PAMUKKALE UNIVERSITY
ENGINEERING FACULTY

SHORT TERM TRAFFIC PREDICTION WITH MACHINE LEARNING AND SIGNALIZATION OPTIMIZATION

B.Sc. THESIS

Osman Doğukan URKAN
(19253502)

Department of Computer Engineering

Thesis Advisor: Assoc. Prof. Dr. Meriç ÇETİN

May 2022

Osman Doğukan URKAN, a B.Sc. student of PAU Engineering and Technology 19253502 successfully defended the thesis entitled "SHORT TERM TRAFFIC PREDICTION WITH MACHINE LEARNING AND SIGNALIZATION OPTIMIZATION", which he/she prepared after fulfilling the requirements specified in the associated legislations, before the jury whose signatures are below.

Thesis Advisor :	**Assoc. Prof. Dr. Meriç ÇETİN**	.....
Jury Members :		.....
	Pamukkale University	
		.....
	Pamukkale University	

Date of Submission : **May 2022**
Date of Defense : **June 2022**

Preface

My esteemed teacher and thesis advisor, Mr. Assoc. Dr. Prof. Meriç ÇETİN for his knowledge and assistance in the field of transportation and traffic engineering. Dr. I would like to thank Soner HALDENBİLEN, Mr. Gökhan Tokmak for helping me to access traffic data and understand the system, and Denizli Municipality Transportation Department.

I would like to express my endless thanks to Mr. Ramazan ÖLMEZ, who contributed greatly to the finalization of my work, to Mr. İbrahim KARAHAN, who opened my horizons with his ideas, and of course to my family for their financial and moral support.

May 2022 Osman Doğukan URKAN

Contents

ABBREVIATIONS

ITS	**:** Intelligent Transportation System
ML	**:** Machine Learning
GB	**:** Gradient Boosting
MAE	**:** Mean Absolute Error
MAPE	**:** Mean Absolute Percentage Error
SVM	**:** Support Vektor Machines
ARIMA	**:** Autoregressive integrated moving average
SARIMA	**:** Seasonal Autoregressive Integrated Moving Average
BPNN	**:** Back Propagation Neural Network
ANN	**:** Artificial Neural Network
XGBoost	**:** Extreme Gradient Boosting
LSTM	**:** Long-Short Term Memory
TVGCN	**:** Time-Variant Graph Convolutional Network
SVR	**:** Support Vector Regression
DCRNN	**:** Diffusion Convolutional Recurrent Neural Network
ASTGCN	**:** Attention Based Spatial–Temporal Graph Convolutional Network

SYMBOLS

$\mathbf{Y}$	**:** Dependent Variable
$\mathbf{X}$	**:** Independent Variable
β_0	**:** Point where the line crosses the y-axis
β_1	**:** Slope of the Line
ε	**:** Error rate
$\mathbf{p}$	**:** Probability that the characteristic exists
$\mathbf{b}$	**:** Vector of logistic regression coefficients
$\mathbf{w}$	**:** SVM Weight Vector
$\mathbf{x}$	**:** SVM Input Vector
σ	**:** RBF Variance Value
$\hat{y}_i$	**:** Predicted Value
y_i	**:** True Value

List of Tables

Page

List of Figures

Page

SHORT TERM TRAFFIC PREDICTION WITH MACHINE LEARNING AND SIGNALIZATION OPTIMIZATION

Abstract

In this study, a short-term traffic prediction model was developed with the data obtained from the intelligent transportation system of the Department of Transportation within Denizli Metropolitan Municipality. The purpose of developing this model is to make it easier for intelligent transportation systems to predict the traffic that will occur much earlier and take action. Due to the short cycle times of the systems, Support Vector Machine (SVM) and Extreme Gradient Boosting (Xgboost) were used to minimize model complexity and computational burden. The hyperparameter optimizations of the models trained with two different intersections, eight different directions, and thirty days of data were performed and the models were compared in terms of speed, absolute mean error (MAE), mean absolute percentage error (MAPE), etc. As a result of the comparison, it was concluded that the Extreme Gradient Boosting model would be more accurate for this problem and data set.

Then, new phase durations were obtained by using the traffic data obtained from the traffic prediction model in the Webster method. These phase durations and ITS durations were simulated on PTV Vissim, a traffic simulation program, and the results were compared. The simulation results show that the queuing delay of the traffic prediction model decreases by $\sim$ %10 to $\sim$ %15 and the intersection density decreases by $\sim$ %3 to $\sim$ %8 at fast increasing and decreasing moments.

Key Words: Short-Term Traffic Prediction, Machine Learning, Gradient Boosting, Support Vector Machine, Signalization Optimization

1. Introduction

As a result of the rapid increase in the world population, rising income levels, improvements in production and the rapid spread of technology, the number of vehicles in traffic is increasing rapidly [4]. The problem of traffic congestion, especially in city centers, has become a very serious issue. The increase in travel time adversely affects the psychology of people and increases the loss of time. In addition to these, the time spent by vehicles in traffic also feeds pollution and global warming [5]. Considering all of these problems, Intelligent Transportation Systems (ITS) are more important than ever.

Intelligent transportation systems are systems developed for increasing traffic safety, efficient use of energy and time, reducing the time spent in traffic and similar purposes. These systems perform tasks such as examining, analyzing and intervening in traffic data. These systems, which have good examples in the world, can make significant improvements in traffic flow [6]. However, many intelligent transportation systems can only predict the traffic a few minutes before it occurs. For this reason, they may not be able to react in some situations. Thus, short-term traffic forecasting becomes much more important, and a hybrid of successful forecasting and real data will increase the efficiency of intelligent transportation systems.

1.1 Purpose of the thesis

The aim of this thesis is to develop a forecasting model using traffic data provided by the Department of Transportation within Denizli Metropolitan Municipality and to make signalization improvements based on these forecasting results. The steps to be followed during the thesis study are as follows:

- Pre-processing of the data set (completing missing data, cleaning, parsing, etc.)
- Creation of selected Machine Learning (ML) models
- Testing and refinement of models (tuning)

- Visualization of results
- Model selection for signaling improvement
- Simulation of traffic after improvement
- Evaluation of results

1.2 Literature Research

Many studies have been conducted in the field of short-term traffic forecasting with various methods such as Machine Learning (SVM, Linear Regression, etc.), Time series (ARIMA, Kalman Filtering, etc.), Deep Learning (KNN, etc.), Neural Networks (BPNN, ANN, etc.). Due to the diversity of the data sets used and the complexity of the problem, it is very difficult to talk about the best method. In recent years, various Machine Learning techniques such as Support Vector Machine (SVM) and Gradient Boosting have produced successful results. It has been observed that research has evolved over time from classical methods and time series methods to methods such as Artificial Neural Networks and Machine Learning. Artificial Neural Networks and Deep Learning methods are frequently used in the literature. Due to the short cycle time of Denizli Transportation Department data to be used in the study, it was decided to use Machine Learning models to avoid complex models, reduce the computational burden and contribute to the literature. In a 1980 paper by Nihan and Nancy L. [7], a Box and Jenkins time series model was used on nine years of traffic data. Very successful results were obtained with an average error rate of %5. In 1991, in a paper written by Davis et al., [8], a member of the American Society of Civil Engineers, time series and ANN methods were compared. The results of the study showed that the ANN method, the K-Nearest Neighbors (KNN) method, produced better results than the Box and Jenkins method. In a 1994 study conducted in the USA, a comparison was made between ANN and K-Nearest Neighbors models. Smith et al. [9] showed that the KNN model gave better results for the Telegraph Road location data, while the ANN model gave better results for the Woodrow Wilson Bridge location data. It is seen that the ANN model has much higher large error rates.

Hinsbergen et al. [10] in 2007 studied Naïve, Parametric and Nonparametric models. The results of the study show that parametric models such as the Integrated

Autoregressive Moving Average (ARIMA) give good results in some cases, while in other cases the results are below the expectations. In the same study, it was concluded that Neural Networks are the most widely used model that gives relatively good results in this problem.

In an ongoing study by Dong et al. [11], a comparison was made between the Support Vector Machine model and the eXtreme Gradient Boosting (XGBoost) model. According to the results of the ongoing study in 2018, it was understood that the XGBoost model works faster and produces more accurate results than the SVM model.

In the study conducted by Yiğit and Haldenbilen [12], it was observed that the Artificial Neural Network (ANN) model was more successful than the ARIMA model. In the study, data taken at certain hours of 3 days of the week were used and it was mentioned that the models may give different results with different parameters.

Sun et al. [13] used an 80-day dataset collected from a highway in China. Convolutional Neural Networks (CNN), Long Short Term Memory (LSTM), Seasonal Autoregressive Integrated Moving Average (SARIMA) and XGBoost models are tested. It is concluded that XGBoost outperforms traditional forecasting methods except SARIMA. However, the study shows that the XGBoost model runs much faster than the SARIMA model. For all these reasons, the XGBoost model is considered to be the best choice among the models used in the study.

