Traffic Prediction

Geromichalou Olga

SID: 3308210012

Supervisor:	Prof. C. Tjortjis
Supervising Committee	
Members:	Prof. P. Bozanis Dr. L. Akritidis

SCHOOL OF SCIENCE & TECHNOLOGY
A thesis submitted for the degree of
Master of Science (MSc) in Data Science
DECEMBER 2022 THESSALONIKI– GREECE

Abstract

This dissertation was written as a part of the MSc in Data Science at the International Hellenic University.

Traffic Prediction is an intelligent scheme of forecasting the traffic flow of a specific place. It is the most critical part of any traffic management system in a smart city. Accurate prediction could decrease accidents and time waste and even increase the quality of life of the citizens. That is why; the research of this topic is of the essence.

In this thesis, a dataset with traffic flow of 6 different Crosses of unknown place is used with Machine Learning and Deep Learning models. Thus, in order to predict the traffic flow regression models as Linear Regression, Random Forest, Multi Layer Perceptron (MLP) and Gradient Boosting are utilized. Other techniques of analyzing the data were adding “time” features and taking another time interval between the observations of the time series, which concluded to better results. Furthermore, the regression problem has been converted into a classification problem and classifiers such as K-Nearest Neighbors (KNN), Support Vector Classifier (SVC), Adaptive Boosting (Adaboost), Decision Tree, Random Forest, Gaussian Naïve Bayes Classifier (GaussianNB) and Extra Trees are used for experimentation. Last, Long short-term memory (LSTM), that the literature review suggests as one of the top deep learning models to predict traffic flow, was utilized and tuned for our case. Indeed, LSTM outperformed the other models with regards to RMSE metric. At each analysis the according statistical metrics have been calculated to compare the different models and choose the optimal one. In our case, for regression as mentioned the LSTM model was the best one and for classification the Extra Trees and the Random Forest classifiers. Cross Validation and Grid Search had also used in search of optimal models.

For the regression problem, a technique that is utilized is that the machine learning models used the data not only of one Cross but of another highly correlated Cross.That results to better models with regards to R^2 metric. Thus, different kind of approaches are examined for this univariable type of problem and acquired better results than the classic regression problem.

Keywords: Mobility, Smart cities, Machine Learning, city intelligence, traffic prediction

Table of Contents

List of Figures

List of Tables

1 Introduction

Time is money and fuel cost a lot of money. According to the Texas A&M Transportation Institute's assessment (2019), the average American commuter wastes 54 hours per year due to traffic delays. That's a two-and-a-half-day trip. That's an extra weekend in a row. That's a full television show. That number may seem priceless to hundreds ofcommuters in big cities, yet it merely represents the average time wasted. With the growth of cities and increasing urbanization, traffic forecasting has become an important and difficult aspect of traffic management. The number of vehicles on the road has increased as a result of population growth, vehicle purchases, and migration to urban areas, resulting in traffic congestion, accidents, and increased travel time. Traffic congestion is expensive in terms of wasted time and energy and that is why this problem should be addressed. If this is done, the improvement of the quality of life around the globe is inevitable.

The aim of this study is to utilize existing traffic flow data with the help of Machine Learning and Deep Learning to predict the number of cars that come through 6 different Crosses and the correlations between them. Generally, the future state of traffic parameters is predicted, in order to take some proactive measures and avoid undesirable traffic circumstances such as congestion and accidents.

Before the case study is explained, a literature part is described in order to clarify all the parts of the problem. Specifically, the definition of the Smart Cities and the different types of traffic data sources, models and problems are pointed. Furthermore, a literature review which depicts the development of the field and the models that have been used at similar cases and how they have progressed are highlighted. Before the data and case study description, the machine and deep learning models that have been used at this study are explained.

The methodology of using the time series observations of lag 13 with the help of sliding windowtechnique as features for the machine learning regression models, the import of new “time” features due to the fact that there is no time stamp, the time series analysis, the converted classification problem and the tuning of the deep learning model LSTM are being presented at chapter 4.

2 Background

2.1 Smart cities

“Smart city, the important strategy of IBM, mainly focuses on applying the next-generation information technology to all walks of life, embedding sensors and equipment to hospitals, power grids, railways, bridges, tunnels, roads, buildings, water systems, dams, oil and gas pipelines and other objects in every corner of the world, and forming the Internet of Things via the Internet”[7]. This is one of the definitions of Smart City Concept that the IBM uses from a technological aspect. Despite the fact that Smart Cities are a state-of-the-art concept and flourish among new ideas and research, its definition it is not clear and without consistent meaning [8]. As the population of earth increase and urbanization is inevitable due to new job opportunities and better infrastructures and generally life, problems like traffic cognation, air pollution, human health, waste and infrastructure management arise. A strategy to alleviate these problems makes a city “Smart”.

The forward-looking way of monitoring and integrating the city’s conditions hold the collective intelligence that makes itself smarter. This means, a more efficient, sustainable and livable city which is accomplished through new intelligent computing technologies [9]. The difference between smart cities and sustainable ones are that former do not only concentrate on environmental sustainability but furthermore on the quality of life and financial growing. Some of the additional benefits are justice in income and job opportunities, basic services, social infrastructure and transportation [8].

Smart mobility is of the essence since it affects the daily life of many citizens. One example of how big data can provide solutions on the field is a road toll system. This provides a plethora of detailed, "real-time" data about the passage of cars through toll gates. The risk of congestion happening in particular city districts can be identified by offline analysis of historical traffic data. When these trends are later discovered in "real-time" data, they offer the opportunity of modification of the traffic management system in order to avoid such issues. Another way of helping the decision making of a driver is the choice of optimal route in the roads of the city. Furthermore, various amenities like public wireless networks, electric vehicle charging stations, and bicycle lanes have become popular in such new developments. Similar instances can be seen in many of the domains that cities are in responsibility of. These data will provide new insight, innovation and opportunities [15].

2.2 Traffic data sources

Historically, at first, traffic data was collected and analyzed to respond to current circumstances, but after that the tendency shifted to proactive traffic management [2]. Asmentioned, the future state of traffic parameters is predicted, in order to take some proactive measures and avoid undesirable traffic circumstances such as congestion and accidents. Traffic management agencies are controlling the timing of intersection signals, monitoring the road network, active demand and traffic management, and managing an emergency [1].

The prediction can also be classified into long-term or short-term. It is obvious what each category is. As their name suggests, the short-term is for predictions that their timeframe is minutes, hours or days and the long-term is for weeks, months or years [26]. Our problem is a short-time traffic flow classification.

What type of data should be taken into account in order to make good predictions? What are the existing tools and techniques for enabling active traffic management? These are some questions that were tried to answer.

First, the research will be affected by the quality of the data you have. Accurate data in a specific format from a trusted source is required for traffic prediction. As Ashwini [1] presents, there are three categories for collecting and managing traffic data: Other-Agency Data Sources, Supervised Data Sources, Unsupervised Data Sources.

If the origin of traffic data is direct and they are collected under the control of road traffic management authorities, then the sources are known as supervised data sources. The collection of traffic datais accomplished by deploying traffic monitoring equipment beside road infrastructure [3] and cars. The structure of the data is the standard format, and these data sources are most extensively utilized.

Unsupervised data sources are sources of traffic data that are not supervised. Unlike supervised data sources there are no traffic agencies collecting traffic data, and the data collection procedure is not meant to collect road traffic data. Unsupervised traffic data sources include data from the Global System for Mobile Communications

(GSM), Long Term Evolution (LTE) connections, and social media. These sources are promising since they do not require infrastructure or conservation.

Other-agency data sources give information that is not immediately connected to traffic but has the potential to effect traffic in the future. Data on planned events like rallies, procession, other public gatherings, and other unexpected occurrences [4] as accidents, shootouts, riots and so onare some examples.Meteorological departments and urban management organizations can collect data about the weather that can affect the traffic in the roadssuch as floods and earthquakes, rainfall, snowfall, and landslides etc. The government or other private portals can extract this information. It can improve real-time traffic prediction accuracy when paired with supervised and unsupervised data sources.

The type of data used for the traffic prediction processes are mainly spatio-temporal data gathered by sensors or other type of data which are collected for computer vision purposes or from crowd sourcing and wireless network data.

A sensor is a device that detects and responds to changes in its environment, such as movement, temperature, humidity, light, pressure, and so on. A sensor's response is a signal that is converted into a readable format and sent to a processor for further processing[1]. Traffic data is collected using a variety of sensors installed on roads. Inductive loop detectors, magnetic and pneumatic tube sensors are examples of intrusive sensors, while radar sensors, laser beams, and infrared sensors are examples of non-intrusive sensors [5]. Based on mobility, sensors in traffic management can be categorized as static sensors or dynamic sensors as Ashwini [1] suggests. For our research the data was indeed collected by static sensors.

2.3 Traffic models

There is a big impact on the tactics we can take to solve our prediction problem if we use real time data which have an efficient good quality. For example, taxis and buses frequently have location systems that can be utilized for traffic organization and tracking [10]. Despite of the application, previous data is required additional to a source of real-time data. When we use data-driven methods to find our prediction, which are techniques that link traffic conditions to external data sources like weather, incidents, constructions, and other special situations, but they ignore network topology in general [12], we rely on the network’s past to predict its evolution. On the other hand, model-driven techniques, that attempt usually through simulation to represent the road network computationally and study the driver's behavior[13], require estimating the traffic stimulation parameters. This topology information is obtained from Traffic Management Bureaus or other public institutions. While it would be great to know the exact location of every car in the network, approaches must realistically deal with incomplete and frequently derived observations. Usual objectives attempt to explain the status of the network in terms of vehicle density, traffic flow, and average velocity at a certain point. Travel time and the length of the line of the cars are two other typical yet frequently derived targets [11]. For road management, it is frequently necessary to predict transmission on road networks as a result of changes in traffic conditions [14], weather, events, or road construction. Ensure that travel time in major routes stays within established levels for emergency crews to arrive at accidents faster and safely.

Model-driven methods may be able to provide beneficial knowledge about unnoticeable sections of the network through simulation, for example, could suggest other routes for routing applications or estimate the number of people that will need to use public transportation for helping public authorities. Additionally, when it is decided that new infrastructure will change the topology of a city, for example if in the place of a road will be constructed a new building or park, then model-driven models can help with the long-term prediction in order to make the optimal choice of the place. Data-driven methods are constrained by the data provided. However, the relationship between the available inputs and the forecast output is crucial. Good forecasts require good data, and the availability of that data influences our options for forecasting future network states.

Some examples of model-driven methods are DynaMIT, DynaSMART-X, VISTA, TRANSIMS and DYNEMO. And data-driven models: An Improved K-nearest Neighbor Model, A Hidden Markov Model for short term prediction, Freeway Traffic Estimation in Beijing based on Particle Filter, Bayesian Combined Neural Network Approach, Adaptive Kalman filter approach [11].

2.4 Traffic problems

According to H. Yuan[6] there are different traffic prediction problems that can be addressed:

Traffic status prediction: When planning a trip from one location to another, it is common to use the navigation system of an electronic map to avoid congested highways. The ability to predict which routes will be crowded in the future is critical to meeting the goal. The worse the traffic situation, the slower the traffic speed or the longer the journey duration. As a result, traffic status prediction can be considered a regression problem, along with traffic speed and trip time prediction. Furthermore, by dividing the traffic speed into several continuous periods, we may measure the traffic status with different types like smooth, mild congestion, and heavy congestion, making predicting the traffic state a classification problem.

Traffic flow prediction: Forecasting traffic flow in the future is critical since excessive traffic has resulted in a number of stomping events. There are two sorts of traffic flows: network-based and region-based. The first type deduces the number of vehicles from loop detector sensors positioned at both route ends. For the second type, we divide the city into various regions and count the number of people moving from one region to the next as region-based traffic flow. As a result, regional traffic flow can be separated into in-flow and out-flow.

Travel demand prediction: Users can order a taxi online through transportation firms. They need to forecast people's travel needs so that vehicles can be dispatched more efficiently to different places. During morning rush hour, for example, they should send more vehicles to residential areas and at the afternoon they should send additional vehicles to office zones. Predicting travel demand is commonly done based on regions, thus we refer to it as region-based travel demand prediction.

Traffic classification: A binary classification problem is, for example, given a taxi's current trajectory, the usage of classification algorithms to determine if it is normal or not, and so advise the driver to alter the path in a timely manner. There are also some problems with multiple classification. Distinct means of transportation, for example, should provide different types of trajectories like walking, bus, subway, and taxi. Additionally, another classification is dividing different sorts of paths into different types of modes. Existing research focuses primarily on machine learning methods to solve the classification problem, such as the hidden Markov model [23], conditional random field, and decision tree, are classified as traditional learning methods, whereas convolutional neural network and recurrent neural network are classified as deep learning methods.

Traffic generation: As deep learning techniques advance, an increasing number of deep learning models are being developed to handle traffic prediction problems, and these models require a considerable amount of training data to increase their accuracy. However, because it is difficult for ordinary people to collect real-world traffic data, producing data is an effective solution to overcome this problem. Also, some applications require a transportation environment to test various ideas. But, due to a scarcity of all types of real-world traffic data, using a real-world setting is unrealistic. As a result, simulating the environment by generating various types of traffic data is beneficial. Moreover, when using real-world data to train traffic prediction models, we must consider privacy protection. As a result, one of the research important issues is how to avoid compromising users' privacy without reducing the effectiveness of trained models. As a conclusion, we have that there are two sections, simulation and completion. The first is when we strive to use collected data to mimic the transportation environment, in which we infer the distribution of traffic data and produce unseen data from other sparse data and then we can fill in missing or sensitive data with fake data. The second one is how to generate unique data in order to solve various prediction problems. Deep learning approaches, such as K-nearest neighbors, generative-adversarial networks, and RNN, are the most common ways of performing these tasks.

In summary, the types of traffic forecast difficulties listed above correlate to the perspectives of the three groups: crowds, governments, and linked businesses. As a result, in the field of transportation, understanding how to tackle these traffic forecast challenges is becoming increasingly crucial [16]. Our research is a traffic flow prediction problem.

2.5 Related Work

Traffic flow prediction as it is described before is an important issue since it can help daily life that is why it is studied from 1970s.The models that have been used are divided into four categories. The first category and more utilized one at the past years are the parameter models (or model-driven methods). These are models with fixed structure whose parameters are trained with regards to the empirical data. ARIMA [17] is the first model that has been used and other modified version like Kohonen ARIMA [18], Seasonal ARIMA [19] and Subset ARIMA [20] have been followed. In order to predict with this type of models, stationarity of mean and variance are obligatory, but traffic data have stochasticity and sheer length of non-linear nature. This type of models and Linear Regression can perform well during normal conditions but do not on external system changes [45]. Despite their simplicity and understandability, they produce bigger prediction error [21]. Other models are Markov chain [25, 36],Bayesian network [27], Kalman filter [28] which also have some preconditions as normal distribution of the residuals and the stationarity of the time series. The non linearity and randomness of the traffic data make this type of models to be not appropriate at transportations [29].

