Traffic Prediction Using Machine Learning Algorithms

Zixin Guo

Ulink College of Shanghai, 559 South Laiting Road, Jiuting Town, Songjiang District, Shanghai, 201615, China
Corresponding Author: Zixin Guo, Email: guozixin912@gmail.com

Abstract
Nowadays, the trend of urbanization is becoming increasingly significant in the metropolis. Traffic congestion is one of the negative consequences caused by this phenomenon, which leads to environmental pollution and a massive amount of potential loss of GDP. The purpose of this research is to propose methods to address this problem using machine learning algorithms such as Decision Tree (DT), multiple layer perceptron (MLP), Support Vector Machine (SVM), and Recurrent Neural Network (RNN) to predict the traffic flow.

Keywords
Traffic prediction; Congestion; Machine learning

Introduction

With a surging proportion of the population pouring into large cities every year, our world is becoming increasingly urbanized. In 2020, a study found out that there over 50 per cent of the world's population is concentrated in urban areas (The World Bank: Urban Population (n.d.)) and this number will hit 70 per cent in the year 2050. With this much population, the sale of cars from 2010 to 2021 reaches 66.7 million worldwide (Number of cars sold worldwide between 2010 and 2021 (n.d.)), it is estimated that in developed countries such as Britain, Germany, and the United States, average money of 975 US dollars is lost per person because of the time they lost during traffic congestion in 2017(The hidden cost of congestion (n.d.)). 1 per cent of the total GPA is lost in the European Union as a result of traffic congestion (Schrank, Lomax, and Eisele (2012)). Not to mention the CO2 emitted by vehicles during traffic congestion that contributes to global warming. As a result, traffic congestion has become a significant problem placing in front us.

However, as the roads are becoming more crowded, technology is also rapidly evolving. With the entering of the era of information explosion, today, the information processed by large corporations is about 1000 times more than 10 years ago, hitting 60 terabytes of information every year (Finding Value in the Information Explosion (n.d.)). Thus, it is much easier for researchers to obtain information from various sources. For example, navigation applications such as Google Maps and Baidu Maps provide information on road conditions; V2X technology enables vehicles to communicate with everything around it such as street surveillance cameras and surrounding infrastructures.

Because of these, the concept of “smart city”

Citation: Zixin Guo. (2022) Traffic Prediction Using Machine Learning Algorithms. The Journal of Young Researchers 4(17): e20220616

Received on May 7, 2022; Accepted on May 10, 2022; Published on June 16, 2022

has grown into a commonplace referring to analysing and evaluating information and data to create systems and structures to function in our city in a more efficient way (Nagy and Simon (2018a))(Qin, Li, and Zhao (2010)). In addition, in recent years, machine learning is a heatedly discussed technology and is considered to be a field which has great potential. Numerous algorithms began appearing and proved to be both accurate and effective in several areas of study including information engineering and analysis.

To achieve smart traffic and smart transportation, traffic prediction is an essential part because if we could obtain a prediction of the road condition for the next period based on our current situation, we can change the traffic signal strategies to handle it in advance. In this essay, I will first summarize other researchers' studies on traffic signal control methods and traffic prediction methods and then carry out my experiment on traffic prediction using real-world data.

Smart City and Smart Mobility

In recent decades, people were becoming increasingly aware of the fact that the major cities in the world have grown much more crowded than at any time in human history. This exerts an impact on the economy, environment, and the daily lives of people. As the result, urbanization is gradually gaining more attention from researchers, trying to deal with social and economic issues(Katz and Bradley, 2013).

Because of this, the concept of a "smart city" is proposed, which aims at making use of the information and data to conduct a management of the city (Townsend, 2013). Among the various problem considered in the area of the smart city such as carbon footprint and waste of energy, traffic congestion is one of the most significant challenges encountered by metropolitan regions(Benevolo, Dameri, and D'auria, 2016). In the past few years, there is an increasing urge for smart mobility(M. Chen, Mao, and Liu (2014)) which has led many operators to engage in traffic network management.

SCOOT

In 1973, the Transport Research Laboratory in the United Kingdom proposed a traffic signal control method called Split Cycle Offset Optimizing Technique (SCOOT) which is an adaptive traffic control system, which has been commonly used in many real-life situations. The principle of SCOOT is that at each intersection, the data of the individual vehicle passing through the detectors are recorded by the system and converted so that it can be used to establish "Cyclic flow profiles". Three parameters (split, offset, and cycle time) are used to optimize the current traffic signal control (Sharma and Gidde, 2014).