The Time-Variant Graph Convolutional Network (TVGCN) developed by Wang et al. [14] was tested on three different real-time datasets and gave much better results than similar models. In addition to models such as Support Vector Regression (SVR) and LSTM, the model using Graph Theory made predictions with very low error rates compared to artificial neural network models such as Diffusion Convolutional Recurrent Neural Network (DCRNN), Attention Based Spatial-Temporal Graph Convolutional Network (ASTGCN). In the test results, SVR: %19, LSTM: %17, DCRNN: %14, ASTGCN: %16, TVGCN: %12 with an average absolute error percentage of %12.

2. SCIENTIFIC BACKGROUND

2.1 Short Term Traffic Prediction

Short-term traffic prediction has been and continues to be used in many areas of transportation research for more than 30 years. It is frequently used in areas such as providing accurate information to drivers, signalization optimization, road feasibility studies, etc. With the increase in data analysis and computational power, its popularity is increasing day by day. In its broadest definition, short-term traffic prediction is the prediction of traffic that will occur in a period ranging from a few seconds to a few hours as a result of data analysis and processing using various models (Machine Learning, Deep learning, Time series, etc.).

2.2 Machine Learning

Machine Learning (ML) can be broadly defined as computational methods originating in the 1950s that use the experience to optimize or make accurate predictions. Experience refers to historical data collected and digitized for analysis by the learning computer. The quality of this data is crucial to the success of ML models [15]. Machine learning can be categorized into three categories according to the way it learns. These are;

- Supervised Learning
- Unsupervised Learning
- Reinforcement learning

Types of problems that can be solved with Machine Learning;

- Text and Document Classification
- Natural Language Processing
- Computer Vision

- Computational Biology
- Short Term Traffic Prediction etc.

2.2.1 Linear Regression

Linear regression is used to determine the relationship between the variable to be predicted (dependent variable) and the variables affecting it (independent variables) and the effects of independent variables on the dependent variable. Simple linear regression analysis is concerned with the nature and degree of the relationship between variables. The mathematical model is shown in equation 2.1.

$$Y = \beta_0 + \beta_1 X + \varepsilon \tag{2.1}$$

Here;

Y : Dependent variable

X : Independent variable

β_0: The point where the line crosses the y-axis

β_1: Slope of the line

ε: The error rate. Linear regression can produce very successful and fast results on linear data sets with relatively low model complexity. However, outliers and over-fitting can adversely affect the model.

2.2.2 Logistic Regression

Logistic regression is a regression method for classification without assuming a linear relationship between dependent and independent variables. Instead of assuming a linear relationship, it assumes a linear relationship between the logits of the explanatory variables and the response. It uses maximum likelihood estimation rather than ordinary least squares to estimate parameters and therefore relies on large sample approaches. Central to Logistic Regression analysis is the task of estimating the log odds of an event. Mathematically, logistic regression estimates a multiple linear regression function defined as follows:

$$logit(p) = b_0 + b_1 X_1 + b_2 X_2 + b_3 X_3 + ... + b_k X_k \tag{2.2}$$

As can be seen in equation 2.2, p is the probability that the characteristic exists.

$$Possibility = (p)/(1-p) \tag{2.3}$$

As can be seen in equation 2.3, the probability is the probability that the characteristic exists divided by the probability that the characteristic does not exist.

$$logit(p) = \ln \frac{p}{1-p} \tag{2.4}$$

Instead of choosing parameters that minimize the sum of square root errors (like linear regression), in logistic regression, the estimator chooses parameters that maximize the probability of observation of the sample values.

2.2.3 Support Vector Machine

Support vector machines are supervised machine learning algorithms used to classify small and medium-sized data, especially in classification problems. Basically, it creates lines to classify points of different classes in the data plane and aims to maximize the distance of these lines to points in different classes.

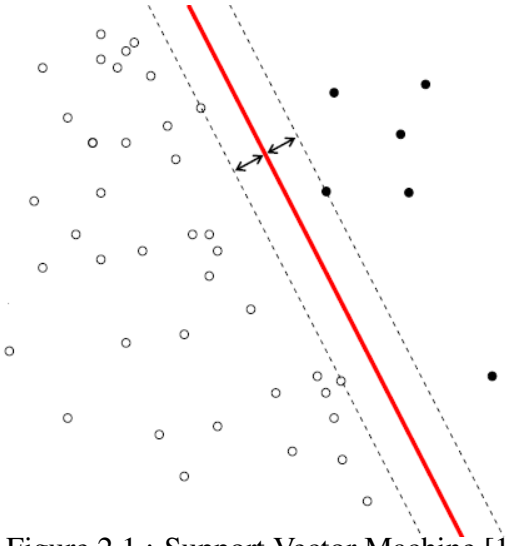


Figure 2.1 : Support Vector Machine [1]

Figure 2.1 shows a line separating two classes. The area between the line and the classes is called the margin, and the larger this area, the more successful the classification.

As shown in Figure 2.2, w is the weight vector, x is the input vector, b is the deviation. If the result for a new value is less than 0, it will be closer to the black dots. If the result is greater than or equal to 0, then it will be closer to the white dots. Not all data sets can

$$\hat{y} = \begin{cases} 0 \text{ if } \mathbf{w}^T \cdot \mathbf{x} + b < 0, \\ 1 \text{ if } \mathbf{w}^T \cdot \mathbf{x} + b \geq 0 \end{cases}$$

Figure 2.2 : SVM Line Equation

be classified in two-dimensional space. In such cases, increasing the dimensionality would increase the processing load, so different kernels are used instead. Some of these are

- Polynomial Kernel
- Gaussian RBF (Radial Basis Function) Kernel
- Sigmoid Kernel etc.

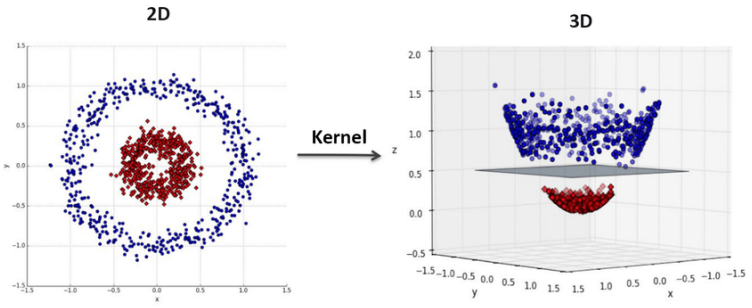


Figure 2.3 : Polynomial Kernel [2]

Figure 2.3 shows the data space after using the Polynomial Kernel.

2.2.3.1 Radial Basis Function Kernel

RBF kernels are one of the most widely used because of their similarity to the Gaussian distribution and because they are the most generalized kernel. It basically calculates how similar two points are or the distance between them. Mathematically, it is represented as follows;

$$K(X_1, X_2) = exp(-\frac{||X_1 - X_2||^2}{2\sigma^2}) \tag{2.5}$$

As seen in the equation σ : variance value

$||X_1 - X_2||^2$: the distance between two points.

2.2.3.2 Polynomial Kernel

For similarity estimation, the polynomial kernel looks not only at the characteristics of the given inputs but also at their combinations. In regression analysis, such combinations are known as interaction features. Mathematically, it is expressed as follows;

$$K(x,y) = (x^t y + c)^d \tag{2.6}$$

In the equation;

x, y: Attribute vectors

c : Hyperparameter

d : Equation degree

2.2.4 Extreme Gradient Boosting

Extreme Gradient Boosting (xgboost) is a decision tree-based machine learning algorithm with gradient boosting. The model, which has gained popularity in recent years, produces very good results and works faster than similar models because it creates decision trees in parallel.

2.2.5 Decision Tree

A Decision Tree consists of a root, decision nodes, and leaf nodes. It is a classification method that creates a model in the form of a tree structure. Basically, the algorithm creates decision structures using labeled training data and induction. It then uses these structures to decide which class the data belongs to. The parameters that affect the model the most are Subsample: What proportion of the training data will be used in training

Column sample(Colsample): How much of the columns in the training data will be used in training

Maximum Depth: Depth of the decision tree

Minimum child weight: The minimum weight that children in a node can receive.

Figure 2.4 shows the regression results with two different decision trees. From these results, it can be seen that the model with more maximum depth can capture outliers better, but this increases model complexity and may lead to over-fitting.

