

Predicting Incident Duration Based on Machine Learning Methods

Zainab Ali Mohammed¹, Mohammed Najm Abdullah², Imad Hussain Al-Hussaini³

¹Computer Engineering Dept, University of Technology, Baghdad, Iraq
Zainab_ali1988@yahoo.com

²Computer Engineering Dept, University of Technology, Baghdad, Iraq
mustafamna@yahoo.com

³University of Baghdad, Baghdad, Iraq
imadhussaini@yahoo.com

Abstract— Traffic incidents don't only cause various levels of traffic congestion but often contribute to traffic accidents and secondary accidents, resulting in substantial loss of life, economy, and productivity loss in terms of injuries and deaths, increased travel times and delays, and excessive consumption of energy and air pollution. Therefore, it is essential to accurately estimate the duration of the incident to mitigate these effects. Traffic management center incident logs and traffic sensors data from Eastbound Interstate 70 (I-70) in Missouri, United States collected during the period from January 2015 to January 2017, with a total of 352 incident records were used to develop incident duration estimation models. This paper investigated different machine learning (ML) methods for traffic incidents duration prediction. The attempted ML techniques include Support Vector Machine (SVM), Random Forest (RF), and Neural Network Multi-Layer Perceptron (MLP). Root mean squared error (RMSE) and Mean absolute error (MAE) were used to evaluate the performance of these models. The results showed that the performance of the models was comparable with SVM models slightly outperforms the RF, and MLP models in terms of MAE index, where MAE was 14.23 min for the best-performing SVM models. Whereas, in terms of the RMSE index, RF models slightly outperformed the other two models given RMSE of 18.91 min for the best-performing RF model.

Index Terms— Incident Duration, Neural Network Multi-Layer Perceptron, Random Forest, Support Vector Machine.

I. INTRODUCTION

Traffic congestion arises when the traffic demand on the highway surpasses its usable capacity. There are two forms of congestion which are: recurrent and non-recurrent. Recurrent congestion, during peak hours, is associated with the physical configuration of the highway, which means that it is mainly caused by high traffic volume in the capacity of the roadway. Moreover, non-recurrent traffic congestion is caused by unplanned events on the highway such as incidents, stranded vehicles, public manifestations, weather, and work zones. Since highway work zones associated with patching, flooring, lane marking, rubble removal, and weeding are followed by temporary reduction of capacity on the highway, and the congestion caused by them can be extremely high portion of the whole traffic congestion [1], [2]. Although non-recurrent congestion is difficult to predict as a result of its random nature, the researches on impact and duration of the traffic incidents are still one of the main focuses for the traffic operators due to the serious social and economic losses generated [3]. Thus, different studies have been initiated to establish mitigation strategies which minimize non-recurrent congestion due to parkway incidents, where traffic incidents

are the main causes of non-recurrent congestion that cause travel delay and can lead to secondary crashes. Therefore, accurate estimation of traffic incident duration plays an essential role to mitigate these effects [4], [5]. To assist a timely response, traffic management centers build workflows are consisting of data collection, analyzing it and implementing the chosen strategy, constantly the usage of updated statistics to monitor traffic, publish information and control incident response resources [6]. The purpose of this study is to investigate various machine learning methods to predict the duration of the incident.

The remainder of this paper is structured as follows: Section II. introduces a review of previous researches on the prediction of incidents duration. Section III. provides an overview of the duration of the incident and its phases. A methodology of the proposed models with the three machine learning methods, namely Support Vector Machine (SVM), Random Forest (RF), Neural Network Multi-Layer Perceptron (MLP) that used to develop the incident duration prediction models is shown in Section IV. Study area details are given in section V. Moreover, the results obtained from the three models are discussed in Section VI. Finally, the conclusion is provided in Section VII.

II. RELATED WORKS

H. Cong, *et al.* [7], J. Tang, *et al.*[8], and X. Li, *et al.* [9] developed models to predict the duration of the incident. A. Khattak, *et al.*, [10] investigated the Ordinary Least Squares (OLS) regression model using a quantile regression method. Root Mean Square Error (RMSE) was used to assess the developed models. Results showed that the RMSE for OLS was 82.29 min, while for the quantile regression with location-based prediction was 57.49 min. H. Park, *et al.* [11] applied a two-step method to develop an incident clearance duration estimation model based on Bayesian Neural Networks (BNN) and TREPAN algorithm. Back-propagation neural network (BPNN), classification and regression tree (CART), and support vector machine (SVM) methods were also used for further investigation of the performance of the developed model. Mean absolute error for the BNN model was 0.6 min. L. Lin, *et al.* [12] proposed a Hybrid model based on the decision tree model M5P tree and the statistical model HBDM to estimate incident duration, which upgraded the M5P tree model to the M5P-HBDM model instead of M5P tree with a linear regression model. Results showed that the proposed model better than the other two, with MAPE equal to 36.20%, and 31.87%. X. Ma, *et al.* [13] developed a Gradient Boosting Decision Trees (GBDTs) method. The proposed model was compared with the back-propagation -neural network (BPNN), support vector machine (SVM), and random forest (RF). It was found that the GBDT method exceeds the performance of the other three methods with clearance time less than or equal 15 min, also longer than 15 min. Y. Zou, *et al.* [14] developed two copula models, an independent copula model, and the Gumble copula model. Mean absolute percentage error (MAPE), and tolerances for the prediction error were used to compare and evaluate the models, and it was found that the copula-based multivariate approach gives better performance, achieving minimum MAPE of 0.74. A. Mihăiță, *et al.*[2] investigated Extreme boosted classification and regression models. Results showed that the extreme boost tree method (XGBoost) outperformed the other models, with MAPE of 68.77, and R2 of 0.78. K. Hamad, *et al.* [15] investigated five machine learning techniques: Regression decision tree, Gaussian Process Regression (GPR), Ensemble trees, Support Vector Machine (SVM), and Artificial Neural Networks (ANN) to predict traffic incidents duration. The results showed that the SVM outperformed

