94%**
NB	**85%**	**73%**	**76%**
SVM (nonlinear)	**88%**	**74%**	68%
SVM (linear)	**84%**	**73%**	69%
NN	**89%**	69%	
DT	**87%**	**88%**	**92%**

Table D. 8. Best Updated Models Average Accuracy

Final Model	Accuracy in S1	Accuracy in S2	Accuracy in S3	Average accuracy
RF	74%	83%	94%	85%
NB	73%	86%	76%	77%
DT	64%	78%	92%	79%

Table D. 9. Global Combined Model Evaluation

Global Combined Model	Model accuracy
Our proposed method	52%
Using average accuracy combination method	**53%**

Table D. 10. Models Evaluation in site 2, site3, and site 1

Models	Accuracy (in site 2)	Accuracy (in site 3)	Accuracy (in site 1)
LR	**83%**	62%	
RF	**76%**	**94%**	**90%**
NB	**85%**	**71%**	69%
SVM (nonlinear)	**88%**	62%	
SVM (linear)	**84%**	65%	
NN	**89%**	**77%**	**72%**
DT	**87%**	**92%**	68%

Table D. 11. Best Updated Models Average Accuracy

Final Model	Accuracy in S1	Accuracy in S2	Accuracy in S3	Average accuracy
RF	90%	77%	78%	82%
NN	72%	83%	75%	76%

Table D. 12. Global Combined Model Evaluation

Global Combined Model	Model accuracy
Our proposed method	52%
Using average accuracy combination method	**53%**

Table D. 13. Models Evaluation in site 3, site1, and site 2

Models	Accuracy (in site 3)	Accuracy (in site 1)	Accuracy (in site 2)
LR	**75%**	**72%**	62%
RF	65%		
NB	**74%**	**73%**	**87%**
SVM (nonlinear)	**73%**	65%	
SVM (linear)	67%		
NN	**76%**	**74%**	**70%**
DT	65%		

Table D. 14. Best Updated Models Average Accuracy

Final Model	Accuracy in S1	Accuracy in S2	Accuracy in S3	Average accuracy
NB	72%	87%	77%	77%
NN	66%	70%	66%	67%

Table D. 15. Global Combined Model Evaluation

Global Combined Model	Model accuracy
Our proposed method	**52%**
Using average accuracy combination method	50%

Table D. 16. Models Evaluation in site 3, site2 and site 1

Models	Accuracy (in site 3)	Accuracy (in site 2)	Accuracy (in site 1)
LR	**75%**	68%	
RF	65%		
NB	**74%**	**87%**	**73%**
SVM (nonlinear)	**73%**	**85%**	63%
SVM (linear)	67%		
NN	**76%**	**86%**	68%
DT	65%		

Table D. 17. Best Updated Models Average Accuracy

Final Model	Accuracy in S1	Accuracy in S2	Accuracy in S3	Average accuracy
NB	73%	86%	76%	77%

Table D. 18. Global Combined Model Evaluation

Global Combined Model	Model accuracy
NB Model	**53%**

Table D.19 shows the models evaluation results for the first sequence (site 1 – site 2 – site 3) for diabetes dataset that non-randomly partitioned. In site 1, all models got high performance, so we sent all models to the next site in the sequence (site 2) and then to site 3. RF and DT models are the best updated models, so we selected and sent these models to all sites for evaluation.

Table D. 19. Models Evaluation in site 1, site2, and site 3

Models	Accuracy (in site 1)	Accuracy (in site 2)	Accuracy (in site 3)
LR	**83%**	68%	
RF	**83%**	**80%**	**84%**
NB	**84%**	**70%**	68%
SVM (nonlinear)	**82%**	66%	
SVM (linear)	**78%**	66%	
NN	**84%**	65%	
DT	**84%**	**82%**	**87%**

Table D.20 presents the updated models evaluation results in all sites. These results are sent to the server with the models and the data size to calculate the average accuracy. The server calculated the average accuracy and then combined the models using the linear combination method to develop the global combined model.

Table D. 20. Best Updated Models Average Accuracy

Final Model	Accuracy in S1	Accuracy in S2	Accuracy in S3	Average accuracy
RF	62%	63%	84%	65%
DT	56%	64%	87%	63%

Table D.21 shows the global combined model result compared with the average accuracy model combining approach. Our method is slightly better than the other combined model.

Table D. 21. Global Combined Model Evaluation

Global Combined Model	Model accuracy
Our proposed method	**62%**
Using average accuracy combination method	60%

Tables D.22 - D.36 show the proposed method for the rest of the sequences to build the global combined model. The tables show that the LR, NB, DT, and FR models are the best updated models that got the best average accuracy and used to build the global combined model. However, for the sequences in Tables D.25, D.28, D.31, and D.34, the local models results got lower accuracy than 70% (a predefined selection metric), so we decreased the accuracy selection threshold to 60%.

Table D. 22. Models Evaluation in site 1, site3, and site 2

Models	Accuracy (in site 1)	Accuracy (in site 3)	Accuracy (in site 2)
LR	**83%**	**70%**	61%
RF	**83%**	**76%**	**87%**
NB	**84%**	68%	
SVM (nonlinear)	**82%**	53%	
SVM (linear)	**78%**	69%	
NN	**84%**	59%	
DT	**84%**	**78%**	**89%**

Table D. 23. Best Updated Models Average Accuracy

Final Model	Accuracy in S1	Accuracy in S2	Accuracy in S3	Average accuracy
RF	72%	87%	66%	77%
DT	66%	89%	64%	73%

Table D. 24. Global Combined Model Evaluation

Global Combined Model	Model accuracy
Our proposed method	**68%**
Using average accuracy combination method	67%

