Vol 6 No 2 (2024) E-ISSN: 2682-860X

International Journal on Robotics, Automation and Sciences

Performance Evaluation of Machine Learning Techniques on Resolution Time Prediction in Helpdesk Support System

Tong-Ern Tai, Su-Cheng Haw*, Wan-Er Kong and Kok-Why Ng

***Abstract* – Estimating incident resolution times accurately is critical to maintaining an effective resource allocation for customer service. In order to meet this need, this paper explores machine learning techniques widely applied in the Resolution Time Prediction and identify the performance of chosen approaches via benchmarking dataset. The proposed method starts with data preprocessing, such as removing outliers and missing values and determining any irregularities in the resolution times distribution. Subsequently, we automatically choose the most relevant features using various statistical techniques. As the last stage of our prediction pipeline, we will apply different machine learning approaches the dataset to find the effectiveness of model and conclude the best technique based on the model accuracy and model fitting time. By applying this strategy, we hope to gain a better understanding of the factors affecting incident resolution times, which will eventually result in better resource allocation and planning for customer support operations.**

Keywords—Resolution Time Prediction, Machine Learning, Ticketing System, Customer Service, Recommender System.

I. INTRODUCTION

Businesses always highly value comprehensive customer service, and customers have the greatest expectations regarding swift resolution. Whenever customers file a service ticket for assistance, they expect a prompt, clear, and practical answer. In order to provide customers with a great experience and maintain the company's reputation, customer service representatives work hard to address issues assigned to them while handling a large number of requests every day. Ayodeji et al. assert that prompt service delivery can boost client happiness and loyalty [1], which in turn can promote repeat and future business [2,3].

The resolution time prediction is the projection of the duration required for a customer support agent to address a customer's problem, question, or grievance. Apart from that, responding to customer inquiries as quickly as feasible would also greatly enhance customer loyalty. The goal of automating this process by estimating the time needed to handle specific issues based on cases similar to previous ones has been made possible by developing cutting-edge technologies such as Artificial Intelligence (AI) and machine learning (ML). ML advancements allow for the automation of ticket classification, which in turn enables the prediction of case resolution times [4,5].

Several services, including banking, meal preparation, tickets, and gadget maintenance, function on a take-turn basis and demand a lengthy wait from their clients. These sectors have accumulated extensive diversified data and case studies throughout

*Corresponding Author email: sucheng@mmu.edy.my

Tong-Ern Tai is with Faculty of Computing and Informatics, Multimedia University, Cyberjaya, Malaysia (e-mail: tai.tong.ern@ student.mmu.edu.my).

Su-Cheng Haw is with Faculty of Computing and Informatics, Multimedia University, Cyberjaya, Malaysia (e-mail: sucheng@mmu.edu.my).

Wan-Er Kong is with Faculty of Computing and Informatics, Multimedia University, Cyberjaya, Malaysia (e-mail: kong.wan.er@ student.mmu.edu.my).

Kok-Why Ng is with Faculty of Computing and Informatics, Multimedia University, Cyberjaya, Malaysia (e-mail: kwng@mmu.edu.my).

International Journal on Robotics, Automation and Sciences (2024) 6, 2:59-68
https://doi.org/10.33093/ijoras.2024.6.2.9
Manuscript received: 19 Apr 2024 | Accepted: 24 Jun 2024 | Published: : 30 Sep 2024
Published by MMU PRESS. URL: http://journals.mmupress.com/ijoras

time, allowing the computer model to forecast the amount of time to resolve an issue. If customers are unaware of how long it will take to resolve their issue, they will not have a positive experience. Resolution Time Prediction (RTP) is hence crucial.

When it comes to the Resolution Time Prediction system, ML is a crucial component that makes the system work and effective [6,7]. The Resolution Time Prediction system frequently apply several types of ML methods, such as gradient boosting [8], Random Forest (RF) [9], and Decision Tree (DT) [10,11]. If the system receives a new ticket, the trained ML model automatically determines how long it will take to respond to this question.

This paper aims to answer the following research questions:

1. What are the factors that contributed to the support of the ticketing system?
    - This problem may be resolved after researching the variables which contribute greatly in efficiency in different support of the ticketing system.
2. How do we propose a new prediction model to predict resolution time?
    - A solid foundation of ML understanding and related knowledge is essential to answer this research question. We need to conduct extensive research in recent research papers which involve resolution time prediction.
3. How to conduct model evaluation for the proposed method?
    - The proposed method can be evaluated using common evaluation metrics in resolution time prediction systems, such as RMSE, MSE and MAE.

II. Material And Method

A. *Background on ML techniques*

Within the more prominent topic of artificial intelligence, machine learning (ML) is a dynamic way to continuously improve outcomes. ML can learn through data learning without direct human interaction. Based on the training data, these algorithms can build models that can make predictions on their own [6]. The resulting models allow for a variety of functions, including decision-making and forecasting, all without requiring human oversight [12].

There are several types of ML techniques: supervised ML, unsupervised ML, semi-supervised ML, and reinforcement Learning [13,14]. This research study will only focus on the supervised ML method. There are two primary types of supervised learning: classification and regression.