the other models with MAE of 14.337 min, but, its training time was relatively long compared to the other models. Regression decision trees rated the best in terms of training time with only 1.26 min, but it was the worst in terms of accuracy with MAE of 16.74 min. K. Hamad, *et al.* [16] developed an Ensemble method, namely Random Forest (RF). Results showed that the best performing RF gives MAE of 36.652 min for the wide range of incidents duration. Moreover, the MAE reduced about 40% for the short range of the best RF performance. Artificial Neural Network (ANN) models were also used for further investigation of the performance of the developed models. ANN slightly outperformed (only a 0.24 percent difference) the RF models in terms of MAE. The RF built-in variable-importance capability showed that the RF model performance slightly downgraded with fewer variables.

III. INCIDENT DURATION

Duration is one of the characteristics of incidents that determines the magnitude of congestion and, thus, has been extensively researched [17].

The incidents duration is generally defined as the time between incident occurrence and clearance of the roadway, which means it's the time elapsed from the occurrence of the incident until all proof of the incident has been removed from the incident scene. This time can be divided into three phases [3], [7]:

1. Reporting time: The time between the occurrence of the incident and incident notification.
2. Response time: The time between incident notification and the arrival of the rescuer.
3. Clearance time: The time between the arrival of the rescuer and incident road clearance.

As shown in Fig. 1:

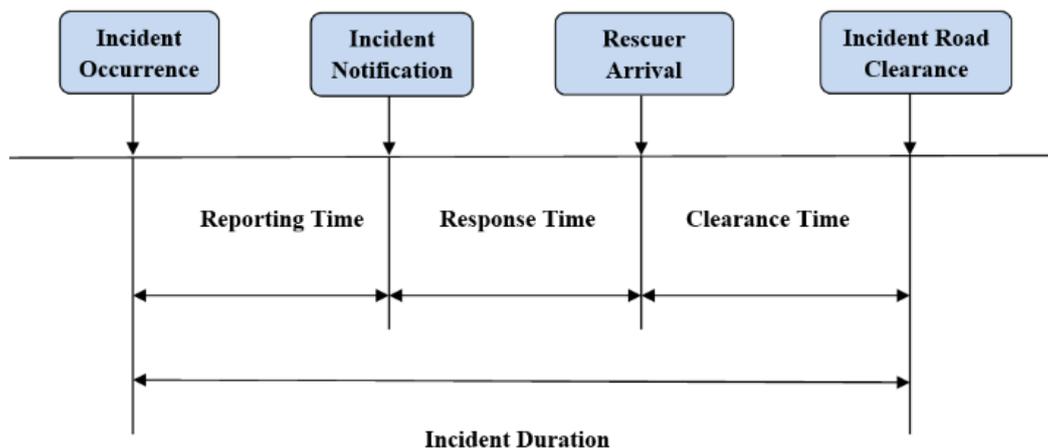


FIG. 1: COMPONENTS OF INCIDENT DURATION [7]

Also, there is an additional phase that can be added to the total time of incident duration which is the recovery time (the time between clearance of the incident and return to normal status). Compared with other phases, incident clearance is the most time-consuming stage in the overall incident management process, a severe incident which was not efficaciously cleared can double or even triple the total duration of the incident [13].

Clearing an incident quickly needs an appropriate managerial assistance in order to make some successful decisions about the resources required to deploy the team and clean the incident scene in a timely manner. This can frequently be accomplished with the aid of operators through among others, a good understanding of the factors influencing incident duration and the efficient use of the predicted incident duration information. Furthermore, any information that is forecasted about the duration of the incident is important for the prevention of congestion caused by the traffic incident where that information warns motorists of the need to re-route or re-schedule their trips. Therefore, incident duration estimation constitutes one of the most essential steps in the incident management process [4].

IV. METHODOLOGY

Three machine learning (ML) methods were used in this study, these includes Support Vector Machine (SVM), Random Forest (RF), and Neural Network Multi-Layer Perceptron (MLP). Measure Indexes, Root Mean Squared Error (RMSE), and Mean Absolute Error (MAE) were used to assess the performance of the developed models. Three models from each one of these three methods were developed with the use of the hyperparameter tuning process and by changing the data split ratio.

The dataset was split into (70&30), and (80&20) for the training and validation processes, where the 80%, and 70% were used as the training datasets, while the 30%, and 20% were the testing datasets. The training dataset was used for calibrating the chosen model, while the testing dataset was used to evaluate the performance of the models using a dataset independent from the one used to train the models. Where the testing dataset is the unseen data which means these data are not used in the training process but only the training dataset is available and the testing dataset will only be available during the evaluation of the developed model. Next, a brief description of each one of the three methods is given.

A. Support Vector Machine

In ML, support vector machines (SVMs, which also called support-vector networks) are supervised learning models with related learning algorithms that analyze data used for classification and regression analysis. SVM is one of the most popular models in Machine Learning in the context of statistical learning theory, based on the Structural Risk Minimization Theory (SRM) developed by Vapnik [18], [19].