Table D. 25. Models Evaluation in site 2, site1 and site 3

Models	Accuracy (in site 2)	Accuracy (in site 1)	Accuracy (in site 3)
LR	**64%**	**86%**	60%
RF	**66%**	**89%**	**81%**
NB	**65%**	**86%**	**70%**
SVM (nonlinear)	59%		
SVM (linear)	**61%**	**77%**	58%
NN	**65%**	**83%**	54%
DT	**67%**	**87%**	**87%**

Table D. 26. Best Updated Models Average Accuracy

Final Model	Accuracy in S1	Accuracy in S2	Accuracy in S3	Average accuracy
RF	70%	89%	81%	78%
NB	73%	70%	70%	71%
DT	66%	91%	87%	80%

Table D. 27. Global Combined Model Evaluation

Global Combined Model	Model accuracy
Our proposed method	52%
Using average accuracy combination method	**61%**

Table D. 28. Models Evaluation in site 2, site3, and site 1

Models	Accuracy (in site 2)	Accuracy (in site 3)	Accuracy (in site 1)
LR	**64%**	62%	
RF	**66%**	**83%**	86%
NB	**65%**	68%	
SVM (nonlinear)	59%		
SVM (linear)	**61%**	61%	
NN	**65%**	67%	
DT	**67%**	**87%**	87%

Table D. 29. Best Updated Models Average Accuracy

Final Model	Accuracy in S1	Accuracy in S2	Accuracy in S3	Average accuracy
RF	86%	61%	60%	74%
DT	87%	60%	60%	74%

Table D. 30. Global Combined Model Evaluation

Global Combined Model	Model accuracy
Our proposed method	**71%**
Using average accuracy combination method	62%

Table D. 31. Models Evaluation in site 3, site1, and site 2

Models	Accuracy (in site 3)	Accuracy (in site 1)	Accuracy (in site 2)
LR	**62%**	72%	63%
RF	**65%**	83%	**82%**
NB	**65%**	86%	**70%**
SVM (nonlinear)	**68%**	82%	65%
SVM (linear)	57%		
NN	**64%**	69%	
DT	**62%**	81%	**82%**

Table D. 32. Best Updated Models Average Accuracy

Final Model	Accuracy in S1	Accuracy in S2	Accuracy in S3	Average accuracy
RF	70%	82%	70%	74%
NB	73%	70%	69%	71%
DT	66%	82%	66%	71%

Table D. 33. Global Combined Model Evaluation

Global Combined Model	Model accuracy
Our proposed method	**67%**
Using average accuracy combination method	63%

Table D. 34. Models Evaluation in site 3, site2, and site 1

Models	Accuracy (in site 3)	Accuracy (in site 2)	Accuracy (in site 1)
LR	52%		
RF	**65%**	75%	**87%**

NB	**65%**	70%	**86%**
SVM (nonlinear)	**68%**	64%	
SVM (linear)	57%		
NN	**64%**	54%	
DT	**62%**	87%	**89%**

Table D. 35. Best Updated Models Average Accuracy

Final Model	Accuracy in S1	Accuracy in S2	Accuracy in S3	Average accuracy
RF	87%	60%	54%	74%
NB	86%	62%	60%	75%
DT	89%	60%	53%	75%

Table D. 36. Global Combined Model Evaluation

Global Combined Model	Model accuracy
Our proposed method	**71%**
Using average accuracy combination method	66%

Table D.37 shows the models evaluation results for the first sequence site 1 – site 2 – site 3 for heart disease dataset that randomly partitioned. RF, NB, and DT models are the best three updated models, so we sent these models to all sites for evaluation.

Table D. 37. Models Evaluation in site 1, site2, and site 3

Models	Accuracy (in site 1)	Accuracy (in site 2)	Accuracy (in site 3)
LR	**75%**	**71%**	85%
RF	**79%**	**85%**	**95%**
NB	**85%**	**81%**	**87%**
SVM (nonlinear)	**78%**	**70%**	77%
SVM (linear)	**80%**	**73%**	83%
NN	**81%**	69%	
DT	**82%**	**92%**	**97%**

Table D.38 presents the updated models evaluation results in all sites and the calculated average accuracy that used to develop the global combined model using the linear combination method by assigning a weight for each model based on its average accuracy.

Table D. 38. Best Updated Models Average Accuracy

Final Model	Accuracy in S1	Accuracy in S2	Accuracy in S3	Average accuracy
RF	84%	76%	95%	83%
NB	81%	75%	87%	80%
DT	80%	71%	97%	80%

Table D.39 shows the global combined model got lower performance than the other combination method.

Table D. 39. Global Combined Model Evaluation

Global Combined Model	Model accuracy
Our proposed method	72%
Using average accuracy combination method	**83%**

Tables D.40 – D.54 show the proposed method for the other sites sequences to build the global combined model. The tables show that the NB, DT, SVM (nonlinear), and FR models are the best updated models that used to build the global combined model.

Table D. 40. Models Evaluation in site 1, site3, and site 2

Models	Accuracy (in site 1)	Accuracy (in site 3)	Accuracy (in site 2)
LR	**75%**	**78%**	72%
RF	**79%**	**87%**	**85%**
NB	**85%**	**82%**	**81%**
SVM (nonlinear)	**78%**	**80%**	78%
SVM (linear)	**80%**	**73%**	73%
NN	**81%**	69%	
DT	**82%**	**90%**	**85%**

Table D. 41. Best Updated Models Average Accuracy

Final Model	Accuracy in S1	Accuracy in S2	Accuracy in S3	Average accuracy
RF	76%	85%	77%	80%
NB	84%	81%	82%	83%
DT	73%	85%	78%	79%

Table D. 42. Global Combined Model Evaluation

Global Combined Model	Model accuracy
Our proposed method	**80%**
Using average accuracy combination method	79%

Table D. 43. Models Evaluation in site 2, site1, and site 3

Models	Accuracy (in site 2)	Accuracy (in site 1)	Accuracy (in site 3)
LR	69%		
RF	**70%**	**89%**	**94%**
NB	**75%**	**83%**	**82%**
SVM (nonlinear)	66%		
SVM (linear)	67%		
NN	**74%**	**83%**	65%
DT	**72%**	**96%**	**94%**