Classification is the process of predicting discrete labels or classes for categorical target data [6]. Classification algorithms acquire the capability of assigning input attributes to one of the predefined classes. One of the examples of classification tasks is determining the spam email. There are many types of categorization algorithms, which include K-Nearest Neighbors (KNN), Naive Bayes, RF, DT, Support Vector Machine, and Logistic Regression. On the other hand, regression is a method which anticipates numerical values for continuous target variables [15]. For example, this may involve anticipating a product's sales or estimating the cost of a property based on features.

This study applies several ML methods: DT, RF, and Extreme Gradient Boosting (XGBoost).

B. *Decision Tree (DT)*

DT is a non-parametric supervised learning approach commonly used in regression and classification tasks. This technique is particularly useful in decision analysis and helps the user identify a strategy most likely to attain a goal [16]. For instance, real life tasks that require DT include predicting potential event outcomes and making financial decisions.

A DT is a hierarchical model which functions based on a tree structure. A DT consists of a root node, branches, internal nodes, and leaf nodes. The structure of the DT is similar to a flowchart, with internal nodes standing in for attribute testing, branching for test results, and leaf nodes for class names. It commences at a root node, utilizes conditional control statements which consists of a different combination of variables to produce output, and concludes with decisions made at the leaf nodes. Figure 1 shows a simple structure of a DT [17]. In the model training stage, a DT will consider measures such as entropy or Gini impurity to decide the best attribute for splitting the data. These measures evaluate the level of random or disorder of the subgroups. Among all attributes, the attribute which has a balance of maximum information acquisition and minimizes impurity reduction after the data split will be chosen as the output.

In the real-life industry, a DT as a decision-making tool is frequently applied in operation management and operation research. In many use cases, this method is integrated with a probability model and selected as the optimal model in choosing real-time decisions when the information is limited.

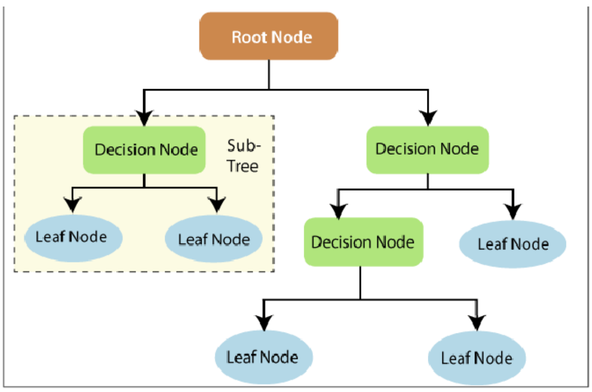

FIGURE 1. DT structure.

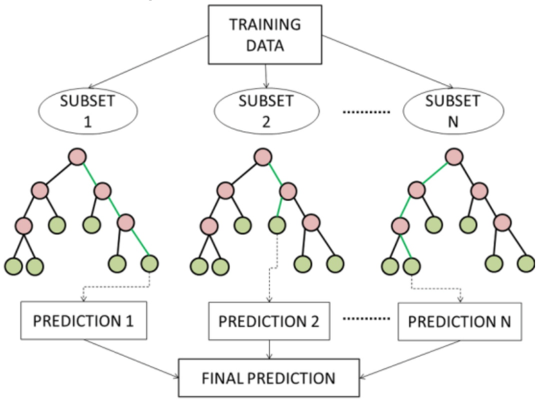

FIGURE 2. Fundamental concept of a RF model.

C. Random Forest (RF)

In ML, the RF technique builds a collection of DTs to produce a forecast or result that is more accurate [18]. This "forest" comprises several DTs trained using the bagging technique [19]. The idea of bagging is combining different learning models to improve the final product. It is well known for being flexible and easy to use, and it works well in solving both regression and classification problems. Figure 2 demonstrates the fundamental concept of an RF model by [20].

As a model made up of DTs, the RF shares nearly identical hyperparameters with both DTs and bagging classifiers. The use of a classifier-class of RF eliminates the necessity for combining a DT and a bagging classifier. Moreover, RF's regressor technique broadens its applicability to regression tasks. The inclusion of more randomization throughout tree growth is one of the RF's distinguishing features. Generally, it optimizes model performance by evaluating the best feature inside a randomly selected subset rather than searching for the most significant feature for node splitting. As such, only a random subset of features is considered for node splitting in a RF classifier. In addition, trees can be made even more diverse by using random thresholds for every characteristic instead of aiming for ideal thresholds.

Although DTs are the building blocks of both RF and DT models, some significant differences exist. A DT creates a collection of rules used to make predictions when it receives a training dataset with features and labels. For example, the DT can develop rules to estimate click probability when sufficient information is provided. In contrast, the RF approach randomly selects observations and features to build a 'forest' of DTs. The trees' output obtained will be averaged as the final result. Besides that, RF also has overfitting issues as it is constructed with "deep" DTs. To overcome this problem, RF creates smaller trees by mixing random chosen feature subsets. However, this approach could result in a lengthy calculation time if the number of trees created in RF is too large.