Machine Learning

In recent years, inspired by machine learning algorithms, modern developed algorithms flourished and brought new opportunities and prospects in the field of smart mobility. According to Jordan and Mitchell (2015), machine learning is the basis of artificial intelligence and big data, a combination of statistics and computer science. According to Singh, Thakur, and Sharma (2016), supervised machine learning is the construction of a set of algorithms that can generate models and recognize patterns to accurately predict future events by analysing data and information. Decision trees (DT), K-nearest neighbours (KNN), Support Vector Machine (SVM), and Neural Networks (NN) can all be used in supervised machine learning. Our project focuses on the use of different machine learning algorithms to predict traffic flow.

Traditional Machine Learning Algorithms

Multilayer Feedforward Perceptron (MLP)

MLP is a feed-forward neural network model and one of the mathematically simpler models. A limited number of layers with neurons constructed the MLP model. And three basic parts are the input layer which receives input data, the hidden layer where the inputs are analysed and processed, and finally the output layer(Pinkus (1999)). The mathematical formula of MLP output is:

$$y = \varphi\left(\sum_{i=1}^{n} \omega_i x_i + b\right) = \varphi(W^T X + b)$$

where X is the vector of inputs, w is the vector of weights, b is the bias and φ is the activation function. Nikravesh, Ajila, Lung, and Ding (2016) proposed a mobile network traffic prediction using MLP as well as Multi-Layer Perceptron with Weight Decay (MLPWD). By using a new risk minimization approach, they showed that MLPWD provided better accuracy in uni-dimensional data in traffic prediction.

Support Vector Machine

Support Vector Machine (SVM) is a classic supervised machine learning algorithm introduced by Vapnik in the 1990s by Cortes and Vapnik (1995) and is widely applied to classification and regression in pattern recognition, data mining in recent years. Cervantes, Garcia-Lamont, Rodríguez-Mazahua, and Lopez (2020). The goal of SVM is to establish a hyperplane and this can be achieved by maximizing the distance from the hyperplane to the nearest points in the data(Pradhan (2012)). This is the general equation of a hyperplane

$$aX + bY = C$$

In the research of Naik and Desai (2017) and Liang, Zhu, and Huang (2017), it was found that SVM has better performance than the other traditional machine learning algorithms. Feng, Ling, Zheng, Chen, and Xu (2018) proposed an innovative short-term traffic flow prediction model to predict both spatial and temporal features using an adaptive multi-kernel support vector machine (AMSVM). They proved that their model outperformed the other existing models and produced a better prediction because their model can make better use of the dynamic patterns of traffic flow.

K-Nearest Neighbours (KNN)

K-Nearest Neighbour (KNN) was first proposed by Cover and Hart (1967) and k represents the number of closest neighbours that are compared to define a new sample. The Nearest Neighbour algorithm can be classified into 2 types: structure-less and structure-based. In structure less KNN, there are training set and sample set, both training and sample's distance are calculated and then the k nearest neighbours are deduced. The second type is structure-based KNN, and it is based on structures of data (Bhatia et al. (2010)). In terms of traffic prediction, Zhang, Liu, Yang, Wei, and Dong (2013) proposed a K nearest model to predict the short-term traffic flow. The model is based on three parts: the database, the design of algorithms and mechanisms, and the prediction model. In their research, they can show that the accuracy is more than 90% Under a more complicated scenario, Cai et al. (2016) proposed an improved KNN model in ototer represent the spatial and temporal correlation in traffic prediction. The model is achieved by first demonstrating the spatial features such as distance and road links and then using a matrix to represent traffic states and finally was of this model are set by the Gaussian function. The result proved that their methods are superior to the others.

