JOIV : Int. J. Inform. Visualization, 8(2) - May 2024 583-591

INTERNATIONAL JOURNAL ON INFORMATICS VISUALIZATION

journal homepage : www.joiv.org/index.php/joiv

Performance Evaluation on Resolution Time Prediction Using Machine Learning Techniques

Tong-Ern Tai [a], Su-Cheng Haw [a,*], Kok-Why Ng [a], Mutaz Al-Tarawneh [b], Gee-Kok Tong [a]

[a] *Faculty of Computing and Informatics, Multimedia University, , Cyberjaya, Selangor, 63100, Malaysia*
[b] *Faculty of Engineering, Mutah University, Karak, 61710, Jordan*

Corresponding author: [*]sucheng@mmu.edu.my

***Abstract*—The quality of customer service emphasizes support tickets. An excellent support ticket system qualifies businesses to provide clients with the finest level of customer support. This enables enterprises to guarantee the consistency of quality customer service delivered successfully, ensuring all clients have a good experience regardless of the nature of their inquiry or issue. To further achieve a higher efficiency of resource allocation, this is when the prediction of ticket resolution time comes into place. The advancing technologies, including artificial intelligence (AI) and machine learning (ML), can perform predictions on the duration required to tackle specific problems based on past similar data. ML enables the possibility of automatically classifying tickets, making it possible to anticipate the time resolution for cases. This paper explores various ML techniques widely applied in the Resolution Time Prediction system and investigates the performance of three selected ML techniques via the benchmarking dataset obtained from the UCI Machine Learning Repository. Implementing selected techniques will involve creating a graphical user interface and data visualization to provide insight for data analysis. The best technique will be concluded after performing the ML technique evaluation. The evaluation metrics involved in this step include Root Mean Square Error (RMSE) and Root Mean Absolute Error (MAE). The experimental evaluation shows that the best performance among the selected ML techniques is Random Forest (RF).**

***Keywords*— Resolution time prediction; machine learning; ticketing system; customer service; recommender system.**

Manuscript received 25 Aug. 2023; revised 31 Oct. 2023; accepted 16 Feb. 2024. Date of publication 31 May. 2024.

I. INTRODUCTION

Comprehensive customer service always plays a crucial role among businesses, and the effectiveness of resolution time is one of the highest demands of the consumer. When customers submit a service ticket to ask for help, they expect a rapid, straightforward, and effective response. The workers from the customer service side, who receive numerous tickets daily, aim to address assigned problems, providing a good customer experience and maintaining the company's positive image simultaneously. Services delivered on time can raise customer satisfaction, develop customer loyalty, and increase potential loyal customers [1], [2], [3].

In many industries, many services run in a take-turn manner, hence requiring customers to spend a long and meaningless waiting time to obtain this service, for example, ticketing, food preparation, and gadget repair. From experience, we believe these industries have accumulated a significant volume and variety of data, which would contain many similar cases and the time required to solve this case. The waiting process is time-consuming, leaving customers with a bad experience with that service. The emergence of Resolution Time Prediction is the solution to this condition.

The anticipation of the typical time required for a customer care provider to address a client's problem, inquiry, or complaint is known as resolution time prediction. Besides that, processing customer queries promptly would contribute significantly to boosting customer satisfaction. Advancements in technology, such as AI and ML, have led to the goal of automating this process by predicting the time needed to resolve specific issues based on similar cases in the past. The emergence of ML [4], [5] opens the possibility for automated ticket classification and, thus, enables the prediction of the resolution time needed to solve the cases [6], [7], [8]. Prediction of resolution times in any field is crucial in several domains.

A predictive model that utilizes ML techniques and past data's underlying pattern to anticipate ticket resolution times will be the ideal solution for quickening ticket assignment and completion [9], [10]. After the system receives a new ticket,

the trained ML model will auto-calculate the time required to resolve this query. The development of ML provides an opportunity to classify tickets, which makes it possible to predict resolution time automatically.

This paper aims to answer the following research questions:

- What ML technique is typically used in the Resolution Time Prediction system? This issue can be solved after research on implementing ML techniques in the available Resolution Time Prediction system.
- Numerous ML approaches should be applied in the Resolution Time Prediction systems. What are the differences between these approaches? Each ML approach holds different advantages and disadvantages, and various ML techniques serve different purposes. Thorough, detailed research is required to understand these approaches' properties.
- All ML techniques have advantages and disadvantages. Which type of ML technique best suits the research project? Before this research, a firm foundation for ML knowledge should be built. The most appropriate strategy can be found by closely monitoring and evaluating each approach's performance from the perspectives of the application outcome.

II. Materials and Method

A. Background on Machine Learning Technique

ML is one of AI's sub-branches, and this technique targets improving deliverables over time through learning data. The programmed ML algorithm will build a model by leveraging the information extracted from the training data and deliver outputs such as forecasting and decision-making without human supervision. The application of ML can be widely found in this digitalized era, and includes visual information processing, voice recognition, signature verification, and predictive analysis.

Under the topic of the Resolution Time Prediction system, the application of ML plays an essential role in making this system possible and successful [11], [12], [13]. A few ML algorithms are commonly used in the Resolution Time Prediction system, including Decision Tree [5], [14] Random Forest [9], [15] and Gradient Boosting [16].

1) Decision Tree: Decision tree (DT) is a common ML technique widely applied in classification and regression tasks. A DT serves as a decision assistance tool that employs a tree-based decision model. This technique uses conditional control statements to produce output. Although they are also a prevalent technique in ML, DT is frequently employed in operations research, notably in decision analysis, to assist in finding a plan that has the highest possibility of success. The concept of a DT algorithm is presented in Fig. 1.