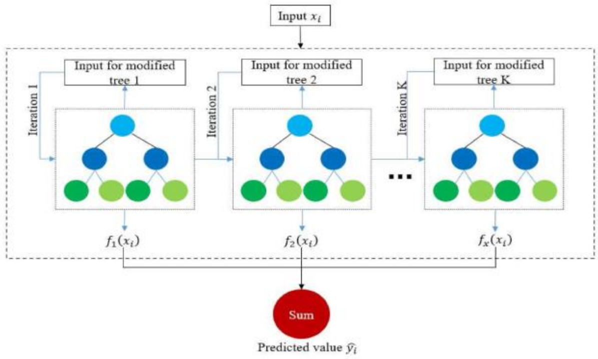

FIGURE 3. Fundamental structure of XGBoost.

D. Extreme Gradient Boosting (XGBoost)

XGBoost is a class of ensemble ML algorithms capable of solving both classification and regression tasks [21]. This method is known for its efficiency and accuracy. It belongs to the class of ensemble learning techniques known as boosting algorithms. In simple words, this algorithm combines the predictions of multiple weaker learners to gain results. Figure 3 shows the fundamental structure of an XGBoost by [22].

In training the model, this algorithm methodically constructs simple and brief DTs. Due to its natural bias, every tree is referred to as a "weak learner" [23]. In order to address forecast mistakes caused by previous models, these trees are gradually added to the ensemble and modified. Model fitting involves an optimization approach based on gradient descent and any arbitrary differentiable loss function. This procedure, appropriately called "gradient boosting," can minimize the loss gradient during model fitting.

Gradient descent is an optimization strategy that minimizes a cost function by iteratively changing the model's parameters depending on the gradients of errors [24]. The method also introduces the concept of "gradient boosting with DTs. This method computes the relevance of each DT added to the ensemble in order to lower the objective function. With the addition of a regularization term and advanced optimization techniques, XGBoost can enhance this approach in reaching a higher precision and effectiveness. Its ability to handle enormous datasets for a variety of ML applications accounts for its broad use and appeal.

E. Comparison of DT, RF, and XGBoost

The most simple approach is the DT algorithm among all techniques. Since this algorithm adopts an information-based approach, it excludes data preprocessing steps like data normalization and data scaling. However, this model is unstable since even a small change in the data will have a big effect on the structure of the DT. However, the RF method may counteract DT's tendency to overfit its training set, often producing an excellent model performance outcome. However, this approach cannot comprehend the results, the potential for overfitting, and the requirement to predefine the number of trees to be incorporated into the model. Regarding XGBoost, this method can speed up data processing time as it does not involve feature normalization. On the other hand, if the created trees are sufficiently deep with noisy data, this strategy may lead to overfitting, just as the RF algorithm.

In short, various ML approaches bring various benefits. There is no such type of ML approach that can fit all kinds of systems; hence, this paper will aim to select a suitable method.

F. Related Works

Borg & Boldt examined the possibility of learning algorithms to predict the time required for email response for customers and customer care representatives using RF [25]. This study involves two experiments with two distinct objectives. The purpose of the first experiment is to find out the possibility of forecasting how long a customer service representative would take to reply to an email. This study chose to focus RF as the learning algorithm and contrasted it with Random Guesser classifier using a uniform random guesser as baseline. The model is trained using a dataset that consists of e-mails delivered by the client in this stage, and the model will then predict the response time from the agent using this data. While the second experiment is identical to the first, the only difference is to forecast the time required for the client to reply to the agent. This experiment uses a dataset which consists of emails received by customer support. The standard evaluation metrics applied in this study included True Positives (TP), False Positives (FP), True Negatives (TN), and False Negatives (FN), and the evaluation metrics used are Accuracy, F1-score, and Area under ROC-curve (AUC). According to the results, it is possible to forecast the email response time of forecast customer and customer support agents. The proposed method achieves AUC of 0.85 and 0.90 for train and test set. These results show that this study can enhance customer service communication efficiency. Future work will involve assessing the practical application of the prediction. This evaluation would encompass two aspects: the degree of customer agent effectiveness and the degree of efficiency by assigning cognitive and emotional loads to different agents based on their experience.

Alsac et al. predicted the amount of time anticipated to handle IT support requests using supervised ML algorithms [26]. This research is predicated on a large dataset that includes over 17,000 tickets from actual events. The researchers preprocess and change the input data using data science approaches, and then apply different supervised ML algorithms to build prediction models for ticket resolution time frames. The ML algorithms involved are Linear Regression, DTs Regression, RF Regression, SVM Regression, and Multiple Regression algorithms. There are three different experiment designs in this study. First, the model is trained and tested using the complete dataset. The dataset is split into two sets in a 70:30 ratio, with 30% for testing and 70% for training. Establishing a 95% confidence interval is the last stage in understanding the Ordinary Least Squares (OLS) table. In order to evaluate the ML techniques, several evaluation metrics are involved, which include MAE, MSE, and MAPE. The outcomes of this research study demonstrate that different supervised ML algorithms perform substantially on this task. DTs and RF Regression stand out among all methods due to their high performance overall. In the future, more classification and regression techniques will be applied to informative data, such as the textual and graphical data related to the ticket.