Decision Tree

The decision tree algorithm is a non-parametric model, which means that the information cannot be represented using a finite number of parameters. Nagy and Simon (2018b) As a result, even if there are some outliers, the result will not be affected much (Singh et al. (2016)). Different splitting standards can be used, for example, Gini Coefficient, Gain Ratio and Info Gain (Rokach and Maimon (2005)). The advantage of the decision tree is the variety of data it can handle, accurate results using limited computing power and good resistance to noise. However, there are also disadvantages such as the decrease of accuracy when classes increase (Xhemali, J HINDE, and G STONE (2009)). Models such as Random Forest (RF), Ada Boost, Gradient Boosting are all derived from traditional Decision Tree (DT). There are several basic components in the decision tree, which are nodes, branches, splitting stopping and pruning(Song and Ying (2015)). For nodes, there are root nodes, internal nodes and leaf nodes which indicates possible choices or final outcomes of decision or events. Branches are the probability of the results or occurrences that derive from nodes. Splitting is the process of dividing parent nodes into child nodes. Stopping is a rule that managed to prevent a decision from growing overly complex. Finally is pruning, pruning is an alternative solution to stopping by removing nodes to reduce their complexity. Compared with traditional methods, M.-Y. Chen (2011) proposed a prediction of corporate financial distress by using the decision tree (DT) classification. Data about the performance of 50 similar companies during a financial crisis is collected and is later used in the analysis. The DT approach achieves a better outcome than the logistic regression (LR) algorithms. Based on the basic decision tree models, Shaikhina et al. (2019) proposed a prediction for high-risk kidney transplantation using Decision Tree (DT) and Random Forest (RF) classification models. The experiment use 80 data samples and reached an accuracy of 85%. DT and RF models enable the researchers to identify risk factors related 7 to acute rejection and further make discoveries about the pattern of their data which was never found by any other statistical models.

Recurrent Neural Network

In recent years, deep learning remains a heatedly In recent years, deep learning remains a popular technology and has been applied to different fields (Krizhevsky, Sutskever, and Hinton (2012))(Yin, Wang, Wang, Chen, and Zhou (2017)). A recurrent Neural Network (RNN) is a type of neural network that belong to the field of deep learning that can capture the pattern of sequences. In 1997, Long-Short Term Memory (LSTM) was first proposed by Hochreiter and Schmidhuber (1997) and the combination of RNN and LSTM largely improved the performance of RNN to memorize long-term sequences (Graves (2012)). There has been a lot of research and experiments carried out by RNN or related LSTM models. R. Yu, Li, Shahabi, Demiryurek, and Liu (2017) proposed an RNN model combined with LSTM and successfully recognize unique patterns in peak-hour traffic and gained a 30% to 50% improvement in their data. In addition, they also conducted accident forecasting with the Mixture Deep LSTM model. Compared with traditional ML methods, Azzouni and Pujolle (2017) proposed an LSTM RNN model and used it in the Network Traffic Matrix (TM) which is the study of approximating future traffic networks by analysing the previous data. The result proved that their model is far superior to other feed-forward neural networks and traditional machine learning algorithms. Based on that, Zhao, Chen, Wu, Chen, and Liu (2017) proposed a novel model based on LSTM. This model, using a two-dimensional network consisting of a various number of memory units, take into consideration of both spatial and temporal relations in traffic prediction and that is what distinguishes it from other models. Vinayakumar, Soman, and Poornachandran (2017) compared the performance of several RNN models by combining them with LSTM, GRU and identity recurrent unit (IRNN) which can identify temporal features in a large sequence of data. All experiments are done under the same 8 conditions and standards with 200 epochs and a learning rate ranging from 0.01 to 0.5. The result showed that LSTM had the best accuracy.