Non-parameter models (or data-driven methods) have not a fixed structure and parameters. Some examples are Random Forest [46], K-nearest neighbors (KNN) and Artificial Neural Network (ANN) [37].This type of models can fit all the functions to arbitrary precision, but they might fail more easily to local minimums and overfitting happens too due to the complexity of the models. Generally, Neural Networks [30] are famous as they include fully connected layers and radial bases functions (RBFs), but they are too shallow to be used with big amount of data [29]. Thus, Deep Learning has a majorsuccess, not only in the field of traffic flow prediction, but generally in image and video classification, natural language processing etc. Some examples areStacked Auto Encoder [31], Deep Belief Network [32], Recurrent Neural Network (RNN), Convolutional Neural Networks (CNN), Feed Forward Neural Network (FFNN) [22], GRU [21],and LSTM [24]. RNN was used first to predict the traffic as it manages correlations between data at different moments efficiently, but since the gradient vanishing or exploding problem exists, other type of Deep Learning models was preferred [21]. LSTM has the ability to control when to forget some information and that is why performs better than ARIMA and generally all non-parameter models when it comes to traffic flow prediction [24]. This happens, also, due to the excellent memorization of long-term dependencies [21].

Combinatory models are models that have been used as a combination of other single models in order to exclude the benefits of each model and resolve their problems. Some examples are combination of: KNN and LSTM [33], Support Vector Regression and LSTM [34], RNN and LSTM [35]. Their accuracy is high, but they are not widely performed due to their complexity and pour real time performance [29]. Other models are combinations of clustering algorithm (unsupervised learning) to separate the data to cluster with regards to their similarity and a supervised model.

Such examples are: Gaussian mixture model clustering algorithm with an artificial neural network [38], DBSCAN and ARIMA, KNN and Support Vector Regression [39], clustering algorithms with CNN and LSTM [40].

New methods are performed with traffic flow data that have been combined with other type of data in order to increase the accuracy like weather data [41], accidents, opening hours of stores [43], or even spatial data that have created a new type of Deep Learning model: Traffic Graph Convolutional Long Short-Term Memory Neural Network (TGC-LSTM) that learns the interactions between highways in the traffic graph [42].

Simulation models are the last type of models that is used for traffic prediction. These models utilize traffic simulations in order to predict the according traffic flow [45].

With regards to preprocessing, missing values are a common problem in traffic prediction. In order to overcome it, techniques like historical average [21] or Bayesian method and Gaussian Mixture model [44] are proposed.

2.6 Machine and Deep Learning models

The machine learning models that have been performed for regressionare Linear Regression, Random Forest, MLP, Gradient Boosting and LSTM and for classification: Decision Tree, Random Forest, KNN, MLP, SVC, AdaBoost, Extra Tree Classifier and Gaussian NB. Next, we present their characteristics and type of modeling.

2.6.1 Linear Regression

In order to model the relationship between two variables, linear regression fits a linear equation to the observed data. The first variable is regarded as an explanatory (independent) variable, whereas the second is regarded as a dependent variable. There can be more than one independent variable. This type of analysis calculates the coefficients of the linear equation. The differences between expected and actual output values are minimized by linear regression by fitting a straight line or surface. The best-fit line for a set of paired data can be found using straightforward linear regression calculators that employ the "least squares" technique. Thus, a minimization of the squared difference of the predicted and actual value is performed [49].

2.6.2 Decision Tree

Decision Tree solves the problem of machine learning by transforming the data into tree representation. They consist of nodes and capture descriptive decision-making knowledge from the supplied data. In their tree structure, the leaves represent the class labels, while the brunches depict the features that have been combines to form the class labels. The con of this model is the understandability of humans, as only by the structure of the model many conclusions can be made [50].

2.6.3 Multi Layer Perceptron (MLP)

Multilayer Perceptron is a supervised classification method fully connected class of feedforward Artificial Neural Network (ANN). The structure of MLP is: an input layer, at least a hidden layer and an output layer. Each node, expect the input layer, applies a nonlinear activation function, which determine if the layer would be "ON" (1) or "OFF" (0), depending on input. Thus, only a small number of nodes are used and that reduce the overfitting. The training technique that has been utilized is backpropagation, with which the gradient of the loss function has been computed with respect to the each weight and gradient methods are used to minimize the loss, such as gradient descent. MLP is preferred for linearly separable data [51].

2.6.4 Random Forest

Random forest is an ensemble method. First, a number of decision trees are trained on a random sample with replacement from the original data with a size of the original training set. At each node split, a subset of the input variables is randomly selected to search for the best split. For classification, the final prediction is given by majority voting and for regression, by averaging the prediction of each decision tree. Some hyperparameters are:

max_feutures (to consider when looking for best split), min_samples_split (minimum number of samples that are needed to split an internal node, it limits the size of the tree), min_sample_leaf (minimum number of samples needed to create a leaf, it removes split candidates that are on the limits) and max_depth (limits the depth of the tree) [46].

2.6.5 Gradient Boosting

Boosting methods produce sequentially base models by emphasizing on the training cases that are hard to estimate. Thus, the combination of a number of weak models into a single high accurate one is performed. Here, the weak learners are decision trees that predict the residuals, which is the difference of the current prediction and the real value. The loss function is the squared error and in order to minimize the loss, gradient descent is being used. With gradient descent, the algorithm takes repeated steps towards the steepest descent, which is the opposite direction of the gradient, to find the local minimum Some hyperparameters are: learning_rate, max_depth,max_features , min_samples_split (of the decision trees), subsample (the size of the random samples) [48].

2.6.6 Extra Trees Classifier

Extremely randomized trees or Extra Trees are an ensemble method of individual trees, but they differ from regular random forests in two ways. First, each tree is trained using the entire learning sample and second, the top-down splitting in the tree learner is randomized. The node is then split using the split with the greatest score out of all the randomly generated splits. The number of randomly picked features to be examined at each node can be specified, just like in standard Random Forests [52].

2.6.7 Adaptive Boosting (Adaboost)

Adaptive Boosting or AdaBoost is a statistical classification meta-algorithm that is used to improve the performance of other models. The results of the other learning algorithms, or "weak learners," are merged to create a weighted total that represents the boosted classifier's final results. Weak learners are updated in favor of the instances that the prior classifiers misclassified, that subsequent classifiers focus more on difficult cases. The scikit-learn model that we use, provides decision trees as weak learners [54].

2.6.8 K-Nearest Neighbors (KNN)

K-Nearest Neighbors or KNN algorithm is a non-parametric supervised learning method that can be used to solve regression and classification problems. The input is the k nearest training observations of the dataset. For classification, the class that is given to the test data is the most common of the k nearest observations (majority voting). For regression, we take the average of the k nearest neighbors [55].

2.6.9 Support Vector Classifier (SVC)

Support vector classifier or SVC is a supervised machine learning model that is used for classification problems. It finds the best hyperplane, which maximizes the margin between the classes that separate the data by mapping the data points to a high-dimensional space. Thus, the data points can be categorized, even when the data are not otherwise linearly separable [56].

2.6.10 Gaussian Naïve Bayes Classifier (GaussianNB)

Gaussian Naïve Bayes Classifier is a probabilistic classification algorithm based on applying Bayes' theorem with strong independence assumptions. The “naïve” assumption that is made is that the continuous values associated with each class are distributed according to a normal (or Gaussian) distribution. Thus, probabilities are used to classify the data, finding the likelihood of each point to be in the according class [57].

2.6.11 Long short-term memory (LSTM)

Long short-term memory or LSTM is an artificial neural network and a deep learning model. Standard neural networks are feedforward, but LSTM has feedback connections too. The parts of this model are: input gate, output gate and forget gate. The forget gate chooses when to forget the output results and thus chooses the optimal time lag for the input sequence. The cell remembers values over arbitrary time intervals. It is preferred for time series prediction as there can be lags of unknown duration between important events [21]. Throughout the training the weight and the bias of each gate are calculated from the historical time series and the features of historical states are recognized and retained [29].

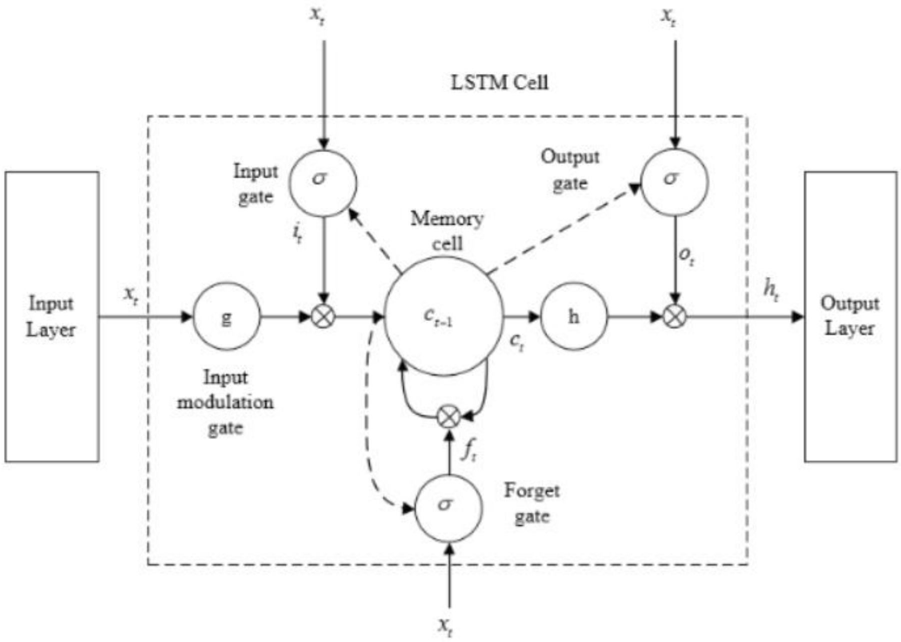


Figure 1: Structure of LSTM Cell [21]

2.7 Statistical Metrics

In order to evaluate the performance of the predictions some statistical metrics are calculated. For regression:R^2, MAE, RMSE, EVS and for classification: Accuracy, Precision, Recall, F1 score and Support.

2.7.1 Statistical Metrics for Regression

The metrics are calculated from the next formulas:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{t=1}^{n}(x_t - \hat{x}_t)^2}$$

$$\text{MAE} = \frac{1}{n}\sum_{t=1}^{n} |\, x_t - \hat{x}_t|$$

$$R^2 = 1 - \frac{\sum_{t=1}^{n}(x_t - \hat{x}_t)^2}{\sum_{t=1}^{n}(x_t - \overline{x_t})^2}$$

$$\text{EVS} = 1 - \frac{Var\,\{x_t - \hat{x}_t\}}{Var\,\{x_t\}}$$

, where x_t is the observed value, $\hat{x}_t$ is the predicted value, $\overline{x_t}$ is the average value of the observed valueand n the number of all the observations.

Root Mean Square Error (RMSE) is sensitive to the extremely large or extremely small error; thus it cannot give a fair assessment in different conditions. Meanabsolute

Error is a scale-dependent accuracy measure and therefore cannot be used to make comparisons between series using different scales. The determination coefficient (R^2) revealsthe degree of similarity between the predicted value and observed value. Explained Variance is used to measure the proportion of the variability of the predictions and the difference with the $R^2$is that it does not account for systematic offset in the prediction.

R^2 and EVS take values between 0 and 1. The higher the metric, the better the model as both represent the proportion of variance that could be explained by the independent variables.

2.7.2 Statistical Metrics for Classification

Accuracy: measures the number of correct predictions made divided by the total number of predictions made, multiplied by 100 to turn it into a percentage.

Precision is the proportion of positive identifications that were actually correct. Recall is the proportion of actual positives that was classified correctly.

Precision = $\frac{TP}{TP+FP}$

Recall = $\frac{TP}{TP+FN}$, where TP is true positives, FP is false positives and FN is false negatives.

Support: the number of actual occurrences of the class in the specified dataset. Imbalanced support in the training data may show structural weaknesses and could signify the need for stratified sampling or rebalancing.

F1 score: combines the precision and recall of a classifier into a single metric by taking their harmonic mean. F1 = 2 $\frac{\text{precision} \quad \text{x recall}}{\text{precision} \quad + \text{recall}}$.

3 Case Study

The dataset that has been used in this research is an open dataset from the site [63]. It is a public (anonymized) road traffic prediction datasets from Huawei Munich Research Center. It is composed of 6 columns each with the traffic volume of a cross every 5 minutes for 56 days. There are no missing values.

	Cross 1	**Cross 2**	**Cross 3**	**Cross 4**	**Cross 5**	**Cross 6**
0	105.0	48.0	30	62.0	31	110.0
1	97.0	41.0	32	55.0	42	103.0
2	76.0	47.0	44	58.0	40	100.0
3	98.0	40.0	39	59.0	43	104.0
4	87.0	41.0	47	49.0	35	112.0
...	...	...	...	...	...	...
16123	85.0	37.0	34	56.0	35	89.0
16124	71.0	45.0	44	50.0	44	53.0
16125	83.0	34.0	34	61.0	44	77.0
16126	89.0	39.0	25	48.0	32	64.0
16127	66.0	36.0	26	50.0	37	55.0

16128 rows × 6 columns

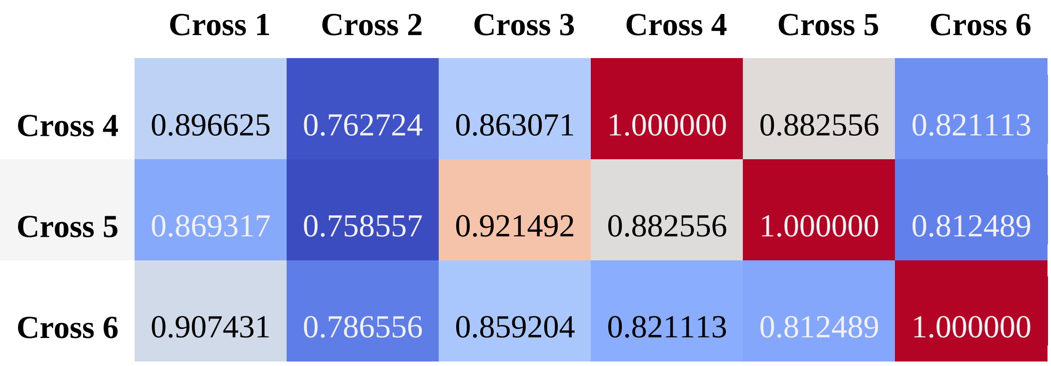
Table1: The dataset

	Cross 1	Cross 2	Cross 3	Cross 4	Cross 5	Cross 6
count	16128.00	16128.00	16128.00	16128.00	16128.00	16128.00
mean	95.806207	45.948444	41.510355	67.656870	36.232453	76.215185
std	87.586717	50.865051	41.849582	68.536141	37.372452	68.101792
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	8.000000	2.000000	1.000000	0.000000	1.000000	5.000000
50%	88.000000	29.000000	32.000000	45.000000	28.000000	66.000000
75%	169.0000	77.0000	73.0000	124.0000	60.0000	139.0000
max	352.5000	302.0000	218.0000	312.0000	193.0000	253.0000

Table2: Dataset description

As we observe for each cross there is difference at the mean and the standard deviation. Thus, the traffic flow is different, and each cross should be predicted with different models, although their correlation hasbeen examined.