Table D. 44. Best Updated Models Average Accuracy

Final Model	Accuracy in S1	Accuracy in S2	Accuracy in S3	Average accuracy
RF	76%	71%	94%	78%
NB	79%	72%	82%	77%
DT	82%	71%	94%	80%

Table D. 45. Global Combined Model Evaluation

Global Combined Model	Model accuracy
Our proposed method	80%
Using average accuracy combination method	**81%**

Table D. 46. Models Evaluation in site 2, site3, and site 1

Models	Accuracy (in site 2)	Accuracy (in site 3)	Accuracy (in site 1)
LR	69%		
RF	**70%**	**93%**	**98%**
NB	**75%**	**87%**	**83%**
SVM (nonlinear)	66%		
SVM (linear)	67%		

NN	**74%**	68%	
DT	**72%**	**94%**	**98%**

Table D. 47. Best Updated Models Average Accuracy

Final Model	Accuracy in S1	Accuracy in S2	Accuracy in S3	Average accuracy
RF	98%	75%	77%	85%
NB	83%	73%	73%	77%
DT	98%	68%	77%	82%

Table D. 48. Global Combined Model Evaluation

Global Combined Model	Model accuracy
Our proposed method	**83%**
Using average accuracy combination method	**83%**

Table D. 49. Models Evaluation in site 3, site1, and site 2

Models	Accuracy (in site 3)	Accuracy (in site 1)	Accuracy (in site 2)
LR	**75%**	**84%**	67%
RF	**82%**	**90%**	**98%**
NB	**77%**	**83%**	**81%**
SVM (nonlinear)	**73%**	**85%**	71%
SVM (linear)	**79%**	**83%**	76%
NN	**82%**	**72%**	67%
DT	**80%**	**97%**	**77%**

Table D. 50. Best Updated Models Average Accuracy

Final Model	Accuracy in S1	Accuracy in S2	Accuracy in S3	Average accuracy
RF	69%	98%	82%	83%
NB	84%	81%	82%	82%
DT	74%	77%	80%	76%

Table D. 51. Global Combined Model Evaluation

Global Combined Model	Model accuracy
Our proposed method	**90%**
Using average accuracy combination method	86%

Table D. 52. Models Evaluation in site 3, site2, and site 1

Models	Accuracy (in site 3)	Accuracy (in site 2)	Accuracy (in site 1)
LR	**75%**	**72%**	79%
RF	**82%**	**89%**	**86%**
NB	**77%**	**81%**	83%
SVM (nonlinear)	**73%**	**73%**	**88%**
SVM (linear)	**79%**	**72%**	83%
NN	**82%**	68%	
DT	**80%**	**94%**	**94%**

Table D. 53. Best Updated Models Average Accuracy

Final Model	Accuracy in S1	Accuracy in S2	Accuracy in S3	Average accuracy
RF	86%	72%	78%	79%
SVM (nonlinear)	88%	67%	80%	78%
DT	94%	68%	77%	80%

Table D. 54. Global Combined Model Evaluation

Global Combined Model	Model accuracy
Our proposed method	**90%**
Using average accuracy combination method	81%

b) Regression:

Table D.55 shows the models evaluation results for the first sequence (site 1 – site 2 – site 3) for Abalone dataset. It shows the evaluation of the local models in site1 and the updated models in site 2 and 3. In site 1, all models got high performance, so we sent the models to the next site in the sequence (site 2) and then to site 3. DTR, NNR, and RFR models are the best three updated models using RMSE metric, so we selected and sent these models to all sites for evaluation.

Table D. 55. Models Evaluation in site 1, site 2, and site 3

Models	RMSE (MAPE) (in site 1)	RMSE (MAPE) (in site 2)	RMSE (MAPE) (in site 3)
LR	**2.23 (14.37)**	**2.08 (11.36)**	2.49 (15.07)
SVR	**2.31 (12.99)**	**2.08 (11.42)**	6.57 (45.09)
DTR	**2.20 (13.44)**	**1.38 (9.39)**	**1.48 (9.34)**
NNR	**2.03 (12.59)**	**1.81 (11.91)**	**2.18 (12.83)**
RFR	**2.19 (14.02)**	**0.80 (4.98)**	**1.08 (5.72)**
Lasso	**2.23 (14.53)**	**2.09 (11.33)**	2.41 (14.35)
Ridge	**2.21 (14.31)**	**2.08 (11.30)**	2.42 (14.57)
ElasticNet	**2.25 (14.64)**	**2.08 (11.26)**	2.61 (16.27)

Table D.56 presents the updated models evaluation results in all sites and the calculated average accuracy for each model. The server combined the models using the linear combination methods to develop the global combined model by assigning a model weight using simple weight average, error-based (RMSE), and performance-based (Accuracy) approaches.

Table D. 56. Best Updated Models Average RMSE (MAPE)

Final Model	RMSE (MAPE) in S1	RMSE (MAPE) in S2	RMSE (MAPE) in S3	Average RMSE (MAPE)
DTR	2.62 (16.99)	2.30 (15.72)	1.48 (9.34)	2.24 (14.81)
NNR	2.33 (15.73)	2.08 (14.90)	2.18 (12.83)	2.19 (14.75)
RFR	2.78 (17.62)	2.47 (16.32)	1.08 (5.72)	2.28 (14.54)

Table D.57 shows that all the linear models combination RMSE results are similar.

Table D. 57. Global Combined Model Evaluation (RMSE)

	Linear Model Combination Method		
	Simple average (RMSE)	Error-based (RMSE)	Performance-based (Accuracy)

The Global Combined Model	2.52	2.52	2.52

Tables D.58 – D.72 show the proposed method for the rest of the sites sequences to build the global combined model. The tables show that the NNR, DTR, and FRR models are the best three updated models that got the best average RMSE and used to build the global combined model.