Further Studies

RNN is an advanced model that can capture temporal correlations since it can handle sequenced data well. However, the spatial feature is also essential for traffic prediction because the location and structure of road networks have effects on traffic flow patterns. (Nagy and Simon (2018b)) This can achieve by using a Convolutional Neural Network (CNN) which applies convolution layers to extract spatial information of road networks. Recent studies combine RNN and CNN to gain satisfactory results. More complicated models are therefore developed to cope with these problems. B. Yu, Yin, and Zhu (2017) proposed a deep learning model called Spatio-Temporal Graph Convolutional Networks (STGCN). The research constructs the model with a complete convolutional structure so that it can be trained faster. This model has greater potential in analysing the spatial and temporal correlations, larger scalability and flexibility as well as a promising result. In terms of traffic prediction, Yao, Tang, Wei, Zheng, and Li (2019) analyse some of the weaknesses of their predecessors such as the lack of consideration of the change of dependencies between locations over time and the possibility of shifts in the peak hours. Therefore, they proposed a new model, Spatial-Temporal Dynamic Network (STDN) to try to fix and modify those deficiencies. This model combined both CNN and LSTM when dealing with both special and temporal features of traffic flow because CNN can effectively summarize road network structure while LSTM can handle sequential data. Noticeably, a Periodically Shifted Attention Mechanism (PSAM) was created for addressing the vulnerability that traffic conditions may not exactly follow daily or weekly patterns. Also using attention and graph neural network, Zheng, Fan, Wang, and Qi (2020a) proposed a Graph Multi-Attention Network for Traffic Prediction. It is designed and trained to conduct long-term traffic prediction in different road networks. It has a 9 encoder and decoder structure with the encoder system taking the input data and then analysing their features, while the decoder system act as an output, predicting the result of future traffic flow using its inputs. Noticeably, there is a "transform attention" layer that lies between the encoder and decoder, which is responsible for the process of converting past traffic features to their corresponding predictions. Their results proved that GMAN outperformed the others.

Method: Data Set

This is the format of my data set:

road_id
day_id
time_id
average speed

There are three independent variables (road_id, day_id and time_id). There are a total of 217 roads in the whole data set. Sine using all 217 roads will increase the training time of my model to a huge extent, I only picked the first several roads in my experiment. The data of every road is recorded over a period of 2 months (61 days) and day_id is used to represent the index of each day. The time interval used in this data set is 10 minutes which means every day, the data is recorded every 10 minutes. This means there are 144 records on each day. The dependent variable here is the average speed which is the main object of my experiment. This indicates the average speed of all vehicles passing through this area in this 10 minutes interval.

Algorithms

Decision Tree

There are three main formulae in the Decision Tree (DT) algorithms. The first one is Information Gain:

$$IG(T,a) = H(T) - H(T|a)$$

The second formula is Information Gain Ratio:

$$IG(T,a) = \frac{-\sum_{i=1}^{n} P(T)logP(T) - (-\sum_{i=1}^{n} P(T|a)logP(T|a))}{-\sum_{i=1}^{n} \frac{N(t_i)}{N(t)} * log_2 \frac{N(t_i)}{N(t)}}$$

And the final formula is the calculation of the Gini Factor:

$$Gini\ Index = 1 - \sum \left(P(x=k)\right)^2$$

It is calculated by subtracting the sum of the squared probabilities of each class from one. It favours larger partitions and is easy to implement whereas information gain favours smaller partitions with distinct values (Entropy, Information Gain, and Gini Index; the crux of a Decision Tree (n.d.)). I used different types of DTs, including Bagging Decision Tree (GDT), Random Forest (RF), Adaptive Boosting (AB), and Gradient Boosting Decision Tree (GBDT). These are all ensemble models and the mian difference between Bagging and Boosting is that the former train a bunch of individual models in a parallel way while the latter train the model in a sequential way(Basic Ensemble Learning (Random Forest, AdaBoost, Gradient Boosting)- Step by Step Explained (n.d.)). In RF, we first train a massive amount of individual decision trees and the prediction of those decision trees is recorded, then these predictions are used to get a final decision. 11 AB learns from the previous error to obtain higher accuracy. When the first decision tree is trained, the weighted error rate is calculated according to how many wrong predictions there are and this is used to train a new decision tree until you reach the final one. In GBDT, we first train a single decision tree and then immediately apply it to our data. The residual of this decision tree is calculated and recorded as "Y". Repeat this step until we get the number of decision trees we want.

Support Vector Machine

The main goal for the Support Vector Machine is to find a hyperplane of all the points in an n-dimensional space. A hyperplane can decide which can decide the boundaries of each class. The points that fall on either side of the plane will be classified into different categories. The hyperplane is not strictly three-dimensional, instead, it depends on the dimensions of input features. If several features are only a line, then the hyperlane is just a line. However, when dimensions become higher, maybe it is hard for us to imagine.