A DT comprises a root node, interior nodes and leaf nodes. In a DT, the internal node indicates a specific condition containing a combination of variables, and the branch will reflect the test's result [17]. In contrast, the leaf node holds the decision of each computation. The routes from root and leaf stand in for categorization principles.

Operations management and operations research frequently employ DT techniques. DT ought to be paralleled by a probability model and chosen as the best option model if the decision must be made online with insufficient information. DT may also be used to compute conditional probabilities descriptively.

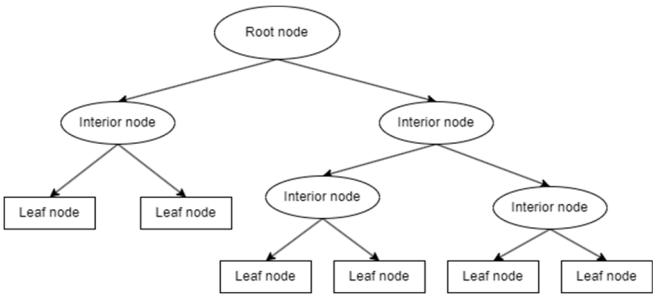

Fig. 1 Concept of DT Algorithm

2) Random Forest: The Random Forest (RF) is one of the ML techniques that is built during the training phase of DT. This technique is an ensemble learning approach for regression, classification, and others. The ensemble learning technique integrates predictions from several ML algorithms to provide forecasts that are more accurate than those from a single model [18]. In simple words, this technique incorporates many DTs to form a "forest" and can perform a very precise classification using the "forest." This is also a significant function for healthcare, business and retail applications. Fig. 2 depicts a typical RF concept.

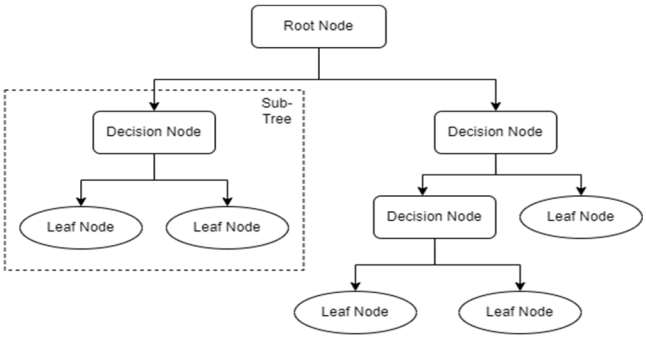

Fig. 2 RF concept

There are three critical hyperparameters for RF algorithms that must be specified before performing model training, and these parameters are tree count, node size, and the sampled feature count. Besides that, one-third of the training data will also be required to be saved prior as test data. The test data is reserved for cross-validation, and the prediction is finalized using the trained model. The method to obtain the result for RF regression and classification tasks differs. The outcome for the regression approach is concluded by averaging the individual DT. In contrast, the outcome for the classification task, which is the target class, should be found by observing the majority categorical variable.

3) Extreme Gradient Boosting: Extreme Gradient Boosting (XGBoost) has been widely utilized to tackle problems with supervised learning. The goal, such as regression or classification, the prediction value can be represented in various ways, such as regression and classification. This approach is commonly seen in a logistic conversion to obtain the probability of a positive class in a logistic regression task and task in ranking outputs by

observing the ranking score. Fig. 3 depicts the concept of the XGBoost algorithm [19].

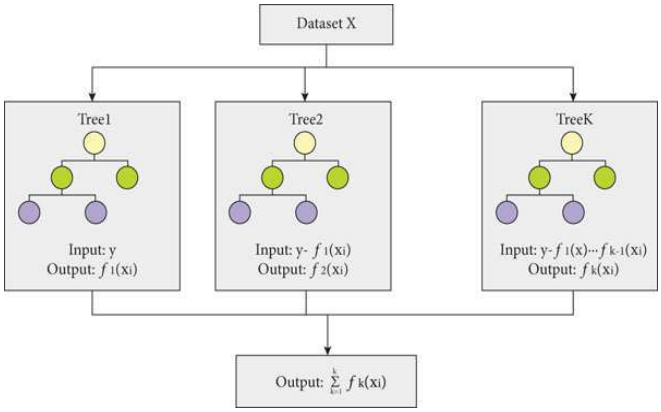

Fig. 3 XGBoost concept

This algorithm is suitable for employing a range of tasks, including regression, classification, and ranking functions by making wise selections. Similar to other approaches, the parameters used in this algorithm are unknown, which users must infer from the data. For example, the parameters would be coefficients for regression tasks. To identify whether the application of this approach suits the provided data, an objective function must be determined before performing data training.

The objective functions in this algorithm contain two components: training loss and regularization. The training loss indicates the model's accuracy in forecasting using train data. For example, logistic loss, utilized in logistic regression, is a frequently used loss function. The regularization component is required to prevent overfitting issues by controlling the model's complexity with the regularization term.

4) Comparison of DT, RF, and XGBoost: The standard type of ML techniques is discussed in detail. Table I investigates and summarizes ML techniques' benefits and drawbacks. Table I shows that each approach is distinct and has its benefits and disadvantages. For instance, the DT algorithm is an easy-to-implement algorithm. This algorithm does not require data pre-processing, such as data normalization and data scaling since this technique serves as an information-based approach. Nonetheless, this model is unstable as a minor modification to the data will cause a significant impact on the DT's structure. On the other hand, the RF algorithm can address the propensity of DT to overfit their training set, which usually leads to an excellent result in model performance. Nevertheless, this method cannot comprehend the findings, the possibility of overfitting, and the necessity of specifying the number of trees to include in the model. As for XGBoost, this technique does not require the features normalization process, which can save time in data processing. In contrast, the drawbacks of this approach are those of the RF algorithm, in which there is the possibility of overfitting if the generated trees are too deep with noisy data. In short, various ML approaches bring multiple benefits. There is no such type of ML approach that can fit all kinds of systems; hence, selecting a suitable method will be the aim of this paper.