2.2.5.1 Gradient Boosting

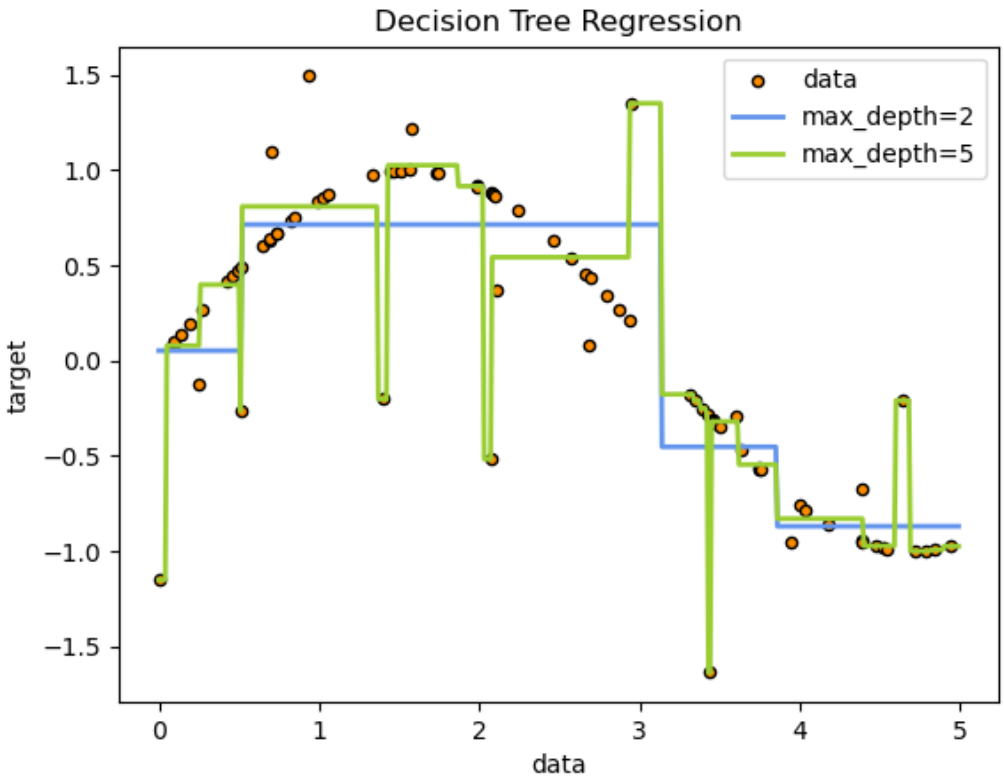


Figure 2.4 : Regression with decision tree [3]

Gradient boosting is a machine learning technique used for regression, classification, and other tasks that produce a predictive model in the form of a collection of weak predictive models such as decision trees. Like other boosting methods, gradient boosting iteratively merges weak students into a single strong student. The mathematical model can be described as follows

$$y = \frac{1}{n}\sum_{i}(\hat{y}_i - y_i)^2 \tag{2.7}$$

Here;

$\hat{y}_i$: Predicted value

y_i : Actual value

n: The total number of samples. As with most supervised machine learning algorithms, the goal is to predict output values from input values. At each step, the model predicts the previous error, improves it, and passes it on to the next step.

3. METHODOLOGY

This chapter provides detailed information about the data set, methods, and technologies used in this thesis.

3.1 Data Set

In the study, traffic data from the Department of Transportation within Denizli Metropolitan Municipality were used. There are 165 intersections connected to the intelligent transportation system used by the Department. Theater Intersection and Halley Intersection were selected among 165 intersections due to their high density and strategic location. The dataset consists of 27 days between 05/09/2021 and 30/09/2021 and includes the date the data was created, hourly vehicle density, and signaling information.

Table 3.1 : Dataset

Intersection name	Number of directions	Number of data
Theater	4	40000
Halley	4	40000

Figure 3.1 shows one-day traffic data for the high school arrival direction at the Theater intersection.

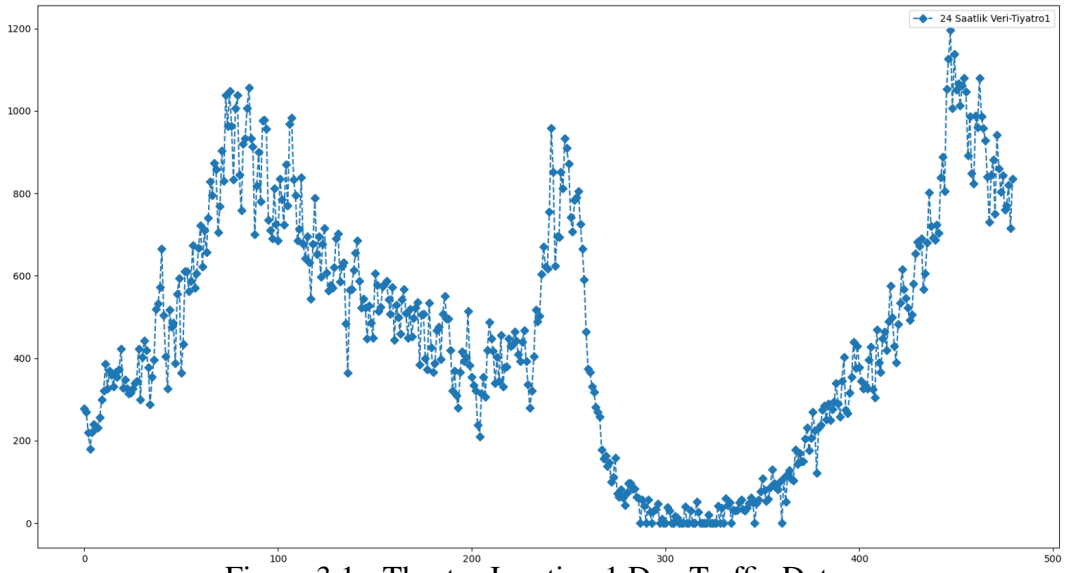


Figure 3.1 : Theater Junction 1 Day Traffic Data

3.2 Preprocessing

Although the dataset is very clean as received from the Department of Transportation, it is not possible to process it or train a model. For this reason, some processing was required before working on it.

3.2.1 Completion of missing values

There were a few missing values in the dataset. In order not to adversely affect the model training, these values were filled by averaging the time intervals in which they were found. Thus, outlier formation was prevented.

3.2.2 Adding intersection and direction features

In the dataset, each direction of each intersection is kept in different datasets. Since I wanted to use all the data together in the model training, I had to combine the data and add the information about which data belongs to which intersection and direction.

3.2.3 Parsing the date column

The creation date of the data was in a single column MM/DD HRS/MIN/SEC and could not be processed as it was. So I split this information into four columns, month, day, hour, and minute so that the final data can be seen in 3.2.

Table 3.2 : Data set after preprocessing

Data Label	Direction	Month	Month	Day	Hour	Hour	Minute
2	8	80000	80000	80000	80000	80000	

3.3 Model Selection

In the literature, there are a wide variety of algorithms used for the short-term traffic prediction problem. Most of them are deep learning and artificial neural networks. Using these complex models may cause computational difficulties both because they are already available in the literature and because the conversion time of the ITS used in Denizli is quite short. For these reasons, it was decided to use Support Vector Machine and Gradient Boosting algorithms that have proven their success in the literature.

4. SUCCESS

There are different ways to measure the success of machine learning models. While classification aims to predict the label of a class, regression aims to predict a numerical value. For this reason, accuracy in classification models can be found by dividing the correctly classified data by all the data, whereas in regression it is necessary to use certain error metrics.

4.1 Error Criteria

4.1.1 Mean Squared Error

Mean square error (MSE) is a widely used error metric in regression problems. MSE is calculated as the square of the difference between predicted and actual values in a data set divided by the number of data.

$$MSE = \frac{1}{N}\sum_{1}^{N}(y_i - \hat{y_i})^2 \tag{4.1}$$

In equation 4.1;

MSE: Mean Square Error

y_i : Actual value

$\hat{y_i}$: The predicted value.

Since the difference between the actual value and the predicted value is squared, the error value is always positive. This method also tends to inflate the error value. It is also very useful when one wants to increase the impact of large errors on the metric.

4.1.2 Root Mean Squared Error

Root mean square error (RMSE) is an extension of MSE. The difference is that the error is squared and then the square root is taken. Thus, errors are used as they are without inflation. In the RMSE metric, as in the MSE, the error is always positive. Its

mathematical formulation is given in 4.2.

$$RMSE = \sqrt{(\frac{1}{N}\sum_{1}^{N}(y_i - \hat{y}_i)^2)}, \quad RMSE = \sqrt{MSE} \tag{4.2}$$

4.1.3 Mean Absolute Error

Like the mean absolute error (MAE), RMSE, the units of the error score match the units of the predicted target value. Unlike RMSE, changes in MAE are linear and are therefore classified as heuristic.

$$MAE = \frac{\sum_{1}^{N}|y_i - \hat{y}_i|}{N} \tag{4.3}$$

5. RESULTS AND EVALUATION

5.1 Application

In this study, a short-term traffic prediction model was developed using Denizli Metropolitan Municipality traffic data. In order to make the data set ready for model training, pre-processing such as completion of missing data, normalization of data, and addition of new attributes were performed. The traffic data obtained from the prediction model were calculated by using the Webster (British) model, which is a phase duration calculation model. As a final process, the calculated phase durations were simulated in PTV Vissim, a traffic flow simulation.

5.2 Results

The results of the study will be analyzed under two headings. These are

- Machine Learning Model Results
- Simulation Results of Calculated Phase Values

5.2.1 Machine Learning Model Results

5.2.2 Support Vector Machine

The support vector machine model underperformed in terms of the length of training and prediction times and its accuracy on the test data set.

Table 5.1 : SVM Results

Model	Training Time(hours)	Prediction Time(seconds)	Accuracy(%)
SVM	~ 7.5	~ 79.2	~ 82.6

Table 5.1 shows the results of the model.