Table D. 58. Models Evaluation in site 1, site 3, and site 2

Models	RMSE (MAPE) (in site 1)	RMSE (MAPE) (in site 3)	RMSE (MAPE) (in site 2)
LR	**2.23 (14.37)**	**2.49 (15.07)**	1.91 (12.29)
SVR	**2.31 (12.99)**	**2.34 (12.49)**	2.14 (12.09)
DTR	**2.20 (13.44)**	**1.46 (9.33)**	**1.38 (9.39)**
NNR	**2.03 (12.59)**	**2.56 (15.14)**	**1.68 (10.99)**
RFR	**2.19 (14.02)**	**1.06 (5.88)**	**0.82 (5.08)**
Lasso	**2.23 (14.53)**	**3.48 (19.78)**	1.94 (12.55)
Ridge	**2.21 (14.31)**	**3.47 (18.72)**	1.91 (12.55)
ElasticNet	**2.25 (14.64)**	**3.50 (18.19)**	1.96 (11.82)

Table D. 59. Best Updated Models Average RMSE (MAPE)

Final Model	RMSE (MAPE) in S1	RMSE (MAPE) in S2	RMSE (MAPE) in S3	Average RMSE (MAPE)
DTR	2.25 (13.18)	1.38 (9.39)	2.66 (13.04)	1.94 (11.44)
NNR	2.11 (12.38)	1.68 (10.99)	2.50 (12.72)	2.00 (11.82)
RFR	2.34 (14.20)	0.82 (5.08)	2.69 (13.95)	1.73 (10.04)

Table D. 60. Global Combined Model Evaluation (RMSE)

	Linear Model Combination Method		
	Simple average (RMSE)	Error-based (RMSE)	Performance-based (Accuracy)
The Global Combined Model	2.85	**2.79**	2.85

Table D. 61. Models Evaluation in site 2, site 1, and site 3

Models	RMSE (MAPE) (in site 2)	RMSE (MAPE) (in site 1)	RMSE (MAPE) (in site 3)
LR	**1.91 (12.25)**	**2.73 (13.65)**	2.49 (15.07)
SVR	**1.96 (11.64)**	**2.71 (13.33)**	6.57 (45.09)
DTR	**1.81 (11.77)**	**1.51 (9.85)**	**1.48 (9.34)**
NNR	**1.74 (11.41)**	**1.95 (12.09)**	**2.18 (12.84)**
RFR	**1.88 (12.35)**	**0.94 (5.62)**	**1.08 (5.72)**
Lasso	**1.87 (12.38)**	**2.71 (13.34)**	2.41 (14.35)
Ridge	**1.90 (12.37)**	**2.72 (13.64)**	2.42 (14.57)
ElasticNet	**1.89 (12.67)**	**2.68 (13.18)**	2.61 (16.28)

Table D. 62. Best Updated Models Average RMSE (MAPE)

Final Model	RMSE (MAPE) in S1	RMSE (MAPE) in S2	RMSE (MAPE) in S3	Average RMSE (MAPE)
DTR	2.63 (16.99)	2.30 (15.72)	1.48 (9.34)	2.24 (14.81)
NNR	2.32 (15.73)	2.08 (14.90)	2.18 (12.84)	2.18 (14.74)
RFR	2.78 (17.62)	2.47 (16.32)	1.08 (5.72)	2.28 (14.54)

Table D. 63. Global Combined Model Evaluation (RMSE)

	Linear Model Combination Method		
	Simple average (RMSE)	Error-based (RMSE)	Performance-based (Accuracy)
The Global Combined Model	2.53	**2.51**	2.53

Table D. 64. Models Evaluation in site 2, site 3, and site 1

Models	RMSE (MAPE) (in site 2)	RMSE (MAPE) (in site 3)	RMSE (MAPE) (in site 1)
LR	**1.91 (12.25)**	**3.48 (18.95)**	2.30 (14.16)
SVR	**1.96 (11.64)**	**3.49 (18.33)**	3.22 (16.01)
DTR	**1.81 (11.77)**	**1.48 (9.34)**	**1.51 (9.85)**
NNR	**1.74 (11.41)**	**2.18 (12.84)**	**1.95 (12.09)**
RFR	**1.88 (12.35)**	**1.08 (5.88)**	**0.98 (5.60)**
Lasso	**1.87 (12.38)**	**2.50 (14.39)**	2.42 (16.47)
Ridge	**1.90 (12.37)**	**2.50 (15.01)**	2.29 (14.34)
ElasticNet	**1.89 (12.67)**	**2.48 (14.06)**	2.30 (14.27)

Table D. 65. Best Updated Models Average RMSE (MAPE)

Final Model	RMSE (MAPE) in S1	RMSE (MAPE) in S2	RMSE (MAPE) in S3	Average RMSE (MAPE)
DTR	1.51 (9.85)	1.95 (13.16)	2.59 (13.64)	1.93 (12.13)
NNR	1.95 (12.09)	1.83 (12.50)	2.48 (13.26)	2.01 (12.52)
RFR	0.98 (5.60)	2.06 (13.91)	2.68 (14.97)	1.82 (11.31)

Table D. 66. Global Combined Model Evaluation (RMSE)

	Linear Model Combination Method		
	Simple average (RMSE)	Error-based (RMSE)	Performance-based (Accuracy)
The Global Combined Model	2.68	**2.64**	**2.64**

Table D. 67. Models Evaluation in site 3, site 1, and site 2

Models	RMSE (MAPE) (in site 3)	RMSE (MAPE) (in site 1)	RMSE (MAPE) (in site2)
LR	**2.28 (13.60)**	**2.73 (13.65)**	1.91 (12.29)
SVR	**2.34 (12.77)**	**2.71 (13.33)**	2.14 (12.09)
DTR	**2.49 (14.25)**	**1.51 (9.85)**	**1.38 (9.39)**
NNR	**2.26 (13.40)**	**2.15 (13.67)**	**1.68 (10.99)**
RFR	**2.54 (14.97)**	**0.93 (5.55)**	**0.82 (5.08)**
Lasso	**2.32 (14.09)**	**2.71 (13.36)**	1.94 (12.55)
Ridge	**2.30 (13.79)**	**2.73 (13.41)**	1.91 (12.55)
ElasticNet	**2.29 (13.91)**	**2.71 (13.36)**	1.96 (11.82)