SVM is a very robust and flexible Machine Learning model, that can perform linear or nonlinear classification, regression, as well as outlier detection. SVMs are particularly suitable for small- or medium-sized datasets. An SVM model is an illustration of the examples as points in space, mapped so that the examples of the individual categories are split by a clear gap which is as large as possible. New examples are then arranged into that same space and forecast to belong to a category based on the side of the gap on which they fall. An SVM estimator (f) on regression can be expressed as:

$$f(x) = w \cdot \Phi(x) + b \quad (1)$$

where Φ denotes a nonlinear transfer function that mapped the input vectors to a high-dimensional feature space in which the sample data are linearly separable. W is the weight and b is the offset [18], [20].

The SVR models were trained by the independent variables. The Kernel parameter with type-rbf and Kernel with type-linear were selected for the development of the three

SVR models. Fig. 2 illustrated the SVR prediction model diagram. The SVR algorithm procedure for predicting the incidents duration is as follows:

Step 1: Determine the input and output of SVR prediction model, where the independent variables, the target variable and the duration of the incidents, were considered as the input and the output of SVR prediction model.

Step 2: Construct the SVR prediction model, this is done by first: determine the training sample set, $T = \{(x_i, y_i), i= 1,2, \dots\}$ according to Step 1, where x_i denotes the i th sample of the independent variables and y_i denotes the i th sample of the dependent variable. Then, select the parameters of SVR algorithm to build and train the model, and finally, validate the model using the test sample set.

Step 3: Predict the incidents duration by the prediction model generated in Step 2.

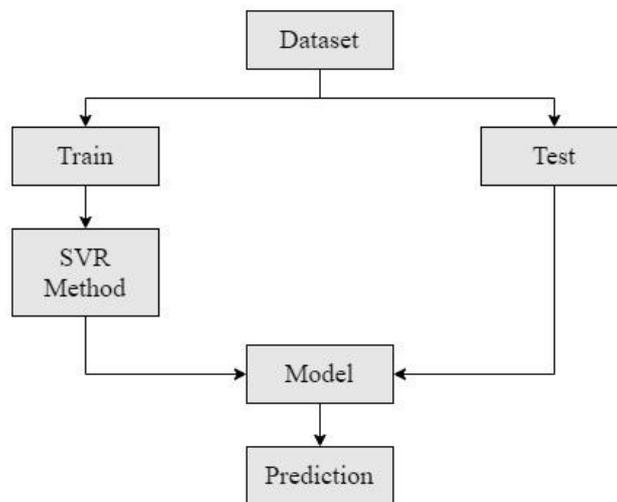


FIG. 2: SVR PREDICTION MODEL

B. Random Forest

Random forests or random decision forests are an ensemble learning method for classification, regression, and other tasks that function by constructing a multitude of decision trees at training time and producing the class in the case of classification or mean prediction in the case of regression of the individual trees. The first algorithm for random decision forests was produced by Tin Kam Ho using the random subspace method.

Random forest, as the name implies, is a combination of decision trees, where each tree is trained using a randomly selected subset of available training data, as it combined the randomness that is used to take the subset of data with having a bunch of decision trees, hence a forest. The RF regression prediction can be expressed as:

$$y_{rf}^B(x) = \frac{1}{B} \sum_{b=1}^B y_b(x) \quad (2)$$

where $y_b(x)$ is the value predicted by the b^{th} tree of the forest, x is the predictor variables, B is the number of trees in the forest, and $y_{rf}^B(x)$ is the average of the values predicted by the trees, as this average is the final value predicted by the forest [21], [22].

Fig. 3 illustrated how RF algorithm works, where the creation of this algorithm can be presented as follows:

Step 1: Creating a bootstrapped subset, where it can be performed by:

- Randomly select “ v ” variables from the training dataset “ t ”,

where $v \ll t$.

Step 2: Creating a decision tree, this can be performed by:

- Among the “ v ” variables, create the node “ n ”, root node, using the best split and the bootstrapped subset created in the previous step.
- Branch the node into daughter nodes using the best split.
- Repeat the previous steps until the desired number of nodes has been reached.

Step 3: Building a forest, where it can be performed by:

- Repeat steps 1 and 2 for “ n ” number times to create the desired number of trees “ B ”.

After the RF model has been created, then the prediction is performed by taking the average of the decision trees outcome, where each tree predicts different incidents duration for the same dataset, and consider that average as the final prediction from the RF model. As shown in equation (2).