TABLE I
BENEFITS AND DRAWBACKS OF ML TECHNIQUES

ML Technique	Benefit	Drawback
DT	• Data normalization and scaling are not required since this technique is an information-based approach. • Implementation of this algorithm is fast and easy.	• Computation power is high, and it takes longer to train the model. • This model is unstable as a minor modification to the data will cause a significant impact on the DT's structure.
RF	• Addressing the propensity of DTs to overfit their training set. • Often produces good results on many issues. Non-linear connection data is no exception.	• The inability to comprehend the findings. • Possibility for overfitting. • It is necessary to Specifying the number of trees to include in the model is necessary.
XGBoost	• Normalized features are not required • Suitable for nonlinear, non-monotonic data or with segregated clusters.	• Have the risk of overfitting if the generated trees are too deep with noisy data.

Pfahl et al. [20] enhanced an existing expert-based Issue Resolution Time (IRT) prediction system using ML techniques. The forecasting quality of expert-based IRT prediction is compared with a few automated ML approaches, including RF, ordered logistic regression (OLR), k-means clustering, k-nearest neighbor (KNN) classification, Naïve Bayes classification, and DT. The performance of chosen ML techniques is evaluated on the data of their case company. The obtained result is compared with the performance of the company's existing IRT prediction system. The R programming language is adopted for all calculations. The prediction accuracy is measured using the Magnitude of Absolute Error (MAE) method and the Magnitude of Relative Error (MRE) method. According to the evaluation, the case company's experts can correctly anticipate IRTs around half of the time, within a range of 10% of the actual IRTs, and the absolute error for 67% of the expert estimates is 0.5 hours or less. It was found that automated approaches obtain lower predictive quality in contrast to expert-based IRT predictions when it comes to working with the data provided by the company, and the highest-performing approaches, OLR and RF, are no exceptions. As the improvement by applying RF and OLR-based techniques failed, models based on text analysis were constructed by the researchers capable of generating a prediction quality equivalent to or greater than the performance of experts-based IRT prediction. In conclusion, the top text-analysis-based approach is Spherical k-means Clustering, whereas the highest-performance meta-information-based models' techniques are OLR and RF. The drawbacks of this approach include an underestimation of IRT in the suggested solution compared with the case of experts on a long-term basis. Future research will incorporate

a hybrid IRT prediction technique combining expert-based and automated estimates.

Rahaman et al. [21] discuss the neighborhood identification issue when several heterogeneous contextual features are present. The researchers frame the research as a queue wait time prediction problem for taxi drivers at airports and examine various times, weather, airline arrival, and taxi trip-related heterogeneous elements. An investigation of the link between the heterogeneous characteristics and the taxi line wait times is undertaken using a massive taxi queue wait time dataset that contains hourly taxi queue wait time and other heterogeneous contextual information. Based on the study, it is suggested that a driver intelligence-biased feature weighting strategy be used to determine a dense quality neighborhood for k-NN regression predictions of taxi line wait times. The performance evaluation of the proposed DI-biassed feature weighting scheme is conducted in two experiments. The first experiment aims to evaluate the Taxi Queue Wait Time Prediction by forecasting the taxi queue wait time and comparing it with various weighting methods, which include baseline, LR-trained weights, equal weights, MI-based weights, and DI-biassed weights, while the second experiment is to evaluate the density and quality of the neighborhood between the baseline and the proposed approach in their paper. In addition to the second experiment, the sample data is split into 30-40-30 randomly for feature selection and feature calculation, model training, and model testing, respectively, and explored with several feature weighting strategies for nearest neighbor estimation. As a result, the experiment's findings suggest that the appropriate weighted, heterogeneous contextual information can drastically raise the identified neighborhood's quality, ultimately enhancing prediction accuracy. In contrast, their paper has a limitation: the restriction in forecasting and the inability to provide the best options for decision-making.

Zuev et al. [22] proposed a predictive technique for estimating incident resolution time to understand event data and forecast the projected resolution time, enabling ML to detect bottlenecks in incident resolution. The researchers have investigated many learning models during development, including the Naive Bayesian classifier, Logistic Regression, and gradient-boosting DT model. The performance of the models is evaluated using the F1 score, which is the harmonic mean of the recall rate and accuracy. This method can consider the two most often used measures in classification: prediction and recall during calculation. Gradient Boosting DT models outperform all other approaches; hence, only the GBDT model is chosen as our chosen model. Researchers adopted a dataset consisting of actual service desk incident data to evaluate forecast accuracy and produce output indicating that the model can help forecast a broad group of problems. Furthermore, the service enhancement technique may be applied to its actual usage. The adoption of ML models in ITSM puts substantial emphasis on customer satisfaction and better coping, resulting in much less work for service desk employees and lower service costs.

Kyritsis and Deriaz [23] proposed a study that examined how ML is applied to anticipate customer wait times in various businesses that need lines. They started by forecasting bank queue wait times and then suggested the possible automation and generalized across different sectors. A fully connected neural network was trained, and this can estimate client wait times that give MAE approximately 3.35 minutes. In addition, they offered a web application that handles queues for various scenarios and sectors. The system may adjust to each queue as it builds an optimally learned neural network for waiting time forecasting for each queue, even if there is a particular possibility that the queues can have different parameters. The proposed system's performance is validated with the help of a simulator. The available dataset is divided into a training set and a test set according to the ratio 8:2 to evaluate the system's performance, with the test set remaining entirely hidden throughout the training phase. A neural network is chosen among all applied ML models in experiments performed due to one of the characteristics of neural networks: this learning model can be trained continuously. This proves that ML outperforms queueing theory when estimating waiting times.