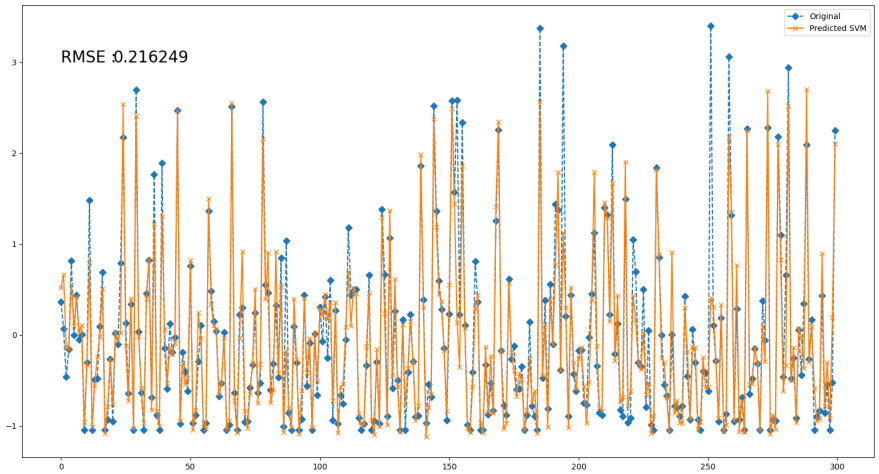


Figure 5.1 : SVM Prediction Results

The prediction values for the test dataset are shown in figure 5.1.

5.2.3 Extreme Gradient Boosting

Xgboost has produced very good results with both training and prediction times and prediction accuracy. It is considered suitable for use in intelligent transportation systems where cycle time is limited.

Table 5.2 : Xgboost Results

Model	Training Time(hours)	Prediction Time(seconds)	Accuracy(%)
Xgboost	~ 0.2	~ 0.003	~ 92

The results of the model are shown in table 5.2.

Figure 5.2 shows the training and test losses of the model.

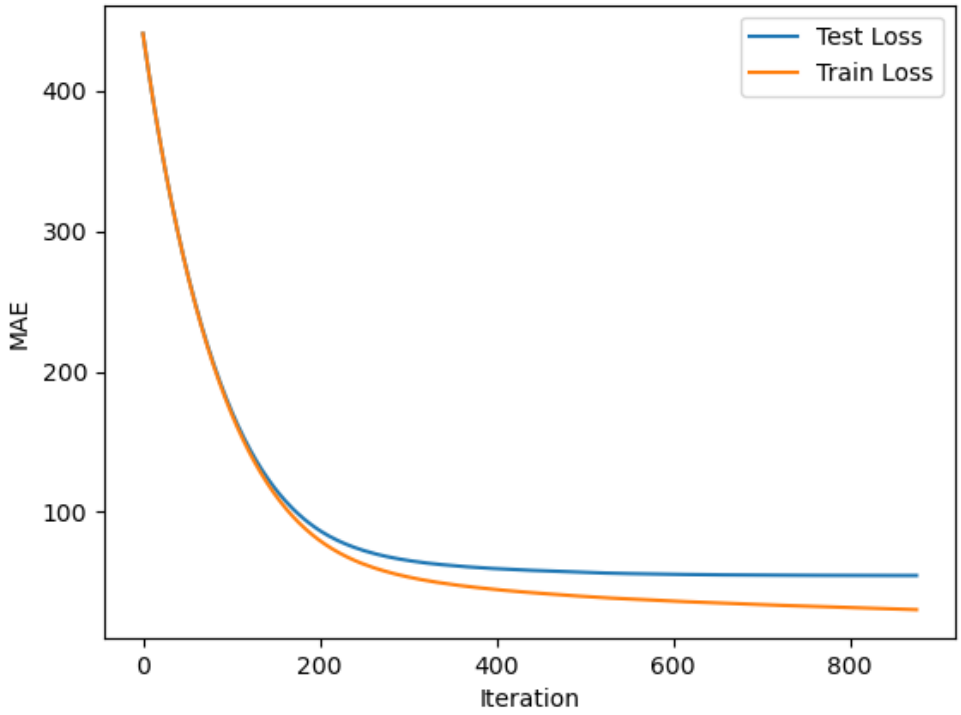


Figure 5.2 : Xgboost Training and Test Loss Values

5.2.4 Simulation Results of Calculated Phase Values

Webster method was selected from three different calculation methods in the study. The reason for this is that the phase durations in Denizli are calculated with this method. At the moments when the traffic increases or decreases rapidly, the signaling data obtained as a result of calculating the density obtained from the traffic prediction model with the Webster method with the signaling data of the ITS was tested in the simulation.

Simulation results are shown in table 5.11 and table 5.12.

Direction	Green Time(sec)	Delay(sec)	Avg. Speed(km/h)	Occ.(%)
1	18	33.70	41.50	0.32
2	28	26.45	45.20	2.14
3	17	41.60	45.67	0.06
4	31	28.36	46	4.87

Table 5.3 : ITS Data Theater Intersection Traffic Simulation Results

Direction	Green Time(sec)	Delay(sec)	Avg. Speed(km/h)	Occ.(%)
1	19	33.65	43	0.25
2	22	31.85	45.3	1.78
3	12	26.30	40.75	0.13
4	41	19.88	46.6	4.63

Table 5.4 : Prediction Model Data Theater Intersection Traffic Simulation Results

5.3 Conclusion

The simulation results show that the traffic prediction model reduces the queuing delay by $\sim$ %10 to $\sim$ %15 and the intersection density by $\sim$ %3 to $\sim$ %8 during fast increasing and decreasing periods. These improvements were between $\sim$ %2 and $\sim$ %5 during the hours of low variation. Considering that there are 164 intersections connected to the system, the results are remarkable.

5.4 Discussion

The results of the study show that short-term traffic prediction provides agility to intelligent transportation systems in times of rapid growth or rapid decline. The tests and simulations were carried out at a single intersection and could not fully reflect the effects of the method considering that traffic is a large system. As the model is used for a certain period of time and data is obtained, its effects will be more clearly understood.