Table D. 68. Best Updated Models Average RMSE (MAPE)

Final Model	RMSE (MAPE) in S1	RMSE (MAPE) in S2	RMSE (MAPE) in S3	Average RMSE (MAPE)
DTR	2.25 (13.18)	1.38 (9.39)	2.66 (13.04)	1.94 (11.44)
NNR	2.11 (12.38)	1.68 (10.99)	2.50 (12.72)	1.99 (11.82)
RFR	2.34 (14.19)	0.82 (5.08)	2.69 (13.95)	1.73 (10.04)

Table D. 69. Global Combined Model Evaluation (RMSE)

	Linear Model Combination Method		
	Simple average (RMSE)	Error-based (RMSE)	Performance-based (Accuracy)
The Global Combined Model	2.85	2.80	**2.79**

Table D. 70. Models Evaluation in site 3, site 2, and site 1

Models	RMSE (MAPE) (in site 3)	RMSE (MAPE) (in site 2)	RMSE (MAPE) (in site 1)
LR	**2.28 (13.60)**	**2.08 (11.36)**	2.30 (14.16)
SVR	**2.34 (12.77)**	**2.08 (11.42)**	3.23 (16.01)
DTR	**2.49 (14.25)**	**1.38 (9.39)**	**1.51 (9.85)**
NNR	**2.26 (13.40)**	**1.81 (11.91)**	**1.95 (12.09)**
RFR	**2.54 (14.97)**	**0.79 (4.95)**	**0.98 (5.60)**
Lasso	**2.32 (14.09)**	**2.09 (11.33)**	2.42 (16.47)
Ridge	**2.30 (13.79)**	**2.08 (11.30)**	2.29 (14.34)
ElasticNet	**2.29 (13.91)**	**2.08 (11.27)**	2.30 (14.27)

Table D. 71. Best Updated Models Average RMSE (MAPE)

Final Model	RMSE (MAPE) in S1	RMSE (MAPE) in S2	RMSE (MAPE) in S3	Average RMSE (MAPE)
DTR	1.51 (9.85)	1.95 (13.16)	2.59 (13.64)	1.93 (12.13)
NNR	1.95 (12.09)	1.83 (12.50)	2.48 (13.26)	2.01 (12.52)
RFR	0.98 (5.60)	2.06 (13.91)	2.68 (14.97)	1.82 (11.31)

Table D. 72. Global Combined Model Evaluation (RMSE)

	Linear Model Combination Method		
	Simple average (RMSE)	Error-based (RMSE)	Performance-based (Accuracy)
The Global Combined Model	2.68	**2.64**	2.68

Table D.73 shows the models evaluation results for the first sequence (site 1 – site 2 – site 3) for Parkinson disease (Total UPDRS) dataset that randomly partitioned. In site 1, SVR, DTR, NNR, RFR, and ElasticNet models got high performance, so we sent the models to the next site in the sequence (site 2) and then to site 3. DTR, RFR, and ElasticNet models are the best updated models using RMSE metric, and we sent these models to all sites for evaluation.

Table D. 73. Models Evaluation in site 1, site 2, and site 3

Models	RMSE (MAPE) (in site 1)	RMSE (MAPE) (in site 2)	RMSE (MAPE) (in site 3)
LR	11.02 (30.4)		
SVR	**10.82 (25.93)**	**7.18 (20.53)**	23.93 (70.16)
DTR	**8.92 (24.06)**	**3.58 (11.77)**	**8.20 (17.75)**
NNR	**7.66 (20.01)**	12.20 (50.65)	
RFR	**9.1 (23.7)**	**2.19 (6.24)**	**7.91 (15.45)**
Lasso	11.17 (30.27)		
Ridge	11.15 (30.53)		
ElasticNet	**10.74 (28.73)**	**7.29 (20.54)**	**10.17 (25.46)**

Table D.74 presents the updated models evaluation results in all sites. These results are sent to the server with the models and the data size to calculate the average RMSE and MAPE and then combined the models using the linear combination method to develop the global combined model.

Table D. 74. Best Updated Models Average RMSE (MAPE)

Final Model	RMSE (MAPE) in S1	RMSE (MAPE) in S2	RMSE (MAPE) in S3	Average RMSE (MAPE)
DTR	17.26 (37.36)	8.82 (33.09)	8.20 (17.75)	11.34 (27.90)
RFR	16.50 (34.69)	8.61 (31.09)	7.91 (15.45)	10.91 (25.55)
ElasticNet	13.12 (37.88)	11.38 (47.94)	10.17 (25.46)	11.43 (34.95)

Table D.75 shows the global combined model RMSE results for all combination methods, and the model combination result using the error-based approach is the best.

Table D. 75. Global Combined Model Evaluation (RMSE)

	Linear Model Combination Method		
	Simple average (RMSE)	Error-based (RMSE)	Performance-based (Accuracy)
The Global Combined Model	8.79	**7.46**	9.48

The following Tables, D.76 – D.90, show the proposed method for the other sites sequences to build the global combined model. The tables show that LR, DTR, RFR, and ElasticNet are the best updated models that got the best average RMSE and used to build the global combined model.