In another study, Benevento et al. [24] proposed a study that proposes Queue-based features for dynamic waiting time prediction in emergency department variables aided by process mining and evaluated their effect on the waiting time prediction accuracy. Such queue-based predictors can track the current status quo of the emergency department (ED), which may dramatically increase the prediction models' accuracy. This study is mainly based on the methodology of the Cross-Industry Standard Process for Data Mining. The four primary steps of the approach for this study include data collection and preparation, predictor variable identification, predictor variable testing, and validation. They developed new queue-based predictors leveraging process mining and the conventional factors affecting ED waiting times. They could estimate the genuine patient flow and indicate the extent of activity crowding using process mining techniques. The performance was assessed using both linear and nonlinear learning approaches. As anticipated, the significant findings demonstrate that the accuracy of waiting time prediction is significantly increased by merging the predictor sets with queue-based characteristics. Using linear and nonlinear learning approaches, the mean square error values decreased by roughly 22% and 23%, respectively. As for the limitation, the quantity of the data collected in this study might be a drawback. Although a seven-month data collection is significant, there may be generalizability problems. As a result, employing a data set that spans one or more years may be more beneficial in accounting for seasonal fluctuations in waiting time and getting reliable outcomes.

Bejarano et al. [25] proposed a new method for forecasting emergency resolution called a Deep Learning-based Emergency Resolution Time Prediction System (DeepER). Deeper is a sequence-to-sequence model that employs recurrent neural networks (RNNs) as the neural network architecture. The model's performance is evaluated through two metrics, RMSE and MAE, and the result obtained is compared with other ML techniques, ARIMA and Linear Regression. Several tests are conducted using the provided data, and the data is adequately preprocessed to address the uneven resolution time distribution, outliers, and incomplete data. The DeepER system better estimates future resolution times by utilizing deep learning technology. Compared to ARIMA and linear regression, the proposed solution has enhanced 3% and 16% in terms of RMSE and 10% and 27%

in terms of MAE, respectively. This proposed system is successful, but this enhanced system also has limitations as the adopted dataset during the experiment has inadequate data points and not enough to fulfill the requirement of training, validation, and testing purposes. The DeepER system does not contribute to better performance after changing different datasets, and the researchers suspect this is due to insufficient data at the subtype level.

Pak et al. [26] proposed a study that enhanced waiting time forecasts for poor acuity emergency department (ED) patients allocated to the waiting area utilizing ML techniques and a broad range of queueing and service flow characteristics. They suggested using the proportion of underpredicted observations in conjunction with the mean squared prediction error (MSPE) and mean absolute prediction error (MAPE). The advantages of applying ML algorithms include flexible data link discovery, important prediction determination, and preventing overfitting of the data that drive their adoption. The researchers also apply quantile regression to develop time predictions that better manage the patient's asymmetric impression of under-predicted and over-predicted ED waiting times. As a result, ML models surpassed the highest rolling mean concerning MSPE by over 20% when queueing and service flow variables are integrated with knowledge of fluctuations. In comparison, quantile regression lowers the rate of patients with substantially underpredicted waiting times by 42%. The researchers conclude that there is strong evidence that the recommended estimation methods produce more precise estimates of ED waiting times than the rolling average. They demonstrated that to improve further predicted accuracy, a hospital ED can offer predictions exclusively to patients who register during the daytime when the ED is fully operational, resulting in higher predictive service rates and a greater demand for treatments. In contrast, this study does come with some constraints. The data employed in this investigation is collected from a single metropolitan tertiary hospital. This reduces the external validity of findings due to the possibility of varied service operations and patient demographics in other hospital EDs. The waiting time prediction models may be more accurate if supply-side elements from the hospital and emergency department (ED), which are used to adjust the dynamics in the utilization of healthcare resources, are included.

Hijry and Olawoyin [27] proposed a solution that studies deep learning techniques for historical queueing variables that will be used in addition to, or instead of, queueing theory to anticipate patient waiting times in a system (QT). They employed four optimization strategies, including Stochastic Gradient Descent (SGD) [28], adaptive moment estimation (Adam) [29],[30] Root Mean Square Propagation (RMSprop), and Adaptive Gradient (AdaGrad) [31], The model is evaluated using the mean absolute error (MAE) method. A conventional mathematical simulation is adopted for further analysis. The findings indicate that the deep learning model can accurately estimate patient wait times with the SGD approach with a minimum MAE of 10.80 minutes. By obtaining the best model, their study makes a theoretical contribution by estimating patients' waiting times using different methods. Their work makes a valuable contribution by utilizing actual ER data. They proposed approaches to forecast patients' wait times with more precise outcomes than a conventional mathematical approach. Furthermore, their newly proposed approach can quickly apply to the existing queuing system across the healthcare industry by leveraging the information extracted from electronic health records (EHRs). In contrast, their study has weaknesses concerning data availability, such as the provided ER information being minimal, for example, patient type of injury, X-ray operation duration, and laboratory test duration. The dataset provided is collected for one year only, leading to the model suffering from data hunger due to the application of deep learning. However, the dataset has been fully utilized.