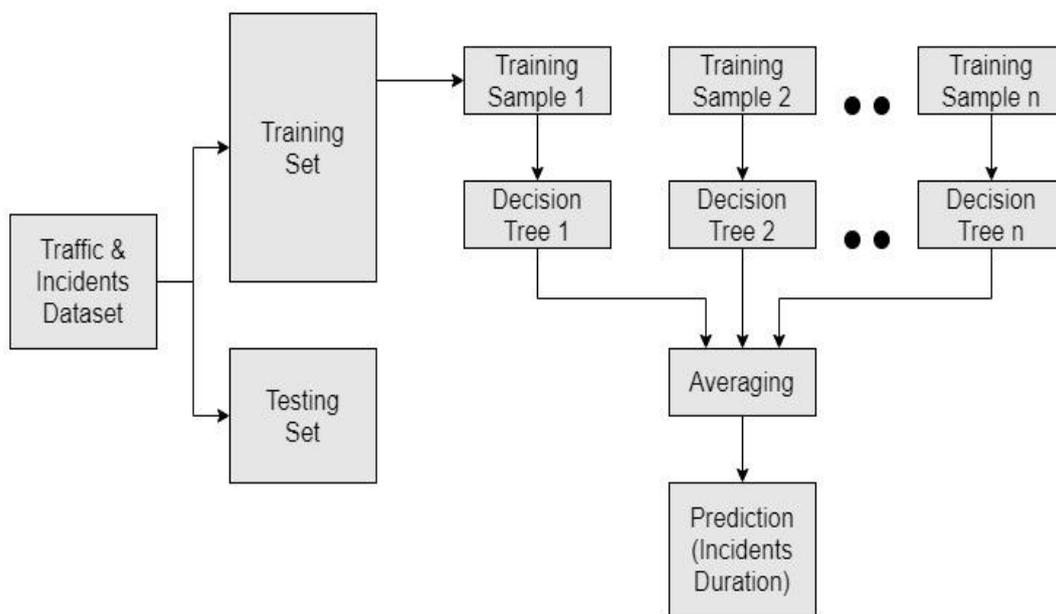


FIG. 3: RF PREDICTION MODEL

C. Neural Network Multi-Layer Perceptron

A multilayer perceptron (MLP) is a deep, artificial neural network. It consists of more than one perceptron. It consists of an input layer for receiving the signal, an output layer which makes a decision or prediction about the input, and in between these two, a random number of hidden layers which represent the true computational engine of the MLP. MLPs are able to approximate any continuous function with one hidden layer.

MLPs are also applied to supervised learning problems, where they train on a set of input-output pairs and learn to model the correlation (or dependencies) between these inputs and outputs. The training process includes changing the parameters, or weights and biases, of the model to reduce errors. Backpropagation is used to make these weight and bias changes relative to the error, and the same error may be calculated in a number of ways, included by root mean squared error (RMSE) [23].

Two hidden layers were used to construct each MLP model with the input and output layers. The number of neurons in each hidden layer was either 100 or 200 neurons. The

DOI: <https://doi.org/10.33103/uot.ijccce.21.1.1>

rectified function was used as the activation function. As illustrated in Fig. 4, the learning process of the perceptrons in the MLP model is demonstrated in the following steps:

Step 1: Takes the traffic and incidents dataset inputs in the input layer that are fed into the perceptron, and multiplies these inputs by their weights, and calculates the sum.

Step 2: Adds the bias weight to the sum.

Step 3: Feeds the sum through the activation function, the relu function was used as the activation function.

Step 4: The result of the relu function is the output.

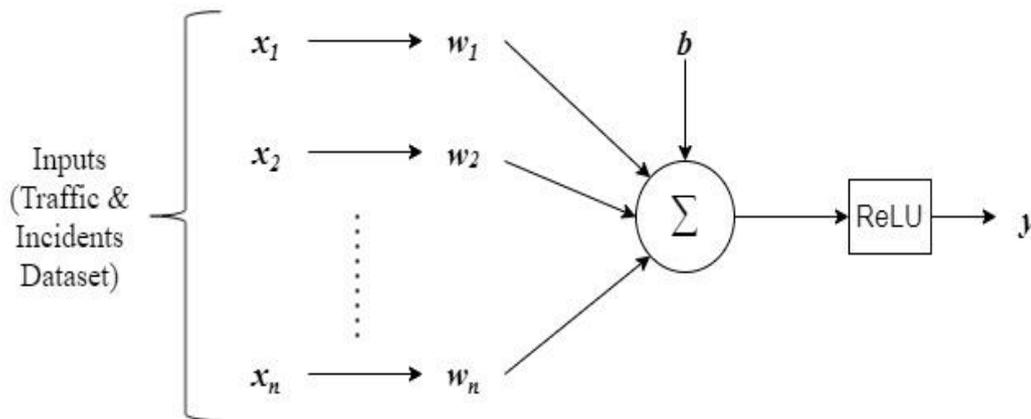


FIG. 4: PERCEPTRON'S LEARNING PROCESS

where $(x_1, x_2 \dots x_n)$ are the input variables, $(w_1, w_2 \dots w_n)$ are the weights, b is the bias, relu is the rectified activation function, and y is the output (it's either an input to the next hidden layer or the final prediction of the incidents duration) and calculated according to the following equation:

$$y = \text{relu activation} (\sum (\text{weight} * \text{input}) + \text{bias}) \quad (3)$$

This paper proposed a framework to develop models which can predict the incident duration to assist the transportation agencies and first responders to trigger the required preventative actions. Datasets were split into two parts, the first part was used to train the model with the best selected hyperparameter, the second unseen part of the dataset was used to evaluate the developed model. This section presented the steps that are used in the models building process, which can be illustrated as: First, data input, which means loading the data. Then, variables identification, this step includes the identification of the (X, Y) variables, where the Xs are the predictor variables, also called the independent variables. Whereas Y is the target, it is also called the response variable or the dependent variable. Then, data split, this step involves the division of the data based on a specific ratio, for example (80% & 20%), or (70% & 30%). Furthermore, method selection, this step involves choosing one of the machine learning methods that have been used in this work, which are: SVR, RF, and MLP methods. Also, parameters initialization, determining the method's

DOI: <https://doi.org/10.33103/uot.ijccce.21.1.1>

parameters is done in this step, where each one of the methods has its own parameters that differ from the other methods. Then, model training, based on the concept of how machine learning methods work, as it trained a model using specific input data and output data (in case of supervised learning) to perform a particular task, this step considers the training of the model by using the training dataset that is provided in the splitting phase. And, model testing (evaluation), it is also called model validation; this phase tests the models. The MSE, RMSE, and MAE error indexes were used to evaluate the model's performance. Finally, the prediction step, which gives the predicted incident duration results. Also there can be a model optimization, where this phase involves the optimization of the model, which can be done either by the changing of the data split ratio, or by the hyperparameter tuning process. *Fig. 5* illustrated the model building stages flowchart.