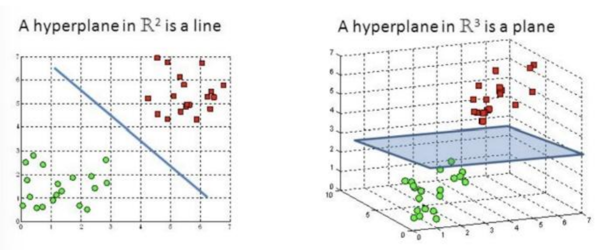


Figure 1. Hyperplanes in 2D and 3D feature space Adapted from Support Vector Machine —Introduction to Machine Learning Algorithms (n.d.)

In order to obtain the hyperplane, the SVM tries to find a way to maximize the margin (distance from the points to the hyperplane).

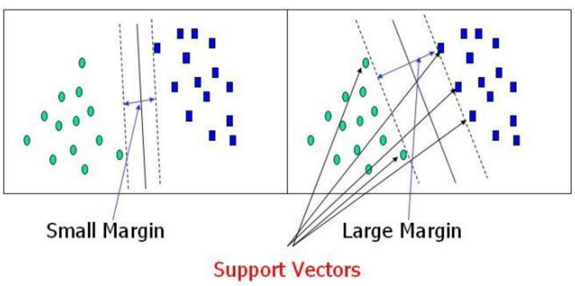


Figure 2. Support Vectors. Adapted from Support Vector Machine — Introduction to Machine Learning Algorithms (n.d.)

The loss function that helps maximize the margin is hinge loss.

$$c(x, y, f(x)) = \begin{cases} 0, \ if \ y * f(x) \geq 1 \\ 1 - y * f(x), \ else \end{cases}$$

$$c(x, y, f(x)) = (1 - y * f(x)) +$$

The cost is 0 if predicted value is the same as the actual value. If they are not, loss function can be calculated. Then, a regulation parameter is added to the cost function to balance the margin maximization and loss. After adding the regulation parameter, the cost function looks as below.

$$\min_w \lambda \|w\|^2 + \sum_{i=1}^{n} (1 - y_i \langle x_i, w \rangle)_+$$

Multilayer Perceptron (MLP)

Multiple Layer Perceptron is a feed-forward neural network containing an input layer, a hidden layer and an output layer. There is only one input layer and one output layer, but there can be a multiple number of hidden layers. In the hidden layers, sigmoid functions are normally used because sigmoid functions provide a smooth, continuous value instead of fixed
boundaries.

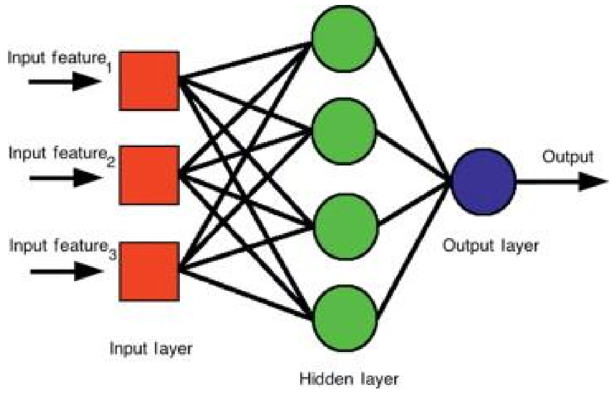


Figure 3. A three-layer MLP with three input features, four hidden neurons and one output. Adapted from Multilayer Perceptron (n.d.)

This picture shows an MLP with only one hidden layer. On the input layer, the input features are passed through an input function u which calculates the weight of the input feature.

$$u(X) = \sum_{i=1}^{n} \omega_i x_i$$

And then this result is passed to the activation function:

$$= f(u(X)) = \begin{cases} 1, & if\ u(X) > 0 \\ 0, & otherwise \end{cases}$$

Recurrent Neural Network (RNN)
RNN is a feedforward neural network that can memorize previous data and information. RNN uses its internal state (memory) to process sequences of inputs. The main difference between RNN and other Neural Networks (NN) is that RNN can memorize the key features of the previous information and thus is effective when handling sequential data. An RNN can be though as many copies of the same network and repeat itself in a loop. RNN is widely used in speech recognition or translation.

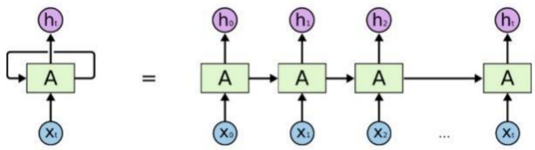


Figure 4. An unrolled recurrent neural network. Adapted from Understanding LSTM Networks (n.d.)