Recently, Ma et al. [32] proposed a study involving a recurrent neural network ML technique, the long short-term memory (LSTM) network. This approach can study the order dependency employed to forecast the passenger flow and passenger flow of metro transit under normal and emergency conditions using transfer learning to tackle the unbalanced dataset issue when the emergency sample size is limited. This work makes innovative utilization of passenger flow data, which can show the transportation capacity of metro transportation more accurately. Moreover, transfer learning is applied in this research instead of the conventional approach to resolving the emergency passenger flow's sample size that is too small. The forecasting outcome without using transfer learning will be more comparable to the typical passenger flow than the emergency passenger flow, reflecting the actual scenario because of the influence of the average passenger flow data size. The final forecast error rate is less than 10%, which might aid the operational better and faster decision. The findings indicate that the average validation loss is below 5% in normal and emergency conditions, which should warn the operating firm to take preventative steps in advance. It demonstrated that the proposed technique has merits in anticipating emergencies compared to the strategy without transfer learning.

Schad et al. [33] demonstrated a method for estimating an incident's resolution difficulty. The approach taken in this study to solve this issue is triaging tickets by recognizing those that are linked to resolution complexity, which can assist in incorporating extra checks and error-proofing techniques. A Graph Convolutional Network is adopted to forecast the complexity of ticket resolution as one of the characteristics of the incident node, and it offers more than just a precise prediction model for the difficulty of ticket resolution. A loss function is applied to assess the model's performance, such as cross-entropy loss. In this experiment, the number of times a ticket is reallocated is a measuring tool for the complexity of event resolution. A flexible workflow is connected to ticket resolution. Regarding ad hoc inquiries, a graph representing ticket resolution can provide various benefits. This leads to the employment of ML in the Relational Graph Convolutional Network for predicting ticket reassignments. The created model has advantages beyond precisely forecasting ticket reassignments. This model offers embeddings that can acquire an understanding of the underlying operation of help desk organization and their user function. For privacy concerns, sensitive information is not included in the provided dataset. Further understanding of the ticket resolution process should result from using the method described in this study in conjunction with a strategy to relate

neural network models to a dataset comprising detailed information on the ticket actions.

B. Proposed Framework

Fig. 4 depicts the flow of our proposed prototype. The prototype will begin by requesting the user select one ML technique for generating recommendations. Only one dataset is needed in this phase, which is digested and cleaned to ensure output accuracy. Next, the prototype will perform model training and evaluation using metrics like RMSE and MAE. The evaluation result is analyzed and compared to conclude the best ML technique.

1) Choosing the ML Techniques: Several techniques that apply in the same domain as these research papers are studied, and three techniques are selected for this study. The first technique chosen is the DT technique, a supervised learning technique capable of performing regression and classification tasks. This technique works by learning straightforward decision rules derived from the data characteristics to build a model that anticipates the target variable value. This technique was chosen due to its capability as an information-based approach. Besides that, this technique does not require any data pre-processing.

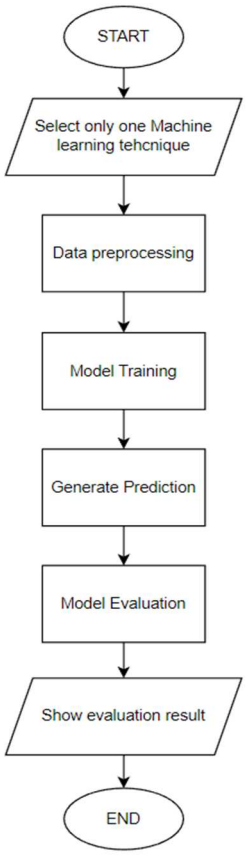

Fig. 4 Prototype implementation flow

The second chosen technique is the RF technique. This technique, also called random decision forests, is an ensemble learning technique for classification, regression, and other problems that work by constructing many DTs during the training phase. The RF output for classification problems is the target class selected by most of the trees, while the mean forecast score of each tree is used for regression problems. This algorithm was chosen due to its ability to minimize the overfitting in DT and hence enhance the model by boosting accuracy. This learning strategy functions well with categorical and continuous variables, making it adaptable to classification and regression issues.

The third technique is XGBoost, a regularized form of gradient boosting algorithm with the application of regularization. When using gradient boosting for regression, the weak learners are regression trees, and each regression tree maps an input data point to one of its leaves that contains a continuous score. This method minimizes a regularised objective function, which integrates a convex loss function with a penalty term for model complexity. The training process is carried out by incorporating new trees that forecast the residuals or mistakes of earlier trees, which are then integrated with earlier trees to get the final prediction. This method can minimize the loss of introducing new models. Hence, this technique is called gradient boosting.

2) Dataset: The dataset chosen for this research project is the Incident management process enriched event log dataset [34]. This multivariate and sequential dataset comprises 36 columns and 141712 rows of records. The information on attributes is stated in Table II.

TABLE II
ATTRIBUTES INFORMATION OF THE EVENT LOG DATASET

No	Attribute	Description
1	number	Incident identifier (24,918 different values)
2	incident state	Incident management process state
3	active	Is the incident still active or closed/canceled
4	reassignment_count	The reassignment of group or support analysts count
5	reopen_count	The rejection count of incident resolution from the caller
6	sys_mod_count	The update count of this incident
7	made_sla	Does the incident surpass the target SLA
8	caller_id	Affected user identifier
9	opened_by	Incident reporter identifier
10	opened_at	Time which incident user open
11	sys_created_by	Incident register identifier
12	sys_created_at	Incident system creation date and time
13	sys_updated_by	User update identifier
14	sys_updated_at	Time which this incident updates
15	contact_type	Contact type of reported incident
16	location	Location identifier
17	category	impacted service explanation (1st level)
18	subcategory	impacted service explanation (2nd level)
19	u_symptom	How users describe the availability of service
20	cmdb_ci	Confirmation item identifier
21	impact	Impact level of incident
22	urgency	Urgency level of incident
23	priority	Priority level of incident
24	assignment_ group	Support team identifier
25	assigned_to	Person in charge