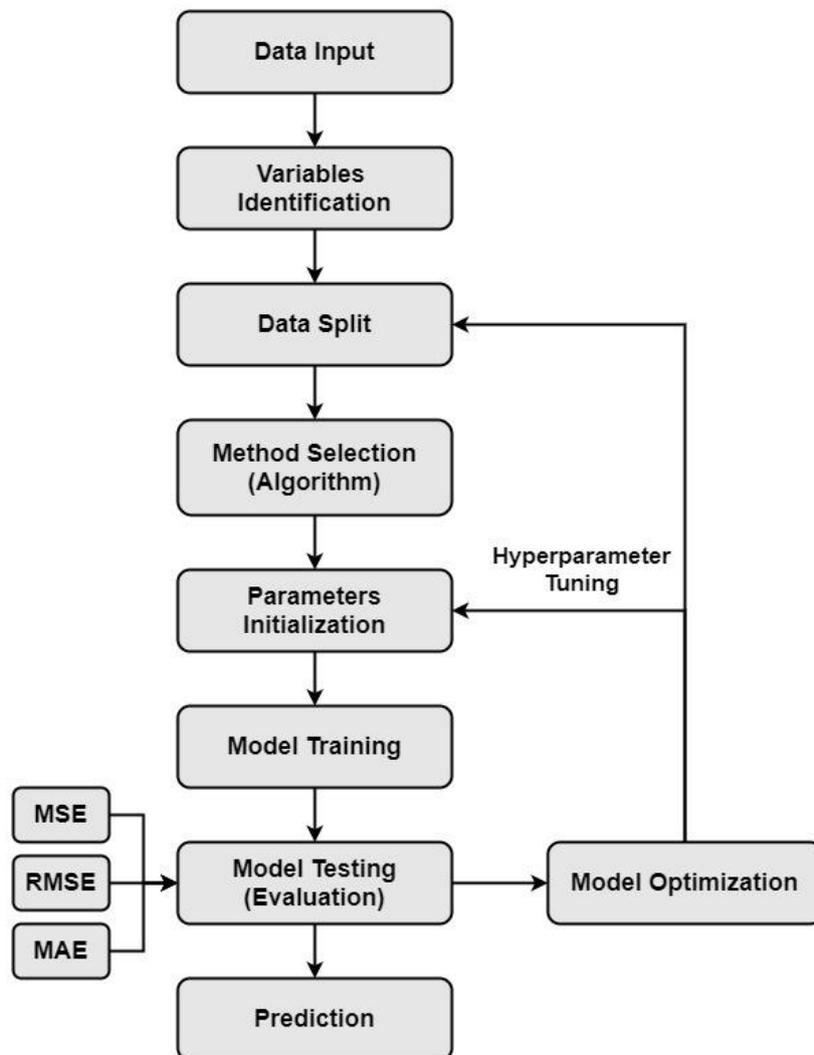


FIG. 5: MODEL BUILDING STAGES

V. STUDY AREA

Traffic management center incident logs and traffic sensors data collected over a 7.3-mile segment of Eastbound Interstate 70 (I-70) in Missouri, United States from January 2015 to January 2017 were used to develop incident duration estimation models. In total, 352 incidents were reported during this period. The average of the incident duration time was 27.6 minutes.

The incident logs included incident type, location and time of the incident, and weather conditions when the incident was reported. The I-70 segment was partitioned into 7 sections. A remote traffic microwave sensor was installed at upstream of each section to collect the traffic flow rate, speed, and occupancy in each lane. Traffic data were collected over 5-min intervals. Traffic data were added to the incident logs based on the location of the incidents and the time when the incidents were reported. Traffic data from a traffic sensor upstream of the incident location were added to the incident logs. Table 1 illustrated all the data variables that are used to build the incident duration prediction models, with description of each variable. Where the target variable is the incident duration variable, also called the response variable, whereas the other variables were the predictor variables.

TABLE 1: DATA DESCRIPTION

Variable	Description
Incident Duration	Incident duration in minutes.
Total Volume	The number of vehicles when an incident occurred.
Average of Occupancy	A vehicle occupancy rate is the number of passengers in the vehicle.
Weighted Average Speed	Vehicle speed collected on the highway.
Road Type	Values: Interstate, Off-Ramp, and On-Ramp.
Time of Day	Time period of the day when the accident occurred.
Peak Hours	Indicates whether the accident occurred during peak hours (Morning Peak: from 6:00 AM to 9:00 AM, Afternoon Peak: from 3:00 PM to 6:00 PM) or non-peak hours. Values: AM Peak, PM Peak, and Off-Peak.
Day of Week	The day of the week when the accident occurred.
Traffic Pattern	Indicates whether the accident occurred on a weekday or a weekend. Values: Weekday, Weekend.
Total Vehicle Count	The number of involved vehicles in an accident.
Weather	The weather condition at the time of the accident. Values: Normal, Winter Storm, and Rain.
Accident Overturned Car	Values: yes, no.

VI. RESULTS

Incident duration prediction models were developed by using SVM, RF, and MLP methods. Tables 2,3, and 4 illustrated the RMSE, and MAE indexes that were used to evaluate each model for training, testing, and prediction datasets. Next, a brief description for each model will be discussed in details.

A. Support Vector Machines Models

Three models were built by using the SVM method, these include SVM-Model-1, SVM-Model-2, SVM-Model-3. In SVM-Model-1, the dataset was split into two parts, 70% for training and 30% for validation, the Kernel parameter with type-rbf was selected in this model. The regularization parameter (C) was equal to 1.0, and the size of the kernel cache was 200 (in MB).