	Cross 1	Cross 2	Cross 3	Cross 4	Cross 5	Cross 6
Cross 1	1.000000	0.829850	0.895144	0.896625	0.869317	0.907431
Cross 2	0.829850	1.000000	0.788421	0.762724	0.758557	0.786556
Cross 3	0.895144	0.788421	1.000000	0.863071	0.921492	0.859204



	Cross 1	Cross 2	Cross 3	Cross 4	Cross 5	Cross 6
Cross 4	0.896625	0.762724	0.863071	1.000000	0.882556	0.821113
Cross 5	0.869317	0.758557	0.921492	0.882556	1.000000	0.812489
Cross 6	0.907431	0.786556	0.859204	0.821113	0.812489	1.000000

Table3: Correlation matrix of the 6 crosses

Generally, we could say that the 6 crosses are highly correlated. This might conclude that the crosses are near each other, since the correlations are high. Specifically, more correlated are the crosses 3 with 5 and 1 with 6. For the most correlated crosses, an extra method of analysis has been implemented and will be discussed at next chapter.

The methods that have been used in order to handle this type of data and how to feed the machine and deep learning models are presented next.

4Methodology and Results

Four different machine learning models are being used to predict the traffic flow. The way the machine learning models were implemented was that for each cross a list of 13 observations were made and then an array with the lists as rows. The 13th elements were the target variable. We have a (1239,13) matrix and the last column is our target. In this way, the models have been used the data from the last one hour to predict the traffic flow of the next 5 minutes. The 33% of the dataset have been used as a test set.

4.1 Regression models of the 6 Crosses

These are the results, rounded at the 3rd decimal point.

Cross 1	Linear Regression	Random Forest	MLP	Gradient Boosting
R^2	0.961	0.916	**0.963**	0.962
MAE	11.140	18.055	**10.441**	10.451
RMSE	305.200	651.636	**289.780**	291.204
EVS	0.961	0.916	**0.963**	0.962

Table4: Regression models' results for Cross 1

Cross 2	Linear Regression	Random Forest	MLP	Gradient Boosting
R^2	0.941	0.883	**0.944**	0.942
MAE	7.547	11.516	**7.201**	7.259
RMSE	153.763	303.203	**145.976**	150.672
EVS	0.941	0.883	**0.944**	0.942

Table5: Regression models' results for Cross 2

Cross 3	Linear Regression	Random Forest	MLP	Gradient Boosting
R^2	0.950	0.893	0.954	**0.957**

MAE	6.113	9.612	5.620	**5.519**
RMSE	88.615	187.480	80.961	**75.729**
EVS	0.950	0.893	0.954	**0.957**

Table6: Regression models' results for Cross 3

Cross 4	Linear Regression	Random Forest	MLP	Gradient Boosting
R^2	0.939	0.882	0.940	**0.946**
MAE	10.384	16.338	10.086	**9.562**
RMSE	284.074	551.426	282.220	**253.272**
EVS	0.939	0.882	0.942	**0.946**

Table7: Regression models' results for Cross 4

Cross 5	Linear Regression	Random Forest	MLP	Gradient Boosting
R^2	0.934	0.865	0.937	**0.941**
MAE	6.234	9.325	5.775	**5.692**
RMSE	93.041	190.863	88.744	**82.700**
EVS	0.934	0.865	0.938	**0.941**

Table8: Regression models' results for Cross 5

Cross 6	Linear Regression	Random Forest	MLP	Gradient Boosting
R^2	0.940	0.906	**0.947**	**0.947**
MAE	10.569	15.322	9.942	**9.859**
RMSE	279.211	441.173	249.785	**249.299**
EVS	0.940	0.906	**0.947**	**0.947**

Table9: Regression models' results for Cross 6

As mentioned previously, for each cross we found the correlations with the other crosses. Thus, after making most correlated pairs of crosses we performed regression as before at the array which had the 12 previous observations of both the crosses.

From the combination of cross 3 and cross 5:

Cross 3	Linear Regression	Random Forest	MLP	Gradient Boosting
R^2	0.953	0.893	**0.959**	0.957
MAE	5.900	9.612	**5.381**	5.490
RMSE	82.613	187.480	**72.721**	74.888
EVS	0.953	0.893	**0.959**	0.957

Table10: Regression models' results for Cross 3 with combined data of Cross 5

The Random Forest and the Gradient Boosting model are the same as when we had only the data from Cross 3. The MLP model with the data of Crosses 3 and 5 has performed better than the model with only the data of Cross 3, specifically these results are the best ones overall.

Cross 5	Linear Regression	Random Forest	MLP	Gradient Boosting
R^2	0.937	0.865	0.941	**0.943**
MAE	6.036	9.325	5.649	**5.556**
RMSE	88.472	190.863	83.215	**80.842**
EVS	0.937	0.865	0.941	**0.943**

Table11: Regression models' results for Cross 5 with combined data of Cross 3

The Gradient Boosting model with the data of Crosses 3 and 5 has performed better than the models with only the data of Cross 5.

It can be concluded that our models perform better in the pair of 3,5 Crosses and might the fact that their score of correlation is 0.921492, which is high, can explain that better performance.

From the combination of cross 1 and cross 6:

Cross 1	Linear Regression	Random Forest	MLP	Gradient Boosting
R^2	0.962	0.916	0.963	**0.963**
MAE	11.048	18.055	10.491	**10.422**
RMSE	297.982	651.636	286.418	**286.518**
EVS	0.962	0.916	**0.964**	0.963

Table12: Regression models’ results for Cross 1 with combined data of Cross 6

As it is observed all the models with data of the Cross 1 and 6 have performed better than the model that have only data of Cross 6. Gradient Boosting is the best model overall.

Cross 6	Linear Regression	Random Forest	MLP	Gradient Boosting
R^2	0.945	0.906	0.946	**0.950**
MAE	10.128	15.322	9.775	**9.501**
RMSE	259.070	441.173	251.399	**234.686**
EVS	0.945	0.906	0.949	**0.950**

Table13: Regression models’ results for Cross 6 with combined data of Cross 1

Here the Gradient Boosting model has performed better than the model with the data of only Cross 6.

From the combination of cross 1 and cross 2:

Cross 1	Linear Regression	Random Forest	MLP	Gradient Boosting

R^2	0.961	0.916	**0.963**	**0.963**
MAE	10.987	18.055	**10.320**	10.381
RMSE	297.638	651.636	**287.315**	289.713
EVS	0.961	0.916	**0.963**	**0.963**

Table14: Regression models’ results for Cross 1 with combined data of Cross 2

Here the MLP model has performed slightly better than the models with only the data of Cross 1.

Cross 2	Linear Regression	Random Forest	MLP	Gradient Boosting
R^2	0.943	0.883	**0.946**	0.945
MAE	7.313	11.516	**7.096**	7.132
RMSE	146.795	303.203	**139.812**	143.576
EVS	0.943	0.883	**0.946**	0.945

Table15: Regression models’ results for Cross 2 with combined data of Cross 1

Here the MLPmodel has performed better than the according model with only Cross 2 data.

As it can be observed, Gradient Boosting and MLP are at the most cases the best models. As for the part of combining two crosses, we can say that the results are better but the combination of the data of two Crosses since the correlation score is high could bring better results. The Random Forest model did not change at any time.

We perform a k-fold Cross Validation with k =10 for the Linear Regression model.We find that the R-squared score is:

Cross 1: 0.9659189, 0.96595484, 0.96936975, 0.96902285, 0.96356757, 0.94956341, 0.93554698, 0.93684901, 0.95037277, 0.92597914

Cross 2: 0.95515022, 0.94456067, 0.9422758, 0.9476057, 0.9291853, 0.89777627, 0.91664063, 0.91819215, 0.92561454, 0.90403003

Cross 3: 0.95336154, 0.94187527, 0.95532162, 0.96557446, 0.95626101, 0.92143707, 0.93046322, 0.9258234, 0.93148068, 0.92182499

Cross 4: 0.94600423, 0.944852, 0.94536174, 0.95127612, 0.94946753, 0.91712499, 0.90569881, 0.90462967, 0.9209083, 0.90246712

Cross 5: 0.94280494, 0.93752784, 0.93729922, 0.95420048, 0.94467008, 0.91086693, 0.90039765, 0.88886706, 0.91600568, 0.8676088

Cross 6: 0.94672541, 0.92058743, 0.94416822, 0.95284576, 0.94078967, 0.9059204, 0.90571004, 0.90505474, 0.89937194, 0.91422574

We observe that the best model is MLP for Cross 1,2 and Gradient Boosting for Cross 3,4,5,6. In order to find if there is overfitting we perform a k-fold Cross Validation with k =10.

We find that the R-squared score is:

Cross 1: 0.9678144, 0.96692069, 0.96940653, 0.97164808, 0.96579447, 0.9525738, 0.93667825, 0.9394977, 0.94906603, 0.92441198

Cross 2: 0.95610703, 0.94715238, 0.94728636, 0.95538401, 0.92749027, 0.89803055, 0.9173128, 0.92540314, 0.92804572, 0.903089

Cross 3: 0.96095947, 0.95099629, 0.96233376, 0.96994253, 0.96305763, 0.92828069, 0.93644997, 0.93389864, 0.93747176, 0.92582487

Cross 4: 0.94668785, 0.95617673, 0.95424368, 0.96158093, 0.95479788, 0.92650693, 0.91932942, 0.91394618, 0.92688988, 0.90420288

Cross 5: 0.94835529, 0.94521332, 0.94759903, 0.9645901, 0.95224472, 0.91995733, 0.91267767, 0.90081031, 0.92206375, 0.87556665

Cross 6: 0.94672541, 0.92058743, 0.94416822, 0.95284576, 0.94078967, 0.9059204, 0.90571004, 0.90505474, 0.89937194, 0.91422574

Moreover, an ensemble method has been performed, taking the average result of the three best models: Linear Regression, MLP and Gradient Boosting.

The results are:

	Cross 1	Cross 2	Cross 3	Cross 4	Cross 5	Cross 6
R^2	0.962	0.942	**0.957**	**0.946**	**0.941**	**0.947**
MAE	10.451	7.259	**5.519**	**9.562**	**5.692**	**9.859**
RMSE	291.204	150.672	**75.729**	**253.272**	**82.700**	**249.299**
EVS	0.962	0.942	**0.957**	**0.946**	**0.941**	**0.999**

Table16: Ensemble model's results for each Cross

Here, for the first 2 Crosses this ensemble model is not better. For the other Crosses, the results are the same as the Gradient Boosting model which was the best one. Generally, this ensemble model does not add any progress to the research.

4.2 Inserting new "time" features

Since the dataset did not provide the time stamp of the observations, another way of analyzing these data was adding different variables to the dataset. A new variable called "Minutes" counts the minutes of each day. For example, for the first observation it was 5, the second 10 etc. for each day as a variable that count the time without the time stamp. Also, another variable was "Day of Week" which was a number between 1 and 7, starting as observation 1 is 1… observation 7 is 7, observation 8 is 1 etc. These 2 variables have been added in order to get the concept of time into the model but without the knowledge of the time stamp and in order to see the impact of the weekends and the time of the day like morning and night. After that a slighting window of 12 and moving average of Cross data were performed and then the data was fed to machine learning models.

16105 rows × 40 columns

	Day of Week	Minutes	Day	Cross 6	Minutes T-1	Minutes T-2	Minutes T-3	Minutes T-4	Minutes T-5	Minutes T-6
12	6	65	1	64.0	60.0	55.0	50.0	45.0	40.0	35.0
13	7	70	1	60.0	65.0	60.0	55.0	50.0	45.0	40.0
14	1	75	1	42.0	70.0	65.0	60.0	55.0	50.0	45.0
15	2	80	1	47.0	75.0	70.0	65.0	60.0	55.0	50.0
16	3	85	1	61.0	80.0	75.0	70.0	65.0	60.0	55.0
...	...	...	...	...	...	...	...	...	...	...
16123	3	1420	56	89.0	1415.0	1410.0	1405.0	1400.0	1395.0	1390.0
16124	4	1425	56	53.0	1420.0	1415.0	1410.0	1405.0	1400.0	1395.0
16125	5	1430	56	77.0	1425.0	1420.0	1415.0	1410.0	1405.0	1400.0
16126	6	1435	56	64.0	1430.0	1425.0	1420.0	1415.0	1410.0	1405.0

16127	7	1440	56	55.0	1435.0	1430.0	1425.0	1420.0	1415.0	1410.0
...	**Cross T-3**	**Cross T-4**	**Cross T-5**	**Cross T-6**	**Cross T-7**	**Cross T-8**	**Cross T-9**	**Cross T-10**	**Cross T-11**	**Cross T-12**
12	48.0	89.0	93.0	98.0	89.0	112.0	104.0	100.0	103.0	110.0
13	61.0	48.0	89.0	93.0	98.0	89.0	112.0	104.0	100.0	103.0
14	54.0	61.0	48.0	89.0	93.0	98.0	89.0	112.0	104.0	100.0
15	64.0	54.0	61.0	48.0	89.0	93.0	98.0	89.0	112.0	104.0
16	60.0	64.0	54.0	61.0	48.0	89.0	93.0	98.0	89.0	112.0
...	...	...	...	...	...	...	...	...	...	...
16123	92.0	77.0	135.0	86.0	84.0	100.0	82.0	104.0	85.0	102.0
16124	82.0	92.0	77.0	135.0	86.0	84.0	100.0	82.0	104.0	85.0
16125	50.0	82.0	92.0	77.0	135.0	86.0	84.0	100.0	82.0	104.0

16126	89.0	50.0	82.0	92.0	77.0	135.0	86.0	84.0	100.0	82.0
16127	53.0	89.0	50.0	82.0	92.0	77.0	135.0	86.0	84.0	100.0

Table17: New dataset with “time” features

These are the results of the same models as before for all the Crosses.