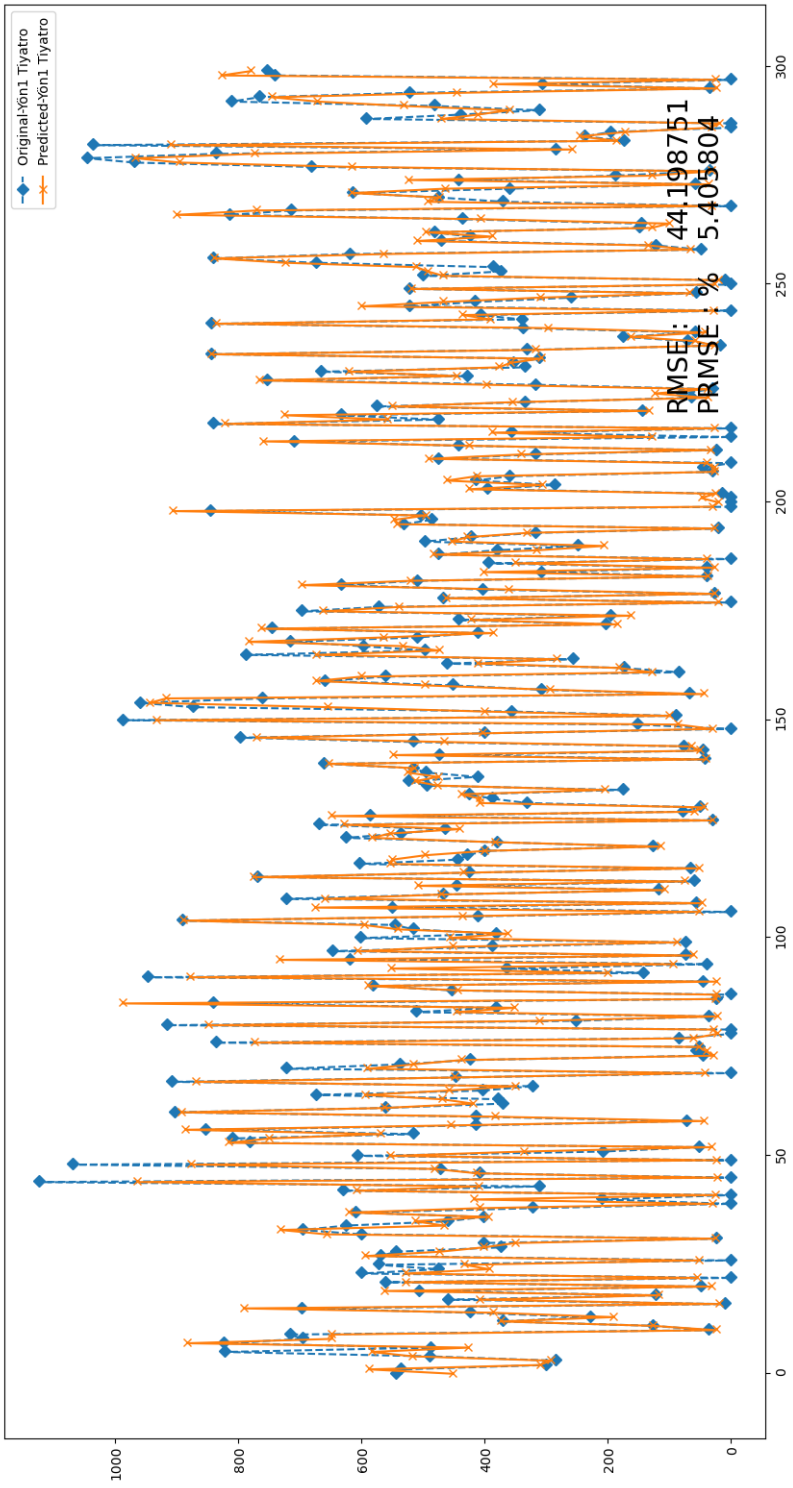


Figure 5.3 : Xgboost Theater Direction 1 Prediction Values

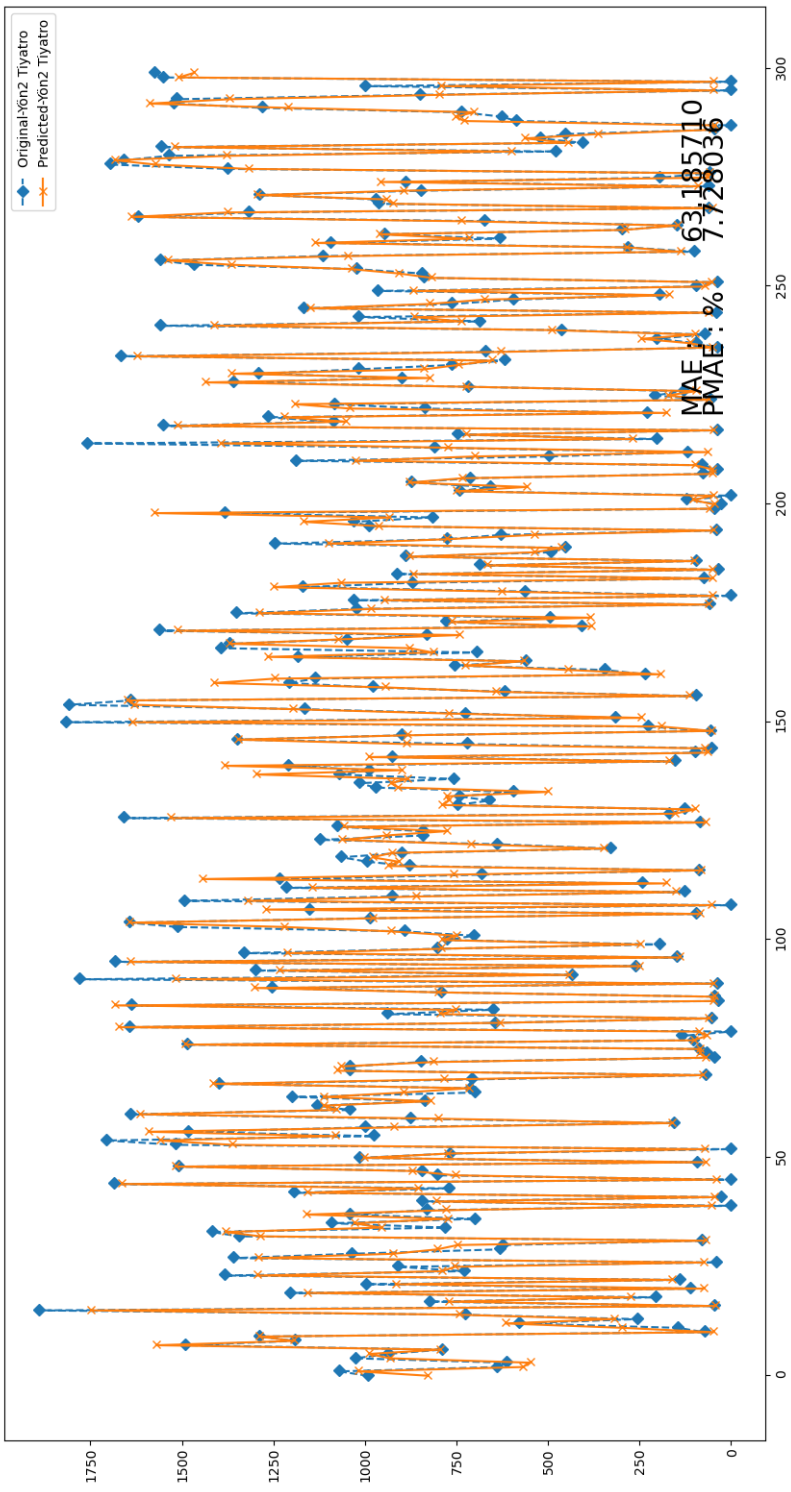


Figure 5.4 : Xgboost Theater Direction 2 Prediction Values

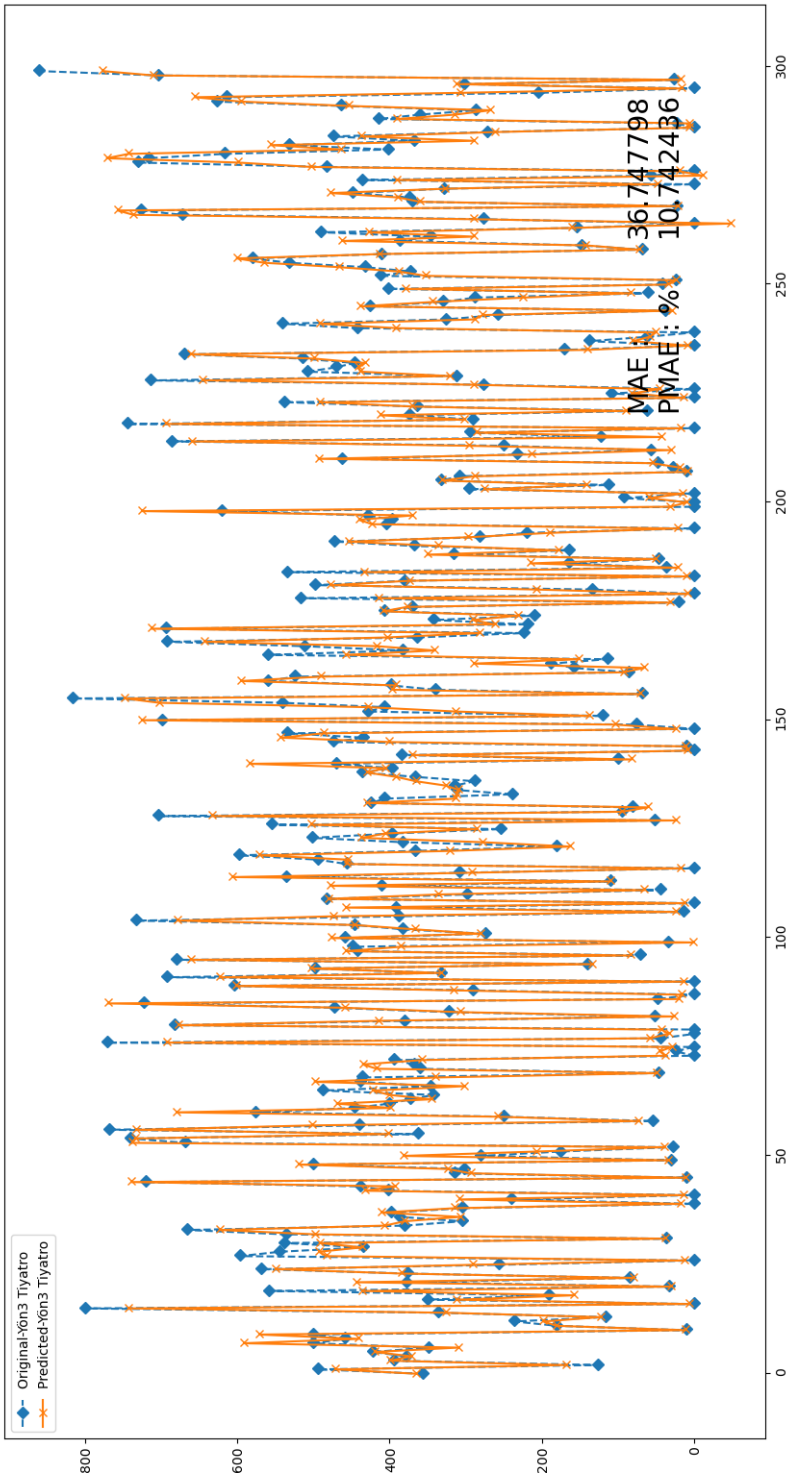


Figure 5.5 : Xgboost Theater Direction 3 Prediction Values

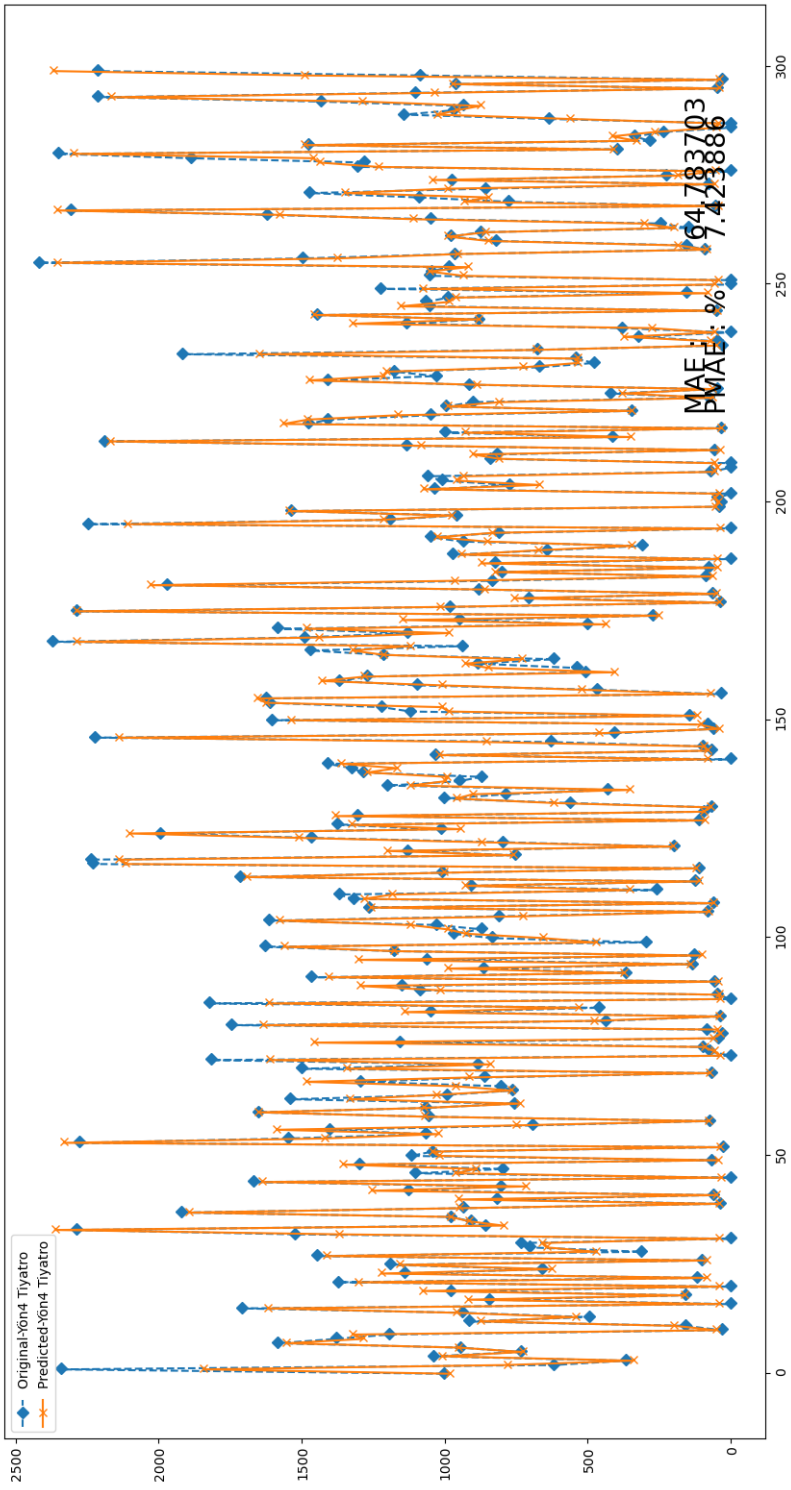


Figure 5.6 : Xgboost Theater Direction 4 Prediction Values

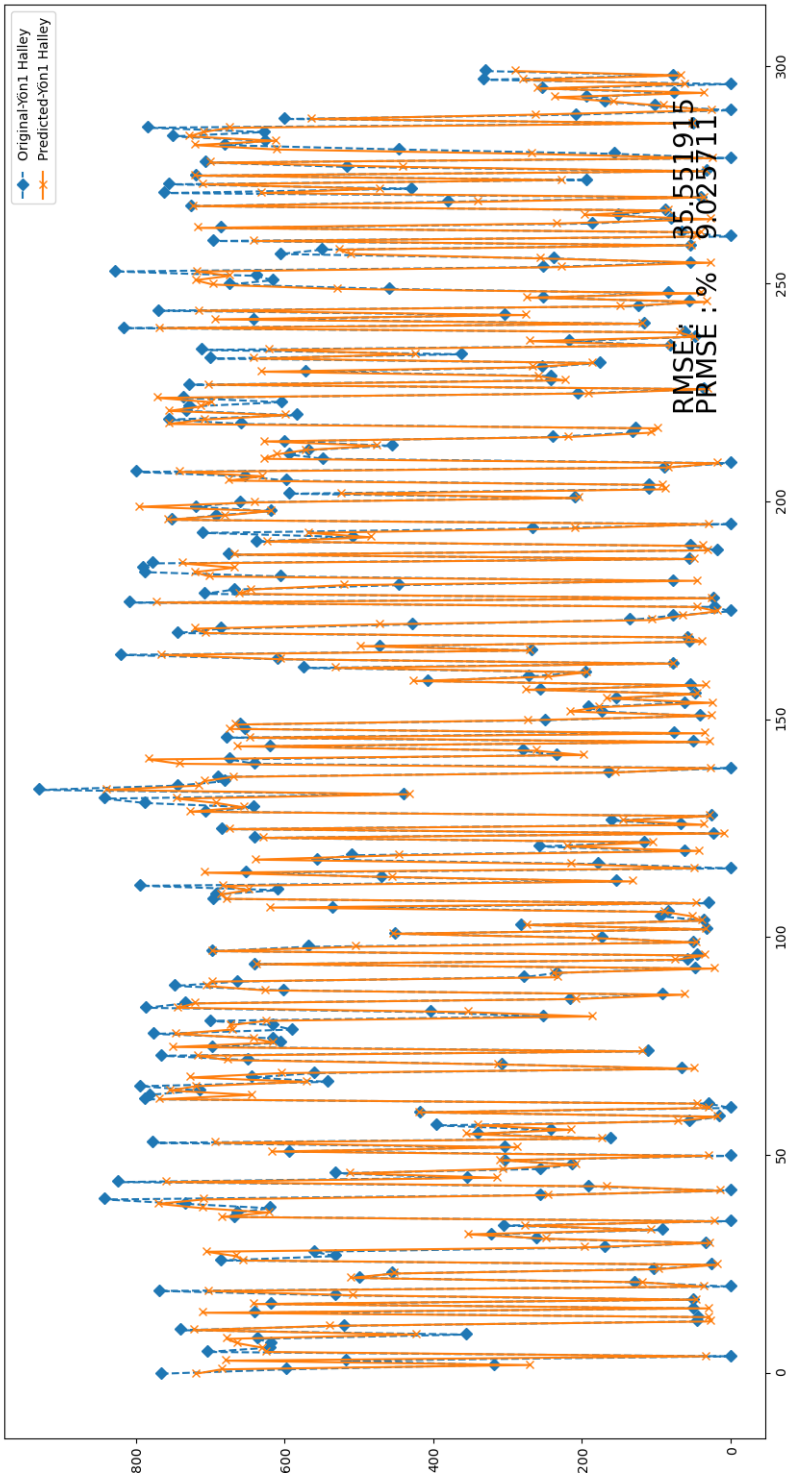


Figure 5.7 : Xgboost Halley Direction 1 Prediction Values

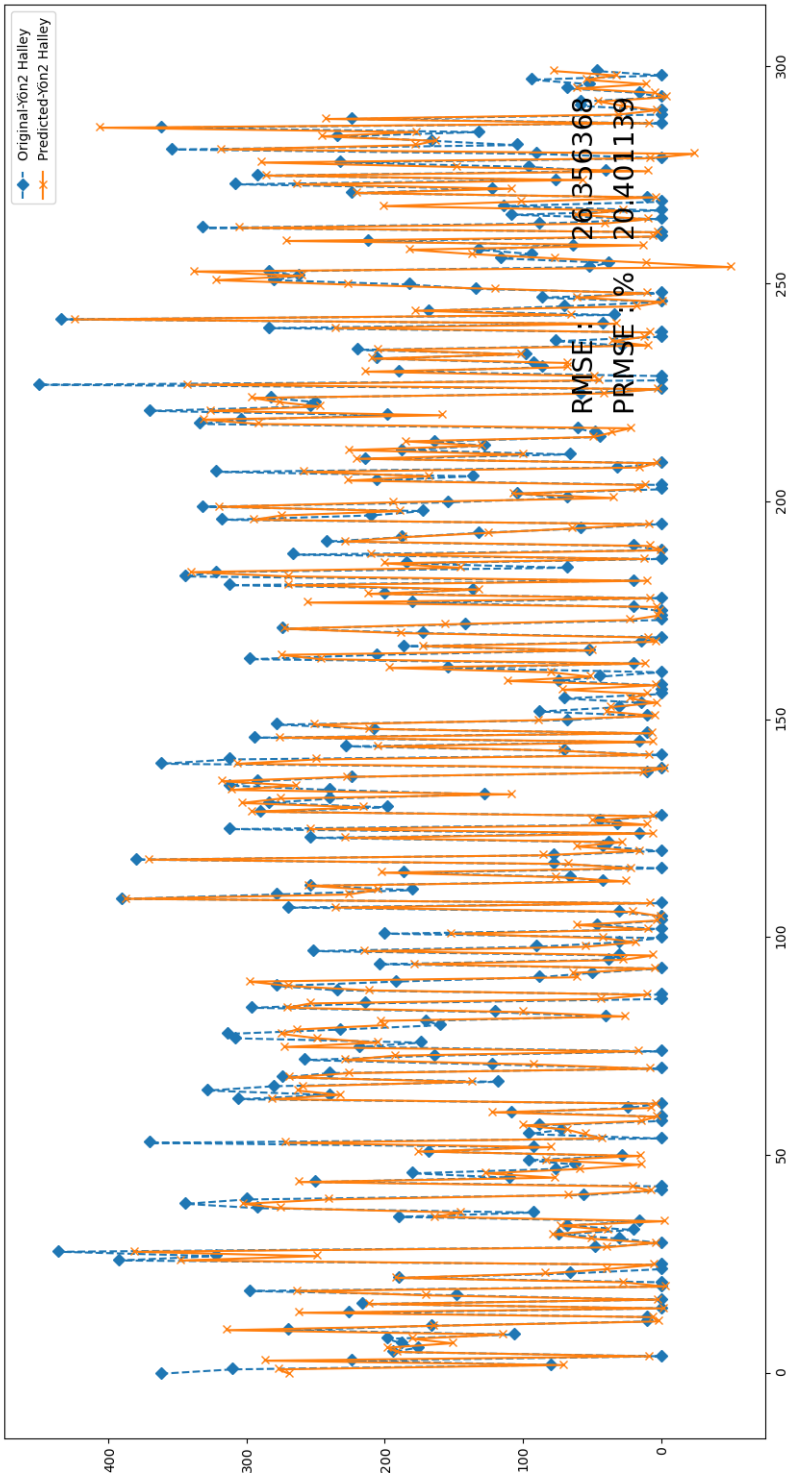


Figure 5.8 : Xgboost Halley Direction 2 Prediction Values

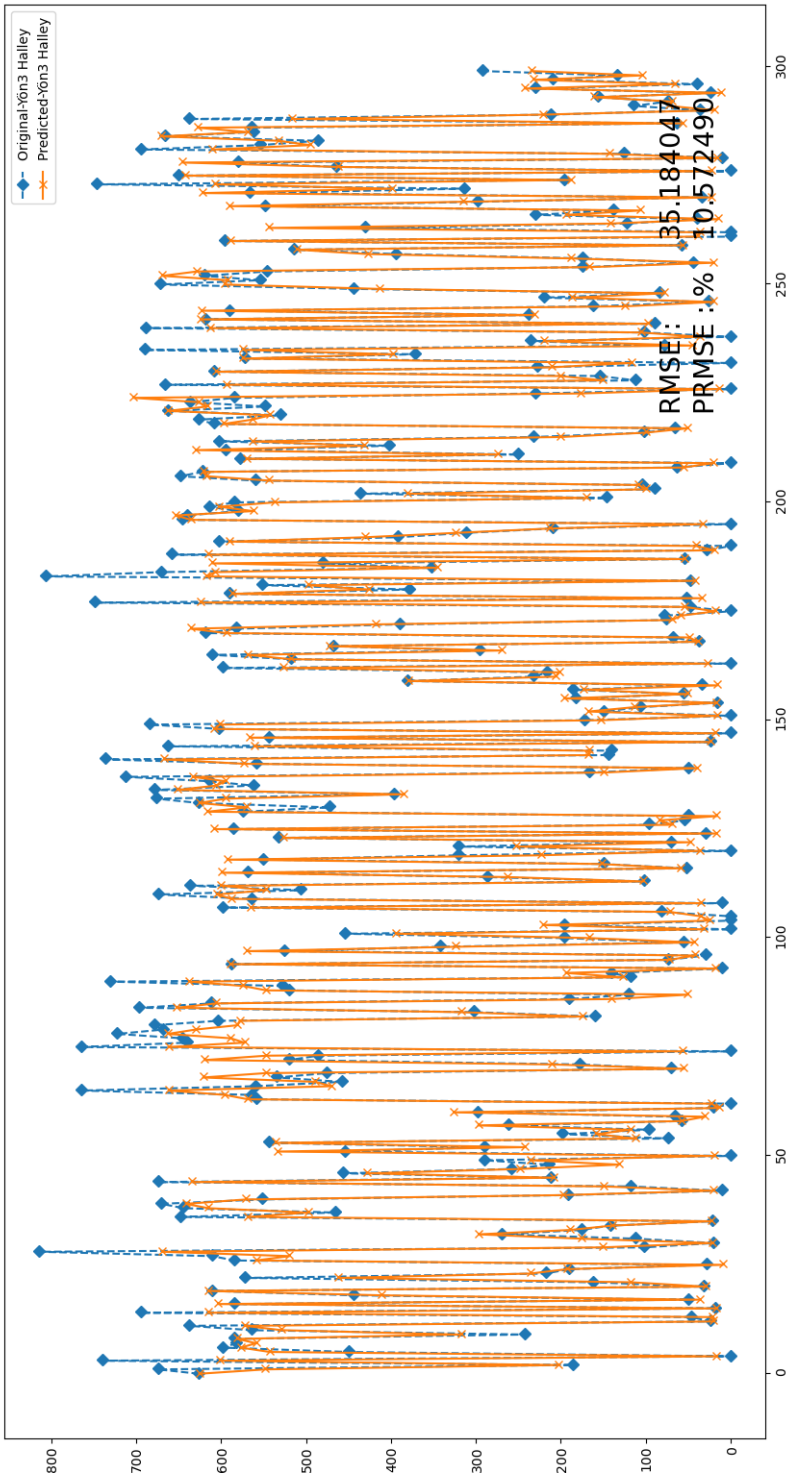


Figure 5.9 : Xgboost Halley Direction 3 Prediction Values

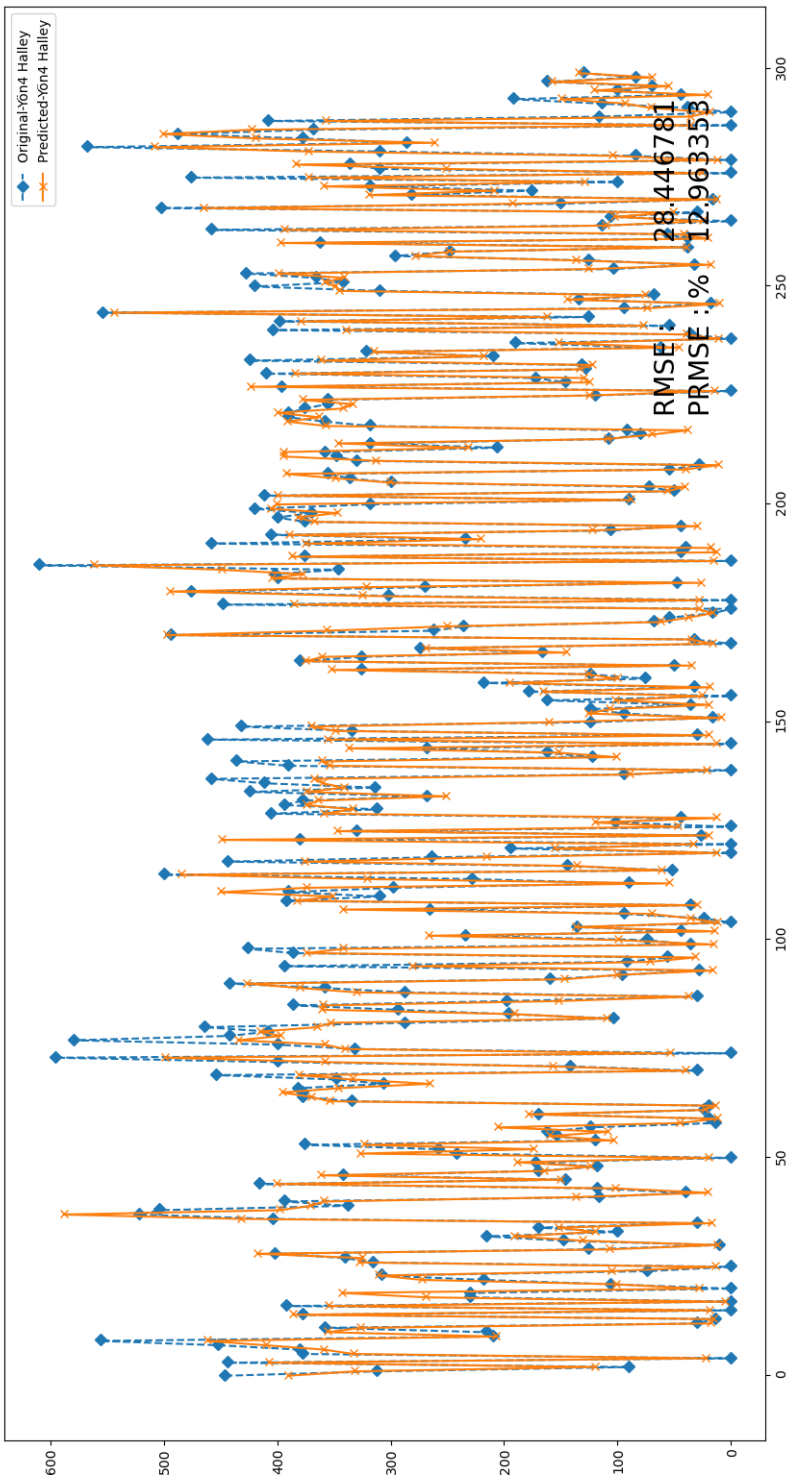
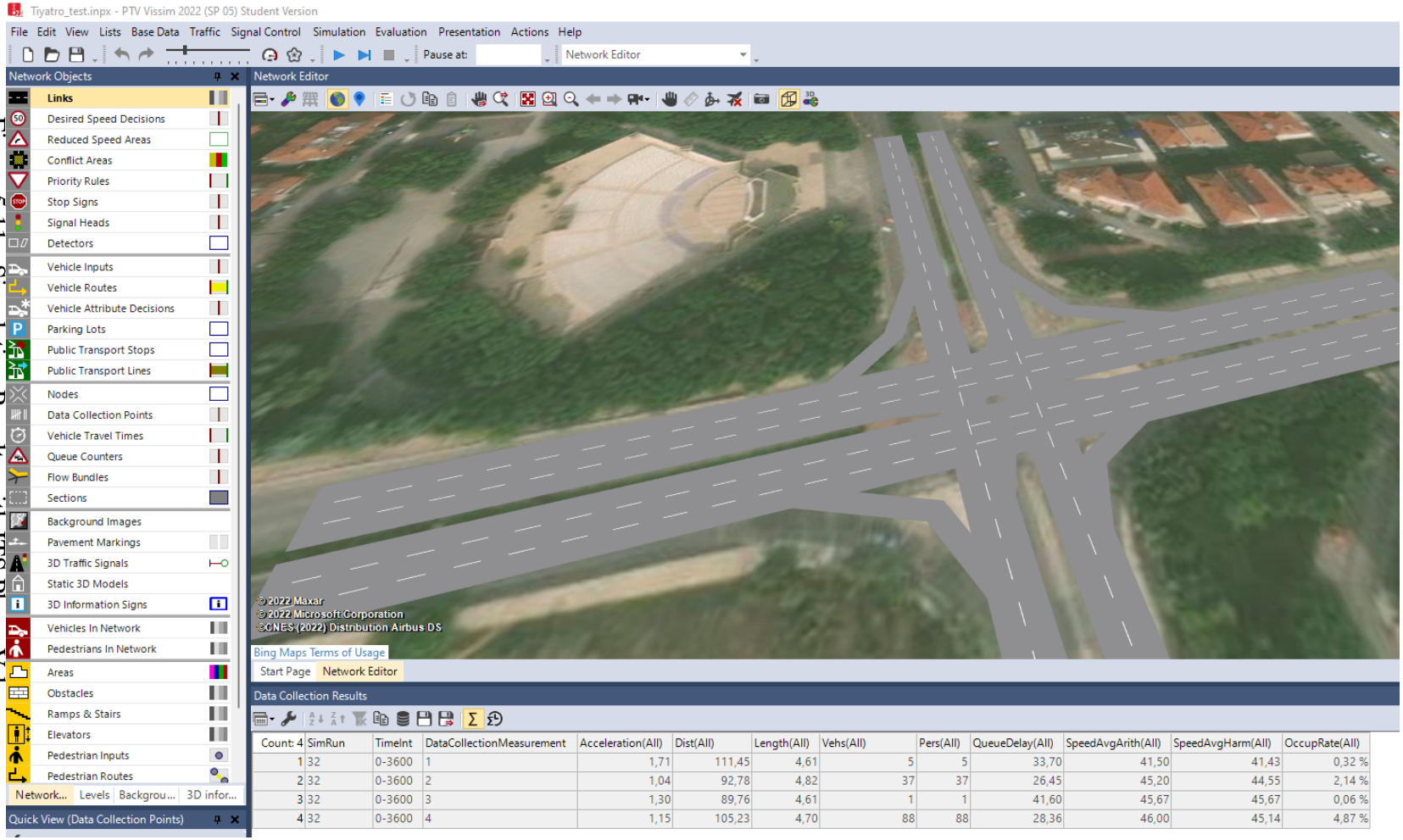


Figure 5.10 : Xgboost Halley Direction 4 Prediction Values



Count: 4	SimRun	TimeInt	DataCollectionMeasurement	Acceleration(All)	Dist(All)	Length(All)	Vehs(All)	Pers(All)	QueueDelay(All)	SpeedAvgArith(All)	SpeedAvgHarm(All)	OccupRate(All)
1	32	0-3600	1	1,71	111,45	4,61	5	5	33,70	41,50	41,43	0,32 %
2	32	0-3600	2	1,04	92,78	4,82	37	37	26,45	45,20	44,55	2,14 %
3	32	0-3600	3	1,30	89,76	4,61	1	1	41,60	45,67	45,67	0,06 %
