TU Clausthal
Clausthal University of Technology

JELENA FIOSINA

# Computationally intensive, distributed and decentralised machine learning: from theory to applications

CUMULATIVE
*HABILITATION THESIS*

2022

**Jelena Fiosina**

Computationally intensive, distributed and decentralised machine learning:
from theory to applications, Cumulative Habilitation Thesis.

Approved by the Faculty of Mathematics, Computer Science and Mechanical Engineering, Clausthal University of Technology, Germany

Clausthal-Zellerfeld, 2022

# Contents

# Part I
# Summary

**Abstract**

Machine learning (ML) is currently one of the most important research fields, spanning computer science, statistics, pattern recognition, data mining, and predictive analytics. It plays a central role in automatic data processing and analysis in numerous research domains owing to widely distributed and geographically scattered data sources, powerful computing clouds, and high digitisation requirements. However, aspects such as the accuracy of methods, data privacy, and model explainability remain challenging and require additional research.

Therefore, it is necessary to analyse centralised and distributed data processing architectures, and to create novel computationally intensive explainable and privacy-preserving ML methods, to investigate their properties, to propose distributed versions of prospective ML baseline methods, and to evaluate and apply these in various applications.

This thesis addresses the theoretical and practical aspects of state-of-the-art ML methods. The contributions of this thesis are threefold.

In Chapter 2, novel non-distributed, centralised, computationally intensive ML methods are proposed, their properties are investigated, and state-of-the-art ML methods are applied to real-world data from two domains, namely transportation and bioinformatics. Moreover, algorithms for 'black-box' model interpretability are presented.

Decentralised ML methods are considered in Chapter 3. First, we investigate data processing as a preliminary step in data-driven, agent-based decision-making. Thereafter, we propose novel decentralised ML algorithms that are based on the collaboration of the local models of agents. Within this context, we consider various regression models. Finally, the explainability of multi-agent decision-making is addressed.

In Chapter 4, we investigate distributed centralised ML methods. We propose a distributed parallelisation algorithm for the semi-parametric and non-parametric regression types, and implement these in the computational environment and data structures of Apache SPARK. Scalability, speed-up, and goodness-of-fit experiments using real-world data demonstrate the excellent performance of the proposed methods. Moreover, the federated deep-learning approach enables us to address the data privacy challenges caused by processing of distributed private data sources to solve the travel-time prediction problem. Finally, we propose an explainability strategy to interpret the influence of the input variables on this federated deep-learning application.

This thesis is based on the contribution made by 11 papers to the theoretical and practical aspects of state-of-the-art and proposed ML methods. We successfully address the stated challenges with various data processing architectures, validate the proposed approaches in diverse scenarios from the transportation and bioinformatics domains, and demonstrate their effectiveness in scalability, speed-up, and goodness-of-fit experiments with real-world data.

However, substantial future research is required to address the stated challenges and to identify novel issues in ML. Thus, it is necessary to advance the theoretical part by creating novel ML methods and investigating their properties, as well as to contribute to the application part by using of the state-of-the-art ML methods and their combinations, and interpreting their results for different problem settings.

# Chapter 1

# Introduction

Machine learning (ML) has become a very important research field, which encompasses computer science, statistics, pattern recognition, data mining, and predictive analytics. It plays a central role in automatic data processing and analytics across numerous research domains for the following reasons: First, contemporary data in various domains are gathered from widely distributed and geographically scattered sources such as sensors, clouds, and Internet of Things devices and their volumes increase continuously over time. Second, modern computers, supercomputers, and computing clouds, as well as powerful edge devices, provide a strong basis for processing large amounts of data. Third, based on the previous facts, application domains resulting from digitisation offer great potential in obtaining useful information from these data and in using them for data-driven decision-making to solve previously unsolvable problems. Finally, this leads to the development of novel ML methods that facilitate the ability of contemporary data-driven AI-based management, control, and decision-making systems to solve domain-specific problems.

State-of-the-art computationally intensive data-driven methods, such as deep learning, and non-parametric resampling- and ensemble-based methods, can replace complex analytical procedures with multiple calculations. Considering their classical ML ancestry, they are easily applicable and, in most cases, more accurate than other methods. Complex artificial networks, such as convolutional, recursive, and generic adversarial networks, as well as autoencoders, are capable of significant achievements in solving previously unsolvable problems including accurate face recognition, autonomous driving assistance, and new drug discovery. In this thesis, we consider the timely application of these methods in bioinformatics and transportation. Thus, modern data-driven AI-based systems that are implemented in transportation can provide more effective control and management of transportation and shared mobility systems, thereby reducing traffic and air pollution. Moreover, proper data processing of bioinformatics data can facilitate new findings in the dependencies between the genome and disease, as well as the promotion of personalised medicine.

Different data processing architectures address the varying requirements of practical applications, which are investigated in this thesis and stimulate the creation of specialised ML methods. However, ML is currently in the developmental stage, and despite the huge progress made in digitisation, several challenges remain.

**Challenge 1**: The wide availability of data increases the need for data processing and analysis. Novel ML methods that are generally computationally intensive should be created for new data-driven statistical problems or adapted for existing ones, thereby providing more effective, accurate, and robust solutions. Novel computationally intensive methods based on resampling, subsampling, ensembling, stacking, hybridisation, and deep learning should be proposed to provide effective solutions for existing and new problem statements.

**Challenge 2**: The challenges in big data relate to the large volumes of distributed data that cannot be processed together in a reasonable time owing to their substantial sizes. Contemporary computing clouds enable the distribution of computational processes across multiple servers and the

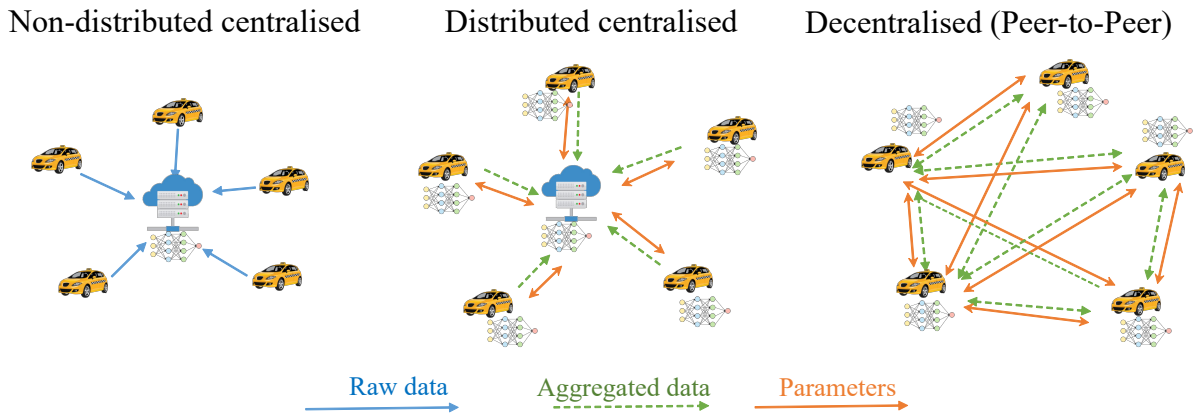



**Fig. 1.1** Data processing architectures

subsequent gathering of the results. Distributed calculations could be addressed through the smart 'artificial' partitioning and parallelisation of data, and computation within a cloud-based architecture or powerful supercomputers. Modern operating systems and frameworks such as Hadoop and SPARK facilitate this process. Many data processing libraries have been developed for distributed versions of contemporary centralised data analysis methods to parallelise data processing in cloud infrastructures. However, these processes cannot be distributed automatically. Therefore, a novel scalable and more rapid distributed version based on prospective ML methods is required to distribute their computations and to increase their execution speed, which will allow more complicated tasks to be solved.

**Challenge 3**: Distributed data sources often cannot be processed centrally. Complex networked applications with a large number of interconnected objects may generate their own data, observe the activities of other objects, and make decisions. Such data-driven systems are frequently modelled using a multi-agent system approach, in which agents can have goals, be intellectual, and possess data processing abilities, as well process the data decentrally and cooperate with one another. The agents often cannot or do not need to use central server capabilities for data processing owing to communication problems or privacy aspects. This leads to data processing in a decentralised manner; however, restricted collaboration may be very profitable. Therefore, novel decentralised ML methods should be created and effective coordination mechanisms should be designed.

**Challenge 4**: Distributed data often cannot be processed together because of privacy reasons and the inability or unwillingness of partners to share the raw local data, which leads to data privacy challenges. Federated learning enables collaborative learning without the exchange of raw data, but only synchronises the model parameters with central cloud server support. Moreover, federated learning decreases the calculation load on the central server and distributes the computations among the participants. Therefore, the creation of novel distributed (e.g. federated architecture-based) ML methods, that satisfy data privacy requirements is necessary.

**Challenge 5**: Most computationally intensive ML methods suffer from unexplainability. Highly accurate, complex, deep learning-based 'black-box' models are typically favoured over those that are less accurate but more interpretable by natural conventional ML models, such as linear regression, decision trees, and support vector machines. A major challenge is explaining the decisions of

multi-agent systems, in which numerous 'black-box' models are used together. These AI-based 'black-box' models are often deemed non-trustworthy because they are susceptible to unexpected errors; that is, they can be fooled in ways that humans cannot. Despite extensive studies having been conducted in the past several years to design techniques to make AI methods more explainable, interpretable, and transparent to developers and users, many open questions remain. Therefore, further research should be conducted to novel explainability approaches for various data analysis models and for different data processing architectures, as well as to investigate the combinations of multiple explainability methods in data-driven applications.

**Challenge 6**: The application of state-of-the-art ML methods for solving real-world problems in different domain scenarios remains topical and challenging. The increase in available data and digitalisation in many domains has resulted in new problem statements in different applications. Therefore, it is very important to propose and investigate alternatives, and to select an appropriate model for each scenario. Special reference architectures should be developed for complex problems that contain numerous stages, at which pipelines of different ML methods should be applied for data processing.

In summary, our purpose was to create novel, computationally intensive, explainable, and privacy-preserving ML methods, to investigate the properties thereof, to propose novel distributed and decentralised ML methods for prospective baselines, and to evaluate and apply these in various domains. This thesis is divided into three research directions, the contributions of which address the aforementioned challenges (Figure 1.2).

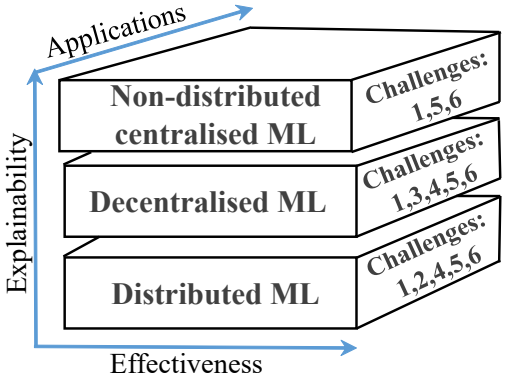



**Fig. 1.2** Research directions

- **Non-distributed, centralised, computationally intensive ML methods**: We propose novel, computationally intensive ML methods, investigate their properties, describe 'black-box' models with explainability methods, and apply these to real-world problems.

- **Decentralised ML methods**: We propose novel decentralised ML methods, investigate the synchronisation mechanisms of local models, apply these in cloud computing-enabled multi-agent system architecture for the data processing step, and evaluate the findings using real-world transportation scenarios.

- **Distributed centralised ML methods**: We investigate distributed ML methods for prospective computationally intensive centralised baselines, propose algorithms for the parallel execution of those methods, apply federated collaborative learning techniques to real-world problems with privacy-sensitive data, and propose an explainable federated learning algorithm.

## 1.1 Contribution and structure of thesis

Research and development in all underlying directions need to be conducted and the results must be combined to achieve progress in data-driven future AI-based systems, as well as to advance and create more sophisticated systems. ML-based data analysis is a pivotal area for the success of AI-based systems. This thesis is based on the contribution made by 11 papers to the theoretical and practical aspects of state-of-the-art and proposed ML methods (Figure 1.3). The recent contributions to each of the highlighted research directions are briefly categorised in this section and are subsequently detailed in the following chapters.

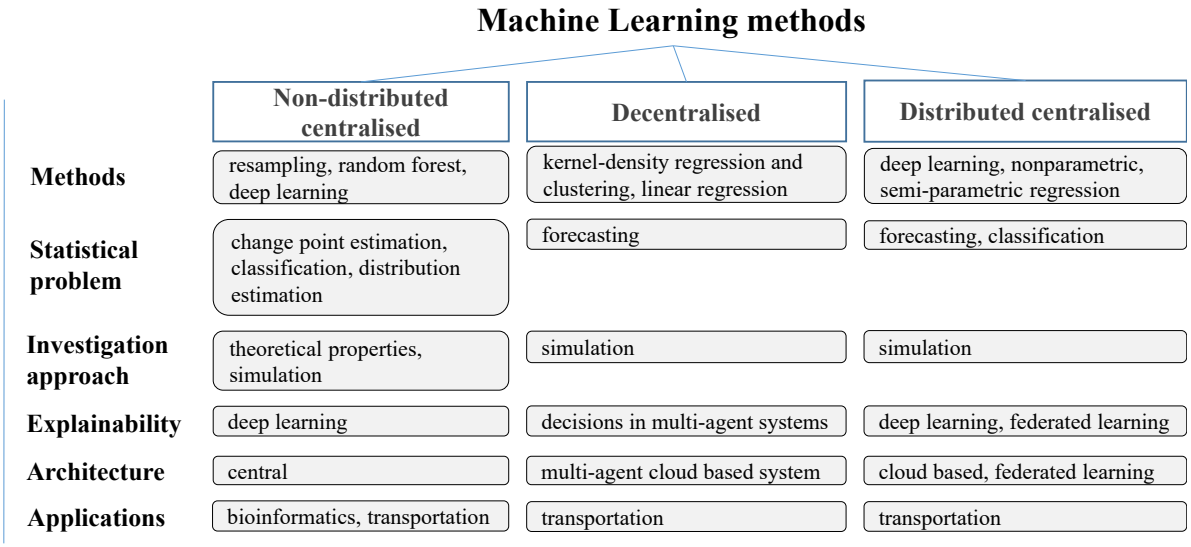



**Fig. 1.3** My contribution areas

- **Non-distributed, centralised, computationally intensive ML methods (Chapter 2)**; challenges: 1, 5, and 6:
  - We introduce a novel resampling-based method for selecting the shortest itinerary and apply this algorithm to model individual routing preferences in a cloud-based, distributed mobility network. We investigate the properties of the proposed algorithm and demonstrate its advantages over other state-of-the-art methods.
  - We present a novel resampling-based change-point estimation algorithm, investigate its properties, validate its results, and compare it with other state-of-the-art methods.
  - We apply deep learning and random forest ML methods to small RNA metadata prediction to solve a classification problem. We address the explainability challenge of the deep learning model and propose a novel explainability algorithm to interpret the results of the considered 'black-box' models.
- **Decentralised ML methods (Chapter 3)**; challenges: 1, 3, 4, 5, and 6:
  - We investigate the data processing step and connect it to the following decision-making steps in the context of a multi-agent system architecture for an intelligent transportation system.

  - We create decentralised linear and kernel-based regression algorithms that facilitate collaboration among the multi-agent system participants. The synchronisation mechanism of the models enables improved prediction accuracy of the individual agent models. We evaluate this approach by solving a travel-time prediction problem.
  - We investigate the explainability of AI-based models by analysing explanation generation techniques to increase user satisfaction and trust in such systems. Thus, we propose the explainability concept for AI-based decisions in multi-agent systems.
- **Distributed centralised ML methods (Chapter 4)**; challenges 1, 2, 4, 5, and 6:
  - We propose distributed parallel versions of several computationally intensive regression types and demonstrate that their parallel cloud computing-based execution increases the speed of those algorithms working with large datasets. We highlight the advantages of parallelisation by conducting various goodness-of-fit, scalability, and speed-up experiments with real-world datasets.
  - We address the data privacy challenge for processing distributed data sources using federated learning. This technique enables partner collaboration without the exchange of raw data, but only parameters that improve the prediction accuracy of the individual models. We propose a deep learning-based federated architecture for taxi travel-time prediction. Moreover, we presente the explainable federated learning concept and investigate the corresponding explainability algorithm.

## 1.2 Background and related work

In this section the state-of-the-art centralised, distributed and decentralised ML methods with the main focus on their computational intensiveness, explainability, data privacy and application are reviewed. These ML methods will be referenced afterwords in the next chapters.

### 1.2.1 Overview of ML methods

ML has a lengthy history, including classical simple linear regression, support vector machines, and decision trees, which remain popular even today owing to their simplicity and interpretability. These methods have been successfully applied in various domains (e.g. transportation [Chowdhury et al., 2017] and econometrics [Henderson and Parmeter, 2015]). With the development of computational techniques, more sophisticated data-driven ML methods have started to appear, such as non-parametric [Henderson and Parmeter, 2015], resampling, ensemble, and deep learning methods. These methods are often 'black-box' models that need to be explained, which has led to the development of various explainability methods. In this subsection, we provide a brief overview of the ML and explainability methods on which our contributions are based.

**Multivariare linear regression**: The classical multivariate linear regression model and its well known least squares estimator $\mathbf{b}$ of $\boldsymbol{\beta}$ are:

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}; \quad \mathbf{b} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y} \tag{1.1}$$

where $\mathbf{Y}$ is an $n \times 1$ vector of dependent variables; $\boldsymbol{\beta}$ is an $m \times 1$ vector of unknown parameters of the system to be estimated; $\mathbf{X}$ is an $n \times m$ matrix of explanatory variables; $\boldsymbol{\epsilon}$ is an $n \times 1$ vector of random errors, $\{\epsilon_i\}$ are mutually independent, have zero expectation, $E[\boldsymbol{\epsilon}] = 0$, and equal variances, $V[\boldsymbol{\epsilon}] = \sigma^2\mathbf{I}$, where $\mathbf{I}$ is an $n \times n$ identity matrix. Note, that the calculation of $\mathbf{b}$ requires information about complete matrix $\mathbf{X}$.

For real-time streaming data the recurrent iterative method for the least squares estimator [Albert, 1972, Andronov et al., 1991] is:

$$\mathbf{b}_{t+1} = \mathbf{b}_t + \mathbf{K}_{t+1}(Y_{t+1} - \mathbf{x}_{t+1}\mathbf{b}_t), \; t = 0, 1, \ldots, \tag{1.2}$$

where $\mathbf{b}_t$ is the estimate after $t$ first observations, $Y_{t+1}$ is the value of a dependent variable, and $\mathbf{x}_{t+1}$ is the values of explanatory variables of the $t + 1$ observation; $\mathbf{K}_{t+1}$ is an $m \times 1$ vector of proportionality, smoothness, or compensation. $\mathbf{K}_{t+1}$ is calculated with an adaptive forecasting method based on exponential smoothness [Andronov et al., 1991]. The forecasted value of the dependent variable at the $k$-th future time moment is:

$$E(Y_k) = \mathbf{x}_k\mathbf{b}, \tag{1.3}$$

where, $\mathbf{x}_k$ is a vector of observed values of explanatory variables for the $k$th future time moment.

**Non-parametric regression**: This type of regression provides a versatile method for exploring a general relationship between two variables. It can predict observations yet to be made without reference to a fixed parametric model and provides a tool for finding spurious observations by studying the influence of isolated points. Moreover, this method constitutes the flexible method of substitution or interpolating between adjacent $\mathbf{X}$-values for missing values [Henderson and Parmeter, 2015].

The formal representation of this regression model [Härdle et al., 2004] with a dependent variable $Y$ and a vector of $d$ regressors $X$ is:

$$Y = m(\mathbf{x}) + \epsilon, \tag{1.4}$$

where $\epsilon$ is the disturbance term such that $E(\epsilon|\mathbf{X} = \mathbf{x}) = 0$ and $Var(\epsilon|\mathbf{X} = \mathbf{x}) = \sigma^2(\mathbf{x})$, and $m(\mathbf{x}) = E(Y|\mathbf{X} = \mathbf{x})$. Further, let $(\mathbf{X}_i, Y_i)_{i=1}^n$ be the observations sampled from the distribution of $(\mathbf{X}, Y)$. Then the Nadaraya-Watson kernel estimator is

$$m_n(\mathbf{x}) = \frac{\sum_{i=1}^n K\left(\frac{\mathbf{x}-\mathbf{X}_i}{h}\right) Y_i}{\sum_{i=1}^n K\left(\frac{\mathbf{x}-\mathbf{X}_i}{h}\right)} = \frac{p_n(\mathbf{x})}{q_n(\mathbf{x})} = \frac{p_{n-1}(\mathbf{x}) + K\left(\frac{\mathbf{x}-\mathbf{X}_n}{h}\right) Y_n}{q_{n-1}(\mathbf{x}) + K\left(\frac{\mathbf{x}-\mathbf{X}_n}{h}\right)}, \tag{1.5}$$

where $K(\bullet)$ is the kernel function of $R^d$ and $h$ is the bandwidth. We considered a multi-dimensional Gaussian kernel function $K(u) = K(u_1, u_2, \ldots, u_d) = K(u_1) \cdot K(u_2) \cdot \ldots \cdot K(u_d)$. The kernel density estimator has a simple recursive windowing method that allows the recursive estimation using the kernel density estimator.