Table D. 76. Models Evaluation in site 1, site 3, and site 2

Models	RMSE (MAPE) (in site 1)	RMSE (MAPE) (in site 3)	RMSE (MAPE) (in site 2)
LR	11.02 (30.4)		
SVR	**10.82 (25.93)**	15.18 (35.54)	
DTR	**8.92 (24.06)**	**10.17 (29.12)**	**8.07 (23.28)**
NNR	**7.66 (20.01)**	31.43 (87.65)	
RFR	**9.1 (23.7)**	**8.27 (15.92)**	**8.12 (22.89)**
Lasso	11.17 (30.27)		
Ridge	11.15 (30.53)		
ElasticNet	**10.74 (28.73)**	15.44 (36.01)	

Table D. 77. Best Updated Models Average RMSE (MAPE)

Final Model	RMSE (MAPE) in S1	RMSE (MAPE) in S2	RMSE (MAPE) in S3	Average RMSE (MAPE)
DTR	14.94 (29.64)	8.07 (23.28)	10.37 (30.56)	11.33 (28.51)
RFR	15.39 (30.63)	8.12 (22.89)	10.39 (30.59)	11.49 (28.75)

Table D. 78. Global Combined Model Evaluation (RMSE)

	Linear Model Combination Method		
	Simple average (RMSE)	Error-based (RMSE)	Performance-based (Accuracy)
The Global Combined Model	**9.9**	**9.9**	**9.9**

Table D. 79. Models Evaluation in site 2, site 1, and site 3

Models	RMSE (MAPE) (in site 2)	RMSE (MAPE) (in site 1)	RMSE (MAPE) (in site 3)
LR	11.65 (32.59)		
SVR	**6.83 (20.80)**	**10.79 (29.17)**	15.50 (37.44)
DTR	**5.15 (16.88)**	**6.94 (18.63)**	**5.10 (14.38)**
NNR	**5.06 (16.66)**	17.06 (39.66)	
RFR	**5.39 (16.74)**	**3.85 (8.98)**	**2.69 (6.70)**
Lasso	15.74 (51.63)		
Ridge	28.34 (83.33)		
ElasticNet	16.63 (52.09)		

Table D. 80. Best Updated Models Average RMSE (MAPE)

Final Model	RMSE (MAPE) in S1	RMSE (MAPE) in S2	RMSE (MAPE) in S3	Average RMSE (MAPE)
DTR	15.68 (39.02)	9.33 (37.59)	5.10 (14.38)	9.61 (28.08)
RFR	15.73 (37.21)	9.01 (34.69)	2.69 (6.70)	8.51 (23.48)

Table D. 81. Global Combined Model Evaluation (RMSE)

	Linear Model Combination Method		
	Simple average (RMSE)	Error-based (RMSE)	Performance-based (Accuracy)
The Global Combined Model	**9.6**	**9.6**	**9.6**

Table D. 82. Models Evaluation in site 2, site 3, and site 1

Models	RMSE (MAPE) (in site 2)	RMSE (MAPE) (in site 3)	RMSE (MAPE) (in site 1)
LR	11.65 (32.59)		
SVR	**6.83 (20.80)**	**8.31 (25.91)**	23.52 (53.80)
DTR	**5.15 (16.88)**	**5.10 (14.38)**	**6.94 (18.63)**
NNR	**5.06 (16.66)**	**9.20 (24.16)**	26.58 (63.26)
RFR	**5.39 (16.74)**	**2.59 (6.48)**	**3.86 (8.94)**
Lasso	15.74 (51.63)		
Ridge	28.34 (83.33)		
ElasticNet	16.63 (52.09)		

Table D. 83. Best Updated Models Average RMSE (MAPE)

Final Model	RMSE (MAPE) in S1	RMSE (MAPE) in S2	RMSE (MAPE) in S3	Average RMSE (MAPE)
DTR	6.94 (18.63)	12.50 (51.34)	11.99 (41.57)	10.44 (36.34)
RFR	3.86 (8.94)	13.05 (51.53)	12.09 (40.99)	9.60 (32.94)

Table D. 84. Global Combined Model Evaluation (RMSE)

	Linear Model Combination Method		
	Simple average (RMSE)	Error-based (RMSE)	Performance-based (Accuracy)
The Global Combined Model	**11.9**	**11.9**	**11.9**

Table D. 85. Models Evaluation in site 3, site 1, and site 2

Models	RMSE (MAPE) (in site 3)	RMSE (MAPE) (in site 1)	RMSE (MAPE) (in site 2)
LR	**8.99 (27.33)**	**10.84 (29.11)**	8.60 (24.87)
SVR	**8.20 (24.86)**	**10.83 (29.22)**	18.39 (57.41)
DTR	**6.37 (18.06)**	**6.94 (18.63)**	**3.58 (11.77)**
NNR	**5.85 (16.49)**	16.38 (43.56)	

RFR	**6.33 (16.79)**	**3.84 (9.13)**	**2.16 (6.18)**
Lasso	**9.48 (28.36)**	**10.80 (28.83)**	7.15 (23.54)
Ridge	**8.85 (27.37)**	**10.95 (28.40)**	10.36 (29.99)
ElasticNet	**9.66 (28.37)**	**10.96 (29.30)**	**6.77 (21.57)**

Table D. 86. Best Updated Models Average RMSE (MAPE)

Final Model	RMSE (MAPE) in S1	RMSE (MAPE) in S2	RMSE (MAPE) in S3	Average RMSE (MAPE)
DTR	17.02 (39.63)	3.58 (11.77)	8.95 (23.12)	10.32 (25.84)
RFR	16.69 (37.85)	2.16 (6.18)	9.19 (23.95)	9.98 (24.27)
ElasticNet	16.97 (37.25)	6.77 (21.57)	10.22 (26.29)	11.62 (28.77)

Table D. 87. Global Combined Model Evaluation (RMSE)

	Linear Model Combination Method		
	Simple average (RMSE)	Error-based (RMSE)	Performance-based (Accuracy)
The Global Combined Model	**10.30**	10.38	10.31