In a separate study, Haw et al. targeted using predictive analytics technology to forecast the resolution time anticipated to tackle a specific task to provide customers with an approximate idea of the amount of time required to resolve their issue [6]. This research starts with data preprocessing by performing one hot encoding on the categorical variables and feature selection using a variety of statistical methods, including chi square correlation coefficient, entropy and point biserial correlation coefficient. The prediction pipeline of this research involves a combination of classification and regression models. The classification model applied in this research is a DT classifier with a one-vs-rest multiclass classification strategy, while the regression model used includes RF, Neural Network (NN) and ADA boost. The evaluation metric used to measure the performance of the ML models in this paper is RMSE. According to the findings, NN has the worst performance, while RF has the best performance. This may be due to the insufficient data diversity. On the other hand, there is a notable improvement in performance in cases where RF is applied with extremity features, in precise, attention. The disadvantage of this research is the application of low data diversity. In future work, it is possible to extend this work by applying data which is larger in size and higher in diversity to gain a higher performance in the accuracy of the prediction model.

Gerunov focused on finding the factors which contribute to excessive delays in IT customer support by analyzing data from a process-aware information system [27]. The effectiveness of the proposed solution is measured by assessing the performance of various manually trained state-of-the-art benchmark models against automated model training using the H2O framework. The standard benchmark models applied in this research include Multiple Linear Regression, Artificial Neural Network (ANN), K-Nearest Neighbors, RF, Support Vector Machine (SVM) while the H2O framework dynamically fits and assesses models with a few algorithms which include Distributed RF, Extremely Randomized Trees, Generalized Linear Model (GLM) with regularization, XGBoost Gradient Boosting Machine, H2O Gradient Boosting Machine, Multi-layer ANN and a 2 Stacked Models which one consist every model trained as well as the best-in class model. Evaluation metrics used in this research are Mean Error, Root Mean Squared Error and Mean Absolute Error. According to the result, AutoML models have the highest performance. The best model is a stacked ensemble model, a combination of 100 different model predictions. AutoML models achieve the lowest RMSE and relatively low mean error and mean absolute error. The most effective individual model in AutoML models is the Gradient Boosting Machine (GBM), which shows high performance very close to the stacked ensemble models. As GBM requires a notably lower computation power and enhancement in model explainability, this model is concluded as the best performer. The results prove that the automated ML models outperform the benchmark models. Further research will clarify the uses of this methodology for assessing and simulating company operations that extend beyond customer service and explore the real-world applications of AutoML.

Jiri et al. offered a novel function that applies the modification history of bug reports and aims to overcome the performance differences that exist in industrial software systems using predictive models created with Open Source Software-derived factors [28]. To be precise, this paper emphasizes efforts to build a predictive model for bug report resolution times under eBay's software ecosystem. The statistical ML methods applied in this paper include RF and eXtreme Gradient Boosting (XGBoost). A comparative performance evaluation between the proposed method and models built in previous studies is also done using several evaluation metrics, including precision, recall, and F1 score. The empirical findings of this research show a significant improvement in the accuracy of model prediction as the proposed method achieves a higher accuracy score compared with the models built in previous studies, which is 29% in precision, 34% in recall, and 33% in the F1 score. This research demonstrates the extent to which the prediction model functions to aid developers in eBay's software ecosystem fulfill specified bug report resolution timelines. Future research could optimize the alerts' effectiveness and further lower the generality of OOSLA bug reports.

G. Proposed Framework

The proposed framework flow is shown using a flowchart. Figure 4 demonstrates the proposed framework workflow. Four selections will be provided in total: DT, RF, XGBoost, and Ensemble Method. DT, RF, and XGBoost techniques have been discussed above, while the ensemble method is built by create a stacking regressor with base and meta regressors. The chosen base regressors are DT, RF, XGBoost while the meta regressor is Linear Regression. All techniques will be integrated to the ensemble method using scikit learn library. The selected ML technique will be adopted to perform a prediction after fitting a cleaned dataset. Only one dataset will be used in this research study. After completing the model fitting, model evaluation will take place. Users can check the accuracy of the ML technique using several evaluation metrics, which include RMSE and MAE. According to the outcome, the user can conclude which method is the most suitable to perform the prediction task and which has the highest accuracy among all.

Choosing the ML Techniques

Users can choose one from the four techniques provided, which are DT, RF, XGBoost, and the ensemble method of all the stated techniques. By default, the DT model is chosen. The user can choose other methods afterwards.

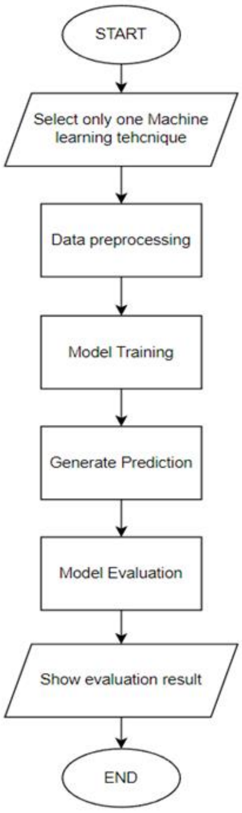

FIGURE 4. Flowchart of proposed framework.