**Partial linear model**: This is semi-parametric regression type [Härdle et al., 2004], [Liang, 2006]. Currently, several efforts have been allocated to developing methods that reduce the complexity of high dimensional regression problems [Henderson and Parmeter, 2015]. The models allow easier interpretation of the effect of each variable and may be preferable to a completely non-parametric model. These refer to the reduction of dimensionality and provide an allowance for partly parametric modelling. Additionally, partial linear models are more flexible combining both parametric and non-parametric components. The partial linear model is:

$$E(Y|\mathbf{U},\mathbf{T}) = \mathbf{U}\boldsymbol{\beta} + m(\mathbf{T}),$$

where $\boldsymbol{\beta}_{p\times 1}$ is a finite dimensional vector of parameters of a linear regression part, $m(\cdot)$ is a smooth function, and the explanatory variables $\mathbf{X}$ are split into two parts: linear part $\mathbf{U}$, and nonparametric part $\mathbf{T}$. Economic theory or intuition should ideally guide the inclusion of the regressors in $\mathbf{U}$ or $\mathbf{T}$, respectively.

**Resampling method**: These methods assume an application of iterative calculations instead of complex analytical models and statistical procedures by using available data in different combinations (resampling, sub-sampling, bootsrtep methods). The resulting solution is approximate; however in many practical situations (too big or too small training datasets, complex and hierarchical structure of analysed system, dependency in data) this may give more robust and precise results as conventional analytical methods or even provide a solution in the situations where conventional methods fail. With the term computationally intensive ML we refer to such methods as sampling [Thompson, 2012], resampling [James et al., 2021], bootstrap [Efron and Tibshirani, 1993] [Davison and Hinkley, 1997], cross validation, kernel density based estimation etc. They showed their efficiency for various applications in forecasting [Afanasyeva and Andronov, 2005, Wu, 1986], clustering [Hinneburg and Gabriel, 2007], change-point analysis [Fiosina and Fiosins, 2011]. For streaming data, computationally intensive methods provide data pre-processing by selection resamples of data and obtaining representative samples, which is the only reasonable way to analyse the data [Leskovec et al., 2020]. Data filtering method based on targeted sequential resampling and model mixtures of distributions using Markov chain Monte Carlo method was introduced in [Manolopoulou et al., 2010].

**Ensemble methods**: These methods are based on the sub-sampling described above, but organised in more complex hierarchical archtecture. Very popular state-of-the-art ensemble methods are random forest [Breiman, 2001] and extreme gradient boosting [Chen and Guestrin, 2016] (XGBoost), which are based on ensembling and resampling of decision are computationally intensive and provide accurate predictions. Moreover, random forest outperforms other conventional classifiers for very high-dimensional data [Breiman, 2001].

A random forest classifier requires lesser training data in comparison with the deep learning classifier and allows the interpretation of features by generating variable importances. However, the random forest classifier is sensitive to class imbalance [O'Brien and Ishwaran, 2019].

**Deep learning**: Deep-learning becomes very popular nowadays, because of the ability of contemporary computers to process quickly very complicated models. Deep learning is especially computationally intensive method. It became especially popular during the past 10 years, with the development of more powerful computers. In many supervised learning problems like forecasting and classification deep learning gives comparable results with such ensemble methods

like random forest or XGBoost. Deep learning has introduced major advances for solving problems that have remained unsolved by the AI community for many years [LeCun et al., 2015]. It is able to analyze big data and is robust enough to treat large amounts of noisy training data [LeCun et al., 2015], [Xiao et al., 2015]. Its disadvantage is that, it requires large amounts of training data [Li et al., 2019c], is prone to overfit for small training sets and is difficult to biologically interpret (feature importance) [Webb, 2018]. In [Kong and Yu, 2018] the random forest and deep learning approaches were used in two stages. For the first stage, the random forest approach was used to extract the most important features and then for the second stage, the deep learning approach was implemented for gene expression data classification based on the selected features. Various applications of deep learning in transportation were discussed in [Varghese et al., 2020].

**Explainability methods:** Conventional ML methods, such as linear regression, decision trees, and support vector machine, are interpretable by nature. Typically, highly accurate complex deep learning-based 'black-box' models are favoured over less accurate but more interpretable conventional ML models. These AI-based 'black box' models are often deemed non-trustworthy since they are susceptible to surprising errors, i.e. they can be fooled in ways humans cannot. Extensive studies have been carried out in the past few years to design techniques to make AI methods more explainable, interpretable, and transparent to developers and users. Post-Hoc methods try to explain the behaviour of 'black box' model (e.g. layer-wise relevance propagation for deep learning [Bach et al., 2015, Montavon et al., 2017]; local interpretable model-agnostic explanation [Molnar, 2019] application for Bayesian predictive models [Peltola, 2018] and for convolutional neural network [Mishra et al., 2017], reversed time attention model of recurrent neural networks [Choi et al., 2016]; 'black box' explanations through transparent approximations [Lakkaraju et al., 2017]). Ante-Hoc methods are conventional ML methods, which are already interpretable from its nature, but often the accuracy of such approach is less: decision trees, linear regression, random forest, generalized additive model.

Model-agnostic and model-specific explanation methods have been reported. Model-agnostic methods, such as LIME [Ribeiro et al., 2016], Sharpley Values, are implementable for each model. However, they require a large number of computations and often are not applicable for big datasets used in deep learning [Molnar, 2019]. Sharpley values method was implemented in [Wang, 2019] to interpret a vertical federated learning model.

Model-specific methods are more suitable for deep learning, which focus on only one type of model and thus are more computationally effective, e.g., DeepLIFT [Shrikumar et al., 2017a], [Ancona et al., 2018], Integrated Gradients [Sundararajan et al., 2017].

### 1.2.2 Distributed ML methods

**Distributed data analysis and multi-agent systems**: A large amount of contemporary generated data in various domains requires proper processing and analysis. An application of the state-of-the-art methods of distributed/decentralised ML methods as well as creation of novel more effective approaches capable of processing distributed big data sources with data privacy restrictions is required.

In a case when data centralisation is available, accurate and computationally effective prediction models can be developed, which addresses the big data challenge through smart 'artificial' partitioning and parallelisation of data and computation within a cloud-based architecture or powerful super computers [Fiosina and Fiosins, 2017]. Computationally intensive methods would seem ideally suited to straightforwardly leveraging parallel and distributed computing architectures: one might imagine using different processors or computing nodes to process different resamples independently in parallel. 'The Bag of Little Bootstraps' procedure [Kleiner et al., 2014] incorporates features of both bootstrap and subsampling to yield a robust, computationally efficient means of assessing the quality of estimators and can be successfully used for ensemble distributed learning [de Viña and Martínez-Muñoz, 2018]. A 'Divide and conquer' approach for distributed data analyses [Chen et al., 2021, Chen, 2013] assumes to divide the data into a number of subsets, process them in parallel and finally merge the results. This approach is especially useful, when datasets are extraordinary large to be processes on a single computer.

Often, data should be physically and logically distributed without transmission of big information volumes, without the need to store, manage, and process massive datasets in one location. This approach enables a data analysis with smaller datasets. However, scaling it up requires novel methods to efficiently support the coordinated creation and maintenance of decentralised data models. Moreover, decentralised systems adapt quicker to situations in real time as well as some nodes of the distributed system prefer rely mostly on their own local information and experience making forecasting process more autonomous [Fiosins et al., 2011].

Specific distributed (e.g., federated learning) or even decentralised (e.g., multi-agent systems) architectures should be applied to support the decentralised data analysis, which requires a coordinated suite of individual local data models, including parameter/data exchange protocols and synchronisation mechanisms [Hinkelmann et al., 2018] among the decentralised data models. Local processing of distributed data sources often lead to less accurate local prediction models, in which proper coordination and synchronisation algorithms should be investigated. Decentralised coordinated methods outperform uncoordinated local ones and are compared to the centralised approach taking accuracy as efficiency criterion [Fiosina et al., 2013a].

The decentralised data analyses based on data-driven multi-agent systems paradigm provides a unified model addressing organizational aspects in application domains (such as ownership and access to data)[Wooldridge, 2009], local processing and decision-making [Klusch et al., 2003, da Silva et al., 2005, Khalil et al., 2015] as well as communication [Hinkelmann et al., 2018], coordination, and cooperation between knowledge sources [Cao et al., 2009]. One key challenge in multi-agent systems is to develop methods and protocols to analyze distributed data sources in a cooperative way and to provide and communicate sufficient information for optimal decisions [Freitas, 2002]. A very significant problem, which arises with parallel execution of separate data models, is connected with future synchronization of these models and corresponding data exchange [Chen, 2013]. Decentralised and adaptive K-Means clustering for Non-independent and/or non-identically distributed data was considered in [Soliman et al., 2020].

**Federated learning**: This is a distributed ML approach after [Konecný et al., 2016] that proposes to organize the inter-organizational collaboration without sharing the data but only model parameters [Yang et al., 2019].

The advantages of this approach are: (i) no need to transmit the original data to the cloud; (ii) the computational load is distributed among the participants, and (iii) distributed model synchronisation ensures more data and more accurate models. However, the application of the federated approach leads to a number of challenges connected with data homogeneity, partner trust and misbehavior or systems reliability ([Bonawitz et al., 2019]).

Federated learning algorithms may use a central server that orchestrates the different steps of the algorithm or they may be peer-to-peer, where no such central server exists.

Some parametric ML methods as support vector machine, linear regression, neural networks can be naturally implemented for federated learning. In deep neural networks a lot of training data is required, thus federated learning can help to gather distributed data keeping its privacy.

Federated learning relies on an iterative process broken down to a set of client–server interactions referred to as round including transmitting the current global model state to participating nodes, training local models on these local nodes to produce a set of potential model updates at each node, and aggregating and processing these local updates into a single global update and applying it to the global model.

We consider $N$ data owners $\{F_i\}_{i=1}^{N}$, who wish to train an ML model by consolidating their respective data $\{D_i\}_{i=1}^{N}$. A centralised approach uses all data together $D = \cup_{i=1}^{N} D_i$ to train a model $M_\Sigma$. A federated system is a learning process in which the data owners collaboratively train a model $M_{FD}$, where any data owner $F_i$ does not expose its data $D_i$ to others. In addition, the accuracy of $M_{FD}$, denoted as $V_{FD}$, should be very close to the performance of $M_\Sigma$, $V_\Sigma$. Formally, we consider $\delta$, a nonnegative real number; if $|V_{FD} - V_\Sigma| < \delta$, we can state that the federated learning algorithm has $\delta$ -accuracy loss [Yang et al., 2019]. Each row of the matrix $D_i$ represents a sample, while each column represents a feature. Some datasets may also contain label data. The feature $X$, label $Y$, and sample IDs $I$ constitute the complete training dataset ($I$,$X$,$Y$). The feature and sample space of the data parties may not be identical. We classify federated learning into horizontal, vertical, and federated transfer learning based on the data distribution among various parties.

**Distributed data analysis in transportation**: The mentioned challenges are topical in the transportation domain, in which the generation and processing of big data are necessary [de la Torre et al., 2021]. Ubiquitous traffic sensors and the Internet of Things concept create a world-wide network of interconnected uniquely addressable cooperating objects, which enable exchange and sharing of information.

The multi-agent system based representation of transportation networks helps overcome the limitations of centralised data analyses, which will enable autonomous vehicles to make better and safer routing decisions [Fiosins et al., 2011] based on the current traffic state [Chlyah et al., 2016]. A combination of centralized and decentralized agent-based approaches to traffic control problems was introduced in [Fiosins et al., 2016], where the agents maintain and share the 'local weights' for each link and turn, periodically exchanging this information with a central traffic information center. There are several cloud or/and multi-agent based architectures for managing traffic networks [Li et al., 2011, Wang, 2008, Lee et al., 2010]. A cloud-based architecture for decentralised big transportation data analyses was proposed in [Fiosina et al., 2013a].

Travel-time estimation, as an important parameter of transportation networks, which accurate prediction helps to reduce delays and transport delivery costs, improves reliability through better selection of routes and increases the service quality of commercial delivery by bringing goods

within the required time window [Büchel and Corman, 2020]. A centralized travel-time prediction was considered in-vehicle route guidance and advanced traffic management systems (e.g., [Lin et al., 2005a]) or for each network link by kernel density estimator [Duan et al., 2019]. A decentralized travel-time forecasting using neural networks, in which travel-time is predicted for each link of the network separately was considered in [Claes and Holvoet, 2011].

Often proper travel-time forecasting model needs a pre-processing step like data filtering and aggregation. Travel-time aggregation models (non-parametric, semi-parametric) for decentralized data clustering and corresponding coordination and parameter exchange algorithms were researched in [Fiosina et al., 2013b]. Two decentralised regression models multivariate linear and kernel-density based for travel-time prediction were proposed in [Fiosina, 2012] , [Fiosina and Fiosins, 2012].

The impact of incorporating decentralised data analysis methods into multi-agent-based applications in traffic and logistic domains has been assessed. Initial requirements and ideas for methods of decentralised data analysis development in the transportation domain operating with big data flows have been identified [Fiosina et al., 2013b]. An optimal route selection based on an analysis of renewal processes has been investigated [Fiosina and Fiosins, 2014].

In the next sections, our contribution will be represented in more technical details.

# Chapter 2

## Computationally intensive ML methods for data analysis

Various state-of-the-art ML methods that assume central realisation are addressed in this chapter. First, we focus on the application of deep learning and ensemble-based (e.g., random forest) methods for solving bioinformatics problems. Moreover, the explainability of these methods is investigated and corresponding algorithms are proposed. Second, we investigate the creation of novel resampling-based ML methods, study their properties, and validate their results using real-world data from the transportation domain.

### 2.1 Deep learning for data augmentation

Data annotations (tissue, age, sex, etc.) are crucial for the re-use of data. A detailed description of the biological conditions in which data has been obtained is required to extract new information from the obtained data. The data should be findable, accessible, interoperable, and reusable, which ultimately facilitates knowledge discovery [Wilkinson et al., 2016]. Annotations are an essential part of semantic data integration systems and allow for a deeper analysis of the data [Madan et al., 2018]. So far, metadata is often not stored together with the expression data and the available metadata is often not normalized, and is unstructured and incomplete. The widely used GEO repository [GEO, ], for example, stores annotations as a number of free-text description fields. This leads to missing and/or inaccurate annotations and requires revisions and manual corrections by an expert [Hadley et al., 2017].

To distinguish between biological conditions, different ML methods were applied.
In [Guo et al., 2017] and [Hadley et al., 2017], the sex in different micro ribonucleic acid (miRNA) tissue samples was defined using differential expression analysis. In [Hadley et al., 2017], the authors used differential expression analysis and analysis of variance to detect the sex differences in several tissues in miRNAs. In [Ellis et al., 2018], the age, sex, and tissue were predicted from mRNA sequencing (mRNA-Seq) expression data using a regression-based approach. massiR [Buckberry et al., 2014] is a method for sex prediction based on gene expression microarrays using clustering. Many studies use a random forest method for the classification of expression data, particularly in disease diagnostics [Statnikov et al., 2008]. [Johnson et al., 2018] provides a good overview of ML methods for expression data analysis.

**Deep Learning and Random Forest-Based Augmentation of sRNA Expression Profiles**, J. Fiosina, M. Fiosins, S. Bonn, In. Proc. of the Int. Symposium on Bioinformatics Research and Applications, LNCS, 11490, 159-170, Springer, 2019

In this study, we aim to predict the metadata based on deep-sequenced small RNAs' (sRNAs') expression profiles by formulating this prediction as a classification problem. Small ribonucleic

acids (sRNAs) are short (less then 200 nt), usually non-coding RNA molecules with many crucial biological functions [Storz, 2002]. The basic rationale for this approach is that data with similar sRNA expressions should have similar metadata. Based on this assumption, we hypothesize that sRNA expression profiles contain enough information to predict the sRNA tissue and sex accurately.

In this study we investigate whether the deep learning-based data augmentation could be superior to conventional ML approaches, such as random forest. The main hypothesis is that deep learning classifier trained on sufficiently large data sets would generalise more efficiently to yet unseen datasets. Whereas single unseen samples might be easy to learn, datasets usually contain a distinct experimental bias that the model has not learnt a priori. We apply deep learning and random forest models on human small RNA-seq datasets from the sRNA expression atlas (SEA, http://sea.ims.bio) [Rahman et al., 2017], a database containing well-structured, manually curated, ontology-based annotations of publicly available sRNA-Seq data. Every sample is semantically annotated and analysed with the same workflow (OASIS [Rahman et al., 2018], https://oasis.dzne.de), increasing data interoperability while reducing analysis bias.

We use 4243 annotated sRNA-Seq samples from the small RNA expression atlas (SEA) database to train and test the augmentation performance (Figure 2.1). In general, the deep learning learner outperforms the random forest method in almost all tested cases. The average cross-validated prediction accuracy of the deep learning algorithm for tissues is 96.5% and for sex is 77%. The average tissue prediction accuracy for a completely new dataset is 83.1% (deep learning) and 80.8% (random forest).

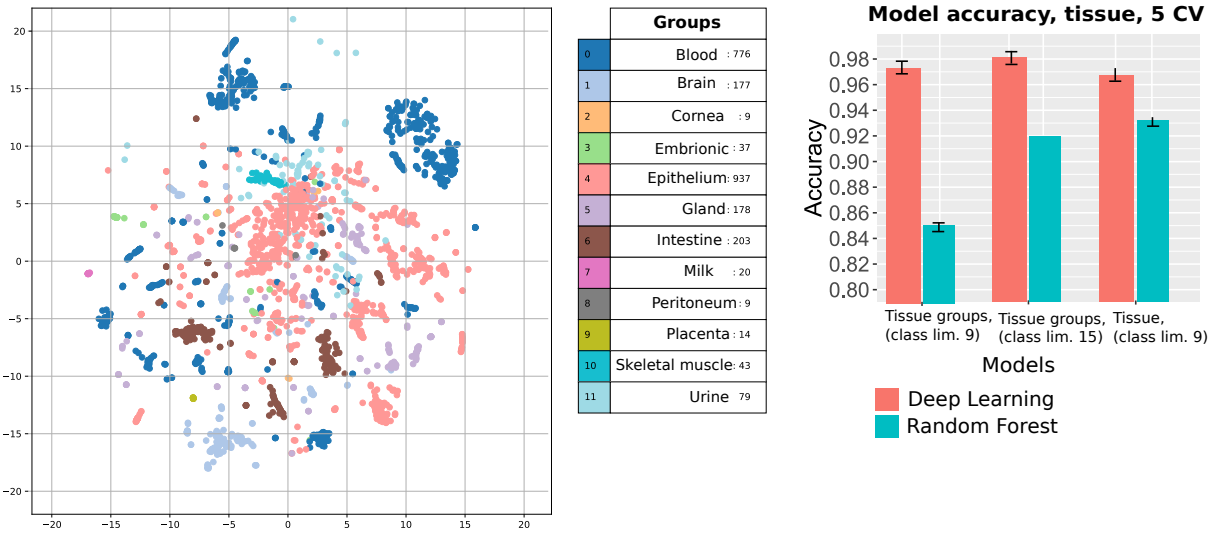



**Fig. 2.1** tSNE distances for tissue group (left) and tissue classification (right)

Often such methods as random forest and deep learning give more accurate results as conventional: linear regression, decision trees, vector support machine, which are interpretable by nature. However, those methods are in-transparent black-box models. We propose a method for interpetation of the results of the deep learning model towards data augmentation.

**Explainable Deep Learning for Augmentation of Small RNA Expression Profiles**, J. Fiosina, M. Fiosins, S. Bonn, Journal of Computational Biology 27 (2), 234-247, 2020.