Cross 1	RF	LR	MLP	GB
R^2	0.916	0.961	0.956	**0.966**
MAE	18.055	11.037	12.583	**9.918**
RMSE	651.636	305.029	338.672	**265.708**
EVS	0.916	0.961	0.961	**0.966**

Table18: Regression models ‘results for Cross 1 with the new dataset with “time” features

Cross 2	RF	LR	MLP	GB
R^2	0.883	0.941	0.935	**0.948**
MAE	11.516	7.504	8.720	**6.917**
RMSE	303.203	153.556	169.375	**136.045**
EVS	0.883	0.941	0.942	**0.948**

Table19: Regression models’ results for Cross 2 with the new dataset with “time” features

Cross 3	RF	LR	MLP	GB
R^2	0.893	0.950	0.944	**0.961**
MAE	9.612	6.081	6.973	**5.281**
RMSE	187.480	88.286	97.608	**69.189**
EVS	0.893	0.950	0.951	**0.961**

Table20: Regression models’ results for Cross 3 with the new dataset with “time” features

Cross 4	RF	LR	MLP	GB
R^2	0.882	0.940	0.941	**0.950**
MAE	16.338	10.287	10.223	**9.173**
RMSE	551.426	282.934	273.986	**236.171**
EVS	0.882	0.940	0.942	**0.950**

Table21: Regression models’ results for Cross 4 with the new dataset with “time” features

Cross 5	RF	LR	MLP	GB
R^2	0.865	0.934	0.938	**0.946**
MAE	9.325	6.198	6.317	**5.457**
RMSE	190.863	92.536	88.030	**75.719**
EVS	0.865	0.934	0.938	**0.946**

Table22: Regression models' results for Cross 5 with the new dataset with "time" features

Cross 6	RF	LR	MLP	GB
R^2	0.906	0.940	0.941	**0.951**
MAE	15.322	10.501	10.880	**9.427**
RMSE	441.173	279.148	275.602	**231.129**
EVS	0.906	0.940	0.942	**0.951**

Table23: Regression models' results for Cross 6 with the new dataset with "time" features

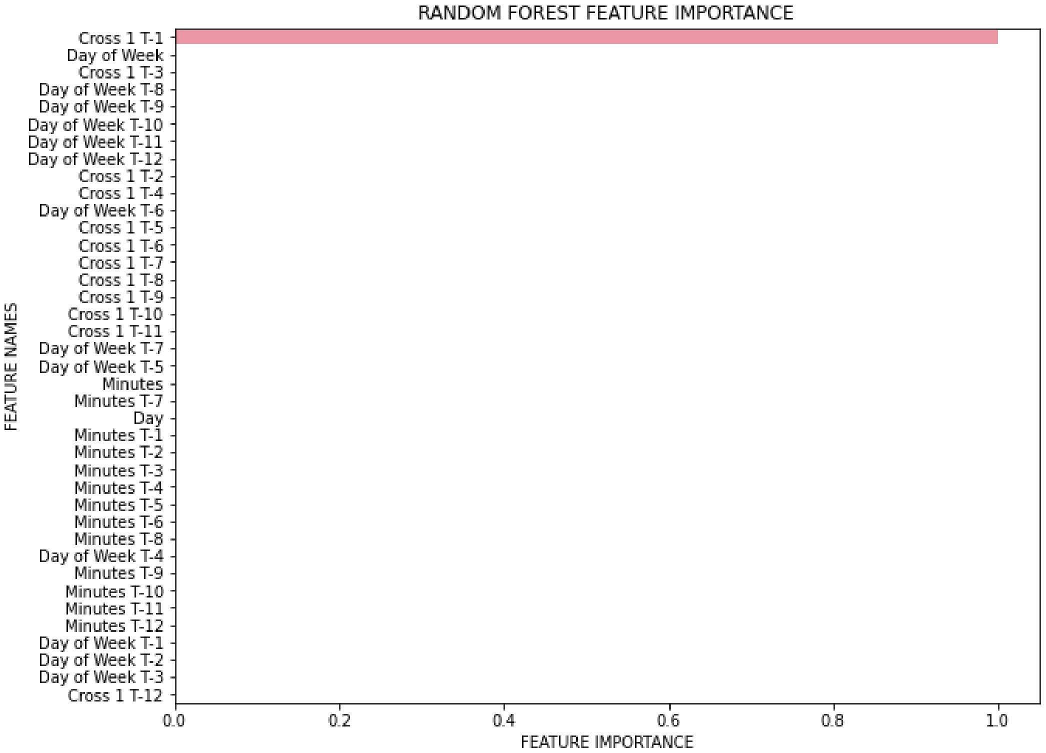


Figure 2: Random Forest feature importance for the dataset with "time" features forCross 1

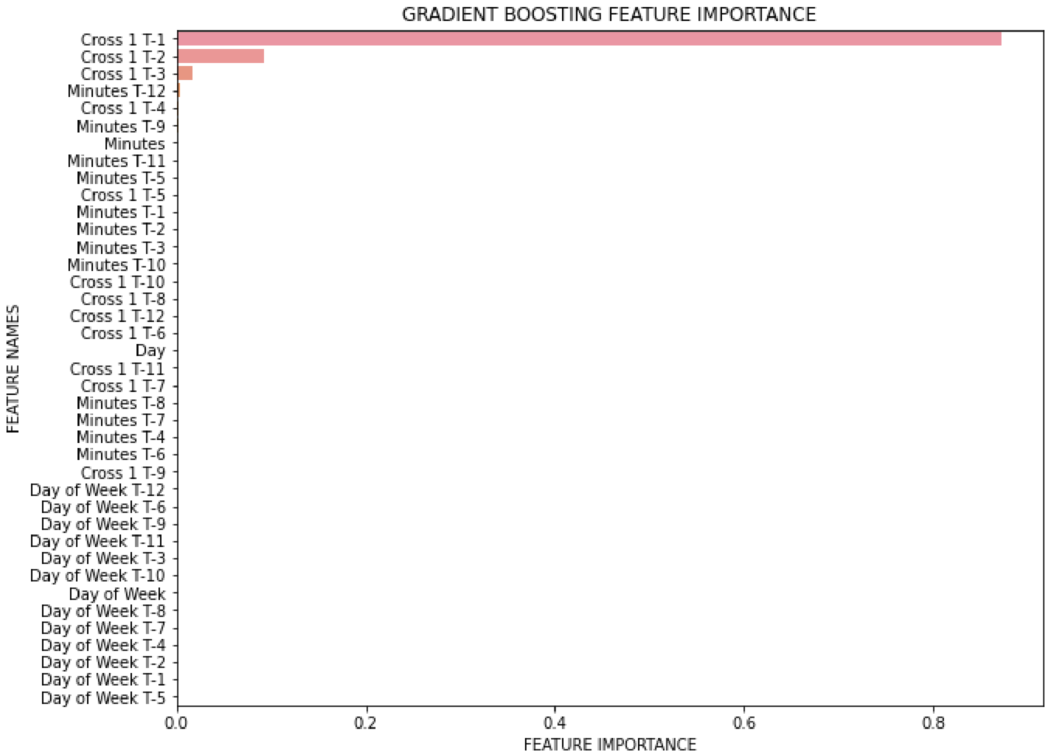


Figure 3:Gradient Boosting feature importance for the dataset with “time” featuresfor Cross 1

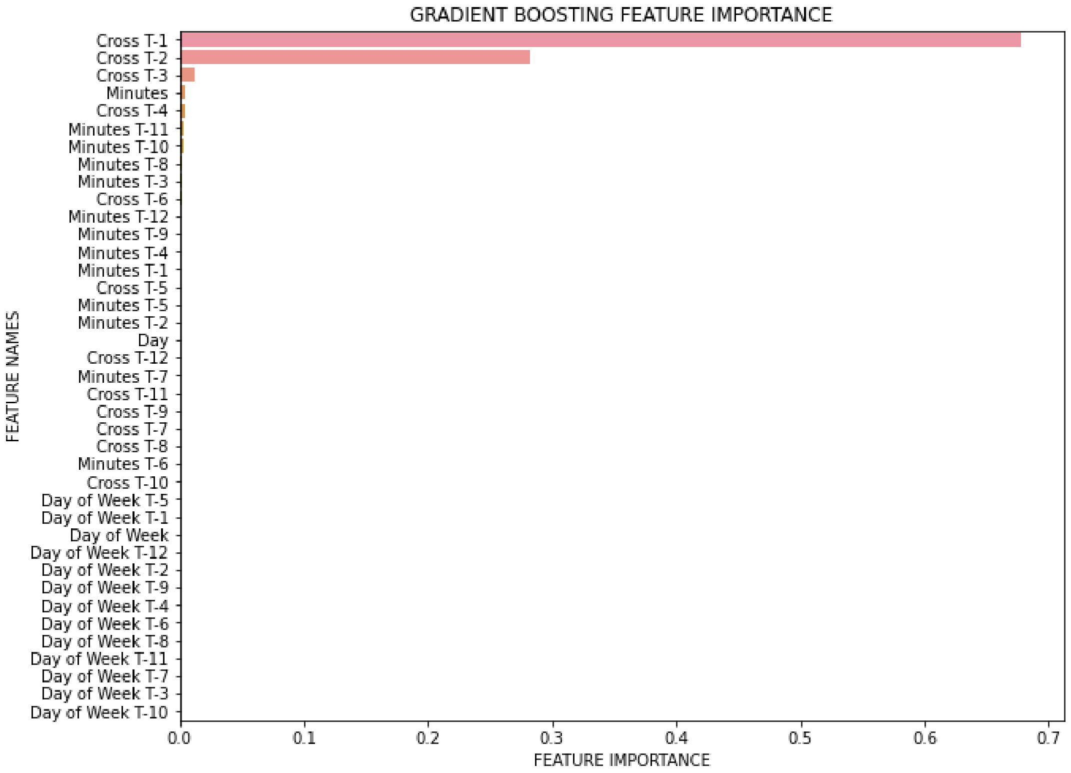


Figure 4: Gradient Boosting feature importance for the dataset with “time” features for Cross 3

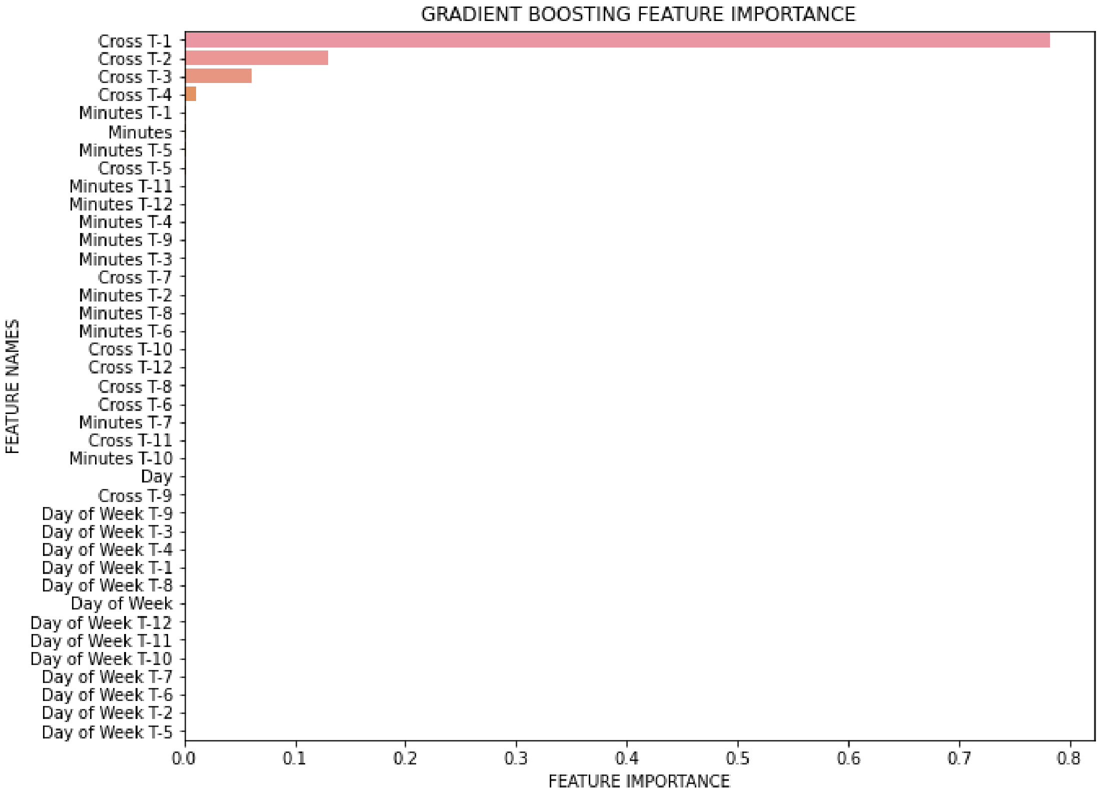


Figure 5: Gradient Boosting feature importance for the dataset with “time” features for Cross 4

From the figures, we could understand that for the models only the T-1 variable was important at the Random Forest model for Cross 1 but that was for the other Crosses too. The 3 last 5-minute intervals were important for the Gradient Boosting model for Cross 1,2,3,5,6. For Cross 3, we could observe that the percent of importance of T-2 is bigger than for the other Crosses. For Cross 4, the 4 previous observations are important. As it is logical since the data is only in a very small interval of 5 minutes. These new variables do not add something to the models and Time Series analysis was the next step. From the models’ results, we conclude that the results are better than the ones of simple regression. However, from the graphs it is understood, that the “time” features do not help enough to improve the models. Thus, time series analysis is the next step of our research.