H. Dataset

The dataset employed in this paper is a public dataset named Analyze Helpdesk tickets dataset. This dataset consists of 100k data records and 10 variables for each data record. The dataset's attributes are explained thoroughly, along with possible values in Table 1.

TABLE 1. Description of attributes on selected dataset.

No	Attribute	Description
1	ticket	A specific identifier for each ticket
2	requestor	A specific identifier of requestor
3	RequestorSeniority	The seniority level of requestor
4	ITOwner	The owner of IT
5	FiledAgainst	The type of file against
6	TicketType	The type of ticket
7	Severity	The level of critical or serious of the ticket
8	Priority	The level of urgent or important of the ticket
9	daysOpen	Length of period which the ticket has been unresolved or active
10	Satisfaction	Level of customer satisfaction of a resolved ticket

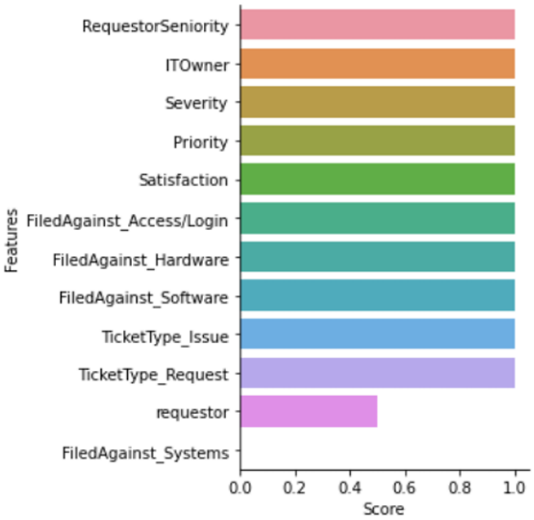

FIGURE 5. Flowchart of proposed framework.

I. Data Cleaning

Several aspects are checked in the data cleaning stage. Firstly, only resolved tickets are chosen in this dataset. All unresolved tickets are removed from the dataset. Next, some impossible values are removed from the dataset. For example, some tickets recorded the wrong resolution time, which is negative four days from the start time. The missing value of the dataset is also checked. Apparently, this dataset has no missing value.

Besides that, the outlier of the attribute 'daysOpen' is also checked. To avoid creating a negative impact in statistical power, these outliers will be removed.

There are also a few attributes that present categorical information. The one-hot-encoding technique removes the textual information, and only the numerical data is kept in the dataset.

Lastly, the data cleaning includes a feature selection using the Boruta feature selection. Some attributes do not contribute much to model accuracy. This can lower the required computational power and further raise the accuracy of the model. Figure 5 shows the Boruta score of each attribute.

After completing the data cleaning phase, the cleaned dataset is saved to a new CSV file and renamed as "cleaned_data.csv".

J. Model Training

Model training is essential for discovering the underlying links and patterns in the data to forecast or decide on new data. All available models in this prototype are adopted from the same library, Scikit-learn. This library is commonly used in research which involves resolution time prediction. The dataset used in this stage should be cleaned and split into two portions, with 80% and 20% for model training and testing, respectively.

K. Model Evaluation

Model evaluation is conducted via a few evaluation metrics, which are model accuracy score, RMSE, MSE and MAE. The test data for each method is set to be the same to reduce any unfairness in the model evaluation stage.

III. RESULTS AND DISCUSSION

According to the Boruta result, most information present in dataset contributes strongly to the support of the ticketing system, especially requestor seniority, IT Owner, severity and priority of ticket. A novel prediction model is built to predict resolution time by integrating the three ML methods in order to achieve diverse predictions as different might capture different patterns and provide a more accurate prediction. The model performance is recorded using different evaluation metrics and compared among each other to identify the best method for this paper. The best method is expected to achieve a balance in both aspects of model accuracy and model fitting time.

The higher the score for model accuracy, the more accurate the model can provide predictions. According to the accuracy score of all models, the best method is XGBoost. This method obtained 81.102384% in accuracy, surpassing other methods. The remaining models also achieve a rather excellent result but are slightly weaker than the XGBoost technique. The second best method is the Ensemble method, and followed by RF as well as DT. The ensemble method achieved 80.012906% accuracy while RF got 79.801571% accuracy. The model with the lowest accuracy is DT.

Figure 6 shows the accuracy score of each technique applied in this paper.

Besides the model accuracy score, several evaluation metrics were also applied to evaluate the model performance. Unlike the model accuracy score, the lower the RMSE, MSE and MAE score gets, the more accurate the model can generate predictions. Overall, XGBoost achieves the best result, which is 2.5081 in RMSE, 6.2907 in MSE and 1.6593 in MAE. The second best method is Ensemble Method, which obtained 2.5794 in RMSE, 6.6534 in MSE, and 1.6916 in MAE. The third best method is RF technique, which got 2.5930 in RMSE, 6.7237 in MSE and 1.6862 in MAE, while the fourth best method is DT, which obtained 2.6728 in RMSE, 7.1443 in MSE and 1.7221 in MAE. The difference between the evaluation scores may not seem significant, but any small difference can have a big impact on the final delivered performance. Figure 7 shows the ML evaluation result.