In this study, we continue the previous problem statement and show that deep learning algorithms outperform random forest-based data augmentation for age annotations using sRNA expression profiles, if enough training data is available. More specifically, the deep learning method performs better than the random forest method for cross-validation experiments as well as on "one dataset out" experiments. The average cross-validated prediction accuracy of the deep learning algorithm for age is 77.2%. Moreover, we have demonstrated how backpropagation can provide a biological interpretation of relevant features for the deep learning classification of tissue, sex, and age.

To understand which sRNAs influence deep learning predictions, we employ backpropagation-based feature importance scores using the DeepLIFT method [Shrikumar et al., 2017b], which enable us to obtain information on biological relevance of small RNAs.

To biologically trace decisions of the deep learning model to the input features, we use DeepLift scores. DeepLift [Shrikumar et al., 2017b] is an approach to assign importance scores, which demonstrate how important the value of each particular input is for each particular output. The scores are assigned according to a difference between a given input with some reference (neutral) input. The DeepLift method overperforms other scoring methods [Shrikumar et al., 2017b]; thus, it has been selected for our analysis. The DeepLift method calculates scores by backpropagating the contributions of all neurons in the network to every feature of the input. Consequently, for each sample $i$, each input neuron $j$, and each output neuron $k$, a score $C_{i,j,k}$ is calculated, which represents an importance of an input $j$ for the output $k$ in the $i$-th input sample (according to a reference input).

We have provided a three-step explanation of our augmentation models. First, we use a heatmap to visualize the DeepLift scores of an individual sample (Figure 2.2). This enabled us to understand, which small RNAs are important for a particular prediction.

Second, we analyse important small RNAs for each class $k$. We select examples, which belonged to the class $k : y_i = k$ and calculate the average difference scores for the correct class and other classes:

$$D1_{j,k} = avg_{y_i=k}(C_{i,j,k} - avg_{k' \neq k} C_{i,j,k'}).$$

Then, we select the top $N$ small RNAs $j$ according to $D1_{j,k}$ for each class $k$.

Finally, we investigate the number of small RNAs to be removed (to set their expression to 0), to change the classification results. For each sample $i$ of class $y_i = k$ and each class $k' \neq k$, we calculate the score differences $D2_{i,j,k'} = C_{i,j,y'} - C_{i,j,k'}$.

We order the differences $D2(i, j, k')$ in descending order and set the expression of small RNAs $j$ one by one, setting their expression to 0. We stop the process when the classification changes from $k$ is $k'$ (similarity analysis) or to any other class $k''$ (doing max 500 steps). We calculate average values for each pair of classes. The obtained matrix demonstrates "stability" of class (or "class similarity"), showing how many sRNAs should be removed to get from class $k$ to class $k' \neq k$ (stability analysis). The corresponding average number of steps has been applied to a matrix, which demonstrated "class stability" (or "class similarity").

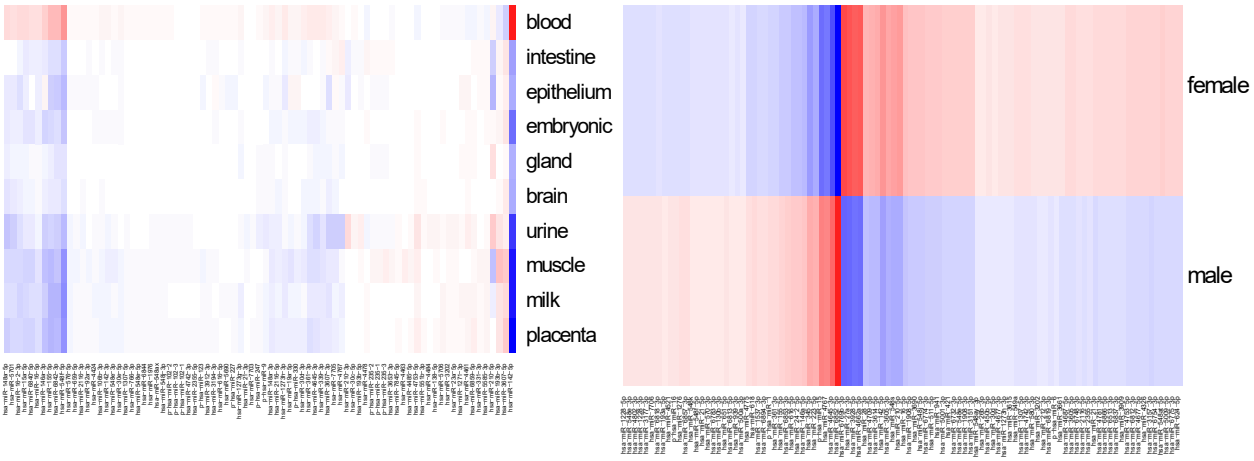



**Fig. 2.2** DeepLIFT scores for tissue group (left) and sex (right) classification.

We demonstrate that DL models can be explained both for individual samples and on average (Figure 2.2). we see some sRNAs clearly voting for the class (red) or against the class (blue).

## 2.2 Resampling-based integrated decision making

In this section, we create novel resampling-based methods, investigate their properties and apply in transportation domain.

We consider a cloud-based traffic control and management system, in which cloud central agents are assisting individual traffic agents in decision making, taking into account common and individual experience. The selection of the shortest itinerary, which requires route comparison on the basis of historical data and dynamic observations for both central cloud and individual agents. Resampling based algorithms can help at the data processing step, which results can be used afterwards for the decision making.

**Resampling based modelling of individual routing preferences in a distributed traffic network**. J. Fiosina, M. Fiosins, Int. Journal of Artificial Intelligence, 12 (1), 79-103, 2014.

This this study, we propose a cloud based architecture, in which we adapt the Markov chain based ranking algorithm [Negahban et al., 2012] for the ranking of routes and calculate the probability distribution $\pi^a$ over a set of the alternative routes $R$ of an agent.

In vehicle navigation routing engines do not customize results based on customer behavior. For example, some users prefer the quickest route while some prefer direct routes. This is because in vehicle navigation systems are traditionally embedded systems [Jin et al., 2020]. For example, during different times of day or weather conditions, drivers may make different routing decisions such as preferring or avoiding highways. In [Guo et al., 2020] the authors propose a leaning architecture based on on-line and off-line part, which uses the similar architecture proposed in out paper using neural networks.

We consider a directed graph $G = (V, E)$, where each edge $\{e_i\}_{i=1}^n \in E$ has an associated weight $\{X_i\}_{i=1}^n$ (e.g., travel-time), which are independent random variables with unknown distribution. A route in the graph is a sequence of edges such that the next edge in the sequence starts from the node, where the previous edge ends. Let $\{r^b\}_{b=1,2}$ be a set of routes. A route is defined as $r^b = \{e_{k_1^b}, e_{k_2^b}, \ldots, e_{k_{n_b}^b}\}$, where $(k_1^b, k_2^b, \ldots, k_{n_b}^b)$ is a sequence of edge indices in the initial graph, thus the route weight is $S^b = \sum_{i \in k^b} X_i$.

The samples of edge weights are collected locally by each agent, so $H_i^a = \{H_{i,1}, H_{i,2}, \ldots, H_{i,m_i^a}\}$ is the $a$-th agent sample of the weights of edge $i = 1, 2, \ldots, c$, $c \leq n$. An (unknown) true cumulative distribution function (cdf) of the sample $H_i$ elements is denoted by $F_i(x)$, $i = 1, 2, \ldots, c$, and the elements of samples $H_i^a$ for all $a$ have the same distribution $F_i(x)$, which means that all agents observe the same system. Each sample may correspond to one or several edges, because observations of two similar edges collected in one sample or no observations about an edge are available and an another edge sample is used instead.

During route selection process, an agent $a$ performs pairwise comparisons of routes, so the probability that the route $b$ has bigger weight than the route $b'$ is estimated: $p_{b,b'}^{*a} = P^*\{S^b > S^{b'}\}$.

Note that the estimates $p_{b,b'}^{*a}$ are consistent estimators of true probabilities $p_{b,b'} = P\{S^b > S^{b'}\}$ ($\lim_{m_i^a \to \infty} p_{b,b'}^{*a} = p_{b,b'}$). We propose a resampling procedure to calculate $p_{b,b'}^{*a}$.

We consider a procedure for a pairwise comparison of two non-overlapping routes. For simplicity we consider only two routes:t $b = 1$ and $b' = 2$ and calculate the probability $\Theta = P\{S^1 > S^2\}$.

Two cases are considered: (1) each edge has different samples, so only one element is extracted from the sample $H_i$; and (2) edges may correspond to common samples, including the common samples for two routes.

We propose an $N$-step resampling procedure. At each step, we randomly without replacement choose $\eta_i^1 + \eta_i^2$ elements from each sample $H_i$: $\eta_i^1$ elements for route 1, and $\eta_i^2$ elements for route 2: $\boldsymbol{\eta}_i = (\eta_i^1, \eta_i^2)$. Let $J_i^b(l)$, $|J_i^b(l)| = \eta_i^b$ be a set of element indices extracted from the sample $H_i$, for a route $b$, $b = 1, 2$, during resampling step $l$, $i = 1, \ldots, c$. Let $\mathbf{X}^{*l} = \bigcup_{i=1}^c \{H_{i,j} : j \in J_i^1(l)\} \cup \bigcup_{i=1}^c \{-H_{i,j} : j \in J_i^2(l)\}$ be the $l$-th resample of the edge weights for both routes, with the weights of route 2 assumed to be negative. Let $\Psi(\mathbf{x})$ be an indicator function, where $\mathbf{x} = (x_1, x_2, \ldots)$ is a vector of real numbers: $\Psi(\mathbf{x})$ is unity if $\sum_i x_i > 0$; otherwise, it is zero.

The average of $\Psi(\mathbf{X}^{*l})$ over all $N$ steps is accepted as the resampling estimator of the probability of interest: $\Theta^* = \frac{1}{N} \sum_{l=1}^N \Psi(\mathbf{X}^{*l})$. The corresponding resampling-based route comparison procedure is presented in Algorithm 1.

**Algorithm 1:** Function RESAMPLING COMPARE

**Function** RESAMPLING COMPARE$H_i, \eta_i, i = 1, \ldots, c, N$
**foreach** $l \in 1, \ldots, N$ **do**
  **foreach** $i \in 1 \ldots c$ **do**
    $X_i^{*l} \leftarrow extract(H_i, \eta_i^1 + \eta_i^2)$
    $X1_i^{*l} \leftarrow subsample(X_i^{*l}, 1, \eta_i^1)$; $X2_i^{*l} \leftarrow subsample(X_i^{*l}, \eta_i^1 + 1, \eta_i^2)$
  $\mathbf{X}^{*\mathbf{l}} = \bigcup X1_i^{*l} \bigcup -X2_i^{*l}$; $\Theta_l \leftarrow \Psi(\mathbf{X}^{*\mathbf{l}})$
$\Theta^* \leftarrow \frac{1}{N} \sum_{l=1}^N \Theta_l$
**return** $\Theta^*$

The function $extract(X, n)$ randomly chooses $n$ elements without replacement from the set $X$. The function $subsample(X, a, n)$ returns $n$ elements from $X$, starting from position $a$. These two cases differ with the parameters of the *extract* procedure.

The estimator $\Theta^*$ is obviously unbiased: $E(\Theta^*) = \Theta$, so we are interested in its variance. Consider the elements extracted at two different steps $l \neq l'$. Moreover, we denote: $\mu = E\ \Psi(\mathbf{X}^{*l})$, $\mu_2 = E\ \Psi(\mathbf{X}^{*l})^2$, $\mu_{11} = E\ \Psi(\mathbf{X}^{*l}) \cdot \Psi(\mathbf{X}^{*l'})$, $l \neq l'$. Then, the variance of the resampling estimator $\Theta^*$ is $V(\Theta^*) = E(\Theta^{*2}) - \mu^2 = \left\{\frac{1}{N}\mu_2 + \frac{N-1}{N}\mu_{11}\right\} - \mu^2$, for the estimation of which we need the mixed moment $\mu_{11}$ depending on the resampling procedure. The analytical expressions for the expectations and variances of the proposed estimators are derived, which allow theoretical evaluation of the estimators' quality. The experimental results demonstrate that the resampling estimator is a suitable alternative to the parametric plug-in estimator. The inferences for the variance of resampling estimator for both cases 'different samples for each edge' (based on $\omega$-pairs notation) and 'common samples for edges' (based on $\alpha$-pairs notation) [Fioshin, 2000], [Afanasyeva, 2005], [Andronov et al., 2009] were proposed. The experiments for a special case of normal distribution were conducted.

When, the agent decides about a route. It has a distribution $\pi^a$, calculated by its individual agent and receives a recommendation $\pi^{a'}$ from the central cloud agent $a'$. A decision idea is to create a mix of distributions $\pi^a$ and $\pi^{a'}$. The agent uses a constant $0 \leq \alpha^a \geq 1$. The following two-step procedure is used:

- A distribution $\pi^a$ is selected with a probability $\alpha$ and a distribution $\pi^{a'}$ with a probability $1 - \alpha$;
- A route is selected according to the selected distribution.

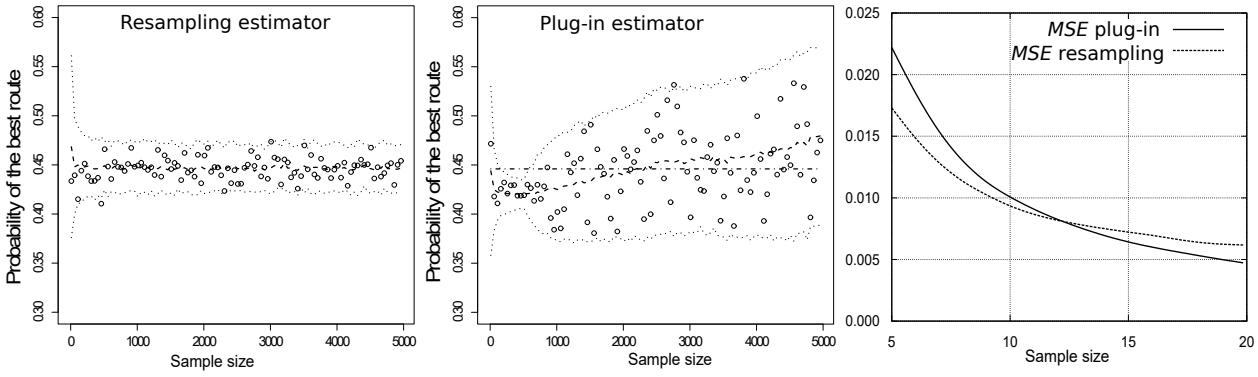



**Fig. 2.3** Resampling (left) and plug-in (center) estimators: values (circles), true value (a dash-dot line), mean (dashed line), deviation (dotted lines); MSE of the estimator for $\Theta = 0.5$ (right)

The results show that resampling estimates of the probability of interest give reliable unbiased forecasts with stable variance, while plug-in estimated are biased and their variance increases with increasing the sample size, so the resampling method outperforms the plug-in for big sample sizes (Figure 2.3, right, center). Comparing the mean squared errors of both estimators we can conclude that resampling estimators are more effective than plug-in estimators for small sample sizes (Figure 2.3, left).

## 2.3 Resampling-based change-point estimation

Change point analysis is an important part of data mining, which purpose is to determine if and when a change in a data set has occurred. Online detection of change point is useful in modeling and prediction of data sequence in application areas such as finance, biometrics [Gavit et al., 2009], robotics [Aroor et al., 2018] and traffic control [Carslaw et al., 2006], cyber attacks in connected vehicles [Comert et al., 2020], climate changes [Arif et al., 2017]. Distributed change-point detection algorithms were discussed in [Tartakovsky and Kim, 2006]. Change-point analysis can be used: 1) for determining if changes in the process control led to changes in an output, 2) for solving a class of problems, such as control, forecasting etc., and 3) trend change analysis [Gavit et al., 2009].

Traditional statistical approach to the problem of change-point detection is maximum likelihood estimation. First, data model is constructed and the likelihood function for change point is derived. Then, the estimator of change point is a result of the likelihood function minimization. This approach requires knowledge of exact data model and its parameters as well as complex analytical or numerical manipulations with likelihood function [Ferger, 2002]. In the case of small samples this approach does not allow to choose the probability distributions correctly and properly estimate their parameters and resampling approach is preferable.

One technique for detecting if and when a change-point (shift) has occurred is a cumulative sum chart (CUSUM chart). A form of a CUSUM chart allows to see visually if there is a change-point [Hinkley, 1971]. Currently a number of novel research works were conducted proposing new algorithms based on CUSUM for change-point identification [Abbasi and Haq, 2019, Otto and Breitung, 2020, Aroor et al., 2018].

A confidence level may be assigned for each detected change. It can be constructed using bootstrap [Efron and Tibshirani, 1993] or our investigated resampling approach.

**Resampling-based change point estimation**, J. Fiosina, M. Fiosins, In. Proc. of Int. Symposium on Intelligent Data Analysis, LNCS, 7014, 150-161, Springer, 2011.

The paper deals with bootstrap-based CUSUM change-point test [Gavit et al., 2009], slightly modified and described in terms of resampling approach [Andronov and Merkuryev, 2000], which allows more accurate analysis by estimating its theoretical properties. First, we derive analytical formulas to estimate the efficiency of this technique by taking expectation and variance as efficiency criteria. Second, we propose another simple resampling test, based on pairwise comparisons of randomly selected data and estimate its efficiency.

Let $\mathbf{X} = \{X_1, X_2, \dots, X_n\}$ be a sequence of random variables, which we divide into as $\mathbf{X} = \{\mathbf{X}^B, \mathbf{X}^A\}$. There exists a change point at a position $k$, if the variables $\mathbf{X}^B = \{X_1, X_2, \dots, X_k\}$ have a distribution $F_B(x, \Theta^B)$, but the variables $\mathbf{X}^A = \{X_{k+1}, X_{k+2}, \dots, X_n\}$ have a distribution $F_A(x, \Theta^A)$, $\Theta^B \neq \Theta^A$. The aim of change point analysis is to estimate the value of $k$. We observe the case, when the distributions $F^B(\cdot)$ and $F^A(\cdot)$ differ with a mean value.

For bootstrap-based CUSUM change-point test: first, a CUSUM chart is constructed, which presents a difference between sample data and a mean. Then, if there is no change-points in mean of sample data, the CUSUM chart will be relatively flat. Alternatively in the case of change-point existence, there will be obvious minimum or maximum in CUSUM chart. (Figure 2.4).

The cumulative sum $S_i$ at each data point $i$ is: $S_i = \sum_{j=1}^{i}(X_i - \bar{X})$, where $i = 1, 2 \ldots n$, $X_i$ is the current value, and $\bar{X}$ is the mean. A CUSUM chart starts at zero ($S_0 = 0$) and always ends at zero ($S_n = 0$). Increasing (decreasing) of the CUSUM chart means that the data $X_i$ are permanently greater (smaller) than the sample mean.

A change in the direction of CUSUM chart allows to spread about the change-point in the mean.

Figure 2.4 presents an example of initial sample (left) and corresponding CUSUMs (right); a bold line represents CUSUMs calculated on initial sample, dotted lines - CUSUMs on bootstrapped data. An initial CUSUM chart well detects a change point at $k = 10$.

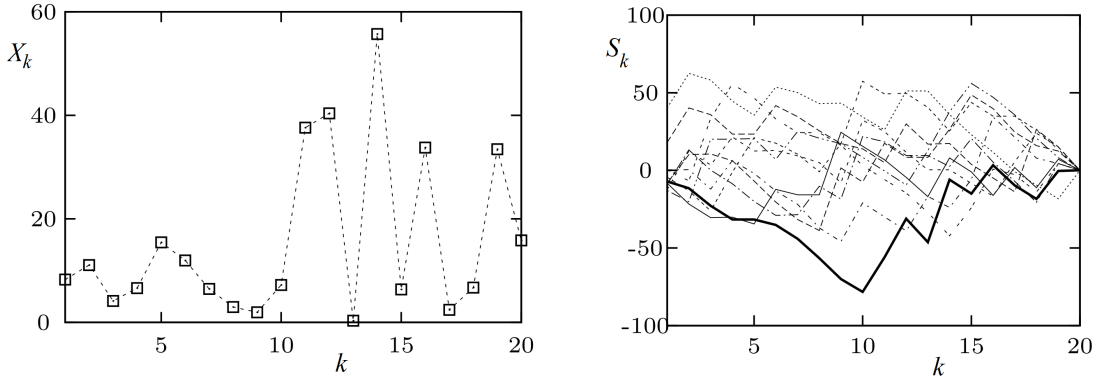



**Fig. 2.4** Sample data (left) and sample CUSUM chart (right)

For each change it is possible to calculate a confidence level using bootstrapping of initial data, which are randomly permuted. $N$ bootstraps are produced with the sample data set. For each bootstrapped data set we construct CUSUMs $S^*(r)$ and estimate its range, so for the $r$-th bootstrap iteration $\Delta S^*(r) = \max_i\{S_i^*(r)\} - \min_i\{S_i^*(r)\}$, $i = 1, 2, \ldots, n$.