Table D. 88. Models Evaluation in site 3, site 2, and site 1

Models	RMSE (MAPE) (in site 3)	RMSE (MAPE) (in site 2)	RMSE (MAPE) (in site 1)
LR	**8.99 (27.33)**	**7.16 (20.54)**	**10.49 (27.06)**
SVR	**8.20 (24.86)**	**7.18 (20.53)**	23.47 (53.75)
DTR	**6.37 (18.06)**	**3.59 (11.77)**	**6.94 (18.63)**
NNR	**5.85 (16.49)**	8.51 (32.19)	
RFR	**6.33 (16.79)**	**2.19 (6.25)**	**3.76 (8.73)**
Lasso	**9.48 (28.36)**	**7.14 (20.56)**	11.30 (32.58)
Ridge	**8.85 (27.37)**	**7.17 (20.36)**	10.99 (29.70)
ElasticNet	**9.66 (28.37)**	**7.29 (20.54)**	10.74 (27.65)

Table D. 89. Best Updated Models Average RMSE (MAPE)

Final Model	RMSE (MAPE) in S1	RMSE (MAPE) in S2	RMSE (MAPE) in S3	Average RMSE (MAPE)
LR	10.49 (27.06)	12.19 (52.06)	10.54 (37.52)	10.92 (37.56)
DTR	6.94 (18.63)	12.50 (51.35)	11.99 (41.57)	10.44 (36.35)
RFR	3.76 (8.73)	13.83 (56.44)	12.86 (44.02)	10.09 (35.35)

Table D. 90. Global Combined Model Evaluation (RMSE)

	Linear Model Combination Method		
	Simple average (RMSE)	Error-based (RMSE)	Performance-based (Accuracy)
The Global Combined Model	11.63	**11.56**	11.76

Table D.91 shows the models evaluation results for the first sequence (site 1 – site 2 – site 3) for Parkinson disease (Motor UPDRS) dataset that partitioned non-randomly. DTR and RFR models are the best updated models using RMSE metric, so we sent these models to all sites for evaluation.

Table D. 91. Models Evaluation in site 1, site 2, and site 3

Models	RMSE (MAPE) (in site 1)	RMSE (MAPE) (in site 2)	RMSE (MAPE) (in site 3)
LR	**6.04 (22.33)**	7.87 (44.44)	
SVR	**6.09 (21.20)**	7.43 (44.84)	
DTR	**5.02 (18.44)**	**4.76 (25.82)**	**4.27 (15.78)**
NNR	**4.49 (15.69)**	6.84 (39.92)	
RFR	**5.06 (17.82)**	**2.61 (12.55)**	**2.30 (7.66)**
Lasso	**6.19 (22.91)**	7.86 (44.22)	
Ridge	**6.05 (22.68)**	7.97 (46.66)	
ElasticNet	**6.15 (23.09)**	7.88 (45.15)	

Table D.92 presents the updated models evaluation results in all sites. These results are sent to the server with the models and the data size to calculate the average RMSE and MAPE, then combined the models using linear combination methods to develop the global combined model.

Table D. 92. Best Updated Models Average RMSE (MAPE)

Final Model	RMSE (MAPE) in S1	RMSE (MAPE) in S2	RMSE (MAPE) in S3	Average RMSE (MAPE)
DTR	4.39 (16.87)	11.42 (74.81)	4.27 (15.78)	6.02 (32.24)
RFR	2.96 (10.53)	11.48 (74.69)	2.30 (7.66)	5.02 (27.86)

Table D.93 shows that the global combined model results using all linear model combination methods got the same RMSE results.

Table D. 93. Global Combined Model Evaluation (RMSE)

	Linear Model Combination Method		
	Simple average (RMSE)	Error-based (RMSE)	Performance-based (Accuracy)
The Global Combined Model	**10.79**	**10.79**	**10.79**

Tables D.94 – D.108 show the proposed method for the other sites sequences to build the global combined model. The tables show that NNR, DTR, and RFR models are the best updated models that got the best average RMSE and used to build the global combined model. For the sequences site2 - site 1 - site 3 and site 2 - site 3 - site 1, we lowered the model selection RMSE threshold to lower than or equal to 7.00 (MAPE is lower than or equal to 40%).

Table D. 94. Models Evaluation in site 1, site 3, and site 2

Models	RMSE (MAPE) (in site 1)	RMSE (MAPE) (in site 3)	RMSE (MAPE) (in site 2)
LR	**6.04 (22.33)**	**6.92 (26.76)**	7.68 (45.36)
SVR	**6.09 (21.20)**	**6.19 (22.59)**	11.81 (45.61)
DTR	**5.02 (18.44)**	**4.26 (15.78)**	**4.76 (25.82)**
NNR	**4.49 (15.69)**	**3.82 (13.38)**	18.45 (70.56)
RFR	**5.06 (17.82)**	**2.25 (7.23)**	**2.65 (12.70)**
Lasso	**6.19 (22.91)**	**6.26 (23.97)**	7.69 (48.94)
Ridge	**6.05 (22.68)**	**7.85 (29.35)**	7.78 (46.29)
ElasticNet	**6.15 (23.09)**	**7.25 (26.53)**	7.99 (45.72)

Table D. 95. Best Updated Models Average RMSE (MAPE)

Final Model	RMSE (MAPE) in S1	RMSE (MAPE) in S2	RMSE (MAPE) in S3	Average RMSE (MAPE)
DTR	8.87 (34.04)	4.76 (25.82)	9.09 (34.02)	6.82 (28.16)
RFR	8.67 (33.36)	2.65 (12.70)	8.88 (33.11)	6.06 (23.75)

Table D. 96. Global Combined Model Evaluation (RMSE)

	Linear Model Combination Method		
	Simple average (RMSE)	Error-based (RMSE)	Performance-based (Accuracy)
The Global Combined Model	7.23	**7.05**	7.22