4.3 Time Series Analysis

The Cross 1 time series is:

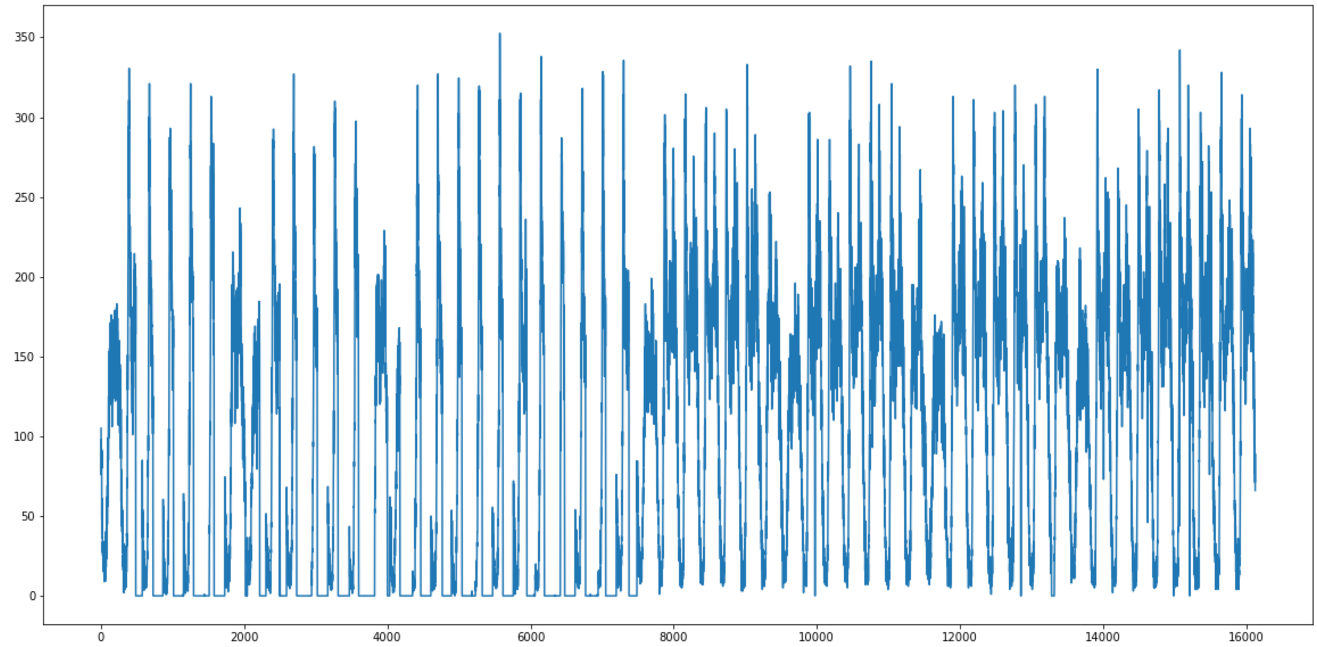
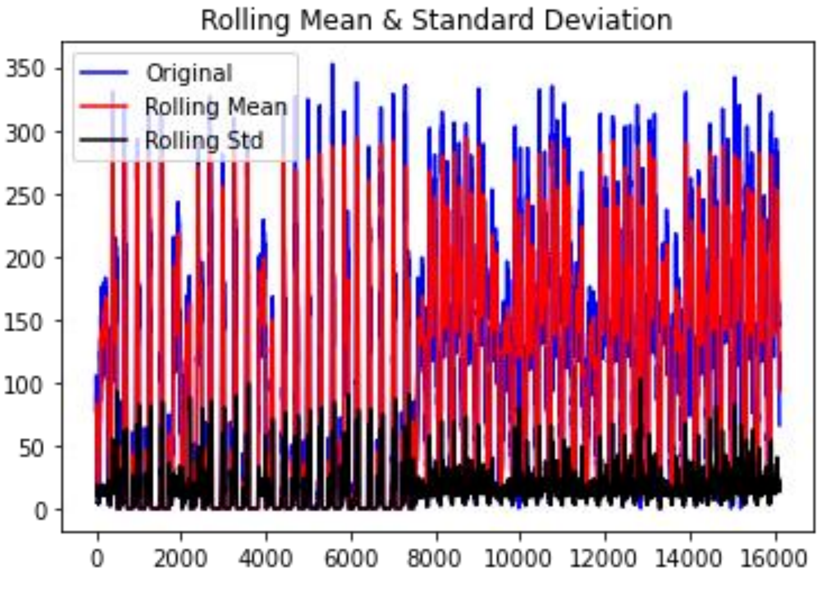
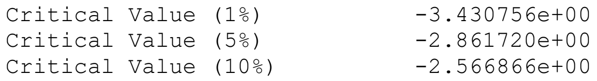


Figure 6: Time series of Cross 1

First (not averaged) data time series analysis:



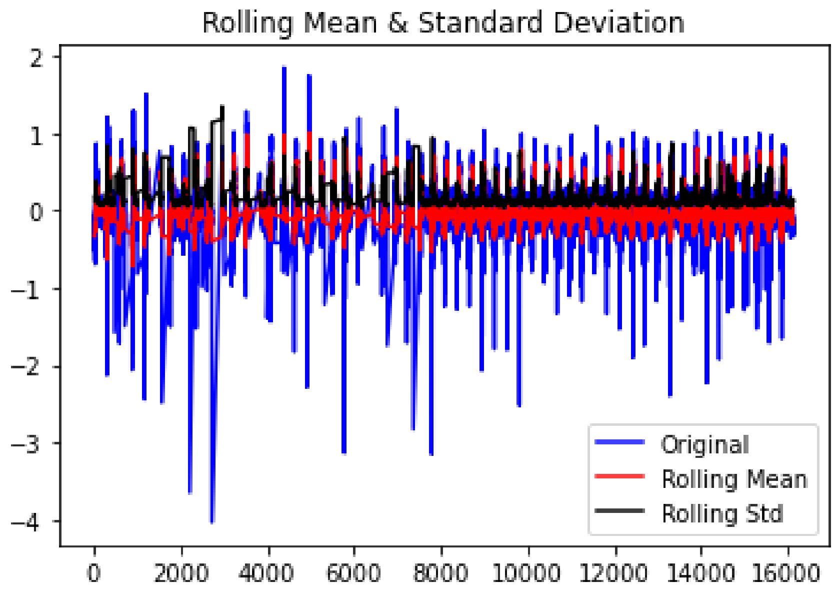
```
Results of Dickey-Fuller Test:
Test Statistic                -1.356987e+01
p-value                        2.225655e-25
#Lags Used                     3.200000e+01
Number of Observations Used    1.609500e+04
```

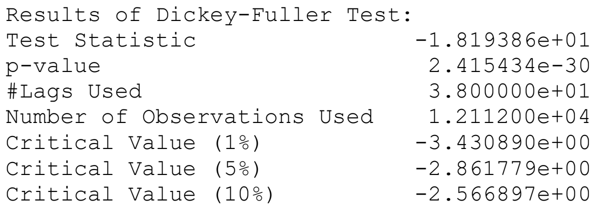


```
Critical Value (1%)           -3.430756e+00
Critical Value (5%)           -2.861720e+00
Critical Value (10%)          -2.566866e+00
```

Figure 7: Rolling mean and Standard Deviation graph with results of Dickey-Fuller Test

And these are the results for the time series after the observations have been into the logarithmic function.





```
Results of Dickey-Fuller Test:
Test Statistic                -1.819386e+01
p-value                        2.415434e-30
#Lags Used                     3.800000e+01
Number of Observations Used    1.211200e+04
Critical Value (1%)           -3.430890e+00
Critical Value (5%)           -2.861779e+00
Critical Value (10%)          -2.566897e+00
```

Figure 8: Rolling mean and Standard Deviation graph with results of Dickey-Fuller Test after using logarithmic function

The pvalue is less than 0.05, so the null hypothesis is rejected. Therefore, the Dickey-Fuller test concludes that there is no unit root, and the time series is stationary. Thus, the ARIMA or AR or MA models can be used.

And after difference for one time:

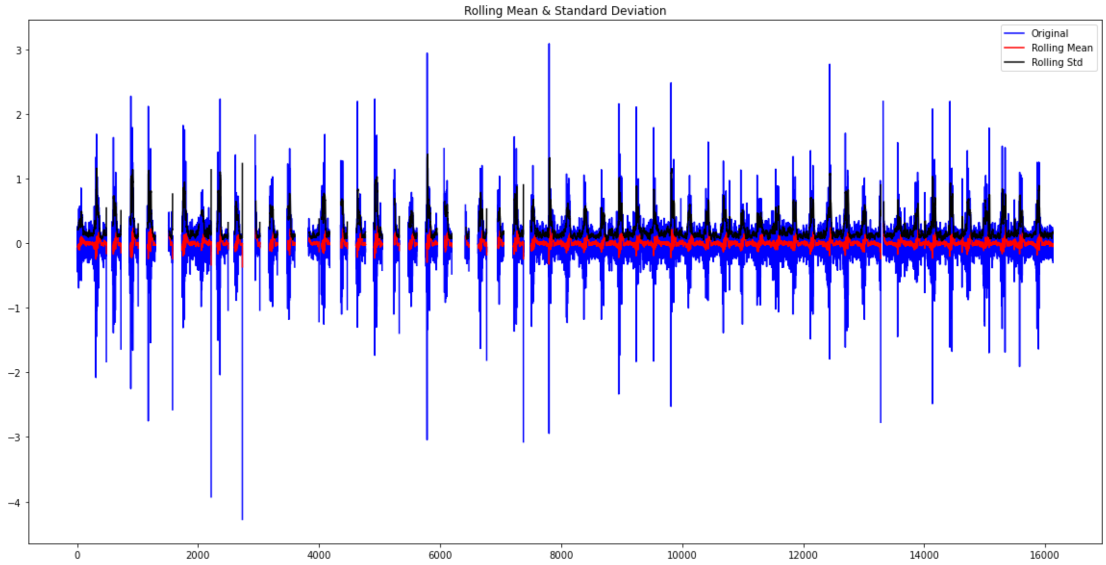


Figure 9: Rolling mean and Standard Deviation graph afterdifferencing

The plots of ACF and PACF are:

The plot of Autocorrelation function (ACF) for all the observations:

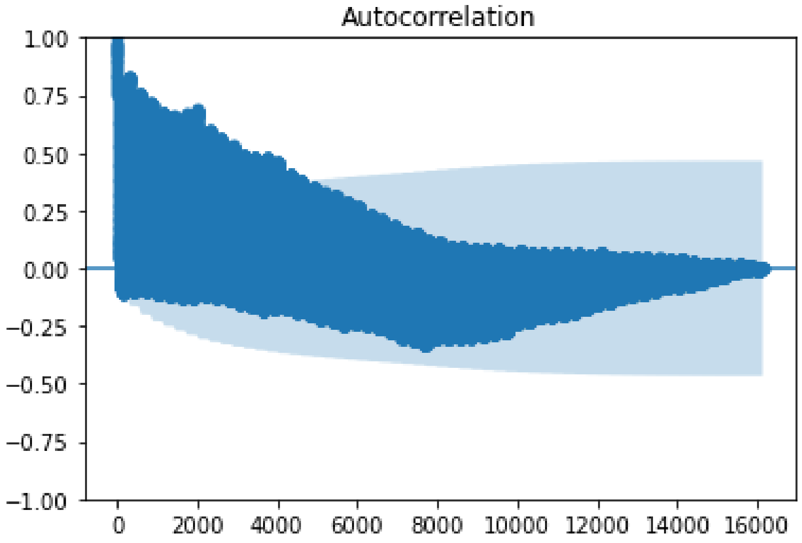


Figure 10: ACF plot for all the observations

The ACF plot for the 5000 first observations:

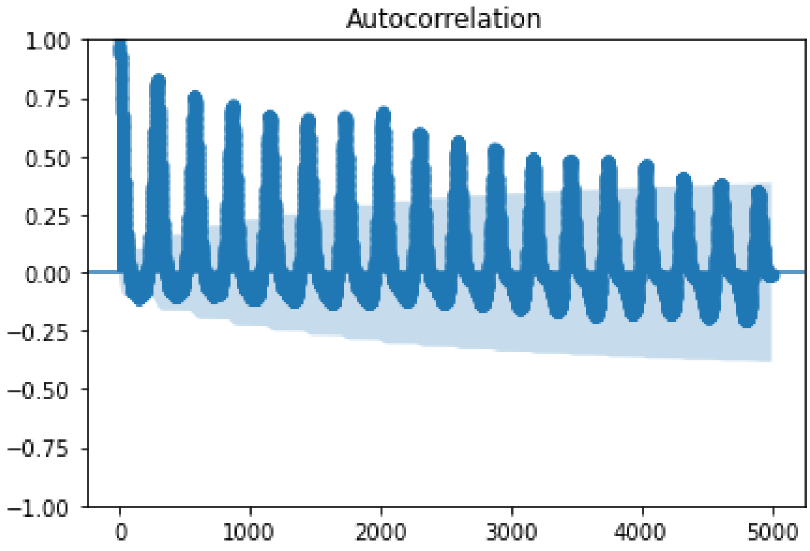


Figure 11: ACF plot for the 5000 first observations

Here we observe some periodicity.

The ACF plot for the 90 first observations:

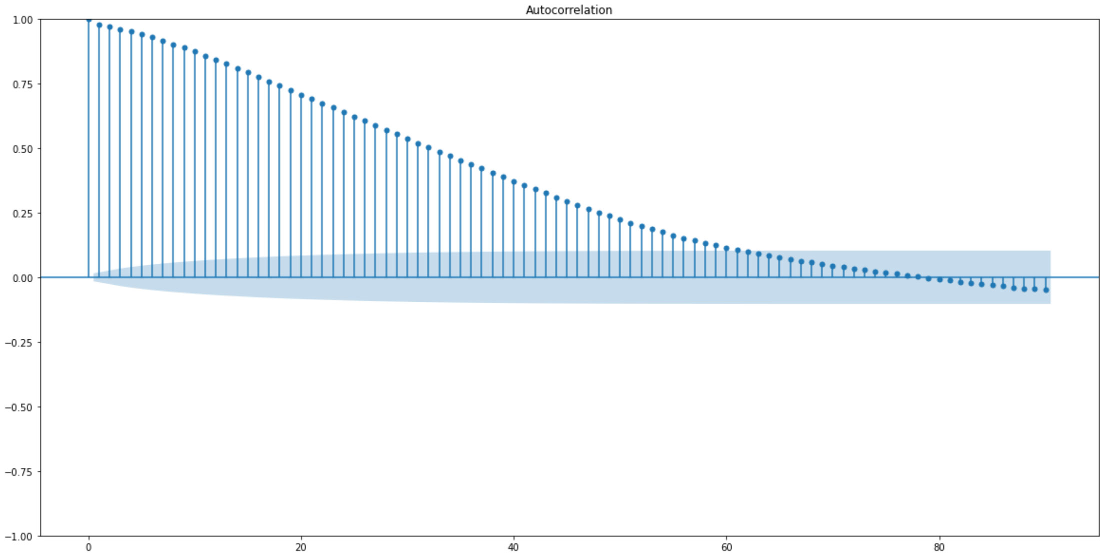


Figure 12: The ACF plot of the 90 first observations

The plot of the Partial Autocorrelation Function (PACF) for the 100 first observations:

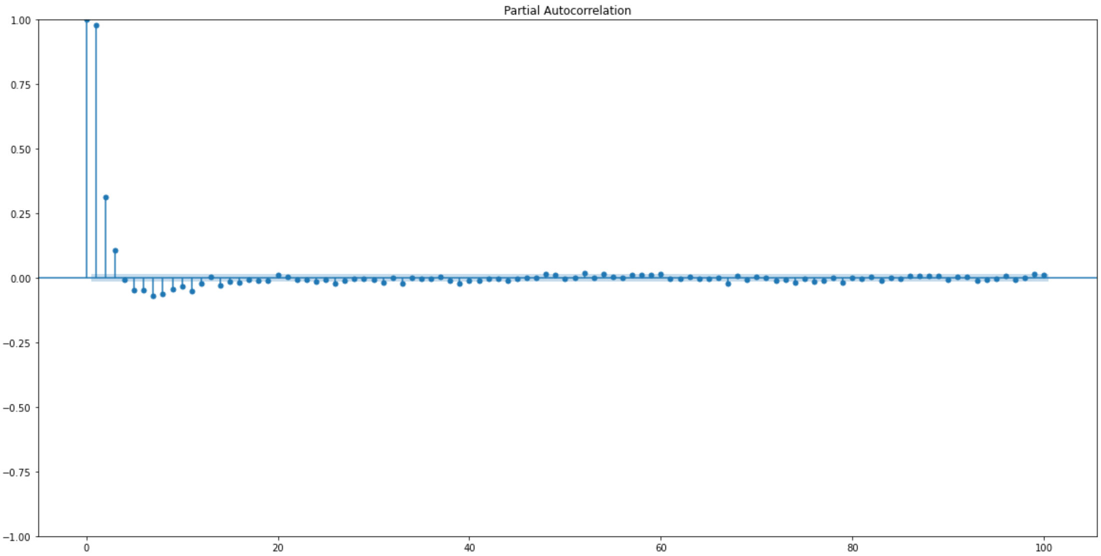


Figure 13: PACF plot of 100 first observations

We observe that only the 3 first lags are important (outside the blue area) with regards their values since they are much bigger than the others. That is something we expected as at the Gradient Boosting model (which gave us the best results) only the 3 of the 12 lags were important features. From this plot, we can say that a AR(3) model would be appropriate. And these are the results when we performed it:

Dep. Variable:	Cross 1	**No. Observations:**	16128
Model:	AutoReg(3)	**LogLikelihood**	-68937.959
Method:	Conditional MLE	**S.D. of innovations**	17.397
Date:	Tue, 27 Sep 2022	**AIC**	137885.918
Time:	14:34:53	**BIC**	137924.359
Sample:	3	**HQIC**	137898.627
	16128		

	coef	**stderr**	**z**	**P>|z|**	**[0.025**	**0.975]**
const	1.3130	0.204	6.437	0.000	0.913	1.713
Cross 1.L1	0.6371	0.008	81.373	0.000	0.622	0.652
Cross 1.L2	0.2424	0.009	26.633	0.000	0.225	0.260
Cross 1.L3	0.1067	0.008	13.631	0.000	0.091	0.122

	Real	**Imaginary**	**Modulus**	**Frequency**
AR.1	1.0095	-0.0000j	1.0095	-0.0000
AR.2	-1.6404	-2.5673j	3.0466	-0.3405

AR.3	-1.6404	+2.5673j	3.0466	0.3405

Table24: AR(3) model’s results

The results of the first machine learning modelsbut only for 3 time lags are:

	Cross 1	**Cross 1 T-1**	**Cross 1 T-2**	**Cross 1 T-3**
3	98.0	76.0	97.0	105.0
4	87.0	98.0	76.0	97.0
5	80.0	87.0	98.0	76.0
6	92.0	80.0	87.0	98.0
7	80.0	92.0	80.0	87.0
...	...	...	...	...
16123	85.0	94.0	77.0	101.0
16124	71.0	85.0	94.0	77.0
16125	83.0	71.0	85.0	94.0
16126	89.0	83.0	71.0	85.0
16127	66.0	89.0	83.0	71.0

16125 rows × 4 columns

Table25: Dataset for only Cross 1 and time lag 3

Cross 1	RF	LR	MLP	GB
R^2	0.915	0.958	0.957	**0.958**
MAE	18.070	11.243	11.068	**10.977**
RMSE	643.445	322.440	327.144	**318.785**
EVS	0.915	0.958	0.957	**0.958**

Table26: Regression Models’ results for 3 time lags

And the feature importance is:

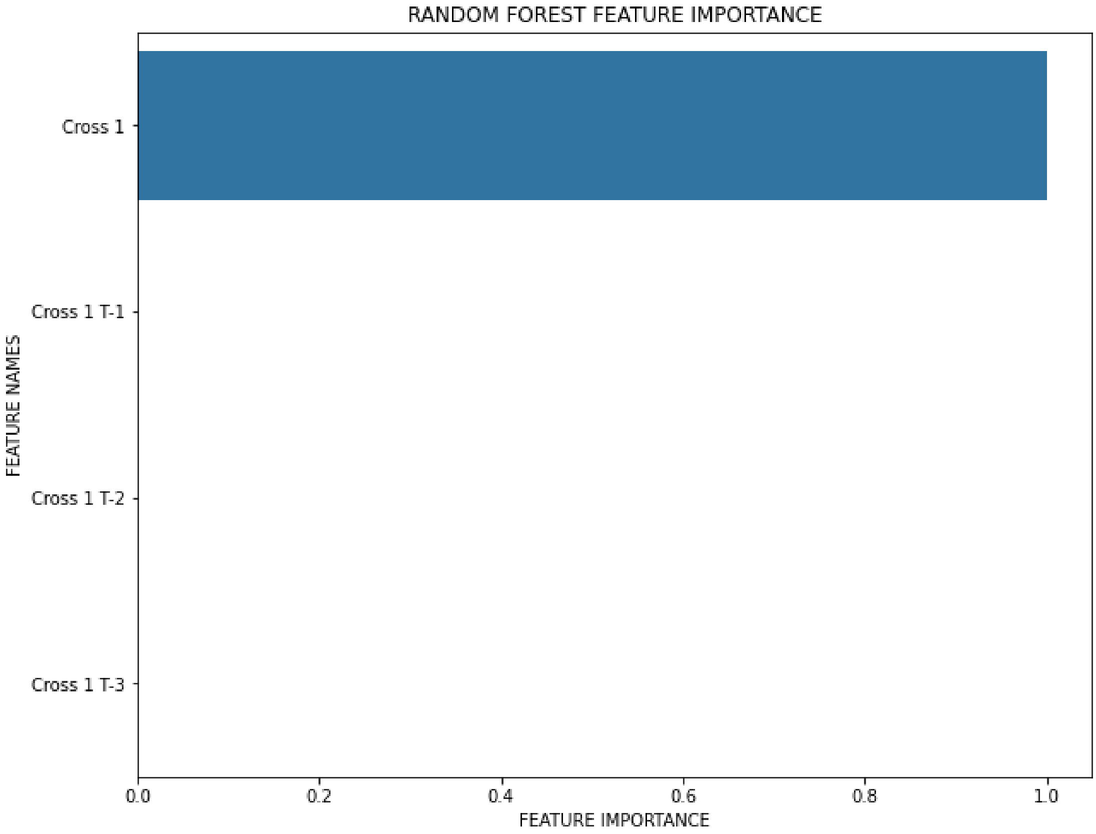


Figure 14: Feature importance of Random Forest for the 3-lag dataset of Cross 1

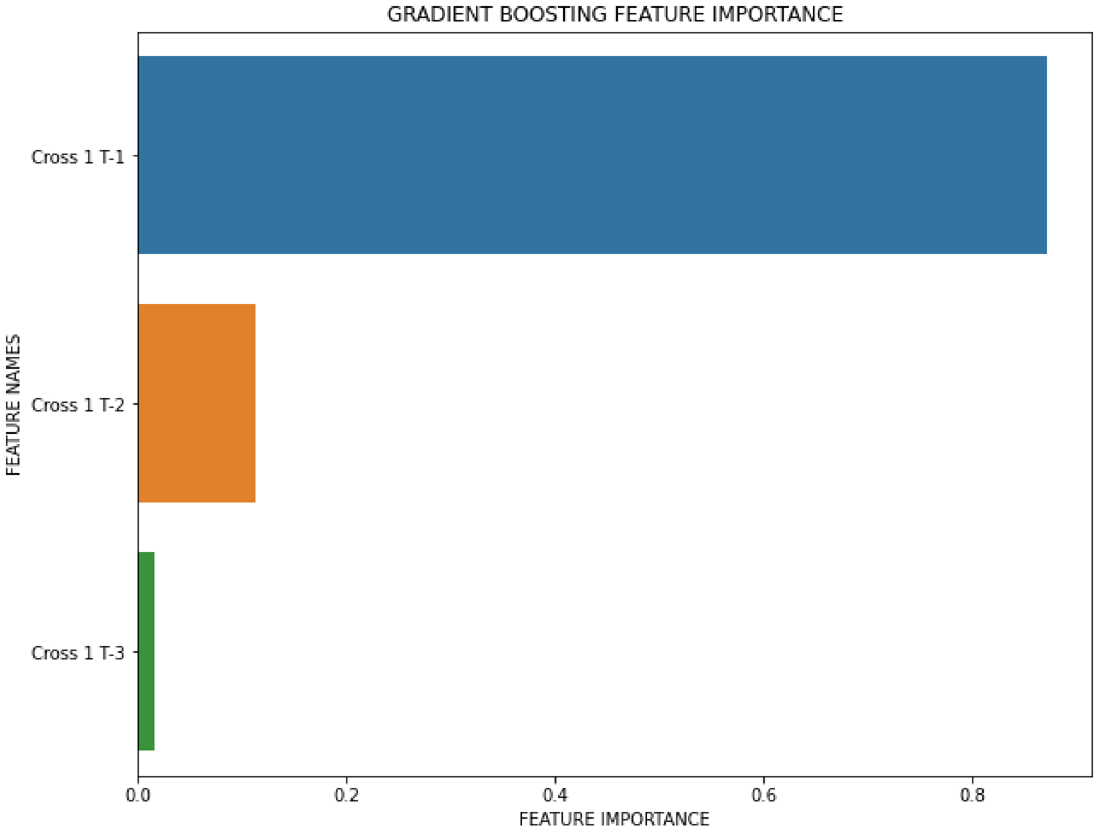


Figure 15: Feature importance of Gradient Boosting for the 3-lag dataset of Cross 1

As it can be observed only the T-1 observation is most important. That is something expected as the time interval is small. Thus, we explore other time intervals to examine if the models can also perform well enough.

4.4 Regression at different time interval

Another technique is to average some observations and make a new dataset of 15 minutes intervals and see the performance of the models. Thus, the new data was at 15 minutes intervals after getting the average of the 3 observations. We take the 12 previous observations as features for our models. These are the results:

Cross 1	RF	LR	MLP	GB
R^2	0.910	0.955	0.959	**0.967**
MAE	18.530	11.531	10.395	**9.140**
RMSE	709.675	351.501	319.353	**263.151**
EVS	0.910	0.955	0.960	**0.967**

Table27: Regression models' results for 15 minutes intervals for Cross 1

These are the graphs of Autocorrelation and Partial Autocorrelation:

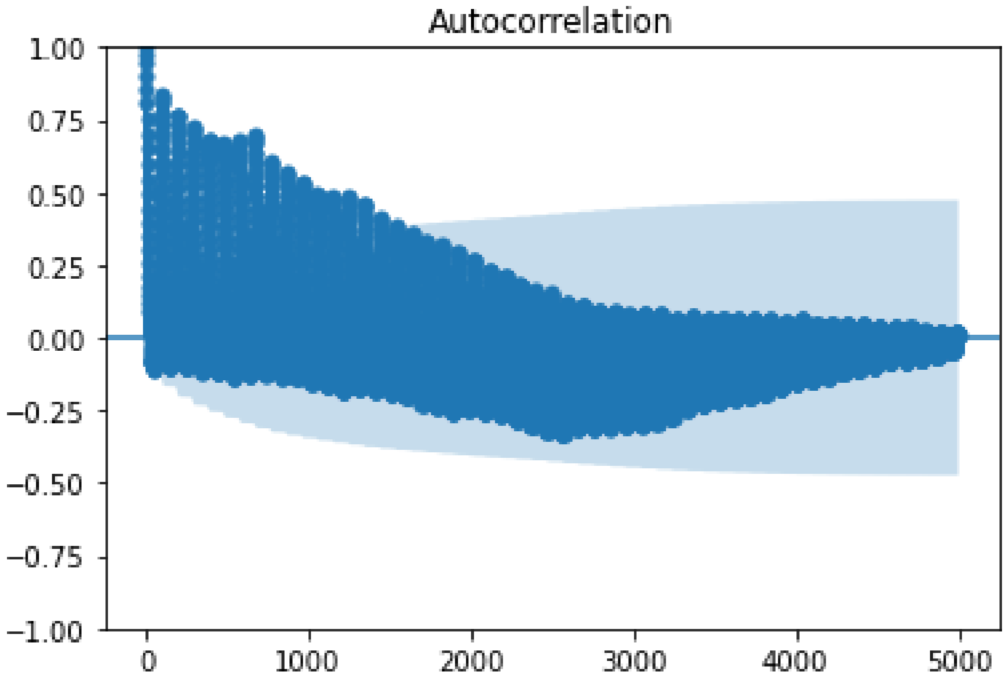


Figure 16: ACF plot for 15 minutes interval dataset of Cross 1

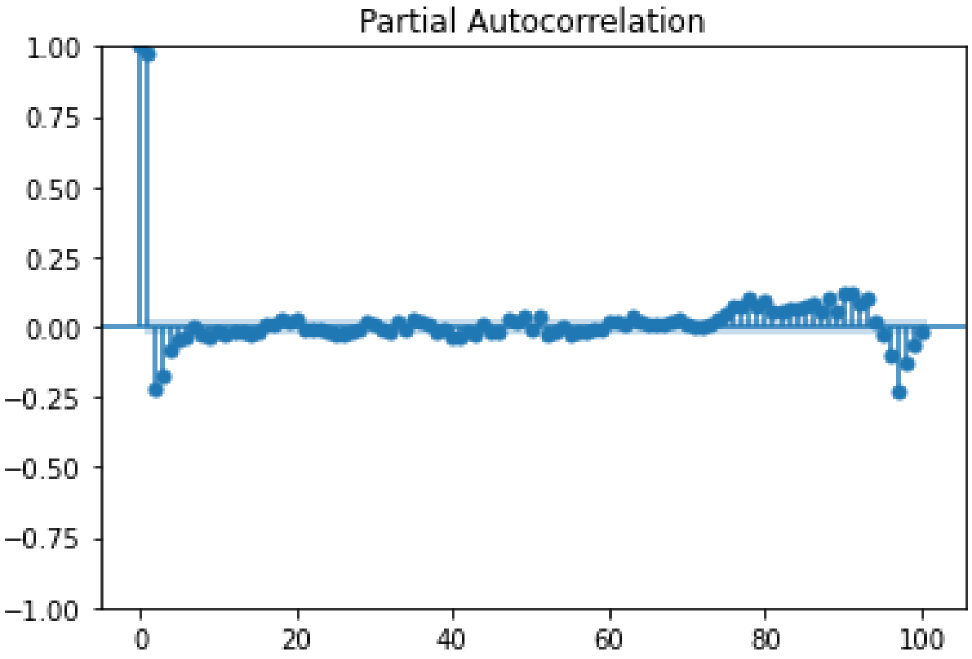


Figure 17: PACF plot of for 15 minutes interval dataset of Cross 1

More than 3 points are outside the blue area, so we can observe that this type of data needs another time series model for prediction.

Cross 2	RF	LR	MLP	GB
R^2	0.893	0.942	0.953	**0.954**
MAE	10.329	7.305	6.498	**6.369**
RMSE	290.326	158.424	126.558	**126.137**
EVS	0.893	0.942	0.954	**0.954**

Table28: Regression models' results for 15 minutes intervals for Cross 2

Cross 3	RF	LR	MLP	GB
R^2	0.876	0.937	0.946	**0.959**
MAE	10.378	6.621	5.869	**5.071**
RMSE	225.462	113.917	97.387	**74.484**
EVS	0.876	0.937	0.948	**0.959**

Table29: Regression models' results for 15 minutes intervals for Cross 3

Cross 4	RF	LR	MLP	GB
R^2	0.893	0.944	0.951	**0.958**
MAE	15.993	10.079	8.709	**8.123**
RMSE	511.236	267.906	232.266	**198.891**
EVS	0.893	0.944	0.951	**0.958**

Table30: Regression models' results for 15 minutes intervals for Cross 4

Cross 5	RF	LR	MLP	GB
R^2	0.864	0.922	0.937	**0.951**
MAE	9.083	6.692	5.815	**5.114**
RMSE	199.453	114.620	71.653	**71.653**
EVS	0.864	0.922	0.938	**0.951**

Table31: Regression models' results for 15 minutes intervals for Cross 5

Cross 6	RF	LR	MLP	GB
R^2	0.916	0.940	0.945	**0.948**
MAE	13.472	9.787	9.045	**8.448**
RMSE	390.011	277.185	252.643	**238.720**
EVS	0.916	0.940	0.947	**0.948**

Table32: Regression models' results for 15 minutes intervals for Cross 6

As it is observed, for each Cross we have almost the same results. If we compare them with the according ones of 5-minutes intervals, we could say that the statistical metrics here are better. Thus, the fact that we average the observations gives better results.