The final step in determining of the confidence level is to calculate the percentage of times that the range for the original CUSUM data $\Delta S = \max_i\{S_i\} - \min_i\{S_i\}$ exceeds the range for the bootstrapped CUSUM data $\Delta S^*(r)$, $r = 1, 2 \ldots N$. Thus, we need to build an empirical cumulative distribution function of bootstrap ranges $\Delta S^*(r)$, as

$$\hat{F}_{\Delta S^*}(x) = \frac{\#\{\Delta S^*(r) \le x\}}{N} = \frac{1}{N}\sum_{r=1}^{N} 1_{\{\Delta S^*(r) \le x\}}. \tag{2.1}$$

Let $H^0$ be a hypothesis about no change-point in data against the alternative $H^1$. It is appropriate to set a predetermined confidence level $\gamma$, beyond which a change is considered significant.

Then using the cdf $\hat{F}_{\Delta S^*}(x)$ (2.1) we construct a bootstrap approximation of a confidence interval for $\Delta S$: $[\hat{F}^{-1}_{\Delta S^*}(\frac{1-\gamma}{2}); \hat{F}^{-1}_{\Delta S^*}(\frac{1+\gamma}{2})]$, where $\hat{F}^{-1}_{\Delta S^*}(\gamma)$ is the quantile of the distribution $\hat{F}_{\Delta S^*}$ of the level $\gamma$. If the interval does not cover the value $\Delta S$, thus initial and bootstrapped data significantly differ, and we reject $H^0$.

CUSUM test interpretation as a resampling [Andronov and Merkuryev, 2000] test allows to derive some its properties and estimate the efficiency on the base of expectation and variance of the estimator.

We consider a test for a change at a point $k$ under $H^0$. We deal with values of CUSUMs instead of the ranges.

We produce $N$ iterations of resampling procedure. At $r$-th iteration, we extract, without replacement, $k$ elements from the sample $\mathbf{X}$, forming the resample $\{X_1^{*r}, X_2^{*r}, \ldots, X_k^{*r}\}$ and construct the CUSUM estimator for the point $k$: $S_k^*(r) = \sum_{i=1}^k (X_i^{*r} - \bar{X}) = \sum_{i=1}^k X_i^{*r} - k\bar{X}$, where $\bar{X}$ is an average over the sample $\mathbf{X}$.

After $N$ such realizations we obtain a sequence $S_k^*(1)$, $S_k^*(2)$, $\ldots$, $S_k^*(N)$ and calculate the resampling estimator $F_k^*(x)$ of the distribution function of bootstrapped CUSUMS as $F_k^*(x) = \frac{1}{N}\sum_{r=1}^N 1_{\{S_k^*(r)\le x\}}$ and find the expressions for its expectation and variance.

Further we propose an alternative resampling change-point test. Let us test a point $k$. The idea behind this method is based of the consideration of the probability $P\{X \le Y\}$, where random variable $X$ is taken randomly from the subsample $\mathbf{X}^B$ and random variable $Y$ from the subsample $\mathbf{X}^A$. If the samples $\mathbf{X}^B$ and $\mathbf{X}^A$ are from one distribution, this probability should be equal to 0.5. However, for our test we scale this value by multiplying to the difference $y - x$ in the case when $x \le y$ and thus, our characteristic of interest is:$\Psi(x, y) = I_{\{x<y\}} \cdot (y - x)$.

Then we produce $N$ realizations of the following resampling procedure. On $r$-th realization we extract one value $X^{*r}$ from the sample $\mathbf{X}^B$ and one value $Y^{*r}$ from the sample $\mathbf{X}^A$, compare them and calculate the value of $\Psi(x, y)$. Thus, the resampling estimator is an average over all realizations of $\Psi(x, y)$:$\Theta^* = \frac{1}{N}\sum_{r=1}^N \Psi(X^{*r}, Y^{*r})$.

Such statistical properties as the expectation and variance of this estimators were also derived in the paper.

Numerical experiments show that the CUSUM test detects change-point very well; however, it may consider as a change-point some point, which is not one. In opposite, the pairwise test is more reliable in the case of a change-point absence; however, it can miss some change-point.

# Chapter 3

# Decentralised data analysis

Multi-agent systems generally represent a complex system that consists of autonomous interacting components. Such systems are usually characterised by big data, which are represented by large volumes of distributed data from various sources. One key challenge in multi-agent systems is the capability of the agents to process such distributed data to provide sufficient information for optimal decisions [Zargayouna, 2019]. Big data processing and mining provides an algorithmic solution for data analysis in a distributed manner to detect the hidden patterns in data and to extract the knowledge that is necessary [Galakatos et al., 2018] for decentralised decision-making [Ponomarev and Voronkov, 2017], [Symeonidis and Mitkas, 2005]. Data processing methods improve the agent intelligence and the performance of multi-agent systems [Rao1 et al., 2010, da Silva et al., 2005], which involve proactive and autonomous agents that perceive their environment, dynamically reason out actions based on the environment, and interact with one another. Furthermore, the coupling of multi-agent systems with data processing methods can be described in terms of ubiquitous intelligence [Cao et al., 2009], with the aim of fully embedding information processing into everyday life. In [Klusch et al., 2003], it was concluded that autonomous data mining agents, as a special type of information agents, may perform various mining operations on behalf of other user(s) or in collaboration with other agents.

In the following, we describe how the prediction and clustering problems of each individual agent can be solved collaboratively with higher accuracy using decentralised data analysis. Furthermore, we propose corresponding distributed (without central authority) ML methods and evaluate their performance on real-world data from the transportation domain.

## 3.1 Cooperative data analysis for data-driven agent-based cloud computing

Agent-based cloud computing is a paradigm that identifies several common problems and provides several benefits by the synergy between multi-agent systems and cloud computing [Shengdong et al., 2019]. Cloud computing is mainly focused on the efficient use of computing infrastructure through reduced cost, service delivery, data storage, scalable virtualization techniques, and energy efficiency. In contrast, multi-object systems are focused on intelligent aspects of agent interaction and their use in developing complex applications. In particular, cloud computing can offer a very powerful, reliable, predictable and scalable computing infrastructure for the execution of multi-agent systems by implementing complex, agent-based applications for modelling and simulation. Also, software agents can be used as basic components for implementing intelligence in clouds, making them more adaptive, flexible, and autonomic in resource management, service provisioning and large scale application executions [Talia, 2011].

**Mining the traffic cloud: Data analysis and optimization strategies for cloud-based cooperative mobility management**, J. Fiosina, M. Fiosins, J.P. Müller, In. Proc. of the Int. Symposium on Management Intelligent Systems, AISC, 220, 25-32, Springer, 2013

In contemporary intelligent transport systems, modeling and forecasting of traffic flows is one of the important techniques that need to be developed [Bazzan and Klügl, 2013]. It is an example of a complex stochastic system, in which many different factors should be estimated. Due to the limitations of centralised approach, decentralised multi-agent systems with autonomous agents allow vehicles to make decisions autonomously, which is fundamentally important for the representation of these networks [Zargayouna, 2019]. We demonstrate the advantages of decentralized architecture focusing on forecasting and clustering problems. We consider sample architecture of a cloud-based intelligent transport system, representing synergy of multi-agent systems, cloud computing and complex stochastic applications and explain the main data flows that appear there. We show as well how the data flows can be processed using data processing methods and provide sufficient information for fulfilling the user requests. The applications executed in the cloud are data-intensive. Therefore, services provided through the cloud require large amounts of data to be processed, aggregated, and analysed. Then, the processed data is used for calculating optimal strategies for traffic participants.

As computation is a bottleneck in cloud computing, a reasonable processing balance between local data sources (clients) and a cloud is required that depends on the client computation power. We consider reference architecture for data processing and decision-making stages in an intelligent transport system 3.1, previously proposed in our another contribution [Fiosina et al., 2013c], focusing on illustrating data flows and their processing as well as using results for optimization of participant strategies and fulfilling their requests.

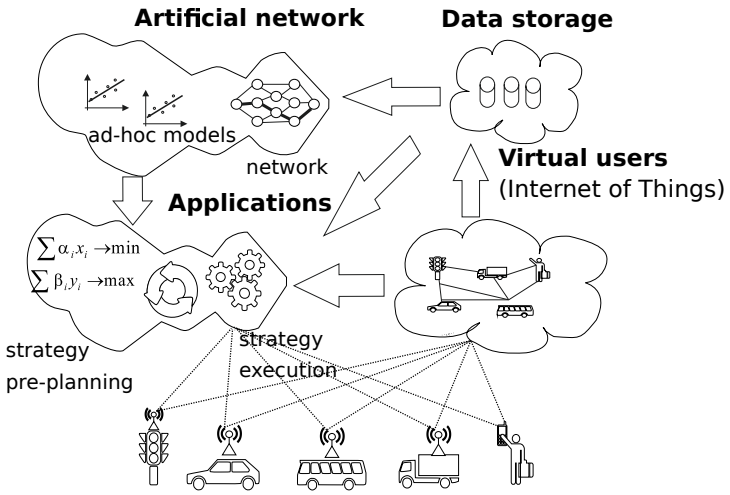



**Fig. 3.1** Reference architecture of data processing and decision-making states in intelligent transport system

Usually many clouds from different providers are available. Some of the problems can be similar to them, and cooperation between them is possible. This problem will be addressed in the next chapter by federated learning.

The users of the intelligent transport systems are permanently connected to the cloud. This allows creating a virtual representation of each user in terms of Internet-of-Things and having in the cloud dynamic sensor data, associated with them (pre-processed or raw). This creates a network of virtual users, which in fact is a mirror of reality in the cloud. This virtual reality contains distributed user data (partly stored in user devices, partly in the virtual storages provided by the cloud, but still associated with users).

On the first stage data should be pre-processed. Raw sensor data requires very much storage space and cannot be stored for a long time. This data can be processed locally or upload to the cloud and pre-processed there. The results of the pre-processing are stored in the user profile and can be uploaded to the cloud at this stage.

The next stage is to organize the virtual cloud information storages. This is made by cloud data mining agents, which collect the information, partially copying it to the storages in the cloud, partially making references to the user profiles, if they are available in the cloud. These agents put special attention to cost of the information, which includes its availability, reliability and precision. These virtual storages are subject of further big data processing and mining.

Cloud-based systems have a big number of users, and should fast react to their requests. For this purpose artificial ad-hoc networks are created, which are oriented to concrete problems, solved by the cloud system. For example, the networks oriented to shortest path calculation, traffic light regulation or passenger transit can be created. Two important problems are solved in the virtual network: estimation of its parameters and pre-calculation of user strategies.

Three sample scenarios are discussed regarding the proposed architecture and the most important stages of data processing, mining, and optimisation: 1) A cooperative intersection control, which optimizes vehicle flows in traffic networks by regulating the intersection controllers. 2) A personal travel companion, which provides dynamic planning and monitoring of multi-modal journey to travellers surface vehicle drivers, and transport operators. 3) A logistic services companion, which provides benefits to clients and stakeholders involved in, affected by, or dependent on the transportation of goods in urban environments.

In the Section 3.2 and 3.3 the corresponding decentralised clustering and regression methods are proposed.

## 3.2 Decentralised regression methods

In this section we consider a data processing step of a conceptual cloud-based architecture of traffic management system (Section 3.1). Distributed regression model with a fusion center for sensor networks was considered in [Gispan et al., 2017], but we focus on a decentralised scenario.

**Decentralised Regression Model for Intelligent Forecasting in Multi-agent Traffic Networks**, J. Fiosina, In Proc. of the Int. Conf. on Distributed Computing and Artificial Intelligence, AINCS, 151, 255-263, Springer, 2012.

In the following we propose a decentralised multivariate linear ML method, which uses resampling approach for its parameter synchronisation procedure.

We consider a traffic network with several vehicles, represented as autonomous agents, which predict their travel-time on the basis of their current observations and history. Each agent locally estimates the parameters of the same traffic network. In order to make a forecast, each agent constructs a regression model, which explains the manner in which different explanatory variables (factors) influence the travel-time. A detailed overview of such factors is provided in [Lin et al., 2005b]. The following information is important for predicting the travel-time [McKnight et al., 2004]: average speed before the current segment, number of stops, number of left turns, number of traffic lights, average travel-time estimated by traffic management centres. We should also take into account the possibility of an accident, network overload ("rush hour") and weather conditions.

We consider a vehicle, whose goal is to drive through the defined road segment under specific environment conditions (day, time, city district, weather, etc.). Let us suppose, that it has no or little experience of driving in such conditions. For accurate travel-time estimation, it contacts other traffic participants, which send their estimated parameters to it. The forecasting procedure of one such vehicle is shown in Fig. 3.2.

We describe the formal model, which is incorporated into each agent's local data processing module. We introduce a notation for the local regression model of each of the $s$ agents in the network. We use index $(i,t)$ for the variables in formula (1.1), to refer to agent $i$ at time $t$: $\mathbf{Y}(i,t) = \mathbf{X}(i,t)\boldsymbol{\beta} + \boldsymbol{\epsilon}(i,t)$, $i = 1,\dots,s$.

Following (1.2), agent $i$ calculates the estimates $\mathbf{b}(i,t)$ of $\boldsymbol{\beta}$ and predicts the travel-time $E[Y(i,t+1)]$ for the future time moment $t+1$, using (1.3).

Prior to forecasting, some agents may adjust their locally estimated parameters with other traffic participants. Let us describe this adjustment procedure more precisely.

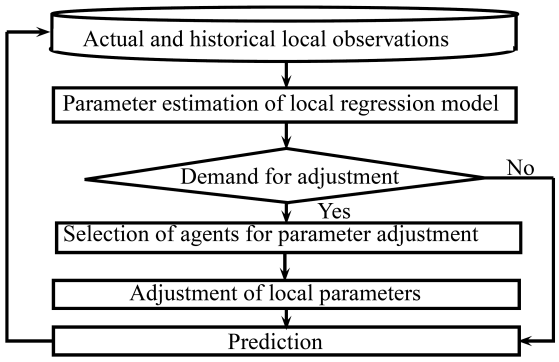



**Fig. 3.2** Algorithm for local travel-time prediction by an individual agent

First, the agent selects the other agents from a given transmission radius, contacts them, and requests them to send their estimated parameters. The agents can be in different situations and their observation may contain outliers. In order to make the adjustment procedure more reliable and robust to outliers, the agent performs the described selection several times in different combinations, forming so-called resamples from the available agents [Afanasyeva and Andronov, 2006].

We implement $N$ realisations of the following resampling procedure for agent $i$. At the realisation $q$, the agent receives the parameter estimates of $r$ randomly chosen neighbour agents. Let vector $\mathbf{L}_i^q$ contain the indices of the selected agents, $|\mathbf{L}_i^q| = r$.

The next step is the adjustment of the parameters. The agent that initialised the adjustment process considers the weighted estimates of other agents. The weights are time-varying and show the reliability level of each agent, depending on its forecasting experience as well as some other factors. Let $\mathbf{c}_i^{*q}(t)$ be a $1 \times r$ vector of the weights at the $q$-th realization at time $t$, $i = 1, \ldots, s$, which is a stochastic vector for all $t$.

Thus, the resampling estimator is an average over all realisations:

$$\mathbf{b}^R(i, t+1) = \frac{1}{N} \sum_{q=1}^{N} \sum_{j=1}^{r} c_{i,j}^{*q}(t) \mathbf{b}(\mathbf{L}_{i,j}^q, t).$$

The algorithm is a combination of the iterative least squares estimator algorithm and resampling-based parameter adjustment. This adjustment procedure aims to increase the reliability of the estimates, especially for insufficient or missing historical data, and to contribute to the overall estimation accuracy [Stankovic et al., 2009].

| Var. | Description | Mod. | Koef. | Est.value |
|---|---|---|---|---|
| $Y$ | travel-time (min) | $Y$ | - | - |
| $X_1$ | route length (km); | $X_1$ | $b_1$ | .614 |
| $X_2$ | avg. speed in system (km/h) | $X_2$ | $b_2$ | -.065 |
| $X_3$ | avg. number of stops (units/min) | $X_3^2$ | $b_3$ | .09 |
| $X_4$ | congestion level (veh/h) | $\sqrt{X_4}$ | $b_4$ | .159 |
| $X_5$ | traffic lights in the route (units); | $X_5^2$ | $b_5$ | .241 |
| $X_6$ | travel-time (units); | $X_6^2$ | $b_6$ | -.058 |

**Table 3.1** Factors and corresponding regression parameters

| Case | $R^2$, whole system | $R^2$ worth agent |
|---|---|---|
| Centralised | 0.66 | - |
| Local | 0.55 | 0.28 |
| Coordinated | 0.64 | 0.58 |

**Table 3.2** Efficiency criteria $R^2$

We simulate a traffic network of the southern part of Hanover (Germany). Vehicles solve a travel-time prediction problem. They receive information about the centrally estimated system variables for this city district from traffic management centre, combine it with their historical information, and make adjustments according to the information of other participants using the proposed consensus algorithm. The prediction influencing factors are listed in table 3.1. To improve the quality of the regression model, some non-linear transformations of the factors are performed. We simulated ten agents and trained them on the observations taken from the available dataset on size 1790. We compared the results for three cases using analysis of variance and adjusted coefficient of determination, $R^2$ (Table 3.2). The results show that agent coordination significantly improves the prediction results and tends to the accuracy of centralised approach.

**Cooperative kernel-based forecasting in decentralized multi-agent systems for urban traffic networks**, J. Fiosina, M. Fiosins, In Proc. of the Workshop on Ubiquitous Data Mining, ECAI 2012, CEUR Workshop Proc.,vol. 960, 3-7, 2012

In this contribution we propose a novel decentralised kernel-density based regression and introduce individual models' collaboration algorithm. We use the same problem statement as in previous study, but used kernel based regression 1.4 instead of linear.

This method allows agent cooperation for sharing their prediction experience. We take into account two main facts. The nodes should coordinate their prediction experience over some previous sampling period and adapt quickly to the changes in the streaming data, without waiting for the next coordination action. Let us first discuss the cooperation technique. We introduce the following definitions.

Let $\mathbf{L} = \{L^j \mid 1 \leq j \leq p\}$ be a group of $p$ agents. Each agent $L^j \in \mathbf{L}$ has a local dataset $D^j = \{(\mathbf{X}_c^j, Y_c^j) \mid c = 1\ldots., N^j\}$, where $\mathbf{X}_c^j$ is a $d$-dimensional vector. In order to underline the dependence of the prediction function (1.5) from the local dataset of agent $L^j$, we denote the prediction function by $m[D^j](\mathbf{x})$.

Consider a case when some agent $L^i$ is not able to forecast for some $d$-dimensional future data point $\mathbf{X}_{new}^i$ because it does not have sufficient data in the neighbourhood of $\mathbf{X}_{new}^i$. In this case, it sends a request to other traffic participants in its transmission radius by sending the data point $\mathbf{X}_{new}^i$ to them. Each agent $L^j$ that has received the request tries to predict $m[D^j](\mathbf{X}_{new}^i)$. If it is successful, it replies to agent $L^i$ by sending its best data representatives $\hat{D}^{(j,i)}$ from the neighbourhood of the requested point $\mathbf{X}_{new}^i$. Let us define $G^i \subset L$, a group of agents, which are able to reply to agent $L^i$ by sending the requested data.

To select the best data representatives, each agent $L^j$ makes a ranking among its dataset $D^j$. It can be seen from (1.5) that each $Y_c^j$ is taken with the weight $w_c^j$ with respect to $\mathbf{X}_{new}^i$, where $w_c^j = \frac{K\left(\frac{\mathbf{X}_{new}^i - \mathbf{X}_c^j}{h}\right)}{\sum_{l=1}^n K\left(\frac{\mathbf{X}_{new}^i - \mathbf{X}_l^j}{h}\right)}$. The observations with maximum weights $w_c^j$ are the best candidates for sharing the experience.