SVM-Model-2 was also built based on a 70%-30% split ratio for training and validation datasets, this model was enhanced by tuning one of the model hyperparameters, this improvement includes the use of Kernel with type-Linear. The regularization parameter (C) was equal to 1.0. The size of the kernel cache was 200 MB.

In SVM-Model-3, the dataset was split into 80%, and 20% for the training and testing process, respectively, with the use of Kernel parameter type-rbf. The size of the kernel cache was also 200 MB, and the regularization parameter (C) was equal to 1.0.

The analysis of these three models have shown that the best performing SVM model was SVM-Model-2, given RMSE of 23.42 min, and MAE of 14.23 min (for the prediction category). The RMSE and MAE indexes decreased by 6.87 min and 0.52 min in the testing in comparison with training. Not falling much behind, the worst SVM model was SVM-Model-1, where the RMSE, and MAE indexes for the prediction model were 23.65 min, and 15.06 min, respectively.

TABLE 2: SVM MODELS PERFORMANCE INDEXES

State	Category	RMSE (min)	MAE (min)
SVM-Model-1	Training	25.25	14.96
	Testing	19.43	15.29
	Prediction	23.65	15.06
SVM-Model-2	Training	25.27	14.39
	Testing	18.40	13.87
	Prediction	23.42	14.23
SVM-Model-3	Training	24.16	14.39
	Testing	21.02	17.16
	Prediction	23.56	14.95

B. Random Forest Models

Three RF models were constructed by using the RF method, which are, RF-Model-1, RF-Model-2, RF-Model-3.

In RF-Model-1 the forest was constructed with 100 trees, and the dataset was split into 70% for training the model, and 30% for the validation process. The maximum depth of the tree was 2, and the minimum number of samples required to be at a leaf node was 1. The prediction RMSE, and MAE indexes were 19.12 min, and 14.61 min, respectively.

While in RF-Model-2, the model included 50 trees, with the same split ratio as the previous model. The maximum depth of the tree was 2, and the minimum number of samples required to be at a leaf node was 1. The minimum number of samples required to split an internal node was 2. RMSE index for the prediction model was 19.13 min, and the prediction MAE index was 14.67 min.

TABLE 3: RF MODELS PERFORMANCE INDEXES

State	Category	RMSE (min)	MAE (min)
RF-Model-1	Training	18.06	14.37
	Testing	21.38	15.17
	Prediction	19.12	14.61
RF-Model-2	Training	18.33	14.46
	Testing	20.86	15.18
	Prediction	19.13	14.67
RF-Model-3	Training	18.46	14.14
	Testing	20.63	16.34
	Prediction	18.91	14.58

TABLE 4: MLP MODELS PERFORMANCE INDEXES

State	Category	RMSE (min)	MAE (min)
MLP-Model-1	Training	24.55	15.82
	Testing	18.57	14.89
	Prediction	22.92	15.54
MLP-Model-2	Training	24.13	15.54
	Testing	18.72	15.14
	Prediction	22.64	15.42
MLP-Model-3	Training	24.50	16.30
	Testing	19.64	16.76
	Prediction	23.60	16.39

RF-Model-3 was constructed with 100 trees, and the model has been enhanced by splitting the dataset into 80%, and 20% for the training and testing process, respectively. The maximum depth of the tree was 2. The number of jobs to run in parallel was 1. Achieving RMSE of 18.91 min, and MAE of 14.58 min, (for the prediction model).

The results have shown that all the three models that built depending on RF method were comparable with slight differences, where RF-Model-3 outperforms the other two. The RMSE and MAE indexes increased by 2.17 min and 2.2 min in the testing in comparison with training.

C. Neural Network Multi-Layer Perceptron Models

Three MLP models have been investigated, these include MLP-Model-1, MLP-Model-2, MLP-Model-3, and each one of these models composed of two hidden layers. Activation function type-relu was used in these three models.

In MLP-Model-1, the number of neurons in the hidden layer was 100 neurons, and the dataset was split into 70% for training the model, and 30% for the validation process. A learning rate type-constant was used. The maximum number of iterations was 200.

While in MLP-Model-2, with the same split ratio as the previous model, the model was enhanced using 200 neurons for each hidden layer. A learning rate type-constant was used. The maximum number of iterations was 200.

MLP-Model-3 was built based on an 80%-20% split ratio for training and validation datasets. The hidden layers included 100 neurons. A learning rate type-constant was used. The maximum number of iterations was 200.

The results of these three MLP models indicated that the best performing MLP model was the one with 200 neurons for each hidden layer, achieving RMSE of 22.64 min, and MAE of 15.42 min (for the prediction model). The RMSE and MAE indexes decreased by 5.41 min and 0.4 min in the testing in comparison with training. while the worst MLP model was MLP-Model-3 with 100 neurons for each hidden layer, and the splitting ratio of (80&20), given the prediction RMSE, and MAE indexes were 23.60 min, and 16.39 min, respectively.

Overall, the performance of the SVM, RF, and MLP models in estimating the duration of the traffic incident was satisfactory. The validation RMSE indexes decreased by 5.82 min, and 5.98 min for SVM-Model-1, and MLP-Model-1, respectively, in comparison with training RMSE indexes. While in RF-Model-1, the validation RMSE index increased by 3.32 min in comparison with training. MAE indexes increased by 0.33 min, and 0.8 min in validation in comparison with MAE training indexes for SVM-Model-1, and RF-Model-1, respectively. While in MLP-Model-1, the validation MAE index decreased by 0.93 min in comparison with the training MAE index. Fig. 6 illustrated the prediction RMSE, and MAE indexes for the SVM-Model-1, RF-Model-1, and MLP-Model-1.