4.5 Classification models of the 6 Crosses

Another way of analysis of our data that have been used is Classification. After separation of our data in 3 same-length buckets with the use of the according percentiles for every cross and using the rolling window technique for 12 places, our data had 18 features. One column was the class of the according observation. The 3 classes were "low", "medium" and "high" as a characterization of the traffic flow at that time. The other 5 was the according observation at the other 5 crosses and the other 12 columns were produced from the from the rolling window technique. Next are presented the results rounded at the 3rd decimal point. Precision, Recall and F1 Score are provided per class, “high”, “low” and “medium” accordingly. The Accuracy is the subset accuracy, which specifies the percentage of samples that have all their labels classified correctly.

Cross 1	DT			MLP			KNN			SVC		
Accuracy	0.849			0.817			0.878			0.888		
Precision	0.87	0.91	0.77	**0.97**	0.91	0.67	0.88	0.91	0.84	0.89	0.90	**0.86**
Recall	0.84	0.92	0.78	0.62	0.96	0.88	0.90	0.95	0.78	0.90	**0.97**	0.79
F1 Score	0.85	0.92	0.77	0.75	0.93	0.76	0.89	0.93	0.81	**0.90**	**0.94**	0.82
Support	1826	1744	1749	1826	1744	1749	1826	1744	1749	1826	1744	1749
	RF			AdaBoost			GaussianNB			ExtraTrees		
Accuracy	**0.892**			0.588			0.788			0.892		
Precision	0.90	0.92	0.85	0.90	**0.94**	0.44	0.80	0.77	0.81	0.90	0.92	**0.86**
Recall	0.90	0.96	0.82	0.84	0.01	**0.90**	**0.93**	0.95	0.48	0.90	0.96	0.81
F1 Score	**0.90**	**0.94**	**0.83**	0.87	0.02	0.59	0.86	0.85	0.60	**0.90**	**0.94**	**0.83**
Support	1826	1744	1749	1826	1744	1749	1826	1744	1749	1826	1744	1749

Table33: Classification models’ results for Cross 1

For Cross 1:

For ExtraTree Classifier Cross k-fold Validation for k = 5 and for the Random Forest Cross k-fold Validation for k = 10 and grid search have been utilized.

For ExtraTree Classifier Cross Validation these are the results for accuracy: 0.90043424, 0.90350605, 0.88830282, 0.89854173, and 0.80359913. So, the 0.904 is the best Accuracy.

For Random Forest, Cross Validation results are: 0.88709677, 0.89854173, 0.89016444, 0.89388768, 0.79584238, so the best is 0.899. For grid search the conditions that have been examined were: 'n_estimators': [100, 1000], 'min_samples_split': [20, 25, 30], 'min_samples_leaf': [5, 8, 10], 'max_leaf_nodes': [18, 19] and the best model has 0.890which is not the best result.

Cross 2	DT			MLP			KNN			SVC		
Accuracy	0.861			0.910			0.895			0.906		
Precision	0.91	0.89	0.78	**0.93**	**0.92**	0.87	0.92	0.91	0.85	**0.93**	0.90	**0.88**
Recall	0.89	0.89	0.79	**0.94**	0.94	0.84	0.92	0.94	0.82	0.93	**0.96**	0.82
F1 Score	0.90	0.89	0.79	**0.94**	0.93	**0.86**	0.92	0.92	0.83	0.93	0.93	0.85
Support	1803	1838	1678	1803	1838	1678	1803	1838	1678	1803	1838	1678
	RF			AdaBoost			GaussianNB			ExtraTrees		
Accuracy	**0.912**			0.891			0.841			0.911		
Precision	**0.93**	**0.92**	**0.88**	0.92	0.89	0.86	0.92	0.80	0.81	**0.93**	**0.92**	**0.88**
Recall	0.93	0.95	**0.85**	**0.92**	0.96	0.79	0.89	0.95	0.67	0.93	0.95	0.84
F1 Score	0.93	**0.94**	**0.86**	0.92	0.92	0.82	0.90	0.87	0.73	0.93	**0.94**	**0.86**
Support	1803	1838	1678	1803	1838	1678	1803	1838	1678	1803	1838	1678

Table34: Classification models' results for Cross 2

For Cross 2:

For ExtraTree Classifier Cross k-fold Validation for k = 5 and for the Random Forest Cross k-fold Validation for k = 5 and grid search have been utilized.

For ExtraTree Classifier Cross Validation these are the results for accuracy: 0.91997519, 0.83276451, 0.94911573, 0.92305306, and 0.86565312. So, the 0.949 is the best Accuracy.

For Random Forest, Cross Validation results are: 0.91811414, 0.84269314, 0.94787465, 0.91405523, 0.86286069, so the best is 0.848. For grid search the conditions that have been examined were: 'n_estimators': [100, 1000], 'min_samples_split': [20, 25, 30], 'min_samples_leaf': [5, 8, 10], 'max_leaf_nodes': [18, 19] and the best model has 0.902 which is not the best result.

Cross 3	DT			MLP			KNN			SVC		
Accuracy	0.836			0.873			0.875			0.880		
Precision	0.87	0.88	0.76	0.84	**0.93**	**0.86**	0.90	0.91	0.82	0.90	0.90	0.84
Recall	0.85	0.90	0.76	**0.95**	0.92	0.75	0.91	0.92	0.80	0.91	**0.94**	0.80
F1 Score	0.86	0.89	0.76	0.89	**0.92**	0.80	0.90	0.91	0.81	**0.91**	**0.92**	0.82

Support	1739	1750	1776	1739	1750	1776	1793	1750	1776	1793	1750	1776
	RF			AdaBoost			GaussianNB			ExtraTrees		
Accuracy	**0.889**			0.855			0.805			0.885		
Precision	**0.91**	0.91	0.85	0.83	0.88	0.85	0.86	0.75	0.82	0.90	0.91	0.84
Recall	0.92	0.93	**0.82**	0.94	**0.94**	0.69	0.91	**0.94**	0.56	0.92	0.93	0.81
F1 Score	**0.91**	**0.92**	**0.83**	0.88	0.91	0.76	0.89	0.84	0.66	**0.91**	**0.92**	**0.83**
Support	1793	1750	1776	1793	1750	1776	1793	1750	1776	1793	1750	1776

Table35: Classification models' results for Cross 3

For Cross 3:

For ExtraTree Classifier Cross k-fold Validation for k = 5 and for the Random Forest Cross k-fold Validation for k = 5 and grid search have been utilized.

For ExtraTree Classifier Cross Validation these are the results for accuracy: 0.8926799, 0.90940118, 0.89078498, 0.8864412, and 0.78312132. So, the 0.909 is the best Accuracy.

For Random Forest, Cross Validation results are: 0.89205955, 0.89947254, 0.89078498, 0.88675147, 0.77970835, so the best is 0.900. For grid search the conditions that have been examined were: 'n_estimators': [100, 1000], 'min_samples_split': [20, 25, 30], 'min_samples_leaf': [5, 8, 10], 'max_leaf_nodes': [18, 19] and the best model has 0.883 which is not the best result.

Cross 4	DT			MLP			KNN			SVC		
Accuracy	0.846			0.860			0.875			0.882		
Precision	0.86	0.90	0.77	0.82	0.89	**0.88**	0.88	0.91	0.82	0.90	0.91	0.83
Recall	0.89	0.89	0.76	**0.97**	0.93	0.67	0.92	0.91	0.80	0.92	0.92	0.80
F1 Score	0.87	0.89	0.77	0.89	**0.91**	0.76	0.90	0.91	0.81	0.91	**0.91**	0.82
Support	1771	1809	1739	1771	1809	1739	1771	1809	1739	1771	1809	1739
	RF			AdaBoost			GaussianNB			ExtraTrees		
Accuracy	0.883			0.862			0.804			**0.886**		

Precision	**0.90**	**0.92**	0.83	0.85	0.88	0.87	0.85	0.76	0.82	**0.90**	**0.92**	0.83
Recall	0.93	0.90	0.82	0.94	**0.95**	0.69	0.93	**0.95**	0.53	0.93	0.90	**0.83**
F1 Score	0.91	**0.91**	0.82	0.89	0.91	0.77	0.88	0.84	0.64	**0.92**	**0.91**	**0.83**
Support	1771	1809	1739	1771	1809	1739	1771	1809	1739	1771	1809	1739

Table36: Classification models' results for Cross 4

For Cross 4:

For ExtraTree Classifier Cross k-fold Validation for k = 5 and for the Random Forest Cross k-fold Validation for k = 5 and grid search have been utilized.

For ExtraTree Classifier Cross Validation these are the results for accuracy: 0.89764268, 0.9050574, 0.88457958, 0.90102389, and 0.76388458. So, the 0.905 is the best Accuracy.

For Random Forest, Cross Validation results are: 0.89578164, 0.89761092, 0.88892336, 0.90071362, 0.76140242, so the best is 0.901. For grid search the conditions that have been examined were: 'n_estimators': [100, 1000], 'min_samples_split': [20, 25, 30], 'min_samples_leaf': [5, 8, 10], 'max_leaf_nodes': [18, 19] and the best model has 0.886 which is not the best result.

Cross 5	DT			MLP			KNN			SVC		
Accuracy	0.848			0.882			0.872			0.888		
Precision	0.83	0.93	0.78	**0.86**	0.95	0.83	0.85	0.94	0.82	**0.86**	0.96	0.85
Recall	0.84	0.93	0.77	0.90	**0.94**	0.81	0.89	0.93	0.79	**0.93**	0.93	0.81
F1 Score	0.84	0.93	0.77	0.88	0.94	0.82	0.87	0.94	0.81	**0.89**	0.94	0.83
Support	1349	1326	1354	1349	1326	1354	1349	1326	1354	1349	1326	1354
	RF			AdaBoost			GaussianNB			ExtraTrees		
Accuracy	**0.893**			0.820			0.80			0.891		
Precision	**0.86**	**0.97**	**0.86**	0.82	0.83	0.81	0.84	0.78	0.78	**0.86**	0.96	**0.86**

Recall	**0.93**	0.93	**0.82**	**0.93**	0.93	0.60	0.91	0.92	0.58	**0.93**	0.93	0.81
F1 Score	**0.89**	**0.95**	**0.84**	0.87	0.88	0.69	0.87	0.85	0.66	**0.89**	**0.95**	0.83
Support	1349	1326	1354	1349	1326	1354	1349	1326	1354	1349	1326	1354

Table37: Classification models' results for Cross 5

For Cross 5:

For ExtraTree Classifier Cross k-fold Validation for k = 5 and for the Random Forest Cross k-fold Validation for k = 5 and grid search have been utilized.

For ExtraTree Classifier Cross Validation these are the results for accuracy: 0.89764268, 0.9050574, 0.88457958, 0.90102389, and 0.76388458. So, the 0.905 is the best Accuracy.

For Random Forest, Cross Validation results are: 0.91811414, 0.84269314, 0.94787465, 0.91405523, 0.86286069, so the best is 0.848. For grid search the conditions that have been examined were: 'n_estimators': [100, 1000], 'min_samples_split': [20, 25, 30], 'min_samples_leaf': [5, 8, 10], 'max_leaf_nodes': [18, 19] and the best model has 0.902 which is not the best result.

Cross 6	DT			MLP			KNN			SVC		
Accuracy	0.823			0.855			0.860			0.874		
Precision	0.82	0.91	0.74	**0.90**	**0.96**	0.74	0.82	0.94	0.82	0.82	0.95	**0.85**
Recall	0.82	0.91	0.74	0.76	0.92	0.88	0.89	0.93	0.77	**0.93**	0.93	0.76
F1 Score	0.82	0.91	0.74	0.82	0.94	0.81	0.86	0.93	0.79	**0.87**	**0.94**	0.81
Support	1725	1787	1807	1725	1787	1807	1725	1787	1807	1725	1787	1807
	RF			AdaBoost			GaussianNB			ExtraTrees		
Accuracy	**0.879**			0.573			0.801			0.878		
Precision	0.84	0.95	**0.85**	0.83	0.67	0.43	0.82	0.80	0.77	0.84	**0.96**	**0.85**
Recall	0.92	0.93	**0.79**	0.90	0.00	0.83	0.87	**0.94**	0.59	**0.93**	0.92	**0.79**
F1 Score	0.88	**0.94**	**0.82**	0.86	0.01	0.57	0.85	0.86	0.67	**0.88**	0.94	**0.82**

Support	1725	1787	1807	1725	1787	1807	1725	1787	1807	1725	1787	1807

Table38: Classification models’ results for Cross 6

For Cross 6:

For ExtraTree Classifier Cross k-fold Validation for k = 5 and for the Random Forest Cross k-fold Validation for k = 5 and grid search have been utilized.

For ExtraTree Classifier Cross Validation these are the results for accuracy: 0.88182382, 0.87868446, 0.89047471, 0.90784983, and 0.82655911. So, the 0.908 is the best Accuracy.

For Random Forest, Cross Validation results are: 0.87655087, 0.87651257, 0.88488985, 0.90443686, 0.82314614, so the best is 0.904. For grid search the conditions that have been examined were: 'n_estimators': [100, 1000], 'min_samples_split': [20, 25, 30], 'min_samples_leaf': [5, 8, 10], 'max_leaf_nodes': [18, 19] and the best model has 0.878 which is not the best result.

4.6 LSTM

As can be concluded from the related work chapter, LSTM models have been used at traffic flow predictions and have outperform all the other models. Thus, it is utilized here with the same percent of train and test set and number of previous observations that have been used as features as the first regression problem that we solved with the help of machine learning models, in order to compare the results.

In order to fine tune the hyperparameters some tests have been made. After finding the optimal parameters, then the same model has performed for the other Crosses too.