All the data $\hat{D}^{(j,i)}$, $L^j \in G^i$ received by agent $L^i$ should be verified, and duplicated data should be removed. We denote the new dataset of agent $L^i$ as $D_{new}^i = \bigcup_{L^j \in G^i} \hat{D}^{(j,i)}$. Then, the kernel function of agent $L^i$ is updated taking into account the additive nature of this function:

$$m[D_{new}^i](\mathbf{x}) = \sum_{L^j \in G^i} m[\hat{D}^{(j,i)}](\mathbf{x}) + m[D^i](\mathbf{x}).$$

Finally, agent $L^i$ can autonomously make its forecast as $m[D_{new}^i](\mathbf{X}_{new}^i)$ for $\mathbf{X}_{new}^i$.

We simulate the same traffic network from previous subsection. We predict travel-time, based on the same factors. We simulate 20 agents having their initial experience represented by a dataset of size 20 till each agent made 100 predictions, thus making their common experience equal to 2400. We assume the maximal number of transmitted observations from a single agent equals 2.

During the simulation, to predict more accurately, the agents use the presented cooperative learning algorithm that supports the communication between agents with the objective of improving the prediction quality. The necessary number of communications depends on the value of the smoothing parameter $h$. The average number of necessary communications is given in Figure 3.3 (left). We can see the manner in which the number of communications decreases with the learning time. We vary $h$ and obtained the relation between the communication numbers and $h$ as a curve. The prediction ability of one of the agents is presented at Figure 3.3 (right). Here, we can also see the relative prediction error, which decreases with time. The predictions that used communication between agents are denoted by solid triangles, and the number of such predictions also decreases with the time. This proofs the efficiency of our proposed learning procedure.

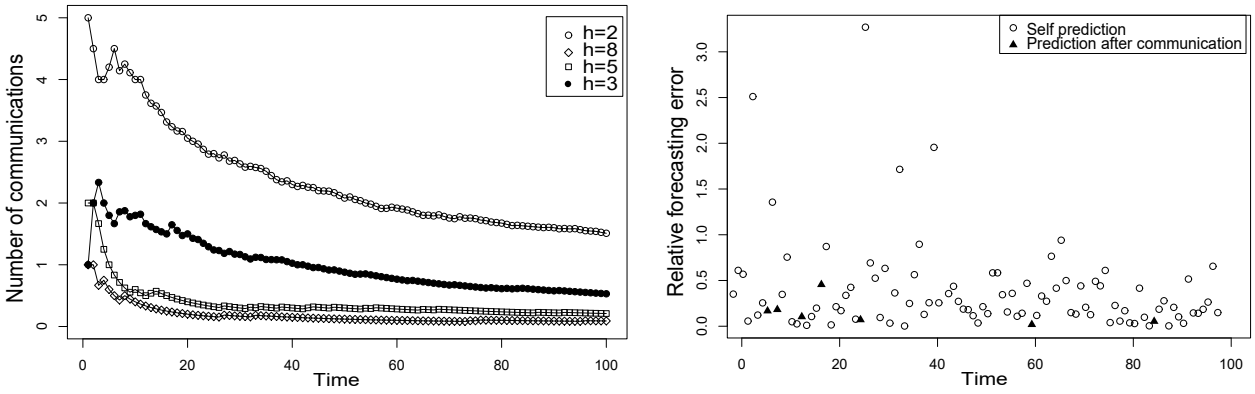



**Fig. 3.3** Average number of communications over time for different $h$ (left). Relative prediction error and communication frequency of a single agent over time (right).

| Characteristic of System | $h$=2 | $h$=3 | $h$=5 | $h$=8 |
|---|---|---|---|---|
| System average $R^2$ | 0.86 | 0.84 | 0.83 | 0.82 |
| Average % of not predicted observations | 3 | 1 | 0.5 | 0.1 |

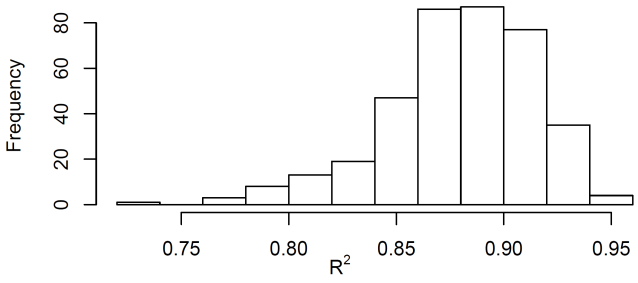


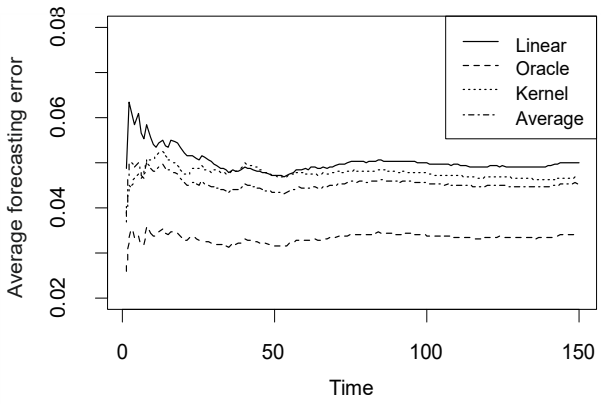

**Fig. 3.4** $R^2$ goodness-of-fit measure using cross-validation for the whole system for different $h$ with its average (left) and distribution (right).

The goodness-of-fit of the system has been estimated using a cross-validation technique. We assume that each agent has its own training set, but it uses all observations of other agents as a test set, so we use 20-fold cross-validation. To estimate the goodness of fit, we use analysis of variance and generalized coefficient of determination $R^2$ that provides an accurate measure of the effectiveness of the prediction of future outcomes by using the non-parametric model [Racine, 1997]. The calculated $R^2$ values and the corresponding number of the observations that could nor be properly predicted by the individual agents depending on $h$ are listed in Figure 3.4 (left). We take into account that the cooperation on the testing step is not assumed. In Figure 3.4 (right) we can also see how $R^2$ is distributed among the system agents. The results suggest that we should find some trade-off between system accuracy (presented by $R^2$) and the number of necessary communications (presented by the percentage of not predicted observations), which depend on $h$. The point of trade-off should depend on the communication and accuracy costs.

A linear regression model from the previous section applied to the same data gives lower average goodness of fit $R^2$=0.77, however, predictions can be calculated for all data points.

**Big data processing and mining for next generation intelligent transportation systems**, J. Fiosina, M. Fiosins, J.P. Müller, Jurnal Teknologi 63 (3), 2013

This contribution summarizes the decentralised cooperative approach for distributed traffic networks. The proposed linear and kernel-based cooperative algorithms are compared with the same experimental settings for different data processing architectures: centralised, uncoordinated, coordinated. The experimental results show that an aggregated "oracle" approach would outperform the both models, if the best model for each observation forecast is selected properly.




| Model | Average forecasting errors | $R^2$ (After cross-validation) |
|---|---|---|
| **Linear** | | |
| Centralised | 0.051 | 0.829 |
| Uncoordinated | 0.054 | 0.811 |
| Coordinated | 0.050 | 0.819 |
| **KD** | | |
| Centralised | 0.045 | 0.913 |
| Uncoordinated | 0.053 | 0.814 |
| Coordinated | 0.047 | 0.859 |
| **'Average'** | | |
| Centralised | 0.046 | --- |
| Uncoordinated | 0.049 | --- |
| Coordinated | 0.046 | --- |
| **'Oracle'** | | |
| Centralised | 0.034 | --- |
| Uncoordinated | 0.037 | --- |
| Coordinated | 0.034 | --- |

**Fig. 3.5** Average forecasting errors using the kernel (KD), linear and combined approaches (left) and Average forecasting errors and goodness-of-fit criteria for the different forecasting models (right).

## 3.3 Explainable multi-agent systems

Explainability is another important aspect for decision-making in multi-agent environment.

**AI for explaining decisions in multi-agent environments**. S. Kraus, A. Azaria, J. Fiosina, M. Greve, N. Hazon, L. M. Kolbe, T. Lembcke, J. P. Müller, S. Schleibaum, and M. Vollrath. In the 34th AAAI Conference on AI (AAAI 2020), New York, USA, February 7-12, 2020, 34(09), pages 13534–13538, AAAI Press, 2020

We propose a novel research direction: explainable decisions in multi-agent environment. This direction formalises the process of system's explanation generation. One of the considered aspects is an explanation provided by combination of various ML methods.

Explanation is necessary for human to understand and accept decisions made by an AI system especially in multi-agent systems when the systems'goals depend on other agents' preferences. In such situations, explanations should aim to increase user satisfaction, taking into account the system's decision, the user's and the other agents' preferences, the environment settings and properties such as fairness, envy and privacy. Generating explanations that will increase user satisfaction

is very challenging; to this end, we proposed a new research direction: Explainable decisions in Multi-agent Environments.

**Explanation generation** The development of AI-based tools that provide the right explanations to the right users at the right time to increase user satisfaction in multi-agent systems is very challenging. We investigate efficient algorithms for generation of explanations, preferably in real time. We propose a two stage procedure: first, a set of possible explanations will be created and then the one that best suits the specific user at the specific settings will be selected. Both stages can be done using ML or any other decision-making procedures based on real user input. If the AI decision is made using neural networks (e.g., [Rosemarin et al., 2019, Li et al., 2019b]) then explainable AI methods can be used to identify important features that led to the decision [Shrikumar et al., 2017b, Bach et al., 2015]. These methods should be adapted to the problems related to Explainable decisions in Multi-agent Environments [Lee, 2019, Selvaraju et al., 2017].

**User Modeling** User satisfaction from an explanation of a given decision strongly depends on the actual decision, the other agents, the environment and the user's beliefs. Thus collecting data on the influence of an explanation on the user's satisfaction must be done in the context of the specific decision it explains and the environment setting. Data collection can be done either using fictitious decisions, their explanations and the multi-agent system environment setting or, much harder to accomplish, in actual settings or at least in simulations. The users can express their preferences on how much they like the explanations. We can use this data to build a generalized model of users' preferences toward explanations. However, this model will not provide us with the explanations that increase the user's satisfaction. Here we will need to let the user express his or her level of satisfaction from a given decision with different variants of explanations and without explanations, and try to build a model that measures the users' satisfaction from the decision.

**Interactive Explanations** When the AI system interacts repeatedly with the same users, the learning phase of the preferences and satisfaction models can be personalized, but more importantly the explanation generated should take long-term satisfaction into consideration. Furthermore, we propose to consider, when interacting with the user, using reinforcement learning to improve the user's model of overtime in a guided way. Recently, there have been a few attempts to consider models for interactive explanations to explainable AI [Madumal et al., 2018] and to value-based agents [Liao et al., 2021], but no system was developed. Interactive explanations can be viewed as argumentation dialogues. It was shown to be beneficial to model the interaction as a partially observable Markov decision process where the uncertainty is about the user's beliefs [Rosenfeld and Kraus, 2016]. Using this approach for explainable decisions in multi-agent environments there is a need to continuously estimate the user's beliefs and sentiment toward the AI system's decision, and to predict how a given explanation statement will modify the user's beliefs and influence its attitude toward the decision.

The system explanation requirement are often dependent on the user preferences. So the individual preferences could be taken in account and then individually explained to the users depending on its importance.

# Chapter 4

## Distributed centralised data analysis

Most current ML methods need to analyse large datasets. As the demand for processing training data has outpaced the increase in the computational power of computing machinery, it is necessary to distribute the ML workload across multiple machines and to transform the centralisation into a distributed system. These distributed systems exhibit new challenges, mainly in terms of the efficient parallelisation of the training process and the creation of a coherent model [Verbraeken et al., 2020]. Cloud computing technologies can be successfully applied to parallelise standard data processing techniques for more feasible working with massive amounts of data. For this purpose, standard algorithms often need to be redesigned for parallel environments to distribute the computations among multiple nodes. One such approach is the use of the MapReduce paradigm. Another means of reducing the computational load of a central data processing server is to use a federated learning approach, which distributes the computations among multiple data owners. However, the main goal of federated learning is to address the data privacy challenge by not sharing the raw data of each participant, and only sharing their model parameters.

### 4.1 Distributed regression for big data forecasting

**Distributed Non-parametric and Semi-parametric Regression on SPARK for Big Data Forecasting**. J. Fiosina, M. Fiosins, Applied Computational Intelligence and Soft Computing, 2017.

In this study, we present distributed parallel versions of some nonparametric and semi-parametric regression models. The forecasting accuracy of the proposed algorithms is compared with the linear, which is the only forecasting model currently having parallel distributed realization within the SPARK framework to address big data problems.

Recently, a new and efficient framework called Apache SPARK [1] was introduces, which allows efficient execution of distributed jobs and therefore is very promising for big data analysis problems. There exist also alternative parallelisation approach as Message Passing Interface [Michailidis and Margaritis, 2013], however, we concentrate on SPARK because of its speed, simplicity, and scalability [Fernández et al., 2014]. We use MapReduce paradigm and describe the algorithms in terms of SPARK data structures to parallelize the calculations.

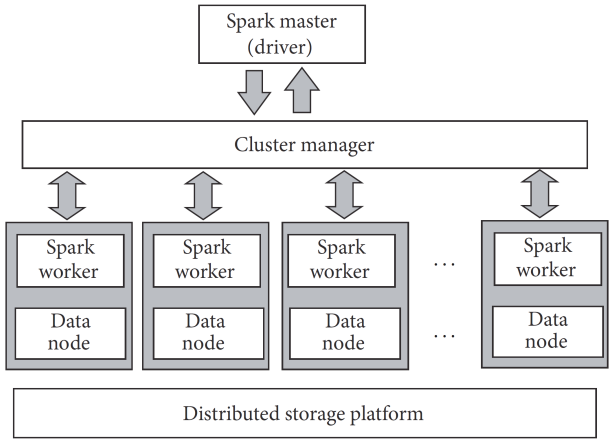
We consider regression-based forecasting for data with nonlinear structure, which is common in real-world datasets [Li et al., 2019a]. Nonparametric and semiparametric regression methods [Henderson and Parmeter, 2015], [Härdle et al., 2004] do not require linearity and are more robust to outliers. The downside of those methods however, is that they are very time-consuming, and

[1] https://spark.apache.org/

therefore the term "big data" for such methods starts much earlier than with parametric approaches. Contemporary nonparallel realizations are not capable of processing all the available data. This requires parallel computation of those methods. There are some approaches to parallelise such models using R add.on packages, MPI [Helwig, 2014], we, however, address Apache SPARK MLlib, which is a promising tool for the efficient realisation of different ML and data mining algorithms [Meng et al., 2016].

The contribution of this study is (i) to design novel distributed parallel kernel density regression and partial linear regression algorithm over the SPARK MapReduce paradigm (Fig. 4.1 (left)) and (ii) to validate that algorithms, analyzing their accuracy, scalability, and speed-up by means of numerical experiments.

An algorithm for the estimation of partial linear model was proposed in [Härdle, 2002], which is based on the likelihood estimator and known as generalised Speckmann [Speckman, 1988] estimator. We reformulated this algorithm (Fig. 4.1 (right)) in terms of functions and data structures, which can be easily parallelizable.




(1) **function** SMOOTHMATRIX($\mathbf{T}, \mathbf{T}', \mathbf{h}$)
(2) $n \leftarrow rows(\mathbf{T}), n' \leftarrow rows(\mathbf{T}')$
(3) **for all** $i \in \overline{1,n}$, $j \in \overline{1,n'}$ **do**
(4) $\mathbf{W} : W_{i,j} \leftarrow K_H(\mathbf{T}_i - \mathbf{T}'_j), \mathbf{S} : S_{i,j} \leftarrow W_{i,j} / \sum_{i=1}^{n} W_{i,j}$
(5) **end for**
(6) **return** $\mathbf{S}$
(7) **end function**
(8)
(9) **function** LINEARCOEFFICIENTS($\mathbf{S}, [\mathbf{Y}, \mathbf{U}, \mathbf{T}]$)
(10) $\widetilde{\mathbf{U}} \leftarrow (\mathbf{I} - \mathbf{S})\mathbf{U}, \widetilde{\mathbf{Y}} \leftarrow (\mathbf{I} - \mathbf{S})\mathbf{Y}$
(11) **return** $\boldsymbol{\beta} \leftarrow (\widetilde{\mathbf{U}}^T \widetilde{\mathbf{U}})^{-1} \widetilde{\mathbf{U}}^T \widetilde{\mathbf{Y}}$
(12) **end function**
(13)
(14) **function** KERNELPART($\boldsymbol{\beta}, [\mathbf{Y}, \mathbf{U}, \mathbf{T}], \mathbf{h}, \mathbf{T}'$)
(15) $\mathbf{S} \leftarrow SmoothMatrix(\mathbf{T}, \mathbf{T}', \mathbf{h})$
(16) **return** $m(\mathbf{T}') \leftarrow \mathbf{S}(\mathbf{Y} - \mathbf{U}\boldsymbol{\beta})$
(17) **end function**
(18)
(19) **function** PLM($[\mathbf{Y}, \mathbf{U}, \mathbf{T}], \mathbf{h}, [\mathbf{U}', \mathbf{T}']$)
(20) Smooth matrix $\mathbf{S} \leftarrow SmoothMatrix(\mathbf{T}, \mathbf{T}, \mathbf{h})$
(21) $\boldsymbol{\beta} \leftarrow LinearCoefficients(\mathbf{S}, [\mathbf{Y}, \mathbf{U}, \mathbf{T}])$
(22) $m(\mathbf{T}') \leftarrow KernelPart(\boldsymbol{\beta}, [\mathbf{Y}, \mathbf{U}, \mathbf{T}], \mathbf{h}, \mathbf{T}')$
(23) **return** $\mathbf{Y}' \leftarrow \mathbf{U}'\boldsymbol{\beta} + m(\mathbf{T}')$
(24) **end function**

**Fig. 4.1** SPARK distributed architecture (left). Partial linear model estimation algorithm, training set: $[\mathbf{Y}, \mathbf{U}, \mathbf{T}]$, test set $[\mathbf{Y}', \mathbf{U}', \mathbf{T}']$ (right).

Taking Hadoop and SPARK distributed architectures and the algorithm from Figure 4.1 into account we developed our distributed partial linear regression algorithm, which assumes parallel executions on several processing nodes for training and forecasting (Figure 4.2).

Kernel-density parallelization can be consider as a special case of partial linear model with missing linear part. To evaluate the performance of the propose solution, we used several datasets, which characteristics are summarized in Table 4.1.