No	Attribute	Description
26	knowledge	Boolean attribute that indicates the necessity of involving knowledge base document to solve this incident
27	u_priority_ confirmation	Has the priority field been checked
28	notify	Have the notifications produced for the incident
29	problem_id	Problem identifier
30	RFC	Alter request of incident identifier
31	vendor	Responsible vendor identifier
32	caused_by	RFC identifier
33	close_code	Incident resolution identifier
34	resolved_by	Identifier for a person who solved this incident
35	resolved_at	Time-resolved (dependent variable)
36	closed_at	Time incident status changes to closed (dependent variable)

3) Data Cleaning: The data cleaning stage begins with filtering the dataset by selecting records for the closed incident state. The dataset is cleaned by removing certain words in a few categorical columns. All missing values are replaced with NaN. Next, the data type of a few columns, which consists of datetime information, is converted to datetime format. Columns "impact," "urgency," and "priority" are cleaned by leaving only scale values. Any missing values present in columns "resolved_at," "sys_created_by," "sys_created_at," and "u_symptom" are removed. The time interval between incident user opening and close time is calculated in seconds, minutes, hours, and days and then saved in new columns. Any record of an incident that shows a negative time gap is removed. The month and day of the ticket opening time are extracted and saved in new columns. In addition, a few unwanted columns are removed from the dataset. Columns "incident_state" and "active" are removed due to all values in these columns being the same while column "caused_by", "vendor", "cmdb_ci", "rfc" and "problem_id" are removed due to the missing value present in these columns has a percentage higher than 98%. The data cleaning process is completed now and saved to a new CSV file named "cleaned_data.csv".

4) Model Training: After the completion of data cleaning, the dataset is ready for the model training. Three ML methods in the Scikit-learn library will be included in this phase: Linear Regression, RF, and Neural Network. This library offers quick and practical tools for predicting data in Python. In addition, this library also provides built-in solid features such as train-test split function and accuracy metrics. In this phase, 80% of the dataset is allocated to fit the ML model and 20% for model evaluation. Hyper-parameter tuning is involved to identify the best parameters for each algorithm. As the model is trained, it can predict the incident resolution time using the test data. The prediction is then evaluated using accuracy metrics MAE and RMSE. Lastly, the accuracy result is plotted using a bar chart.

III. Results and Discussion

The prototype main page is shown in Fig. 5. All data present in the dataset will be displayed, followed by the size of the dataset. The second tab shows the evaluation result of applied ML approaches. The prediction of incident resolution time will be compared in seconds, minutes, hours, and days to the actual time to find the most accurate prediction, as shown in Fig. 6. At the same time, the evaluation result will also be illustrated in Fig. 7. The evaluation score of each approach is recorded in Table III.

Fig. 5 Program main page

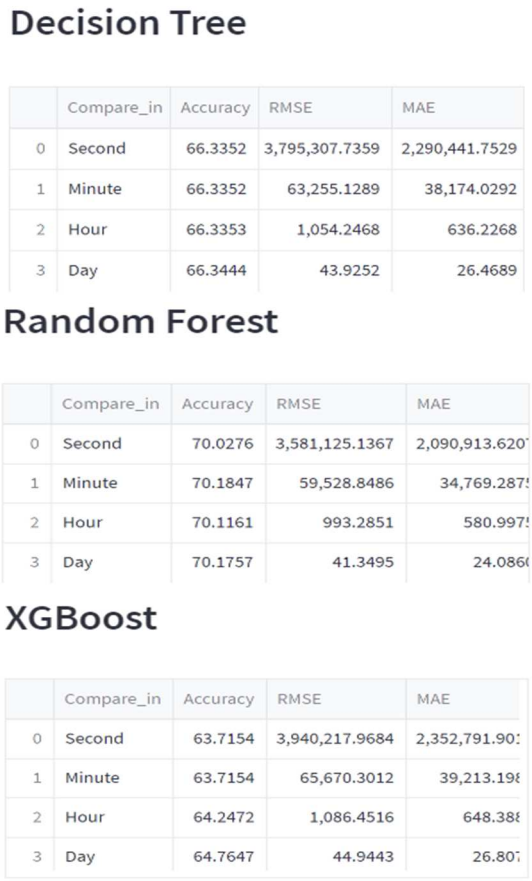

Fig. 6 Evaluation result of ML techniques

According to Table III, the values of model accuracy and evaluation metrics scores, including MAE and RMSE, obtained vary across approaches.

- Accuracy: The RF algorithm achieves the highest accuracy score, 70.04. The second highest score is obtained by the DT algorithm, which is 66.34, while the third highest is the XGBoost algorithm, which is 64.76.
- MAE: The RF algorithm achieves the lowest MAE score, 24.17. The DT algorithm, 26.47, obtains the second lowest score, while the third lowest is the XGBoost algorithm, 26.81.

- RMSE: The RF algorithm achieves the lowest RMSE score, 41.45. The second lowest score is obtained by the DT algorithm, 43.93, while the third lowest is the XGBoost algorithm, 44.94.

In general, the RF algorithm outperforms all other approaches.