4	32	0-3600	4	1,15	105,23	4,70	88	88	28,36	46,00	45,14	4,87 %

Figure 5.11 : Simulation Results with ITS Phase Values

Count: 4	SimRun	TimeInt	DataCollectionMeasurement	Acceleration(All)	Dist(All)	Length(All)	Vehs(All)	Pers(All)	QueueDelay(All)	SpeedAvgArith(All)	SpeedAvgHarm(All)	OccupRate(All)
1	21	0-3600	1	1,77	111,70	4,55	4	4	33,65	43,04	42,84	0,25 %
2	21	0-3600	2	1,12	92,75	4,36	33	33	31,85	45,31	44,56	1,78 %
3	21	0-3600	3	1,50	89,27	4,49	2	2	26,30	40,74	40,18	0,13 %
4	21	0-3600	4	0,99	105,33	4,39	92	92	19,88	46,59	45,63	4,63 %

Figure 5.12 : Simulation Results with Prediction Model Phase Values

Bibliography

[1] **ŞİRİN, E.**, (2019), Support Vector Machine (SVM) Ile Sınıflandırma: Python örnek uygulaması, https://www.veribilimiokulu.com/support-vector-machine-svm-ile-siniflandirma-python-ornek-u

[2] **AKCA, M.F.**, (2020), Nedir Bu Destek Vektör Makineleri? (Makine öğrenmesi serisi-2), https://medium.com/deep-learning-turkiye/nedir-bu-destek-vekt%C3%B6r-makineleri-makine-%C3%B6%C4%9Frenmesi-serisi-2-94e576e4223e.

[3] 1.10. decision trees, https://scikit-learn.org/stable/modules/tree.html.

[4] **Placek, M.**, (2021), Car production: Number of cars produced worldwide, https://www.statista.com/statistics/262747/worldwide-automobile-production-since-2000/.