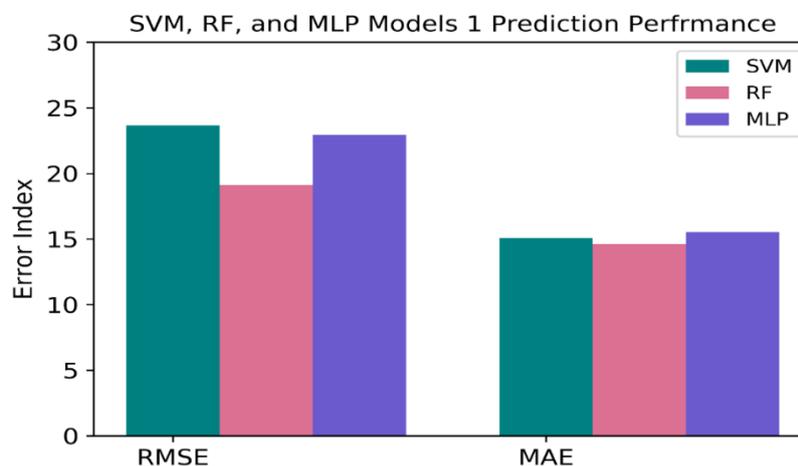


FIG. 6: SVM, RF, AND MLP MODELS 1 PREDICTION PERFORMANCE

The SVM, RF, and MLP models, for the training, testing, and prediction datasets, have almost the same error range in terms of RMSE and MAE.

When comparing the best-performing models of these three techniques, it can be found that the SVM model slightly outperformed the other two models, for the prediction dataset, the best-performing SVM model scored MAE of 14.23 min as compared to 14.58 min, and 15.42 min MAE for the best-performing RF and MLP models, respectively.

Fig.7 illustrated the prediction RMSE, and MAE indexes for the SVM-Model-2, RF-Model-2, and MLP-Model-2.

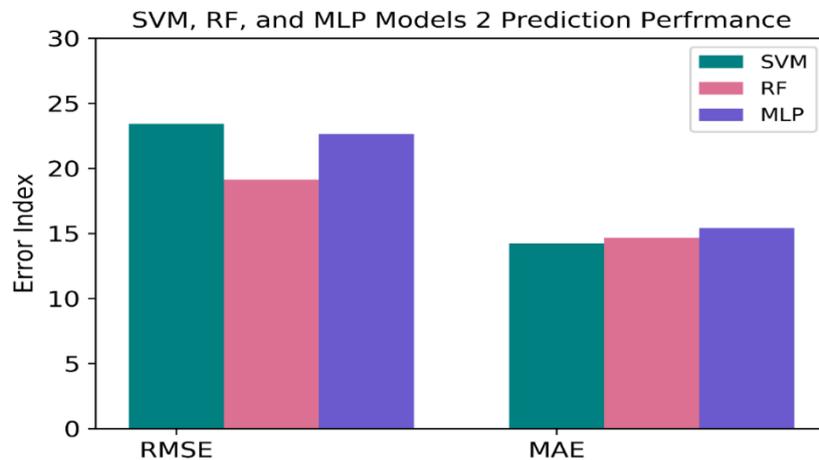


FIG. 7: SVM, RF, AND MLP MODELS 2 PREDICTION PERFORMANCE

On the other hand, in terms of RMSE indexes for the prediction dataset of these three methods, it has been shown that RF model slightly outperformed the SVM and MLP models, given RMSE of 18.91 min for the best-performing RF model as compared to 23.42 min and 22.64 min RMSE for the best-performing SVM and MLP models, respectively.

Fig. 8 illustrated the prediction RMSE, and MAE indexes for the SVM-Model-3, RF-Model-3, and MLP-Model-3.

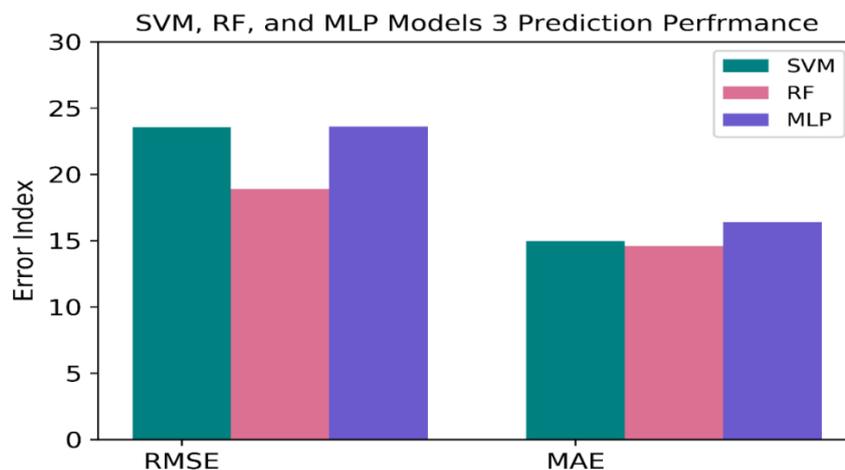


FIG. 8: SVM, RF, AND MLP MODELS 3 PREDICTION PERFORMANCE

VII. CONCLUSION

Traffic incidents are one of the primary reasons for the Non-recurrent congestion which in turn can lead to secondary accidents. Accurately predicting the duration of an incident plays an important role in reducing the impact of the Non-recurrent congestion on road capacity reduction and massive travel time loss. Traffic incident logs and traffic sensors data from Eastbound Interstate 70 (I-70) in Missouri, United States were collected over two years with a total of 352 incidents. This study investigated different machine learning algorithms to predict the duration of the incident. Support Vector Machine, Random Forest, and Neural Network Multi-Layer Perceptron were developed for the estimation of the duration. Root mean squared error and mean absolute error were used to assess the models' performance and it was found that the SVM model slightly outperformed the other two models, where the best-performing SVM model scored MAE of 14.23 min. While, in terms of RMSE indexes for the prediction dataset of these three methods, it has been shown that the RF model slightly outperformed the SVM and MLP models, achieving RMSE of 18.91 min for the best-performing RF model.