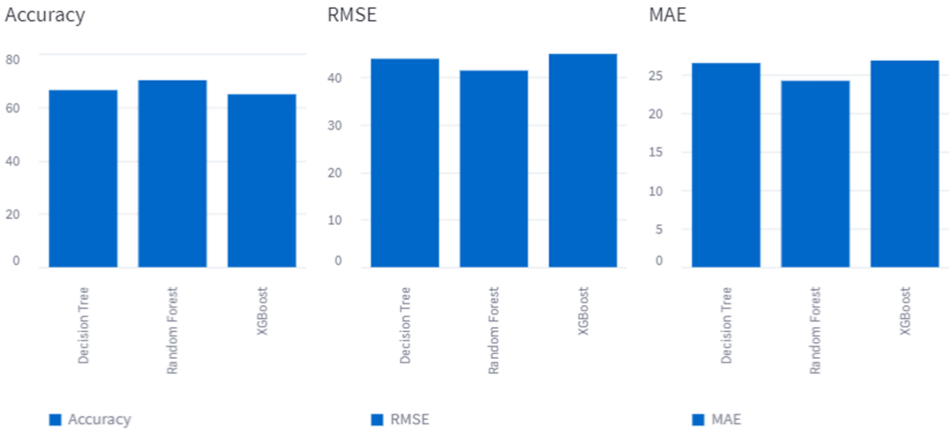

Fig. 7 Evaluation results

TABLE III
EVALUATION RESULTS

ML Technique	Accuracy	Evaluation metric	
		MAE	RMSE
DT Regressor	66.34	26.47	43.93
RF Regressor	70.04	24.17	41.45
XGBoost Regressor	64.76	26.81	44.94

IV. CONCLUSION

ML technology has been proven to be an effective tool in prediction across several domains, including customer service. Multiple ML approaches are explored in this paper, and three ML techniques have been chosen to be implemented: DT, RF, and XGBoost. The evaluation metrics applied are MAE and RMSE. Based on the preliminary experimental evaluation, RF performs best among applied ML techniques. As for future work, we intend to conduct a more extensive evaluation using the F1 score, Precision, and Recall metrics. In addition, a dashboard will be created for better visualization and analysis.

REFERENCES

[1] Y. Ayodeji, H. Rjoub, and H. Özgit, "Achieving sustainable customer loyalty in airports: The role of waiting time satisfaction and self-service technologies," Technology in Society, vol. 72, p. 102106, Feb. 2023, doi: 10.1016/j.techsoc.2022.102106.

[2] L. Jenneboer, C. Herrando, and E. Constantinides, "The Impact of Chatbots on Customer Loyalty: A Systematic Literature Review," Journal of Theoretical and Applied Electronic Commerce Research, vol. 17, no. 1, pp. 212–229, Jan. 2022, doi: 10.3390/jtaer17010011.

[3] H. Ng, M. S. Jalani, T. T. V. Yap, and V. T. Goh, "Performance of Sentiment Classification on Tweets of Clothing Brands," Journal of Informatics and Web Engineering, vol. 1, no. 1, pp. 16–22, Mar. 2022, doi: 10.33093/jiwe.2022.1.1.2.

[4] A. A. Zaveri, R. Mashood, S. Shehmir, M. Parveen, N. Sami, and M. Nazar, "AIRA: An Intelligent Recommendation Agent Application for Movies," Journal of Informatics and Web Engineering, vol. 2, no. 2, pp. 72–89, Sep. 2023, doi: 10.33093/jiwe.2023.2.2.6.

[5] N. Ahmed Mahoto, R. Iftikhar, A. Shaikh, Y. Asiri, A. Alghamdi, and K. Rajab, "An Intelligent Business Model for Product Price Prediction Using Machine Learning Approach," Intelligent Automation & Soft Computing, vol. 29, no. 3, pp. 147–159, 2021, doi:10.32604/iasc.2021.018944.

[6] P. Ambardekar, A. Jamthe, and M. Chincholkar, "Predicting defect resolution time using cosine similarity," 2017 International Conference on Data and Software Engineering (ICoDSE), Nov. 2017, doi: 10.1109/icodse.2017.8285884.

[7] M. F. Bergeron et al., "Machine Learning in Modeling High School Sport Concussion Symptom Resolve," Medicine & Science in Sports & Exercise, vol. 51, no. 7, pp. 1362–1371, Jan. 2019, doi: 10.1249/mss.0000000000001903.

[8] F. Al-Hawari and H. Barham, "A machine learning based help desk system for IT service management," Journal of King Saud University - Computer and Information Sciences, vol. 33, no. 6, pp. 702–718, Jul. 2021, doi: 10.1016/j.jksuci.2019.04.001.

[9] S. Fuchs, C. Drieschner, and H. Wittges, "Improving Support Ticket Systems Using Machine Learning: A Literature Review," Proceedings of the 55th Hawaii International Conference on System Sciences, 2022, doi: 10.24251/hicss.2022.238.

[10] C. I. L. Sharon and V. Suma, "Predictive Analytics in IT Service Management (ITSM)," Data Mining and Machine Learning Applications, pp. 175–193, Jan. 2022, doi:10.1002/9781119792529.ch7.

[11] "Improving the Prediction Resolution Time for Customer Support Ticket System," Journal of System and Management Sciences, Dec. 2022, doi: 10.33168/jsms.2022.0601.

[12] L. S. B. Pereira, R. Pizzio, S. Bonho, L. M. F. De Souza, and A. C. A. Junior, "Machine Learning for Classification of IT Support Tickets," 2023 International Conference On Cyber Management And Engineering (CyMaEn), Jan. 2023, doi:10.1109/cymaen57228.2023.10051041.

[13] M. F. bin Harunasir, N. Palanichamy, S.-C. Haw, and K.-W. Ng, "Sentiment Analysis of Amazon Product Reviews by Supervised Machine Learning Models," Journal of Advances in Information Technology, vol. 14, no. 4, pp. 857–862, 2023, doi:10.12720/jait.14.4.857-862.

[14] C. Miloudi, L. Cheikhi, A. Abran, and A. Idri, "Maintenance Effort Estimation for Open-Source Software: Current trends," in *CEUR Workshop Proceedings*, 2022.

[15] M. Mamedov, K. Vorontsova, E. Treshcheva, and I. Itkin, "Building a Reusable Defect Resolution Time Prediction Model Based on a Massive Open-Source Dataset: An Industrial Report," 2021 IEEE International Conference on Artificial Intelligence Testing (AITest), Aug. 2021, doi: 10.1109/aitest52744.2021.00023.