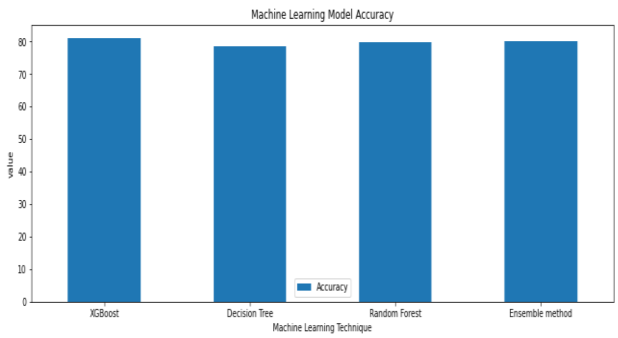

FIGURE 6. ML model accuracy score.

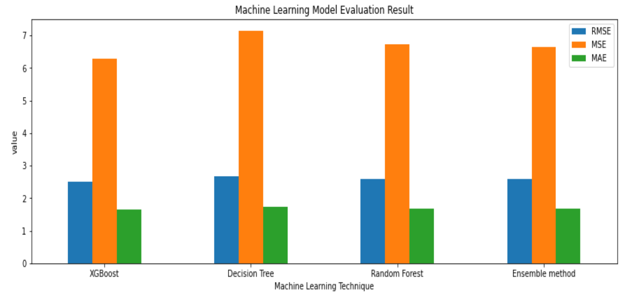

FIGURE 7. ML evaluation result.

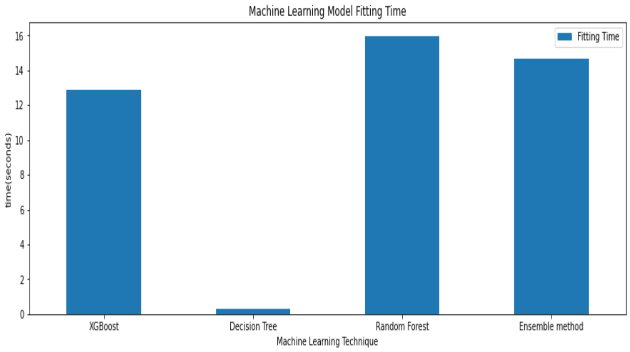

FIGURE 8. ML model training time.

TABLE 2. ML evaluation result.

ML Technique	Accuracy	RMSE	MSE	MAE
XGBoost	81.1023	2.5081	6.2907	1.6593
DT	78.5380	2.6728	7.1443	1.7221
RF	79.8015	2.5930	6.7237	1.6862
Ensemble Method	80.0129	2.5794	6.6534	1.6916

Table 2 presents the model accuracy score followed by other evaluation metrics scores.

Aside from the model prediction accuracy, model fitting time also plays an essential role in evaluating model performance. A shorter model fitting time under the same computational power environment indicates better model efficiency and higher user satisfaction. A cross-validation technique is also employed in this stage to let the training and validation processes be parallelized to reduce the total wall-clock time and ensure a reliable estimate of model performance. The result of model training time is presented in Figure 8.

From the aspect of model fitting time, the best method is DT. This technique obtained a fitting time that was significantly shorter than that of other methods, which was 0.2854 seconds in precision. The second best method is XGBoost, which got 12.8610 seconds. The third best method is Ensemble Method, which only requires 14.6554 seconds for model fitting, while the fourth best method is RF which requires 15.9628 seconds. The average fitting time of all methods is recorded in Table 3.

TABLE 3. ML model fitting time.

No	ML Technique	Model Fitting Time (in seconds)
1	XGBoost	12.8610
2	DT	0.2854
3	RF	15.9628
4	Ensemble Method	14.6554

Upon comparing the evaluation scores across several techniques, it is often discovered that the RMSE score obtained by each approach is typically higher than the MAE score. Since RMSE and MAE both assess the average magnitude of mistakes in the predicted values, there is a small difference in the calculation of these metrics. RMSE is able to penalize greater errors more strongly due to the squaring process.

The higher the RMSE scores, the wider the errors are expected between predicted and actual values. This can be more serious, particularly when the predictions deviate significantly from the actual values. On the other hand, MAE provides a more straightforward measure of the average magnitude of errors, regardless of direction. Thus, the result shows that the employed ML techniques can function in high accuracy but with a certain range of errors.

It is clear that the result concludes that the XGBoost model is the most promising model. This method achieved the best score in model accuracy evaluation and second best in model fitting evaluation. While RF offers significant advantages over single decision trees by using bagging to reduce variance, the XGBoost model outperforms RF in nature of building model sequentially, with each new model correcting errors made by the previous ones. This can reduce bias and further improve the model accuracy. Moreover, the XGBoost technique stands out due to its incorporation of internal regularization techniques, successfully reducing the risk of overfitting. Compared to the ensemble method, as this model is built with these three models, combining several models can sometimes introduce conflicts in their predictive capabilities, leading to suboptimal performance. Although the model fitting time is not the best, the result is still acceptable. This comprehensive set of advantages positions XGBoost model as the best method out of all the techniques in this paper.