REFERENCES

- [1] Y. Chung, H. Kim, and M. Park, "Quantifying non-recurrent traffic congestion caused by freeway work zones using archived work zone and ITS traffic data," *Transportmetrica*, vol. 8, no. 4, pp. 307–320, 2012.
- [2] A.-S. Mihaita, Z. Liu, C. Cai, and M.-A. Rizoiu, "Arterial incident duration prediction using a bi-level framework of extreme gradient-tree boosting," no. October, pp. 21–25, 2019.
- [3] K. Fu, T. Ji, L. Zhao, and C. T. Lu, "TITAN: A spatiotemporal feature learning framework for traffic incident duration prediction," *GIS Proc. ACM Int. Symp. Adv. Geogr. Inf. Syst.*, pp. 329–338, 2019.
- [4] Y. Chung, "Development of an accident duration prediction model on the Korean Freeway Systems," *Accid. Anal. Prev.*, vol. 42, no. 1, pp. 282–289, 2010.
- [5] Q. Shang, D. Tan, S. Gao, and L. Feng, "A Hybrid Method for Traffic Incident Duration Prediction Using BOA-Optimized Random Forest Combined with Neighborhood Components Analysis," *J. Adv. Transp.*, vol. 2019, 2019.
- [6] F. C. Pereira, F. Rodrigues, and M. Ben-Akiva, "Text analysis in incident duration prediction," *Transp. Res. Part C Emerg. Technol.*, vol. 37, pp. 177–192, 2013.
- [7] H. Cong, C. Chen, P. S. Lin, G. Zhang, J. Milton, and Y. Zhi, "Traffic Incident Duration Estimation Based on a Dual-Learning Bayesian Network Model", *Transp. Res. Rec.*, vol. 2672, no. 45, pp. 196–209, 2018.
- [8] J. Tang, L. Zheng, C. Han, F. Liu, and J. Cai, "Traffic Incident Clearance Time Prediction and Influencing Factor Analysis Using Extreme Gradient Boosting Model," *J. Adv. Transp.*, vol. 2020.
- [9] X. Li, J. Liu, A. Khattak, and S. Nambisan, "Sequential Prediction for Large-Scale Traffic Incident Duration: Application and Comparison of Survival Models", *Transp. Res. Rec.*, vol. 2674, no. 1, pp. 79–93, 2020.
- [10] A. J. Khattak, J. Liu, B. Wali, X. Li, and M. W. Ng, "Modeling traffic incident duration using quantile regression," *Transp. Res. Rec.*, vol. 2554, no. 2554, pp. 139–148, 2016.
- [11] H. Park, A. Haghani, and X. Zhang, "Interpretation of Bayesian neural networks for predicting the duration of detected incidents ABSTRACT," vol. 2450, 2016.
- [12] L. Lin, Q. Wang, and A. W. Sadek, "A combined M5P tree and hazard-based duration model for predicting urban freeway traffic accident durations," *Accid. Anal. Prev.*, vol. 91, pp. 114–126, 2016.
- [13] X. Ma, C. Ding, S. Luan, Y. Wang, and Y. Wang, "Prioritizing Influential Factors for Freeway Incident Clearance Time Prediction Using the Gradient Boosting Decision Trees Method," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 9, pp. 2303–2310, 2017.
- [14] Y. Zou, X. Ye, K. Henrickson, J. Tang, and Y. Wang, "Jointly analyzing freeway traffic incident clearance and response time using a copula-based approach," *Transp. Res. Part C Emerg. Technol.*, vol. 86, no. October 2017, pp. 171–182, 2018.
- [15] K. Hamad, M. A. Khalil, and A. R. Alozi, "Predicting Freeway Incident Duration Using Machine Learning," *Int. J. Intell. Transp. Syst. Res.*, 2019.

- [16] K. Hamad, R. Al-Ruzouq, W. Zeiada, S. Abu Dabous, and M. A. Khalil, "Predicting incident duration using random forests," *Transp. A Transp. Sci.*, vol. 16, no. 3, pp. 1269–1293, 2020.
- [17] Y. Qi and H. Teng, "An information-based time sequential approach to online incident duration prediction," *J. Intell. Transp. Syst. Technol. Planning, Oper.*, vol. 12, no. 1, pp. 1–12, 2008.
- [18] C. Cortes and V. Vapnik, "Support-Vector Networks," *Mach. Learn.*, 1995.
- [19] B. Yu, Y. T. Wang, J. B. Yao, and J. Y. Wang, "A comparison of the performance of ANN and SVM for the prediction of traffic accident duration," *Neural Netw. World*, vol. 26, no. 3, pp. 271–287, 2016.
- [20] A. Géron, "*Hands-On Machine Learning with Scikit-Learn & TensorFlow*," USA, 2017.
- [21] T. K. Ho, "The Random Subspace Method for Constructing Decision Forests" vol. 20, no. 8, pp. 832–844, 1998.
- [22] A. A. B. Ruíz, "Machine Learning With Random Forests And Decision Trees," vol. 3, no. 2, pp. 54–67, 2015.
- [23] M. Minsky and S. Papert, *Perceptrons, Expanded Edition An Introduction to Computational Geometry*. 1969.