Table D. 97. Models Evaluation in site 2, site 1, and site 3

Models	RMSE (MAPE) (in site 2)	RMSE (MAPE) (in site 1)	RMSE (MAPE) (in site 3)
LR	9.25 (50.90)		
SVR	7.73 (46.84)		
DTR	**6.17 (33.83)**	**3.88 (14.22)**	**4.27 (15.77)**
NNR	**5.95 (32.14)**	**5.22 (19.33)**	19.98 (71.47)
RFR	**6.40 (33.45)**	**2.14 (7.08)**	**2.22 (7.21)**
Lasso	7.88 (47.01)		
Ridge	7.81 (46.12)		
ElasticNet	7.99 (47.05)		

Table D. 98. Best Updated Models Average RMSE (MAPE)

Final Model	RMSE (MAPE) in S1	RMSE (MAPE) in S2	RMSE (MAPE) in S3	Average RMSE (MAPE)
DTR	4.39 (16.87)	11.42 (74.76)	4.27 (15.77)	6.02 (32.22)
RFR	2.94 (10.13)	11.78 (75.72)	2.22 (7.21)	5.08 (27.92)

Table D. 99. Global Combined Model Evaluation (RMSE)

	Linear Model Combination Method		
	Simple average (RMSE)	Error-based (RMSE)	Performance-based (Accuracy)
The Global Combined Model	12.01	**11.98**	11.99

Table D. 100. Models Evaluation in site 2, site 3, and site 1

Models	RMSE (MAPE) (in site 2)	RMSE (MAPE) (in site 3)	RMSE (MAPE) (in site 1)
LR	9.25 (50.90)		
SVR	7.73 (46.84)		
DTR	**6.17 (33.83)**	**4.26 (15.78)**	**3.88 (14.22)**
NNR	**5.95 (32.14)**	**5.49 (20.59)**	**3.92 (13.98)**
RFR	**6.40 (33.45)**	**2.30 (7.34)**	**2.17 (6.75)**
Lasso	7.88 (47.01)		
Ridge	7.81 (46.12)		
ElasticNet	7.99 (47.05)		

Table D. 101. Best Updated Models Average RMSE (MAPE)

Final Model	RMSE (MAPE) in S1	RMSE (MAPE) in S2	RMSE (MAPE) in S3	Average RMSE (MAPE)
DTR	3.88 (14.22)	11.06 (71.39)	4.96 (16.73)	5.97 (30.70)
NNR	3.92 (13.98)	10.92 (70.48)	4.79 (15.49)	5.89 (29.98)
RFR	2.17 (6.75)	11.24 (71.16)	4.09 (11.52)	5.25 (26.83)

Table D. 102. Global Combined Model Evaluation (RMSE)

	Linear Model Combination Method		
	Simple average (RMSE)	Error-based (RMSE)	Performance-based (Accuracy)
The Global Combined Model	**10.51**	10.60	10.61

Table D. 103. Models Evaluation in site 3, site 1, and site 2

Models	RMSE (MAPE) (in site 3)	RMSE (MAPE) (in site 1)	RMSE (MAPE) (in site 2)
LR	**6.38 (23.61)**	**6.19 (21.78)**	8.68 (49.75)
SVR	**6.23 (22.93)**	**6.11 (22.16)**	11.99 (45.13)
DTR	**5.36 (19.95)**	**3.88 (14.22)**	**4.76 (25.82)**
NNR	**5.04 (17.72)**	**5.23 (19.34)**	5.99 (35.30)
RFR	**5.42 (19.26)**	**2.19 (6.94)**	**2.57 (12.04)**
Lasso	**7.33 (26.97)**	**6.18 (22.24)**	8.09 (45.92)
Ridge	**6.52 (24.99)**	**6.14 (21.82)**	8.04 (46.45)
ElasticNet	**6.60 (24.76)**	**6.13 (22.15)**	7.64 (44.51)

Table D. 104. Best Updated Models Average RMSE (MAPE)

Final Model	RMSE (MAPE) in S1	RMSE (MAPE) in S2	RMSE (MAPE) in S3	Average RMSE (MAPE)
DTR	8.83 (33.84)	4.76 (25.82)	9.07 (33.83)	6.80 (28.04)
RFR	8.70 (34.28)	2.57 (12.04)	8.99 (34.17)	6.08 (24.15)

Table D. 105. Global Combined Model Evaluation (RMSE)

	Linear Model Combination Method		
	Simple average (RMSE)	Error-based (RMSE)	Performance-based (Accuracy)
The Global Combined Model	7.98	**7.93**	7.97

Table D. 106. Models Evaluation in site 3, site 2, and site 1

Models	RMSE (MAPE) (in site 3)	RMSE (MAPE) (in site 2)	RMSE (MAPE) (in site 1)
LR	**6.38 (23.61)**	7.86 (45.08)	
SVR	**6.23 (22.93)**	7.87 (44.68)	
DTR	**5.36 (19.95)**	**4.76 (25.82)**	**3.88 (14.22)**
NNR	**5.04 (17.72)**	6.93 (40.97)	
RFR	**5.42 (19.26)**	**2.72 (13.24)**	**2.11 (6.88)**
Lasso	**7.33 (26.97)**	7.84 (44.80)	
Ridge	**6.52 (24.99)**	7.86 (44.37)	
ElasticNet	**6.60 (24.76)**	7.92 (45.21)	

Table D. 107. Best Updated Models Average RMSE (MAPE)

Final Model	RMSE (MAPE) in S1	RMSE (MAPE) in S2	RMSE (MAPE) in S3	Average RMSE (MAPE)

DTR	3.88 (14.22)	11.06 (71.37)	4.97 (16.74)	5.97 (30.70)
RFR	2.11 (6.88)	11.12 (70.12)	3.93 (11.29)	5.14 (26.49)

Table D. 108. Global Combined Model Evaluation (RMSE)

	Linear Model Combination Method		
	Simple average (RMSE)	Error-based (RMSE)	Performance-based (Accuracy)
The Global Combined Model	**10.47**	**10.47**	**10.47**