We compare the goodness-of-fit metric ($R^2$) varying the size of training set (Figure 4.3). For all the datasets, (non)semi-parametric models show more accurate results. Kernel regression experiences problem with increasing the dimensionality, because it is difficult to find the points in the neighborhood of the specific point in big dimensions. It could be a reason, why semi-parametric

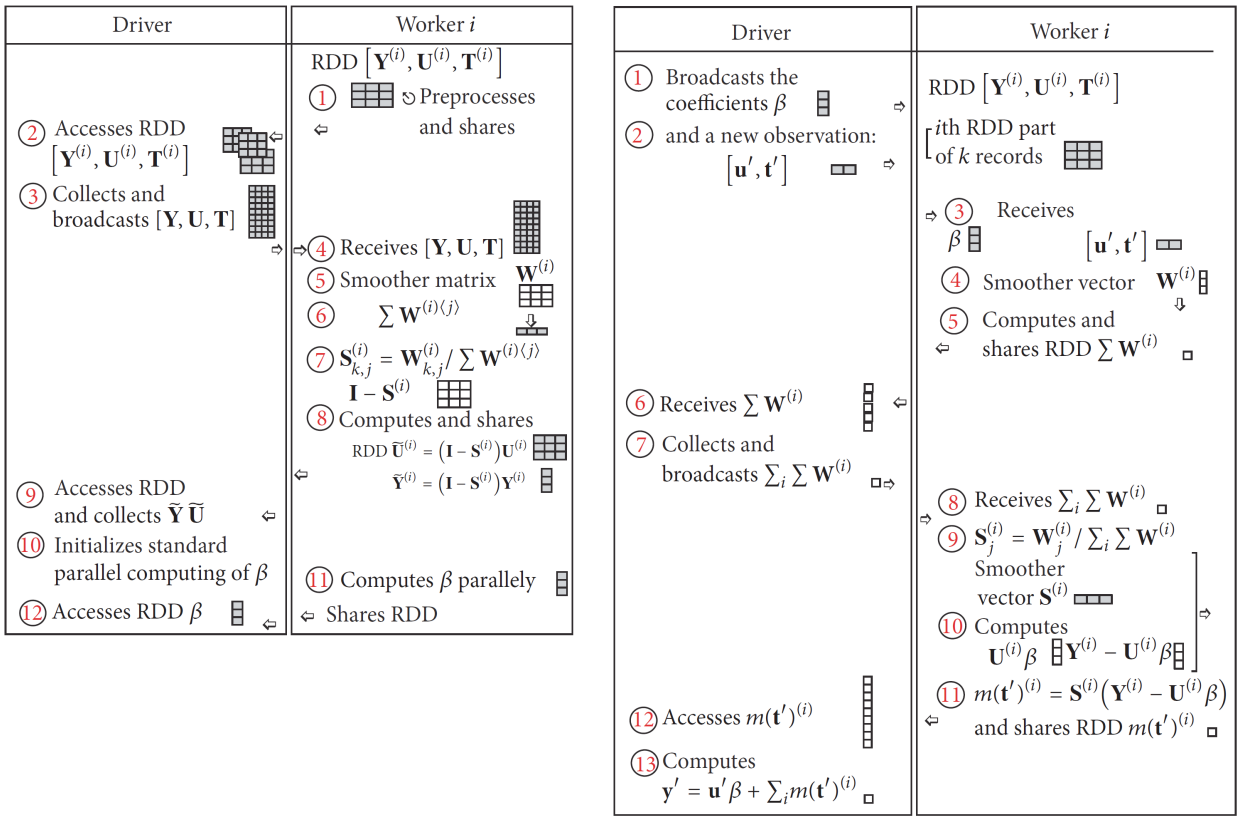



**Fig. 4.2** Partial linear model training (left) and forecasting (right), where RDDs are SPARK distributed memory abstractions.

| Dataset | Number of records | Number of factors |
|---|---|---|
| Synthetic data: $y = 0.5x^1 + x^2 sin(x^2) + \epsilon$ | 10,000 | 2 |
| Hanover traffic data | 6,500 | 7 |
| Airlines delays data[2] | 120,000,000 (we used 13,000) | 29+22(10) |

**Table 4.1** Characteristics of the datasets.

models showed more accurate results. We compare the speed (execution time) in the scalability

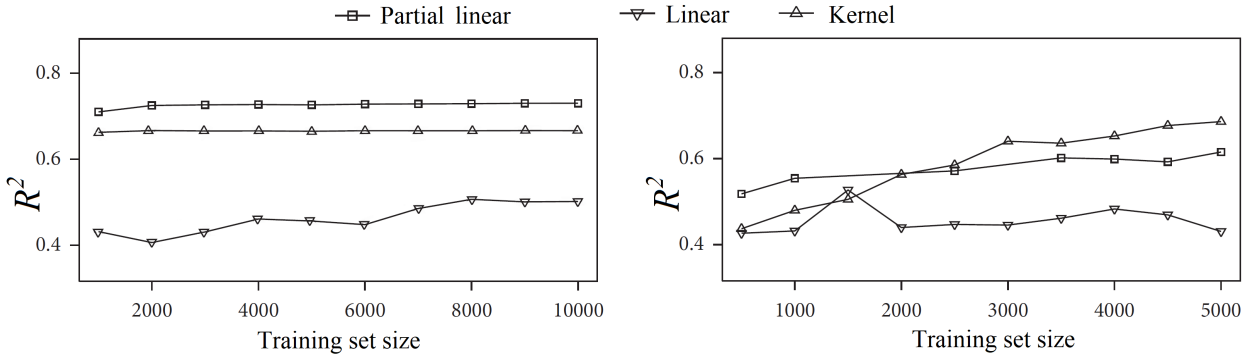



**Fig. 4.3** Forecasting quality of regression models for airline data (left) and traffic data (right).

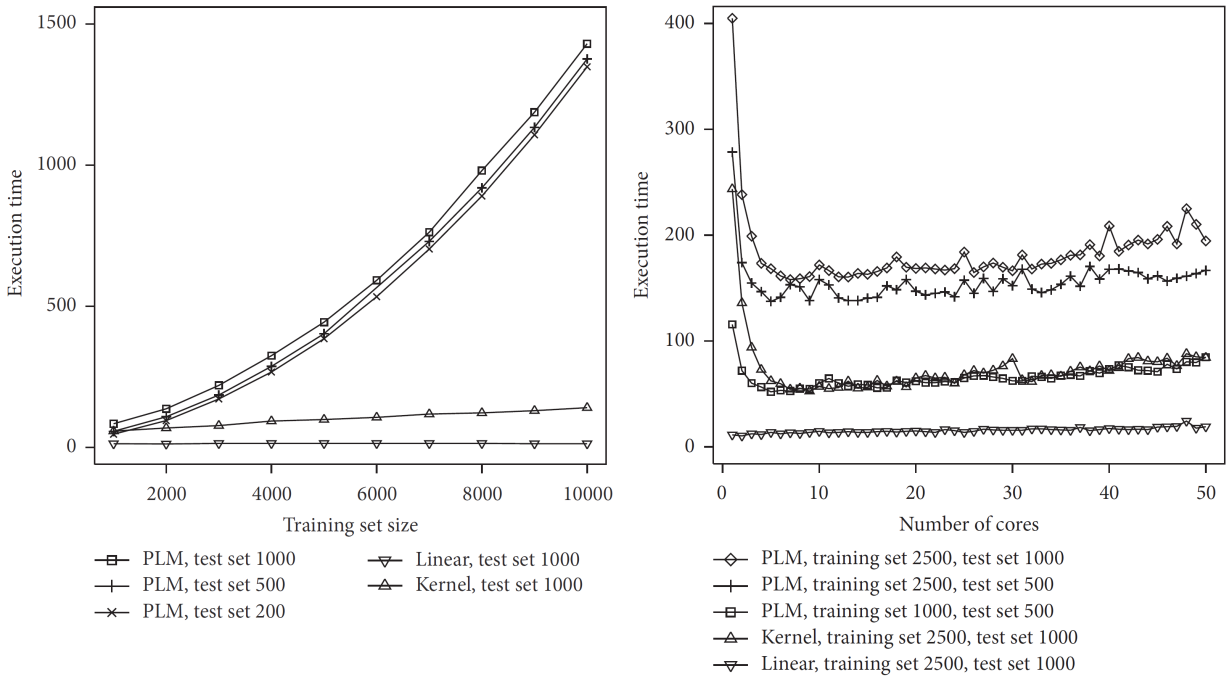



**Fig. 4.4** Execution time dependence on training set size for airlines data (left); PLM algorithm execution time depending on the number of processing cores for traffic data (right).

experiments varying the size of training and test set (Figure 4.4 (left)). All the experiments show that the training set influenced the execution time non-lineary, but the test set influence the time linearly. Finally, we examine how the execution time changes the number of available cores (Figure 4.4 (right)). We demonstrate the feasibility of processing datasets of varying sizes that are otherwise not feasible to process with a single machine. An interesting aspect is that for each combination (dataset, algorithm) we could find the optimal amount of resources (number of cores) to minimize the algorithms execution time.

## 4.2 Federated learning in distributed transportation networks

The accuracy and interpretability are two dominant features present in successful predictive models. However, more accurate black-box models are not sufficiently explainable and transparent. This feature complicates the user acceptance of AI-driven systems and can be troublesome even for AI model developers.

With increased transportation availability increases the amount of traffic and this leads to a large number of available decentralised data sources. Thus, the AI technologies implemented there should be capable of processing these data in a decentralised manner, according to the data privacy regulations. We address this challenge with federated learning approach.

The main assumption is that the federated model should be parametric (e. g., mainly deep learning based) because the algorithm synchronises the models by synchronising the parameters.

A known limitation of deep learning is that neural networks inside it are unexplainable black-box models.

**Explainable federated learning for taxi travel time prediction**. J. Fiosina, In Proc. of the 7th Int. Conf. on Vehicle Technology and Intelligent Transport Systems - VEHITS 2021, pages 670-677, INSTICC, SciTEPress.

In this study we propose [Fiosina, 2021] a privacy-preserving explainable federated model, which achieves a comparable accuracy to that of the centralised approach on the considered real-world dataset. We predict the Brunswick taxi travel-time based on floating car data trajectories obtained from different taxi service providers, which should remain private. The explainable federated learning model makes predictions for the stated problem and allows a joint learning process over different users, processing the data stored in each of them without exchanging their raw data, but only parameters, as well as providing joint explanations about variable importance.

Therefore, we aim to address several research questions. 1) Which is the most accurate ML prediction method for the given data in a centralised manner? We identify the best hyper-parameters for each method. 2) Under which conditions federated learning is effective? We distribute the dataset among more providers, and analyse after which point the distributed and non-synchronised models lose their accuracy and federated learning is beneficial. We define an optimal synchronisation plan for parameter exchange, identifying the hyper-parameters and frequency of parameter exchange that is acceptable and beneficial. 3) Do existing black-box explanation methods successfully explain federated learning models? We investigate how the state-of-the-art explainability methods can explain federated models.

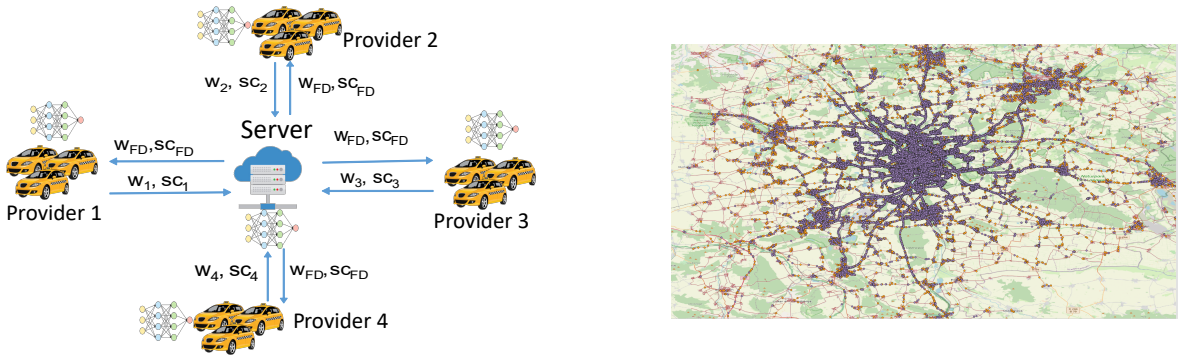



**Fig. 4.5** Explainable federated learning architecture (left) and Road network of Brunswick, (latitude: 51.87 - 52.62, longitude: 10.07 - 11.05 (right).

We propose a strategy for explainability of the federated model and illustrate it on a travel-time prediction problem. Our aim is to describe the application of state-of-the-art explainability methods to federated learning, while maintaining data privacy. We apply the federated architecture and explainability methods to the focal problem and consider what information and how often should be exchanged. Moreover, the application of each explainability method to a concrete task only produces baseline results because the result interpretation is specific to the particular task or application at hand.

**Algorithm 2:** Explainable federated learning training process

**Result:** Trained $M_{FD}$ model
Define the same initial $w_i$ for $M_i$, $i$=1...$N$, $epoch$=1 ;
**while** *The loss function does not converge* **do**
  **foreach** *batch of data* **do**
    **foreach** $F_i$ *in parallel* **do**
      $Train(M_i^{<epoch,batch>}, D_i^{TR,batch})$;
      **if** *synchronisation* **then**
        $F_i$ sends $w_i^{<epoch,batch>}$ to the server;
    **if** *synchronisation* **then**
      Server aggregates the parameters/gradients and broadcasts them:
      $w_{FD}^{<epoch,batch>} = FedAggregation(w_i^{<epoch,batch>}, i = 1 \dots N)$;
      **foreach** $F_i$ *in parallel* **do**
        $F_i$ receives updated parameters from the server and updates its model:
        $M_i^{<epoch,batch>} = M_{FD}^{<epoch,batch>}$;
  $epoch = epoch + 1$
Training process is over. The last obtained $M_{FD}^{<epoch,batch>}$ is the final model: $M_i^{<localFD>}$ of each $F_i$;
**foreach** *participant* $F_i$ *in parallel* **do**
  **foreach** *instance* $j$ *of* $D_i^{TE}$ *dataset* **do**
    Calculates attribution scores: $sc_{i,j} = ScoringAlgorithm(M_i^{<localFD>}, D_{i,j}^{TE})$;
  $F_i$ calculates its average scores and sends the result to the server: $sc_{i*} = \frac{\sum_j sc_{i,j}}{|D_i^{TE}|}$;
Server aggregates the participant scores and broadcasts the result: $sc_{FD} = \frac{\sum_i sc_{i*}|D_i^{TE}|}{|\cup_i D_i^{TE}|}$;
**foreach** *participant* $F_i$ *in parallel* **do**
  Each $F_i$ updates its attribution scores: $sc_{i*} = sc_{FD}$;

Let us give a formal definition of the federated learning concept. Let $N$ federated learning participants $\{F_i\}_{i=1}^N$ own datasets $\{D_i\}_{i=1}^N$ as previously defined. Each participant $F_i$ divides its dataset $D_i = D_i^{TR} \cup D_i^{TE}$ into training set $D_i^{TR}$ and test set $D_i^{TE}$. The individual local models are trained on $D_i^{TR}$ and their explainability attribution scores are computed on $D_i^{TE}$. $\{M_i\}_{i=1}^N$ is the local models of the participants, while $M_{FD}$ is the federated model. As we consider learning on batches, $M_i^{<epoch,batch>}$ is the local model of the participant $F_i$ for the current $epoch$ and $batch$ of data, while $M_{FD}^{<epoch,batch>}$ is the current federated model. $w_i^{epoch,batch}$ are the current parameters of the model $M_i^{<epoch,batch>}$: $w_i^{epoch,batch} = w(M_i^{<epoch,batch>})$. The training process of such a system contains the steps presented in Algorithm 2. The scoring algorithm can be one of the explainability methods e.g., DeepLIFT, Integrated gradients (Section 1.2.1). Note that we start the federated variable explanation process when the federated training process is finished and a copy of the common federated model $M_i^{<localFD>}$ of each $F_i$ is locally available.

**Experiments**: We predict the Brunswick taxi travel-time based on floating car data (FCD) trajectories obtained from two different taxi service providers (January 2014 - January 2015) (Figure 4.5 (right)) and the corresponding weather data. First, we transform the raw FCD data trajectories and de-noised them obtaining 542066 trajectories. Additionally, we divide the map

into different size grids (e.g., 200m*200m) to determine whether this aggregation can improve our forecasts.

We predict the travel-time using different methods (Table 4.2) and find the corresponding best hyper-parameters by the grid search.

| **Model** | **Hyper-parameters** |
|---|---|
| Regression | Linear (no); Ridge ($\alpha = 0.09$); Lasso ($\alpha = 1e-9$) |
| XGBoost | $colsample\_bytree = 0.7$; $learning\_rate = 0.12$; $max\_depth = 9$, $\alpha$=15; $n\_estimators = 570$ |
| Random forest | $num\_trees = 100$; $max\_depth$ and $min\_samples\_leaf$ are not restricted |
| Deep learning | fully conn. percentron with 2 hidden layers, number of neurons: 64-100, Re-Lu act. function; 0.2 dropout between hidden layers; optimiser SGD; MSE loss function; $NN\_batchsize = 128$, $epochs = 800$; $learning\_rate = 0.02$ |
| Federated learning | synchronisation each 2nd batch, $NN\_batch\_size$ is proportional to the size of each provider's dataset, the sum of all provider's $NN\_batch\_size = 128$. |

**Table 4.2** Optimal model hyper-parameters

We divide the dataset into training (80%) and test (20%) sets. We use the mean squared error (MSE) as an efficiency criterion and 5-fold cross validation for model comparison. The accuracy with an MSE of .0010 corresponds to 5 min, while that with an MSE of .0018 to 7.5 min.

We identify the best ML prediction model (Table 4.3). For one data provider (centralised approach, second column), the best results is obtained by the XGBoost and random forest methods (.00097 and .0010). Conventional regression methods such as linear, Lasso and Ridge regressions provide the same inaccurate results. Deep learning exhibit a slightly lower performance than those of the best models.

| Model | Number of data providers | | | | | |
|---|---|---|---|---|---|---|
| | 1 | 2 | 4 | 8 | 16 | 32 |
| Linear, Ridge, Lasso regression | .0019 | .0019 | .0019 | .0019 | .0020 | .0020 |
| XGBoost | .00097 | .0011 | .0011 | .0012 | .0012 | .0013 |
| Random forest | .0010 | .0011 | .0011 | .0012 | .0012 | .0013 |
| Deep learning | .0011 | .0012 | .0012 | .0013 | .0014 | .0015 |
| Federated Learning | — | .0011 | .0011 | **.0011** | **.0011** | **.0011** |

| For 8 providers synch. each $n$-th batch | Average MSE |
|---|---|
| 1 | **.0011** |
| 2 | **.0011** |
| 3 | .0012 |
| 4 | .0012 |
| 5 | .0013 |

**Table 4.3** MSE of travel-time prediction with different ML methods (left) and for different synchronisation frequencies (right)

Despite the fact that XGBoost and random forest methods provide the most accurate results for the centralised approach, federated learning could be implemented only on the deep learning bases. We equally distribute the dataset among several providers, and execute the models locally

on each provider without synchronisation. Then we analyse after which point the distributed nonsynchronised models lose their accuracy and federated learning becomes beneficial.

We investigate the effect of the synchronisation frequency on the accuracy of the federated model. Thus, the accuracy decreases with the step-wise decrease in the synchronisation frequency (Table 4.3) (right). With synchronisation performed in each batch or even each second batch, the accuracy remains the same as that of the centralised approach. With a rarer synchronisation, the accuracy decreases.

Next, we compare the results of variable importances for local, federated and centralised approaches. We select the Integrated gradients method as an explainability scoring algorithm because of its simplicity and speed. The baseline in this algorithm is taken equal to the average value of each feature. Figure 4.6 (left) contains variable importance calculated with the federated model for each of eight data providers locally using their test data. Despite the fact that the main tendency in variable importance by all of eight providers remains the same, the locally obtained results differ from the importance scores, calculated with all test data. This may lead to inaccurate explainability by some local providers, especially with small testset sizes. The proposed attribution scores averaging mechanism allows to avoid this inaccuracy without transmitting the local testsets.

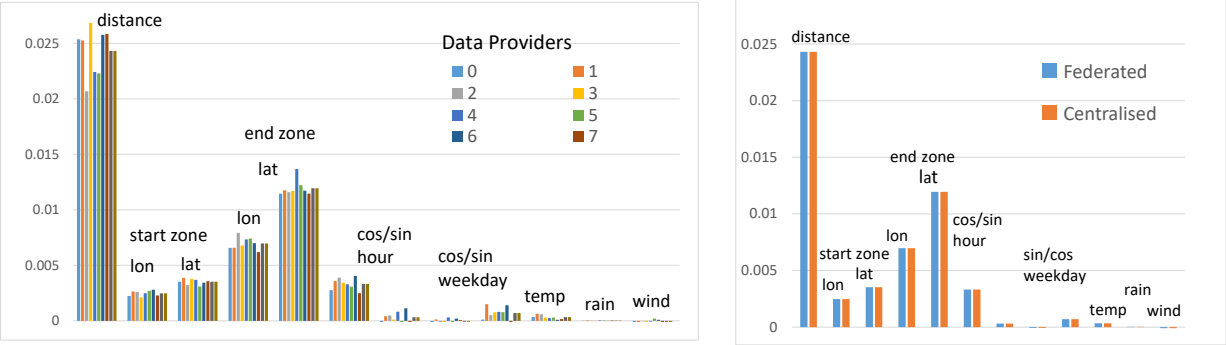



**Fig. 4.6** Explainability of individual models (left) and FL vs centralised approach (right) using Integrated gradients with baseline equal to average value of each feature

Figure 4.6 (right) presents variable importance calculated for the centralised and federated learning approaches using aggregated test datasets (centralised) or aggregation of scores (federated), which leads to the same results. We observe that without raw data transfer our approach allows more accurate calculation of variable importance than one each provider can obtain using only its local test set.

We investigate which parameters have the biggest influence on the results. According to Figures 4.6 (left) and 4.6 (right) the most important variable for all the models is FCD distance, which is expected. The next important variables are zones' coordinates, sine and cosine of the traveling hour and day of the week. This is clear that the distance could not completely determine the travel-time. However, we can conclude that despite of our expectations, almost all weather parameters do not significantly influence the predictions.

# Chapter 5

## Outlook and future work directions

Despite the significant progress and constant development of contemporary state-of-the-art ML methods, numerous challenges remain, such as the widely distributed data sources, limited computing power of single servers, data privacy requirements, model explainability, and applicability in various domains. Therefore, novel, computationally intensive, distributed ML methods and tools are required. This thesis is based on the contribution of 11 papers to the theoretical and practical aspects of state-of-the-art and proposed ML methods. In particular, the work in this thesis addresses three main research directions: computationally intensive ML methods, decentralised ML methods, and distributed centralised ML methods, in which the stated challenges are addressed.