IV. Conclusion

This study has achieved its objective in exploring and analyzing ML techniques commonly employed in Customer Support Ticket System. A few ML methodologies were introduced through investigation, including DT, RF, and XGBoost. The main goal was to gain a nuanced understanding of their efficacy and applicability within this specific domain. A comprehensive literature analysis was carried out to enhance our understanding by highlighting the latest developments and prevalent patterns in the use of ML techniques in related fields. This review was the foundation, guiding the choice and accompanying detailed analysis of three different ML approaches. A prototype is also developed to physically present useful implementation of ML methods inside the ticketing system framework. This prototype provided insights into the relative efficacy of each ML technique by graphically representing its performance. The effectiveness of the methods is carefully assessed using performance evaluation metrics such as RMSE, MSE, and MAE. The results indicate that XGBoost performed exceptionally well, as seen by low RMSE, MSE, and MAE scores and an excellent model accuracy score. Although the model fitting time for XGBoost is not the shortest of all strategies, its performance is still excellent and validates its position as the best technique. As a result, it is unquestionably shown that XGBoost is the best option compared to other approaches, which maintain its effectiveness and accuracy in customer support ticket systems.

The future work of this research involves creating more interesting and insightful data visualization for users. This would help users conduct a deeper data analysis. In addition, more assessment measures may also be considered in this research to provide a wider data analysis for users.

ACKNOWLEDGMENT

This work is supported by the funding of TM R&D from the Telekom Malaysia, Malaysia (Ref MMUE/240068).

REFERENCES

[1] Y. Ayodeji, H. Rjoub and H. Ozgit, “Achieving sustainable customer loyalty in airports: The role of waiting time satisfaction and self-service technologies”, *Technology in Society*, vol. 72, no. 102106, 2023.
DOI: https://doi.org/10.1016/j.techsoc.2022.102106

[2] T.E. Tai, S.C. Haw, K.W. Ng, M. Al-Tarawneh and G.K. Tong, “Performance evaluation on resolution time prediction using machine learning techniques”, *JOIV: International Journal on Informatics Visualization*, vol. 8, no. 2, pp. 583-591, 2024.
DOI: https://doi.org/10.62527/joiv.8.2.2305

[3] 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, 2022.
DOI: https://doi.org/10.3390/jtaer17010011

[4] P. Ambardekar, A. Jamthe and M. Chincholkar, “Predicting defect resolution time using cosine similarity,” *International Conference on Data and Software Engineering (ICoDSE),* pp. 1-6, 2017.
DOI: https://doi.org/10.1109/ICODSE.2017.8285884

[5] 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, 2021.
DOI: https://doi.org/10.1016/j.jksuci.2019.04.001

[6] S.C. Haw, K. Ong, L.J. Chew, K.W. Ng, P. Naveen and E.A. Anaam, “Improving the prediction resolution time for customer support ticket system”, *Journal of System and Management Sciences*, vol. 12, no. 6, pp. 1-16, 2022.
DOI: https://doi.org/10.33168/JSMS.2022.0601

[7] L.S.B. Pereira, R. Pizzio, S. Bonho, L.M.F. De Souza and A.C. Arnoni Junior, “Machine learning for classification of IT support tickets,” *International Conference On Cyber Management And Engineering*, pp. 210-213, 2023.
DOI: https://doi.org/10.1109/CyMaEn57228.2023.10051041

[8] M.K. Yucel and A. Tosun, “Measuring bug reporter's reputation and its effect on bug resolution time prediction,” *7th International Conference on Computer Science and Engineerin*g, pp. 110-115, 2022.
DOI: https://doi.org/10.1109/UBMK55850.2022.9919454

[9] 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,” *IEEE International Conference on Artificial Intelligence Testing*, pp. 73-80, 2021.
DOI: https://doi.org/10.1109/AITEST52744.2021.00023

[10] H. Wu, L. Shi, C. Chen, Q. Wang and B. Boehm, “Maintenance effort estimation for open source software: a systematic literature review,” *IEEE International Conference on Software Maintenance and Evolution*, pp. 32-43, 2016.
DOI: https://doi.org/10.1109/ICSME.2016.87

[11] M.N. Omri and W. Mribah, “Towards an intelligent machine learning-based business approach”, *International Journal of Intelligent Systems and Applications*, vol. 14, no. 1, pp. 1-23, 2022.
DOI: https://doi.org/10.5815/ijisa.2022.01.01

[12] B.C. Holtom and T.C. Burch, “A model of turnover-based disruption in customer services”, *Human Resource Management Review*, vol. 26, no. 1, pp. 25-36, 2016.
DOI: https://doi.org/10.1016/j.hrmr.2015.09.004