[16] M. K. Yucel and A. Tosun, "Measuring Bug Reporter's Reputation and Its Effect on Bug Resolution Time Prediction," 2022 7th International Conference on Computer Science and Engineering (UBMK), Sep. 2022, doi: 10.1109/ubmk55850.2022.9919454.

[17] Y. Yang and W. Chen, "Taiga: performance optimization of the C4.5 decision tree construction algorithm," Tsinghua Science and

Technology, vol. 21, no. 4, pp. 415–425, Aug. 2016, doi:10.1109/tst.2016.7536719.
[18] P. Mahajan, S. Uddin, F. Hajati, and M. A. Moni, “Ensemble Learning for Disease Prediction: A Review,” Healthcare, vol. 11, no. 12, p. 1808, Jun. 2023, doi: 10.3390/healthcare11121808.
[19] J.-J. Liu and J.-C. Liu, “Permeability Predictions for Tight Sandstone Reservoir Using Explainable Machine Learning and Particle Swarm Optimization,” Geofluids, vol. 2022, pp. 1–15, Jan. 2022, doi:10.1155/2022/2263329.
[20] D. Pfahl, S. Karus, and M. Stavnycha, “Improving expert prediction of issue resolution time,” Proceedings of the 20th International Conference on Evaluation and Assessment in Software Engineering, Jun. 2016, doi: 10.1145/2915970.2916004.
[21] M. S. Rahaman, Y. Ren, M. Hamilton, and F. D. Salim, “Wait Time Prediction for Airport Taxis Using Weighted Nearest Neighbor Regression,” IEEE Access, vol. 6, pp. 74660–74672, 2018, doi:10.1109/access.2018.2882580.
[22] D. Zuev, A. Kalistratov, and A. Zuev, “Machine Learning in IT Service Management,” Procedia Computer Science, vol. 145, pp. 675–679, 2018, doi: 10.1016/j.procs.2018.11.063.
[23] A. I. Kyritsis and M. Deriaz, “A Machine Learning Approach to Waiting Time Prediction in Queueing Scenarios,” 2019 Second International Conference on Artificial Intelligence for Industries (AI4I), Sep. 2019, doi: 10.1109/ai4i46381.2019.00013.
[24] E. Benevento, D. Aloini, N. Squicciarini, R. Dulmin, and V. Mininno, “Queue-based features for dynamic waiting time prediction in emergency department,” Measuring Business Excellence, vol. 23, no. 4, pp. 458–471, Nov. 2019, doi: 10.1108/mbe-12-2018-0108.
[25] G. Bejarano, A. Kulkarni, X. Luo, A. Seetharam, and A. Ramesh, “DeepER: A Deep Learning based Emergency Resolution Time Prediction System,” 2020 International Conferences on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData) and IEEE Congress on Cybermatics (Cybermatics), Nov. 2020, doi: 10.1109/ithings-greencom-cpscom-smartdata-cybermatics50389.2020.00090.
[26] A. Pak, B. Gannon, and A. Staib, “Predicting waiting time to treatment for emergency department patients,” International Journal of Medical Informatics, vol. 145, p. 104303, Jan. 2021, doi:10.1016/j.ijmedinf.2020.104303.
[27] H. Hijry and R. Olawoyin, “Predicting Patient Waiting Time in the Queue System Using Deep Learning Algorithms in the Emergency Room,” International Journal of Industrial Engineering and Operations Management, vol. 03, no. 01, pp. 33–45, Oct. 2021, doi:10.46254/j.ieom.20210103.
[28] J. Azimjonov and T. Kim, “Stochastic gradient descent classifier-based lightweight intrusion detection systems using the efficient feature subsets of datasets,” Expert Systems with Applications, vol. 237, p. 121493, Mar. 2024, doi: 10.1016/j.eswa.2023.121493.
[29] Y. Arouri and M. Sayyafzadeh, “An adaptive moment estimation framework for well placement optimization,” Computational Geosciences, vol. 26, no. 4, pp. 957–973, Feb. 2022, doi:10.1007/s10596-022-10135-9.
[30] K. Gayathri Devi, K. Balasubramanian, C. Senthilkumar, and K. Ramya, “Accurate Prediction and Classification of Corn Leaf Disease Using Adaptive Moment Estimation Optimizer in Deep Learning Networks,” Journal of Electrical Engineering & Technology, vol. 18, no. 1, pp. 637–649, Aug. 2022, doi: 10.1007/s42835-022-01205-0.
[31] Z. W. Wang, J. W. Lu, and J. Zhou, “Learning Adaptive Gradients for Binary Neural Networks,” *Tien Tzu Hsueh Pao/Acta Electronica Sinica*, vol. 51, no. 2, 2023, doi: 10.12263/DZXB.20211084.
[32] J. Ma, X. Zeng, X. Xue, and R. Deng, “Metro Emergency Passenger Flow Prediction on Transfer Learning and LSTM Model,” Applied Sciences, vol. 12, no. 3, p. 1644, Feb. 2022, doi: 10.3390/app12031644.
[33] J. Schad, R. Sambasivan, and C. Woodward, “Predicting help desk ticket reassignments with graph convolutional networks,” Machine Learning with Applications, vol. 7, p. 100237, Mar. 2022, doi:10.1016/j.mlwa.2021.100237.
[34] M.F. Claudio, S.M. Peres, “Incident management process enriched event log Data Set”, UCI Machine Learning Repository. Retrieved January 28, 2023, from https://archive.ics.uci.edu/ml/datasets/Incident%2Bmanagement%2Bprocess%2Benriched%2Bevent%2Blog#.