However, substantial future research is required to advance the proposed method in terms of creating novel techniques and applying state-of-the-art methods in different problem settings, scenarios, and domains. We determined numerous possible future investigations for the formulated challenges.

Our contribution and future work directions for these challenges are summarised below:

- **Challenge 1 (Novel effective ML methods)**:
  - *Contribution*: We propose novel resampling-based methods for change-point estimation [Fiosina and Fiosins, 2011], [Fiosins et al., 2012], and stochastic graph route comparisons [Fiosina and Fiosins, 2014]. Several resampling-based algorithms are created, and formulas for estimating their properties are derived. The experimental results demonstrate that the proposed approaches outperform their classical alternatives, in which the distributions are estimated directly from the data.
  - *Future research directions*:
    - The combination of bootstrapping and resampling techniques with other data analysis procedures (e.g. change-point trend analysis and autoregression), the derivation of their theoretical properties, and validation of the proposed approaches on real-world data.
    - Improvements to prediction and classification techniques by constructing hybrid models and investigating their properties (e.g. improving the classification accuracy by stacking a combination of various models for data augmentation and improving travel-time forecasting models by ensembling various models).
- **Challenge 2 (Distributed computations)**:
  - *Contribution*: We propose a distributed parallelisation algorithm for the semi-parametric and non-parametric regression types, and implement these in the Apache SPARK computation environment and data structures [Fiosina and Fiosins, 2017]. Scalability, speed-up, and goodness-of-fit experiments using the proposed methods demonstrate the excellent performance of the proposed approach.
  - *Future research directions*:

  - The creation of novel distributed versions of state-of-the-art ML methods based on resampling, 'divide and conquer', and MapReduce to artificially divide the data and parallelise computations for cloud computing (e.g. on SPARK, GPU, and MPI).

- **Challenge 3 (Decentralised networked architectures)**:
  - *Contribution*: We propose a decentralised linear regression method [Fiosina, 2012] and introduce a resampling-based technique for the synchronisation of individual models. We propose non-parametric collaborative kernel-based regression algorithm [Fiosina and Fiosins, 2012], which effectively solves the travel-time prediction problem. The proposed synchronisation enables us to obtain more accurate predictions/clusters than when using the individual models of each agent [Fiosina et al., 2013a].
  - *Future research directions*:
    - The development of decentralised versions of other state-of-the-art ML methods (e.g., logistic regression, multiple regression, change-point analysis, time series, classification, and clustering algorithms).
    - Improvement of existing decentralised ML methods by introducing more efficient data or parameter exchange and synchronisation algorithms (e.g., involving new reputation or agent reliability level computing schemes and novel data exchange rules; a combined approach, which enables selection between parametric and non-parametric estimators; the use of the median resampling approach, which is more resistant to outliers).

- **Challenge 4 (Data privacy)**:
  - *Contribution*: We apply federated deep learning for travel-time forecasting [Fiosina, 2021]. The proposed collaborative model enables us to obtain more accurate prediction models than the individual non-cooperative models without transmitting raw data. Moreover, the proposed federated model yields comparable results to a centralised model that is fitted on all the data.
  - *Future research directions*:
    - The investigation of different federated learning architectures, and the development of novel federated coordinated ML approaches based on state-of-the-art methods and their explanations (e.g. other deep learning architectures, other parametric models, and decision tree-based methods).

- **Challenge 5 (Explainability)**:
  - *Contribution*: We explore an approach for interpreting the results of a deep learning model that is applied to a classification problem [Fiosina et al., 2020], introduce an explainable federated architecture for the travel-time forecasting problem [Fiosina, 2021], and propose a novel research direction towards explainable decision-making in multi-agent systems [Kraus et al., 2020].
  - *Future research directions*:
    - The interpretation of black-box models of ML methods; the use, comparison, combination, and proposal of novel explainability methods.
    - An extension of the research direction 'Explainable decision-making in multi-agent systems' by proposing novel explainability ML algorithms.

  - The creation of explainability algorithms for data analysis in distributed and decentralised architectures.

- **Challenge 6 (ML applications)**:
  - *Contribution*: We apply state-the-art and our proposed ML methods to solve prediction [Fiosina, 2012], [Fiosina, 2021], classification [Fiosina et al., 2020], and change-point detection [Fiosins et al., 2012] problems for transportation and bioinformatics applications.
  - *Future research directions*:
    - The integration of the proposed algorithms into complex data-driven AI system architectures as a data processing step (e.g. integrating the proposed route comparison and individual user preference algorithms into cloud-based intelligent transportation systems).
    - The implementation of the state-of-the-art and proposed decentralised ML algorithms for solving problems in the transportation domain (e.g. the vehicle routing problem) and transferring the knowledge to other domains.
    - The application of centralised ML methods for other scenarios in transportation (ride-sharing), bioinformatics, etc. For example, investigating models of other types of expression data, improving the classification results by using more accurate variables and sample filtering, and performing deeper result interpretation, including the enrichment of non-miRNAs and contaminants.
    - The application of distributed ML and federated learning architectures to problems in various domains to ensure rapid, reliable, robust, and privacy-preserving data analysis.

# References


Abbasi and Haq, 2019. Abbasi, S. and Haq, A. (2019). Enhanced adaptive CUSUM charts for process mean. *Journal of Statistical Computation and Simulation*, 89(13):2562–2582.

Afanasyeva, 2005. Afanasyeva, H. (2005). Resampling-approach to a task of comparison of two renewal processes. In *Proc. of the 12th Int. Conf, on Analytical and Stochastic Modelling Techniques and Applications*, pages 94–100, Riga.

Afanasyeva and Andronov, 2005. Afanasyeva, H. and Andronov, A. (2005). On robustness of resampling estimators for linear regression models. In *Proc. of the Int. Symposium on Stochastic Models in Reliability, Safety and Logistics*, pages 6–11.

Afanasyeva and Andronov, 2006. Afanasyeva, H. and Andronov, A. (2006). On robustness of resampling estimators for linear regression models. *Communications in Dependability and Quality Management: An international Journal*, 9(1):5–11.

Albert, 1972. Albert, A. (1972). *Regression and the Moor-Penrose Pseudoinverse*. Academic Press, New York and London.

Ancona et al., 2018. Ancona, M., Ceolini, E., Öztireli, C., and Gross, M. (2018). Towards better understanding of gradient-based attribution methods for deep neural networks. In *In Proc. of 6th Int. Conf. on Learning Representations, ICLR*.

Andronov et al., 2009. Andronov, A., Fioshina, H., and Fioshin, M. (2009). Statistical estimation for a failure model with damage accumulation in a case of small samples. *Journal of Statistical Planning and Inference*, 139(5):1685 – 1692.

Andronov et al., 1991. Andronov, A., Kiselenko, A., and Mostivenko, E. (1991). *Forecasting of the development of Regional Transport System*. KNZ UrO RAN, Sivtivkar. (In Russian).

Andronov and Merkuryev, 2000. Andronov, A. and Merkuryev, Y. (2000). Optimization of statistical sample sizes in simulation. *Journal of Statistical Planning and Inference*, 85(1-2):93 – 102.

Arif et al., 2017. Arif, S., Mohamad Mohsin, M. F., Abu Bakar, A., Hamdan, A., and Syed Abdullah, S. (2017). Change point analysis: A statistical approach to detect potential abrupt change. *Jurnal Teknologi*, 79.

Aroor et al., 2018. Aroor, A., Epstein, S. L., and Korpan, R. (2018). Online learning for crowd-sensitive path planning. In *Proc. of the 17th Int. Conf. on Autonomous Agents and MultiAgent Systems*, AAMAS '18, page 1702–1710, Richland, SC. Int. Foundation for Autonomous Agents and Multiagent Systems.

Bach et al., 2015. Bach, S., Binder, A., et al. (2015). On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PLoS ONE*, 10(7).

Bazzan and Klügl, 2013. Bazzan, A. and Klügl, F. (2013). A review on agent-based technology for traffic and transportation. *The Knowledge Engineering Review*, FirstView:1–29.

Bonawitz et al., 2019. Bonawitz, K., Eichner, H., Grieskamp, W., Huba, D., Ingerman, A., Ivanov, V., Kiddon, C., Konecný, J., Mazzocchi, S., McMahan, H. B., Overveldt, T. V., Petrou, D., Ramage, D., and Roselander, J. (2019). Towards federated learning at scale: System design. *CoRR*, abs/1902.01046.

Breiman, 2001. Breiman, L. (2001). Random forests. *Machine Learning*, 45(1):5–32.

Buckberry et al., 2014. Buckberry, S., Bent, S. J., Bianco-Miotto, T., and Roberts, C. (2014). massir: a method for predicting the sex of samples in gene expression microarray datasets. *Bioinformatics*, 30:2084–2085.

Büchel and Corman, 2020. Büchel, B. and Corman, F. (2020). Review on statistical modeling of travel time variability for road-based public transport. *Frontiers in Built Environment*, 6:70.

Cao et al., 2009. Cao, L., Luo, D., and Zhang, C. (2009). Ubiquitous intelligence in agent mining. In *Agents and Data Mining Interaction. Lecture Notes in Computer Science*, volume 5680, pages 23–35.

Carslaw et al., 2006. Carslaw, D. C., Ropkins, K., and Bell, M. C. (2006). Change-point detection of gaseous and particulate traffic-related pollutants at a roadside location. *Environmental Science & Technology*, 40(22):6912–6918.

Chen and Guestrin, 2016. Chen, T. and Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In *Proc. of the 22nd ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, KDD '16, page 785–794, New York, NY, USA. Association for Computing Machinery.

Chen, 2013. Chen, X. (2013). *A dissertation: Analysis of Big Data by Split-and-Conquer and Penalized Regressions: New Methods and Theories*. New Brunswick, New Jersey, USA: Rutgers, The State University of New Jersey.

Chen et al., 2021. Chen, X., Cheng, J. Q., and Xie, M. (2021). Divide-and-conquer methods for big data analysis. *ArXiv*, abs/2102.10771.

Chlyah et al., 2016. Chlyah, M., Dardor, M., and Boumhidi, J. (2016). Multi-agent system based on support vector machine for incident detection in urban roads. In *In Proc. of the 11th Int. Conf. on Intelligent Systems: Theories and Applications (SITA)*, pages 1–6.

Choi et al., 2016. Choi, E., Bahadori, M., et al. (2016). RETAIN: An interpretable predictive model for healthcare using reverse time attention mechanism. In Lee, D. D., Sugiyama, M., Luxburg, U. V., Guyon, I., and Garnett, R., editors, *Advances in Neural Information Processing Systems 29*, pages 3504–3512. Curran Associates, Inc.

Chowdhury et al., 2017. Chowdhury, M., Apon, A., and Dey, K. (2017). *Data Analytics for Intelligent Transportation Systems*. Elsevier Science Publishers B. V., NLD, 1st edition.

Claes and Holvoet, 2011. Claes, R. and Holvoet, T. (2011). Ad hoc link traversal time prediction. In *IEEE Conf. on ITS*, pages 1803–1808.

Comert et al., 2020. Comert, G., Rahman, M., Islam, M., and Chowdhury, M. (2020). Change point models for real-time cyber attack detection in connected vehicle environment. *CoRR*, abs/2003.04185.

da Silva et al., 2005. da Silva, J. C., Giannella, C., Bhargava, R., Kargupta, H., and Klusch, M. (2005). Distributed data mining and agents. *Eng. Appl. of AI*, 18(7):791–807.

Davison and Hinkley, 1997. Davison, A. and Hinkley, D. (1997). Bootstrap methods and their application. *Journal of the American Statistical Association*, 94.

de la Torre et al., 2021. de la Torre, R., Corlu, C. G., Faulin, J., Onggo, B. S., and Juan, A. A. (2021). Simulation, optimization, and machine learning in sustainable transportation systems: Models and applications. *Sustainability*, 13(3).

de Viña and Martínez-Muñoz, 2018. de Viña, P. and Martínez-Muñoz, G. (2018). Using bag-of-little bootstraps for efficient ensemble learning. In Kůrková, V., Manolopoulos, Y., Hammer, B., Iliadis, L., and Maglogiannis, I., editors, *Artificial Neural Networks and Machine Learning – ICANN 2018*, pages 538–545, Cham. Springer International Publishing.

Duan et al., 2019. Duan, P., Mao, G., Huang, B., and Kang, J. (2019). Estimating link travel time distribution using network tomography technique. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pages 2598–2603.

Efron and Tibshirani, 1993. Efron, B. and Tibshirani, R. (1993). *An introduction to the Bootstrap*. Chapman and Hall, New York.

Ellis et al., 2018. Ellis, S. et al. (2018). Improving the value of public RNA-seq expression data by phenotype prediction. *Nucleic Acids Res.*, 46(9).

Ferger, 2002. Ferger, D. (2002). On the almost sure convergence of maximum likelihood-type estimators for a change-point. *Theory Stoch. Processes*, 8(1–2):81–87.

Fernández et al., 2014. Fernández, A., del Río, S., López, V., Bawakid, A., del Jesus, M. J., Benítez, J. M., and Herrera, F. (2014). Big data with cloud computing: An insight on the computing environment, mapreduce, and programming frameworks. *Wiley Int. Rev. Data Min. and Knowl. Disc.*, 4(5):380–409.

Fioshin, 2000. Fioshin, M. (2000). Efficiency of resampling estimators of sequential-parallel systems reliability. In *Proc. of 2nd Int. Conf. Simulation, Gaming, Training and Business Process Reengineering in Operations*, pages 112–117, Riga.

Fiosina, 2012. Fiosina, J. (2012). Decentralised regression model for intelligent forecasting in multi-agent traffic networks. In *AISC - DCAI'12*, volume 151, pages 255–263. Springer.

Fiosina, 2021. Fiosina, J. (2021). Explainable federated learning for taxi travel time prediction. In *Proc. of the 7th Int. Conf. on Vehicle Technology and Intelligent Transport Systems - VEHITS*, pages 670–677. INSTICC, SciTePress.

Fiosina and Fiosins, 2011. Fiosina, J. and Fiosins, M. (2011). Resampling-based change point estimation. In Gama, J., Bradley, E., and Hollmén, J., editors, *Advances in Intelligent Data Analysis X*, volume 7014 of *Lecture Notes in Computer Science*, pages 150–161. Springer Berlin / Heidelberg.

Fiosina and Fiosins, 2012. Fiosina, J. and Fiosins, M. (2012). Distributed cooperative kernel-based forecasting in decentralized multi-agent systems for urban traffic networks. In *In Proc. of Ubiquitous Data Mining (UDM) Workshop in conjunction with ECAI 2012*, pages 3–7, Montpellier, France.

Fiosina and Fiosins, 2014. Fiosina, J. and Fiosins, M. (2014). Resampling based modelling of individual routing preferences in a distributed traffic network. *Int. Journal of AI*, 12(1):79–103.

Fiosina and Fiosins, 2017. Fiosina, J. and Fiosins, M. (2017). Distributed nonparametric and semiparametric regression on spark for big data forecasting. *Applied Comp. Int. and Soft Computing*, 2017:13.

Fiosina et al., 2020. Fiosina, J., Fiosins, M., and Bonn, S. (2020). Explainable deep learning for augmentation of sRNA expression profiles. *Journal of Computational Biology*, 27(2):234–247.

Fiosina et al., 2013a. Fiosina, J., Fiosins, M., and Müller, J. (2013a). Big data processing and mining for next generation intelligent transportation systems. *Jurnal Teknologi*, 63(3).

Fiosina et al., 2013b. Fiosina, J., Fiosins, M., and Müller, J. (2013b). Decentralised cooperative agent-based clustering in intelligent traffic clouds. In et al., M. K., editor, *Multiagent System Technologies – Proc. of 11th German Conf. on MAS Technologies*, volume 8076 of *LNAI*, pages 59–72. Springer.

Fiosina et al., 2013c. Fiosina, J., Fiosins, M., and Müller, J. (2013c). Mining the traffic cloud: Data analysis and optimization strategies for cloud-based cooperative mobility management. In Casillas, J., Martínez-López, F., Vicari, R., and De la Prieta, F., editors, *Management Intelligent Systems*, volume 220 of *Advances in Intelligent Systems and Computing*, pages 25–32. Springer.

Fiosins et al., 2011. Fiosins, M., Fiosina, J., Müller, J., and Görmer, J. (2011). Agent-based integrated decision making for autonomous vehicles in urban traffic. In Demazeau, Y., Pechoucek, M., Corchado, J., and Perez, J., editors, *Advances on PAAMS*, volume 88 of *Advances in Intelligent and Soft Computing*, pages 173–178. Springer.

Fiosins et al., 2012. Fiosins, M., Fiosina, J., and Müller, J. P. (2012). Change point analysis for intelligent agents in city traffic. In Cao, L., Bazzan, A., Symeonidis, A., Gorodetsky, V., Weiss, G., and Yu, P., editors, *Agents and Data Mining Interaction, ADMI 2011*, volume 7103 of *LNCS*. Springer Berlin / Heidelberg.

Fiosins et al., 2016. Fiosins, M., Friedrich, B., Görmer, J., Mattfeld, D., Müller, J. P., and Tchouankem, H. (2016). A multiagent approach to modeling autonomic road transport support systems. In *Autonomic Road Transport Support Systems*.

Freitas, 2002. Freitas, A. A. (2002). *Data Mining and Knowledge Discovery with Evolutionary Algorithms*. Springer-Verlag, Berlin, Heidelberg.

Galakatos et al., 2018. Galakatos, A., Crotty, A., and Kraska, T. (2018). *Distributed Machine Learning*, pages 1196–1201. Springer New York, New York, NY.

Gavit et al., 2009. Gavit, P., Baddour, Y., and Tholmer, R. (2009). Use of change-point analysis for process monitoring and control. *BioPharm International*, 22.

GEO, . GEO. Gene Expression Omnibus. https://www.ncbi.nlm.nih.gov/geo/.

Gispan et al., 2017. Gispan, L., Leshem, A., and Be'ery, Y. (2017). Decentralized estimation of regression coefficients in sensor networks. *Digital Signal Processing*, 68:16–23.

Guo et al., 2020. Guo, C., Yang, B., Hu, J., Jensen, C. S., and Chen, L. (2020). Context-aware, preference-based vehicle routing. *The VLDB Journal*, 29:1149–1170.

Guo et al., 2017. Guo, L. et al. (2017). miRNA and mRNA expression analysis reveals potential sex-biased miRNA expression. *Scientific Reports*, 7.

Hadley et al., 2017. Hadley, D., Pan, J., et al. (2017). Precision annotation of digital samples in ncbi's gene expression omnibus. *Sci. Data*, 4:170125.

Härdle, 2002. Härdle, W. (2002). *Applied Nonparametric Regression*. Cambridge University Press, Cambridge.

Härdle et al., 2004. Härdle, W., Müller, M., Sperlich, S., and Werwatz, A. (2004). *Nonparametric and Semiparametric Models*. Springer, Berlin/Heidelberg.

Helwig, 2014. Helwig, N. E. (2014). Semiparametric regression of big data in r. In *Proc. of the CSE Big Data Workshop*.

Henderson and Parmeter, 2015. Henderson, D. and Parmeter, C. (2015). *Applied Nonparametric Econometrics*. Applied Nonparametric Econometrics. Cambridge University Press.

Hinkelmann et al., 2018. Hinkelmann, D., Schmeink, A., and Dartmann, G. (2018). Distributed learning-based state prediction for multi-agent systems with reduced communication effort. In *Proc. of the 15th ACM Int. Conf. on Computing Frontiers*, CF '18, page 376–380, New York, NY, USA. Association for Computing Machinery.

Hinkley, 1971. Hinkley, D. V. (1971). Inference about the change-point from cumulative sum tests. *Biometrika*, 58(3):509–523.

Hinneburg and Gabriel, 2007. Hinneburg, A. and Gabriel, H.-H. (2007). Denclue 2.0: Fast clustering based on kernel density estimation. *Adv. in Intelligent Data Analysis VII, LNCS*, 4723:70–80.

James et al., 2021. James, G., Witten, D., Hastie, T., and Tibshirani, R. (2021). *An Introduction to Statistical Learning with Applications in R, 2nd Edition*. Springen.

Jin et al., 2020. Jin, X., Takayama, T., Yashiro, A., and Nakamura, T. (2020). Analysis of personal routing preference from probe data in cloud. In *SAE Technical Paper 2020-01-0740*. SAE.