[13] K.K. Jha, R. Jha, A.K. Jha, M.A.M. Hassan, S.K. Yadav and T. Mahesh, “A brief comparison on machine learning algorithms based on various applications: a comprehensive survey,” *IEEE International Conference on Computation System and Information Technology for Sustainable Solutions*, pp. 1-5, 2021.
DOI: https://doi.org/10.1109/CSITSS54238.2021.9683524

[14] J. Zhang, Y. Zeng and B. Starly, “Recurrent neural networks with long term temporal dependencies in machine tool wear diagnosis and prognosis,” *Discover Applied Sciences*, vol. 3, no. 442, 2021.
DOI: https://doi.org/10.1007/s42452-021-04427-5

[15] J. Xue, R. Birke, L. Y. Chen and E. Smirni, “Managing data center tickets: prediction and active sizing,” *IEEE/IFIP International Conference on Dependable Systems and Networks*, pp. 335-346, 2016.
DOI: https://doi.org/10.1109/DSN.2016.38

[16] N.A. Harun, S.H. Huspi and N.A. Iahad, “Question classification framework for helpdesk ticketing support system using machine learning,” *7th International Conference on Research and Innovation in Information Systems,* pp. 1-7, 2021.
DOI: https://doi.org/10.1109/ICRIIS53035.2021.9617077

[17] B.T. Jijo and A.M. Abdulazeez, “Classification based on decision tree algorithm for machine learning”, *Journal of Applied Science and Technology Trends*, vol. 2, no. 1, pp. 20 - 28, 2021.
DOI: https://doi.org/10.38094/jastt20165

[18] S. Shah, “Automatic ticket assignment using machine learning and deep learning techniques”. Master of Science in Data Analytics thesis, School of Computing, National College of Ireland, Ireland, 2021.
URL: https://norma.ncirl.ie/4416/

[19] S.P. Paramesh, C. Ramya and K.S. Shreedhara, “Classifying the unstructured IT service desk tickets using ensemble of classifiers,” *3rd International Conference on Computational Systems and Information Technology for Sustainable Solutions*, pp. 221-227, 2018.
DOI: https://doi.org/10.1109/CSITSS.2018.8768734

[20] G. Laudato, R. Oliveto, S. Scalabrino, A..R Colavita, L. De Vito, F. Picariello and I. Tudosa, “Identification of R-peak occurrences in compressed ECG signals,” *IEEE International Symposium on Medical Measurements and Applications*, pp. 1-6, 2020.
DOI: https://doi.org/10.1109/MeMeA49120.2020.9137207

[21] S. Fatima, A. Hussain, S. bin Amir, S.H. Ahmed and S.M.H. Aslam, “XGBoost and random forest algorithms: An in depth analysis”, *Pakistan Journal of Scientific Research*, vol. 3, no. 1, pp. 26–31, 2023.
DOI: https://doi.org/10.57041/pjosr.v3i1.946

[22] M. Zou, W.G. Jiang, Q.H. Qin, Y.C. Liu and M.L. Li, “Optimized XGBoost model with small dataset for predicting relative density of Ti-6Al-4V parts manufactured by selective laser melting”, *Materials*, vol. 15, no. 15, 5298, 2022.
DOI: https://doi.org/10.3390/ma15155298

[23] R.K. Dhanaraj, V. Ramakrishnan, M. Poongodi, L. Krishnasamy, M. Hamdi, K. Kotecha and V. Vijayakumar, “Random forest bagging and X-Means clustered antipattern detection from SQL query log for accessing secure mobile data”, *Wireless Communications and Mobile Computing*, vol. 2021, no.1, pp. 1-9, 2021.

DOI: https://doi.org/10.1155/2021/2730246

[24] R.Y. Sun, “Optimization for deep learning: an overview”, *Journal of the Operations Research Society of China*, vol. 8, pp. 249–294, 2020.
DOI: https://doi.org/10.1007/s40305-020-00309-6

[25] A. Borg, J. Ahlstrand and M. Boldt, “Predicting e-mail response time in corporate customer support”, *22nd International Conference on Enterprise Information Systems,* vol. 1, pp. 305-314, 2020.
DOI: https://doi.org/10.5220/0009347303050314

[26] M. Yıldız, A. Alsaç, T. Ulusinan, M.C. Ganiz and M.M. Yenisey, “IT support ticket completion time prediction,” *7th International Conference on Computer Science and Engineering*, pp. 198-203, 2022.
DOI: https://doi.org/10.1109/UBMK55850.2022.9919591

[27] A.A. Gerunov, “Forecasting customer support resolution times through automated machine learning”, *Economics and Management*, vol. 19, no. 2, pp. 1-11, 2022.
URL:http://em.swu.bg/images/SpisanieIkonomikaupload/SpisanieIkonomika2022/1.1.%20AGerunov_ITSupport_AutoML_20221129.pdf

[28] J. Gesi, H. Wang, B. Wang, A. Truelove, J. Park and I. Ahmed, “Out of time: a case study of using team and modification representation learning for improving bug report resolution time prediction in eBay”, *Information and Software Technology*, unpublished.
DOI: https://doi.org/10.2139/ssrn.4571372