[5] **Zhang, K. and Batterman, S.** (2013). Air pollution and health risks due to vehicle traffic, *The Science of the total environment*, *450-451*, 307–316.

[6] **Makino, H.**, **Tamada, K.**, **Sakai, K. and Kamijo, S.** (2018). Solutions for urban traffic issues by ITS technologies, *IATSS research*, *42*(2), 49–60.

[7] **Nihan, N.L. and Holmesland, K.O.** (1980). Use of the box and Jenkins time series technique in traffic forecasting, *Transportation*, *9*(2), 125–143, https://doi.org/10.1007/BF00167127.

[8] **Davis, G.A. and Nihan, N.L.** (1991). Nonparametric regression and short-term freeway traffic forecasting, *Journal of Transportation Engineering*, *117*(2), 178–188.

[9] **Smith, B. and Demetsky, M.** (1994). Short-term traffic flow prediction models-a comparison of neural network and nonparametric regression approaches, *Proceedings of IEEE International Conference on Systems, Man and Cybernetics*, volume 2, pp.1706–1709 vol.2.

[10] **Hinsbergen, C.**, **Lint, J. and Sanders, F.** (2007). Short Term Traffic Prediction Models, *14th World Congress on Intelligent Transport Systems, ITS 2007*, *7*.

[11] **Dong, X.**, **Lei, T.**, **Jin, S. and Hou, Z.** (2018). Short-term traffic flow prediction based on XGBoost, *2018 IEEE 7th Data Driven Control and Learning Systems Conference (DDCLS)*, IEEE, pp.854–859.

[12] **Ravza Nur YİĞİT, S.H.** (2021). Kısa Zamanlı Trafik Tahmini ile Devre Süresi Optimizasyonu ve Gecikme Analizi, *Teknik Dergi*.

[13] **Sun, B.**, **Sun, T. and Jiao, P.** (2021). Spatio-Temporal Segmented Traffic Flow Prediction with ANPRS Data Based on Improved XGBoost, *Journal of Advanced Transportation*, *2021*.

[14] **Wang, Y.**, **Fang, S.**, **Zhang, C.**, **Xiang, S. and Pan, C.** (2021). TVGCN: Time-Variant Graph Convolutional Network for Traffic Forecasting, *Neurocomputing*.

[15] **Mohri, M.**, **Rostamizadeh, A. and Talwalkar, A.** (2018). *Foundations of machine learning*, MIT press.

CURRICULUM VITAE

Name Surname: Osman Dogukan URKAN

Doğ Date and Place of Birth: 06.01.1995 - Uşak

Email: osman.urkan@hotmail.com

Status of Education:

- **High School:** 2013, Usak High School
- **License:** Pamukkale University, Faculty of Engineering, Computer Engineering

Professional Experiences and Awards:

- Internship I: Leonardo Turkey Aerospace Defense and Security Systems Inc., 2020
- Internship II: Eski16 Technology Industry and Trade Limited Company, 2021