Johnson et al., 2018. Johnson, N., Dhroso, A., Hughes, K., and Korkin, D. (2018). Biological classification with rna-seq data: Can alternatively spliced transcript expression enhance machine learning classifiers? *RNA*, 24:1119–1132.

Khalil et al., 2015. Khalil, K. M., Abdel-Aziz, M., Nazmy, T. T., and Salem, A.-B. M. (2015). Machine learning algorithms for multi-agent systems. In *Proc. of the Int. Conf. on Intelligent Information Processing, Security and Advanced Communication*, IPAC '15, New York, NY, USA. Association for Computing Machinery.

Kleiner et al., 2014. Kleiner, A., Talwalkar, A., Sarkar, P., and Jordan, M. I. (2014). A scalable bootstrap for massive data. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 76(4):795–816.

Klusch et al., 2003. Klusch, M., Lodi, S., and Moro, G. (2003). Agent-based distributed data mining: The KDEC scheme. In *AgentLink*, pages 104–122.

Konecný et al., 2016. Konecný, J., McMahan, H. B., Yu, F., Richtárik, P., Suresh, A., and Bacon, D. (2016). Federated learning: Strategies for improving communication efficiency. *CoRR*, abs/1610.05492.

Kong and Yu, 2018. Kong, Y. and Yu, T. (2018). A deep neural network model using random forest to extract feature representation for gene expression data classification. *Scientific Reports*, 8.

Kraus et al., 2020. Kraus, S., Azaria, A., Fiosina, J., Greve, M., Hazon, N., Kolbe, L. M., Lembcke, T., Müller, J. P., Schleibaum, S., and Vollrath, M. (2020). AI for explaining decisions in multi-agent environments. In *The 34th AAAI Conference on AI, AAAI 2020, 2020, New York, NY, USA, February 7-12, 2020*, pages 13534–13538. AAAI Press.

Lakkaraju et al., 2017. Lakkaraju, H., Kamar, E., Caruana, R., and Leskovec, J. (2017). Interpretable & explorable approximations of black box models. In *Proc. of KDD Workshop on Fairness, Accountability, and Transparency in Machine Learning (FAT ML)*.

LeCun et al., 2015. LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *Nature*, 521.

Lee, 2019. Lee, J. (2019). Complementary reinforcement learning towards explainable agents. *CoRR*, abs/1901.00188.

Lee et al., 2010. Lee, W., Tseng, S., and Shieh, W. (2010). Collaborative real-time traffic information generation and sharing framework for the intelligent transportation system. *Information Sciences*, 180:62–70.

Leskovec et al., 2020. Leskovec, J., Rajaraman, A., and Ullman, J. D. (2020). *Mining of Massive Datasets, 3dr Edition*. Cambridge University Press, Cambridge, UK.

Li et al., 2019a. Li, M., Ma, Y., and Li, R. (2019a). Semiparametric regression for measurement error model with heteroscedastic error. *Journal of Multivariate Analysis*, 171:320–338.

Li et al., 2019b. Li, M., Qin, Z., Jiao, Y., Yang, Y., Wang, J., Wang, C., Wu, G., and Ye, J. (2019b). Efficient ridesharing order dispatching with mean field multi-agent reinforcement learning. In *The World Wide Web Conference*, WWW '19, page 983–994, New York, NY, USA. Association for Computing Machinery.

Li et al., 2011. Li, Q., Zhang, T., and Yu, Y. (2011). Using cloud computing to process intensive floating car data for urban traffic surveillance. *Int. Journal of Geographical Information Science*, 25:1303–1322.

Li et al., 2019c. Li, Y., Huang, C., Ding, L., Li, Z., Pan, Y., and Gao, X. (2019c). Deep learning in bioinformatics: Introduction, application, and perspective in the big data era. *Methods*, 166:4–21. Deep Learning in Bioinformatics.

Liang, 2006. Liang, H. (2006). Estimation in partially linear models and numerical comparisons. *Computational statistics and data analysis*, 50:675–687.

Liao et al., 2021. Liao, B., Anderson, M., and Anderson, S. L. (2021). Representation, justification and explanation in a value driven agent: An argumentation-based approach. *AI Ethics*, 1:1–19.

Lin et al., 2005a. Lin, H., Zito, R., and Taylor, M. (2005a). A review of travel-time prediction in transport and logistics. In *Proc. of the Eastern Asia Society for Transportation Studies*, volume 5, pages 1433 – 1448, Hamburg.

Lin et al., 2005b. Lin, H.-E., Zito, R., and Taylor, M. A. (2005b). A review of travel-time prediction in transport and logistics. In *Proc. of the Eastern Asia Society for Transportation Studies*, volume 5, pages 1433 – 1448, Hamburg.

Madan et al., 2018. Madan, S., Fiosins, M., et al. (2018). A semantic data integration methodology for translational neurodegenerative disease research. In *Proc. of the 11th Int. Conf. Semantic Web Applications and Tools for Life Sciences (SWAT4HCLS 2018)*, volume 2275. CEUR Workshop Proceedings.

Madumal et al., 2018. Madumal, P., Miller, T., Vetere, F., and Sonenberg, L. (2018). Towards a grounded dialog model for explainable artificial intelligence. In *1st Int. workshop on socio-cognitive systems at IJCAI 2018*.

Manolopoulou et al., 2010. Manolopoulou, I., Chan, C., and West, M. (2010). Selection sampling from large data sets for targeted inference in mixture modeling. *Bayesian analysis*, 5(3):1–22.

McKnight et al., 2004. McKnight, C. E., Levinson, H. S., Kamga, C., and Paaswell, R. E. (2004). Impact of traffic congestion on bus travel time in northern new jersey. *Transportation Res. Record Journal*, 1884:27–35.

Meng et al., 2016. Meng, X., Bradley, J., Yavuz, B., Sparks, E., Venkataraman, S., Liu, D., Freeman, J., Tsai, D., Amde, M., Owen, S., Xin, D., Xin, R., Franklin, M. J., Zadeh, R., Zaharia, M., and Talwalkar, A. (2016). Mllib: Machine learning in apache spark. *Journal of Machine Learning Research*, 17(34):1–7.

Michailidis and Margaritis, 2013. Michailidis, P. and Margaritis, K. (2013). Accelerating kernel density estimation on the gpu using the cuda framework. *Applied mathematical sciences*, 7:1447–1476.

Mishra et al., 2017. Mishra, S., Sturm, B., and Dixon, S. (2017). Local interpretable model-agnostic explanations for music content analysis. In *Proc. of International Society for Music Information Retrieval Conference*.

Molnar, 2019. Molnar, C. (2019). *Interpretable Machine Learning. A Guide for Making Black Box Models Explainable*. Lulu.

Montavon et al., 2017. Montavon, G., Lapuschkin, S., et al. (2017). On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *Pattern Recognition*, 65:211–222.

Negahban et al., 2012. Negahban, S., Oh, S., and Shah, D. (2012). Iterative ranking from pair-wise comparisons. *Advances in Neural Information Processing Systems*, 3.

Otto and Breitung, 2020. Otto, S. and Breitung, J. (2020). Backward CUSUM for Testing and Monitoring Structural Change. VfS Annual Conference 2020 (Virtual Conference): Gender Economics 224533, Verein für Socialpolitik / German Economic Association.

O'Brien and Ishwaran, 2019. O'Brien, R. and Ishwaran, H. (2019). A random forests quantile classifier for class imbalanced data. *Pattern Recognition*, 90:232 – 249.

Peltola, 2018. Peltola, T. (2018). Local interpretable model-agnostic explanations of bayesian predictive models via kullback-leibler projections. In *Proc. of the 2nd Workshop on XAI at IJCAI/ECAI*.

Ponomarev and Voronkov, 2017. Ponomarev, S. and Voronkov, A. E. (2017). Multi-agent systems and decentralized artificial superintelligence. *CoRR*, abs/1702.08529.

Racine, 1997. Racine, J. (1997). Consistent significance testing for nonparametric regression. *Journal of Business and Economic Statistics*, 15:369–379.

Rahman et al., 2018. Rahman, R.-U. et al. (2018). Oasis 2: improved online analysis of small rna-seq data. *BMC Bioinformatics*, 19(54).

Rahman et al., 2017. Rahman, R.-U., Sattar, A., Fiosins, M., et al. (2017). SEA: The small RNA expression atlas. *bioRxiv*.

Rao1 et al., 2010. Rao1, V. S., Vidyavathi, S., and G.Ramaswamy (2010). Distributed data mining and agent mining interaction and integration: A novel approach. *IJRRAS*, 4(4):388–398.

Ribeiro et al., 2016. Ribeiro, M. T., Singh, S., and Guestrin, C. (2016). "Why should I trust you?": Explaining the predictions of any classifier. In *Proc. of the 22nd ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, pages 1135–1144.

Rosemarin et al., 2019. Rosemarin, H., Rosenfeld, A., and Kraus, S. (2019). Emergency department online patient-caregiver scheduling. In *The 33th AAAI Conference on AI, AAAI-19*.

Rosenfeld and Kraus, 2016. Rosenfeld, A. and Kraus, S. (2016). Strategical argumentative agent for human persuasion. In *ECAI*, pages 320–328.

Selvaraju et al., 2017. Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., and Batra, D. (2017). Grad-cam: Visual explanations from deep networks via gradient-based localization. In *2017 IEEE Int.Conf. on Computer Vision (ICCV)*, pages 618–626.

Shengdong et al., 2019. Shengdong, M., Zhengxian, X., and Yixiang, T. (2019). Intelligent traffic control system based on cloud computing and big data mining. *IEEE Transactions on Industrial Informatics*, 15(12):6583–6592.

Shrikumar et al., 2017a. Shrikumar, A., Greenside, P., and Kundaje, A. (2017a). Learning important features through propagating activation differences. In Precup, D. and Teh, Y. W., editors, *Proc. of ML Research*, volume 70, pages 3145–3153, Int. Convention Centre, Sydney, Australia. PMLR.

Shrikumar et al., 2017b. Shrikumar, A., Greenside, P., and Kundaje, A. (2017b). Learning important features through propagating activation differences. In *Proc. of 34th Int. Conf. on ML, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, pages 3145–3153.

Soliman et al., 2020. Soliman, A., Girdzijauskas, S., Bouguelia, M.-R., Pashami, S., and Nowaczyk, S. (2020). Decentralized and adaptive k-means clustering for non-iid data using hyperloglog counters. In Lauw, H. W., Wong, R. C.-W., Ntoulas, A., Lim, E.-P., Ng, S.-K., and Pan, S. J., editors, *Advances in Knowledge Discovery and Data Mining*, pages 343–355, Cham. Springer International Publishing.

Speckman, 1988. Speckman, P. (1988). Kernel smoothing in partial linear models. *Journal of the Royal Statistical Society, Series B*, 50:413–436.

Stankovic et al., 2009. Stankovic, S. S., Stankovic, M. S., and Stipanovic, D. M. (2009). Decentralized parameter estimation by consensus based stochastic approximation. *IEEE Trans. Automatic Controll*, 56.

Statnikov et al., 2008. Statnikov, A., Wang, L., and Aliferis, C. F. (2008). A comprehensive comparison of random forests and support vector machines for microarray-based cancer classification. *BMC Bioinformatics*, 9.

Storz, 2002. Storz, G. (2002). An expanding universe of noncoding rnas. *Science*, 296(5571):1260–1263.

Sundararajan et al., 2017. Sundararajan, M., Taly, A., and Yan, Q. (2017). Axiomatic attribution for deep networks. In *Proc. of the 34th Int. Conf. on ML*, volume 70 of *ICML'17*, page 3319–3328. JMLR.org.

Symeonidis and Mitkas, 2005. Symeonidis, A. and Mitkas, P. (2005). A methodology for predicting agent behavior by the use of data mining techniques. In Gorodetsky, V., Liu, J., and Skormin, V., editors, *Autonomous Intelligent Systems: Agents and Data Mining*, pages 161–174, Berlin, Heidelberg. Springer.

Talia, 2011. Talia, D. (2011). Cloud computing and software agents: Towards cloud intelligent services. In Fortino, G., Garro, A., Palopoli, L., Russo, W., and Spezzano, G., editors, *WOA*, volume 741 of *CEUR Workshop Proceedings*, pages 2–6. CEUR-WS.org.

Tartakovsky and Kim, 2006. Tartakovsky, A. G. and Kim, H. (2006). Performance of certain decentralized distributed change detection procedures. In *In Proc. of the 9th Int. Conf. on Information Fusion*, pages 1–8.

Thompson, 2012. Thompson, S. (2012). *Sampling*. CourseSmart. Wiley.

Varghese et al., 2020. Varghese, V., Chikaraishi, M., and Urata, J. (2020). Deep learning in transport studies: A meta-analysis on the prediction accuracy. *Journal of Big Data Analytics in Transportation*, 2:199–220.

Verbraeken et al., 2020. Verbraeken, J., Wolting, M., Katzy, J., Kloppenburg, J., Verbelen, T., and Rellermeyer, J. S. (2020). A survey on distributed machine learning. *ACM Comput. Surv.*, 53(2).

Wang, 2008. Wang, F. (2008). Toward a revolution in transportation operations: AI for complex systems. *IEEE Intelligent Systems*, 23:8–13.

Wang, 2019. Wang, G. (2019). Interpret federated learning with shapley values. *CoRR*, abs/1905.04519.

Webb, 2018. Webb, S. (2018). Deep learning for biology. *Nature*, 554:555–557.

Wilkinson et al., 2016. Wilkinson, M. D. et al. (2016). The fair guiding principles for scientific data management and stewardship. *Sci. Data*, 3:160018.

Wooldridge, 2009. Wooldridge, M. (2009). *An Introduction to MultiAgent Systems*. John Wiley and Sons.

Wu, 1986. Wu, C. F. J. (1986). Jackknife, bootstrap and other resampling methods in regression analysis. *The Annals of Statistics*, 14(4):1261–1295.

Xiao et al., 2015. Xiao, T. et al. (2015). Learning from massive noisy labeled data for image classification. In *2015 IEEE Conf. on Comp. Vision and Pattern Recognition (CVPR)*, pages 2691–2699.

Yang et al., 2019. Yang, Q., Liu, Y., Chen, T., and Tong, Y. (2019). Federated machine learning: Concept and applications. *ACM Trans. Intell. Syst. Technol.*, 10(2).

Zargayouna, 2019. Zargayouna, M. (2019). *Multi-Agent Approaches for Dynamic Transportation Problems. Habilitation Thesis.* Université Paris Dauphine.

# Part II
# Appendix

# Appendix A

## List of Own Publications and Description of my Contribution

The following scientific contributions have been presented in this manuscript.

### A.1 Computationally intensive ML for data analysis

**Deep Learning and Random Forest-Based Augmentation of sRNA Expression Profiles**, J. Fiosina, M. Fiosins, S. Bonn, In. Proc. of the Int. Symposium on Bioinformatics Research and Applications, LNCS, 11490, 159-170, Springer, 2019

**Contribution**: The work has been conducted jointly with M. Fiosins and S. Bonn. I was the main contributor (more than **70%**) to all stages of research. I was the main contributor to the machine learning part of the paper, selected and constructed the machine learning models, partially prepared the data, conducted the experiments and wrote the main parts of the paper. M. Fiosins and S. Bonn contributed to the biological problem statement, data preparation, interpretation of the results and wrote paragraphs to the paper.

**Resampling based modelling of individual routing preferences in a distributed traffic network**. J. Fiosina, M. Fiosins, Int. Journal of Artificial Intelligence, 12 (1), 79-103, 2014.

**Contribution**: This paper is joint work with M. Fiosins. I was the main contributor (more than **70%**) to all stages of research. I was responsible for the theoretical part of the paper and formal inferences connected with an estimation of stochastic processes, the implementation of the proposed methods, conducted experiments and wrote the paper. M. Fiosins contributed to the Markov Chain based route ranking algorithm description, executed data preparation stage and wrote paragraphs to the paper.

**Resampling-based change point estimation**, J. Fiosina, M. Fiosins, In. Proc. of Int. Symposium on Intelligent Data Analysis, LNCS, 7014, 150-161, Springer, 2011.

**Contribution**: This paper is joint work with M. Fiosins. I was responsible for the theoretical part of the paper and formal inferences connected with the estimation of theoretical properties of the compared CUSUM tests. I contributed to implementation, conduction of experiments and paper writing. We co-jointly contributed to the problem statement and evaluation of the results. My total contribution is more than **70%**.

**Explainable Deep Learning for Augmentation of Small RNA Expression Profiles**, J. Fiosina, M. Fiosins, S. Bonn, Journal of Computational Biology 27 (2), 234-247, 2020.

**Contribution**: The work has been conducted jointly with M. Fiosins and S. Bonn. I mostly contributed to the machine learning part of the paper. M. Fiosins and S. Bonn contributed to the biological part and the interpretation of the results. The explainability algorithm was constructed co-jointly with M. Fiosins. My total contribution is ca. **50%**.

## A.2 Decentralised data analysis

**Decentralised Regression Model for Intelligent Forecasting in Multi-agent Traffic Networks**, J. Fiosina, In Proc. of the Int. Conf. on Distributed Computing and Artificial Intelligence, AINCS, 151, 255-263, Springer, 2012.

**Contribution:** I am fully responsible for all contributions of this paper.

**Cooperative kernel-based forecasting in decentralized multi-agent systems for urban traffic networks**, J. Fiosina, M. Fiosins, In Proc. of the Workshop on Ubiquitous Data Mining, ECAI 2012, CEUR Workshop Proc.,vol. 960, 3-7, 2012

**Contribution**: This paper is joint work with M. Fiosins. I was the main contributor to all stages of the research. I proposed the distributed kernel-density regression algorithm, contributed to the implementation, conducted the experiments and wrote the main parts of the paper. M. Fiosins contributed to experimental evaluation, data preparation and wrote paragraphs to the paper. My contribution in total is more than **70%**.

**Big data processing and mining for next generation intelligent transportation systems**, J. Fiosina, M. Fiosins, J.P. Müller, Jurnal Teknologi 63 (3), 2013

**Contribution**: This paper is joint work with M. Fiosins and J.P. Müller. This paper aggregates our contribution to decentralised data processing. I was the main author of the paper (about **60%**). I was the main contributor to the decentralised forecasting, its implementation, conduction of experiments and writing the paper. The reference architecture was developed co-joitly with M. Fiosins.

M. Fiosins also contributed to decentralised clustering, its implementation, and wrote paragraphs to the paper. J.P. Müller contributed to discussions of concepts and language revision.

**Mining the traffic cloud: Data analysis and optimization strategies for cloud-based cooperative mobility management**, J. Fiosina, M. Fiosins, J.P. Müller, In. Proc. of the Int. Symposium on Management Intelligent Systems, AISC, 220, 25-32, Springer, 2013

**Contribution**: This paper is joint work with M. Fiosins with whom all stages of the research have been conducted conjointly. I was the main contributor to the description of cloud traffic scenarios and writing the paper. J.P. Müller contributed to discussions of concepts and language revision. My contribution is ca. **60%**.

**AI for explaining decisions in multi-agent environments**. S. Kraus, A. Azaria, J. Fiosina, M. Greve, N. Hazon, L. M. Kolbe, T. Lembcke, J. P. Müller, S. Schleibaum, and M. Vollrath. In the 34th AAAI Conference on AI (AAAI 2020), New York, USA, February 7-12, 2020, 34(09), pages 13534–13538, AAAI Press, 2020

**Contribution**: This paper is an outcome of the 'EC-Rider: Explainable AI methods for human-centric ride-sharing' project and is a joint work all of ten international participants from Israel and Germany. I was responsible for the overview of the state of the art explainability methods for black-box ML models, their place in the proposed research direction of explainable decisions in multi-agent environments and their application for ride-sharing domain. My contribution is ca. **10%**.

## A.3 Distributed centralised data analysis

**Explainable federated learning for taxi travel time prediction**. J. Fiosina, In Proc. of the 7th Int. Conf. on Vehicle Technology and Intelligent Transport Systems - VEHITS 2021, pages 670-677, INSTICC, SciTEPress.

**Contribution**: I am fully responsible for all contributions of this paper.

**Distributed Non-parametric and Semi-parametric Regression on SPARK for Big Data Forecasting**. J. Fiosina, M. Fiosins, Applied Computational Intelligence and Soft Computing, 2017.

**Contribution**: I was the main contributor to the proposition of the algorithms in SPARK terms, their implementation and conducted the main part of experiments. M. Fiosins prepared the cloud environment and data as well as contributed paragraphs to the paper. My contribution is ca. **70%**.