Cross 1	4 units, batch size =1,epochs= 100	64 units, batch size =1,epochs= 100	64 units, batchsize =50,epochs= 100	100units, batchsize =100,epochs= 100	100units, batchsize =100,epochs=1000
RMSE (train)	34.82	33.48	17.05	15.83	15.24
RMSE (test)	41.03	38.06	20.89	19.65	20.22

R^2(train)	0.84	0.85	0.96	0.97	0.97
R^2(test)	0.72	0.76	0.93	0.94	0.93

Table39: LSTM’s results for Cross 1

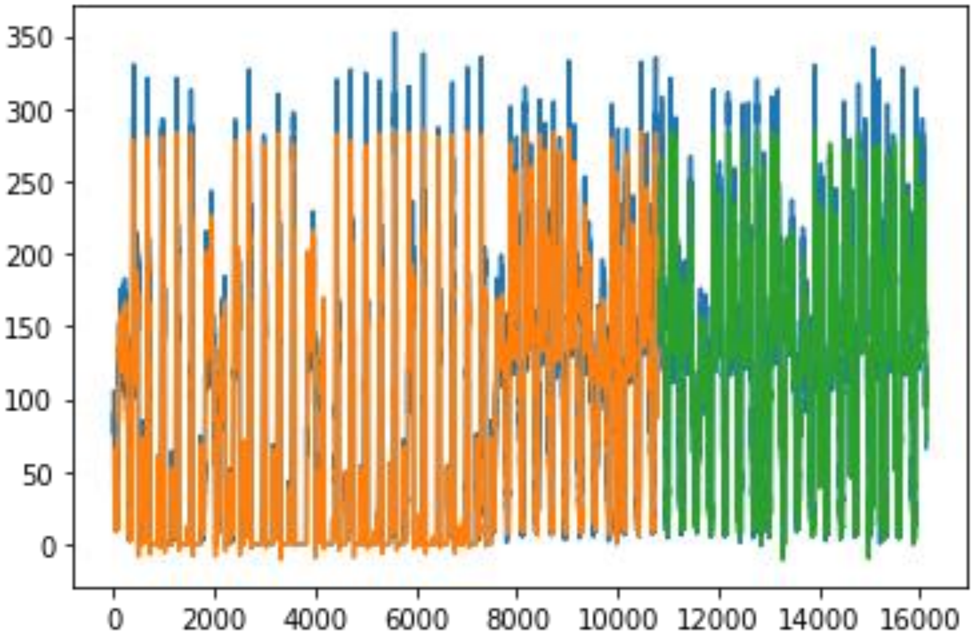


Figure18: Time Series of Cross 1(blue) and results of LSTM train (orange) and test (green) sets

For 100 units, batch size =100, epochs=1000 for all the other Crosses:

	Cross 2	Cross 3	Cross 4	Cross 5	Cross 6
RMSE (train)	8.59	7.86	15.24	6.90	15.29
RMSE (test)	17.22	11.12	20.22	11.97	20.82
R^2(train)	0.96	0.97	0.97	0.96	0.95
R^2(test)	0.90	0.92	0.93	0.88	0.88

Table40: LSTM’s results for all the other Crosses

For Cross 1, we have RMSE 20.22 which is much less than 289.780 from the MLP model.

For Cross 2, the RMSE 17.22 is much less than 145.976 from the MLP model.

For Cross 3, the RMSE 11.12 is much less than 75.729 from the Gradient Boosting model.

For Cross 4, the RMSE 20.22 is much less than 253.272 from the Gradient Boosting model.

For Cross 5, the RMSE 11.97 is much less than 82.700 from the Gradient Boosting model.

For Cross 6, the RMSE 20.82 is much less than 249.299 from the Gradient Boosting model.

For all the crosses, the results are almost same. But generally, we could say that LSTM performed better than the other models and has highlighted the power and the good performance that we expected from the literature review.

5 Conclusions and future work

5.1 Conclusions

The research that had been contacted here is an approach of dealing with a univariable traffic prediction problem without timestamp. The classical time series analysis and prediction had been converted into a machine and deep learning problem. Several models for regression and classification have been utilized compared. We conclude that the LSTM and Gradient Boosting are the best ones for regression. As it is expected the LSTM model, after hyperparameter tuning, outperforms the others with respect to RMSE. With the classification problem, Extra Trees and Random Forest classifier are the best ones as in many traffic prediction problems. Cross Validation and Grid Search are also utilized. Techniques like trying different time intervals or getting new “time” features have been examined and had better results than the simplest methods. Furthermore, when the data from 2 Crosses that have been highly correlated are used the models have better results than the ones with only the data from one Cross. It could be highlighted that the 6 Crosses have similar results and that indicates that they are near.

5.2 Threats to validity

Since all the methods that have been used are correctly utilized without misconceptions, the threats that the research have are in the part of the data quality. The data are acquired from sensors that can faultily register some observation due to several reasons like bad weather conditions. For example, here our data have parts that all the Crosses have zero traffic flow for several observations. This might be correct for night timebut it is also noticed at other timestoo. For the particular dataset, another approach of handling the series of continuing zeros is to use the moving average instead of the zeros as suggested at the [64] which will help the regression models. However, then we replace some real zeros with the moving average.

5.3 Future work

The traffic prediction problem that is tried to be solved here can generate new work for research. One way of working is the use of this type of data, which is traffic flow data to manage a smart traffic light in order to reduce congestion and accomplish better traffic management. Some other idea for future work is to highlight the use of autonomous cars. As their number increase, more complicated models that take care of this type of transporting are of the essence.

References

[1] B. P. Ashwini, R. Sumathi, Data Sources for Urban Traffic Prediction: A Review on Classification, Comparison and Technologies (2020)

[2] F. Schimbinschi, X. V. Nguyen, J. Bailey, C. Leckie, H. Vu, and R. Kotagiri, Traffic forecasting in complex urban networks: Leveraging big data and machine learning (2015)

[3] G. Leduc, Road Traffic Data: Collection Methods and Applications (2008)

[4] A. Pande, B. Wolshon, D. Matherly, P. Murray-Tuite, and B. Wolshon, Traffic Management for Planned, Unplanned, and Emergency Events.

[6] H. Yuan, G. Li, A Survey of Traffic Prediction: from Spatio-Temporal Data to Intelligent Transportation (2021)

[7]A. Nuaimi Applications of big data to smart cities, Journal of Internet Services and Applications (2015)

[8] H. Ahvenniemi, What are the differences between sustainable and smart cities? (2017)

[9] H. Chourabi,Understanding Smart Cities: An Integrative Framework, 45th Hawaii International Conference on System Sciences (2012)

[10] E. Bolshinsky and R. Freidman, Traffic Flow Forecast Survey (2012)

[11] J. Barros, M. Araujo, R.J.F. Rossetti, Short-term real-time traffic prediction methods: a survey (2015)

[12] W. Min, L. Wynter, Real-time road traffic prediction with spatiotemporal correlations (2011)

[13] S. Blatnig, Microscopic Traffic Simulation with Intelligent Agents (2008)

[14] F. Maier, R. Braun, F. Busch, and P. Mathias, Pattern-based short-term prediction of urban congestion propagation and automatic response (2008)

[15] C. Harrison and I. A. Donnelly, A theory of Smart Cities (2011)

[16] Zhang K., Chuai G., Zhang J., Chen X., Si Z., Maimaiti S., DIC-ST: A Hybrid Prediction Framework Based on Causal Structure Learning for Cellular Traffic and Its Application in Urban Computing (2022)

[17] Ahmed M.S., Cook A.R., Analysis of freeway traffic time-series data by using Box-Jenkins techniques (1979)

[18] M. vanderVoort , M. Dougherty,S. Watson , Combining Kohonen maps ARIMA time series models to forecast traffic flow (1996)
[19] B.M. Williams, L.A. Hoel , Modeling and forecasting vehicular traffic flow as a seasonal ARIMA process: Theoretical basis and empirical results (2003)
[20]S. Lee, D. Fambro, Application of subset autoregressive integrated moving average model for short- term freeway traffic volume forecasting (1999)
[21] F. Rui,Z.M. Zuo, Using LSTM and GRU Neural Network methods for Traffic Flow Prediction (2016)
[22] Y.L.Y. Duan,W. Kang , Z. Li ,F. Wang, Traffic Flow Prediction with Big Data: A Deep Learning Approach (2015)

[23] J. Krumm,E. Horvitz, Locadio: inferring motion and location from wi-fi signal strengths (2004)

[24] Y. Tian, L. Pan, Predicting Short-Term Traffic Flow by Long Short-Term Memory Recurrent Neural Network (2015)

[25]S. Zhang, Z. Kang, Z. Zhang, C. Lin, C. Wang, J. Li, A Hybrid Model for Forecasting Traffic Flow: Using Layerwise Structure and Markov Transition Matrix (2019)
[26] A. Chen, J. Law,M. Aibin, A Survey on Traffic Prediction Techniques Using Artificial Intelligence for Communication Networks (2021)

[27] Z. Li, S. Jiang, L. Li, Y. Li, Building sparse models for traffic flow prediction: an empirical comparison between statistical heuristics and geometric heuristics for Bayesian network approaches (2019)
[28] T. Zhou, D. Jiang, Z. Lin, G. Han, X. Xu, J. Qin, Hybrid dual Kalman filtering model for short-term traffic flow forecasting (2019)
[29] J. Zheng, M. Huang, Traffic Flow Forecast through Time Series Analysis Based on Deep Learning (2020)
[30] M. X. Liu, L. Xia, J. Zhong, N. N. Dou, B. Li, Is it necessary to approach the compressed vertebra bilaterally during the process of PKP?, Journal of Spinal Cord Medicine (2020)
[31] Y. Lv, Y. Duan, W. Kang, Z. Li, F. Y. Wang, Traffic Flow Prediction with Big Data: A Deep Learning Approach (2015)
[32] L. Li, L. Qin, X. Qu, J. Zhang, Y. Wang, B. Ran, Day-ahead traffic flow forecasting based on a deep belief network optimized by the multi-objective particle swarm algorithm, Knowledge-Based Systems, vol. 172, pp. 1-14 (2019)
[33] X. Luo, D. Li, Y. Yang, S. Zhang, Spatiotemporal Traffic Flow Prediction with KNN and LSTM, Journal of Advanced Transportation (2019)
[34] J. Guo, Z. Xie, Y. Qin, L. Jia, Y. Wang, Short-Term Abnormal Passenger Flow Prediction Based on the Fusion of SVR and LSTM (2019)
[35] X. Wang, L. Xu, K. Chen, Data-Driven Short-Term Forecasting for Urban Road Network Traffic Based on Data Processing and LSTM-RNN, Arabian Journal for Science and Engineering (2019)
[36] H. Qi, X. Hu, Real-time headway state identification and saturation flow rate estimation: a hidden Markov Chain model (2020)
[37] K. Kumar,M. Parida,V. K. Katiyar,Short-term TFP for a non-urban highway usingartificial neural network (2013)

[38] Kim, Y. J., & Hong, J. S., Urban TFP system using a multifactor pattern recognition model, IEEE Transactions on Intelligent Transportation Systems (2015)
[39] Wu, Y., Tan, H., Qin, L., Ran, B., & Jiang, Z., A hybrid deep learning based TFP method and its understanding (2018)
[40] D. Ma, B. Sheng, S. Jin, X. Ma, P. Gao, Short-term traffic flow forecasting by selecting appropriate predictions based on pattern matching (2018)
[41]F. I. Rahman, Short-Term Tfp Using Machine Learning-Knn, Svm, And Ann With Weather Information (2020)
[42] Z. Cui, K. Henrickson, R. Ke, Y. Wang, Traffic graph convolutional recurrent neural network: A deep learning framework for network-scale traffic learning and forecasting (2019)
[43] A. Mystakidis, C. Tjortjis, Big Data Mining for Smart Cities: Predicting Traffic Congestion using Classification Proc.11th IEEE Int'lConf. on Information, Intelligence, Systems and Applications (IISA 20) (2020)

[44] A. Abadi, T.Rajabioun, P. A. Ioannou , Traffic Flow Prediction for Road Transportation Networks With Limited Traffic Data (2015)
[45] Y. Lv, Y.Duan, W. Kang, Z. Li, F. Wang, Traffic Flow Prediction With Big Data: A Deep Learning Approach (2014)
[46] Z. Mei, W. Ding, C. Feng, L. Shen, Identifying commuters based on random forest of smartcard data (2020)
[47]L.Breiman, Random forests (2001)
[48]J. H. Friedman, Greedy function approximation: a gradient boosting machine (2001)
[49] S. L. Zeger, A regression model for time series of counts (1988)
[50] L.Rokach, O.Maimon,Decision Trees (2005)
[51] F. Murtagh, Multilayer perceptrons for classification and regression (1990)

[52] P.Geurts,D. Ernst,L. Wehenkel,Extremely randomized trees (2005)

[53] C.Tianqi ,C. Guestrin, XGBoost: A Scalable Tree Boosting System (2016)

[54] J. Zhu, H. Zou, S. Rosset, T. Hastie, Multi-class AdaBoost (2009)

[55] T. Cover,P.Hart, Nearest neighbor pattern classification,*IEEE Transactions on Information Theory*. **13** (1): 21–27 (1967)
[56] D.Srivastava, L. Bhambhu,Data classification using support vector machine, Journal of Theoretical and Applied Information Technology 12. 1-7 (2010)
[57] H. Zhang, The optimality of Naive Bayes, Proc. FLAIRS(2004)
[58] S. Liapis, K. Christantonis, V. Chazan-Pantzalis, A. Manos, D.E. Filippidou C. Tjortjis,A methodology using classification for traffic prediction: Featuring the impact of COVID-19, Integrated Computer-Aided Engineering (ICAE),), Vol. 28, pp. 417-435, IOS Press(2021)

[59] K. Christantonis, C. Tjortjis, A. Manos, D.Filippidou, E. Christelis, Smart Cities Data Classification for Electricity Consumption & Traffic Prediction, Automatics & Software Enginery, 31(1) (2020)

[60] K. Christantonis, C. Tjortjis, A. Manos, D.E. Filippidou, E. Mougiakou , E. Christelis, Using Classification for Traffic Prediction in Smart Cities, 16th Int'l Conf. on Artificial Intelligence Applications and Innovations (AIAI 20) (2020)

[61] D. Tasios D., C. Tjortjis , A. Gregoriades , Mining Traffic Accident Data for Hazard Causality Analysis, 4th IEEE SE Europe Design Automation, Computer Engineering, Computer Networks, and Social Media Conf. (SEEDA-CECNSM) (2019)

[62]T. I. Theodorou, A. Salamanis , D. Kehagias , D. Tzovaras , C. Tjortjis , Short-Term Traffic Prediction Under both Typical and Atypical Traffic Conditions using a Pattern Transition Model, *3rd Int'l Conf. Vehicle Technology and Intelligent Transport Systems* (VEHITS 17), pp. 79-89 (2017)

[63]C.Axenie, S. Bortoli, Road Traffic Prediction Dataset (2020). Available online: https://zenodo.org/record/3653880#.Y02UoHZBy3D

[64] A. Navarro-Espinoza,O. R. López-Bonilla, E. E. García-Guerrero, E. Tlelo-Cuautle,D. López-Mancilla, C. Hernández-Mejía, E. Inzunza-González, Traffic Flow Prediction for Smart Traffic Lights Using Machine Learning Algorithms (2022)