Long Short Term Memory (LSTM) is a type of RNN that is able to have long-term memory. In standard RNN, the repeating unit only has a tan function. However, in the LTSM network, it is much more complicated. And this is a structure of the repeating unit of LSTM.

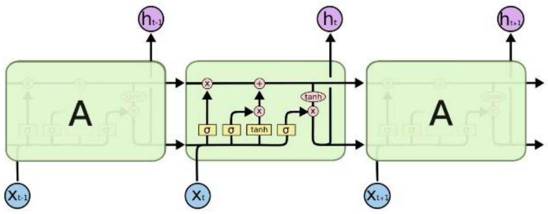


Figure 5. The repeating module in an LSTM contains four interacting layers. Adapted from Understanding LSTM Networks (n.d.)

There are mainly three gates in an LSTM network. 1. Input values first go through the input gate. The gate uses a sigmoid function to decide which memory should pass through and modify memory. Tanh function gives weight to every memory that passes through the sigmoid function.

$$i_t = \sigma(W_i * [h_{t-1}, x_t] + b_i)$$

$$\widetilde{C}_t = \tanh\,(W_C * [h_{t-1}, x_t] + b_i$$

2. The second gate is the forget gate where some unimportant details are forgotten.
It is decided by sigmoid function

$$f_t = \sigma(W_f * [h_{t-1}, x_t] + b_f)$$

3. Finally, there is the output gate. Output is decided using input values and the previous memory. Sigmoid function decided which value to pass through and tanh function give weights.

$$O_t = \sigma(W_o[h_{t-1}, x_t] + b_o)$$

$$h_t = O_t * tanh(C_t)$$

Procedure

The input data is firstly trained using Decision Tree-based algorithms, SVM and MLP and RNN algorithms. Let Vi be the average speed of cars in a specific time interval i, which i = 10 minutes. Therefore, we have 144 samples every day. The predicted algorithms we denoted as s. The prediction results are then classified into s1, s2 and s3. s1 ∈ [0,20], s2 ∈ (20,40], s3 ∈ (40,60].

Results

Regression
The graph below demonstrates the change in speed according to the change in time.

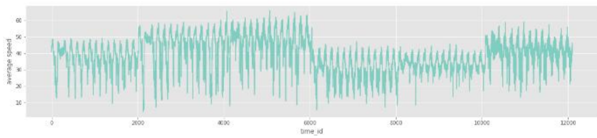
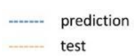


Figure 6. Average speed of my data set for 6 roads, each in a 14-day period

6 roads are selected in my experiment, each in a 2-week period. Consider on each day, there is a total of 144 data, the total number of records is 12096. As can be observed in the graph, different roads have different average speeds. For example, the average speed of road 2 and 3 are quicker than that of roads 4 and 5. A clear pattern can be observed every day with the average speed fluctuates from highest to lowest.

The graph below illustrates my regression results using different ML algorithms such as:



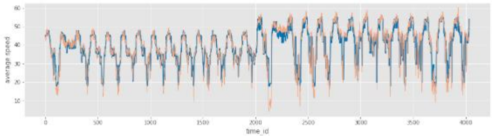


Figure 7. Average speed prediction using Bagging Tree

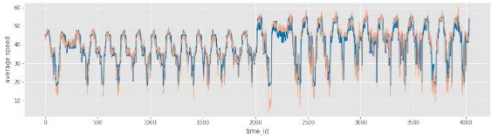


Figure 8. Average speed prediction using Ada Boost

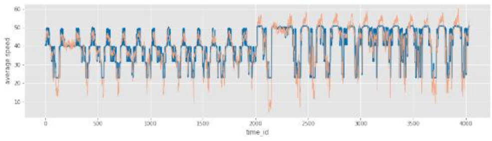


Figure 9. Average speed prediction using Random Forest

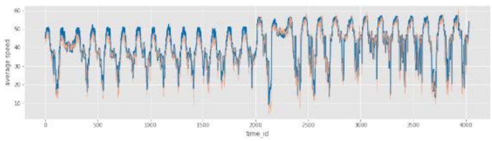


Figure 10. Average speed prediction using Gradient Boosting

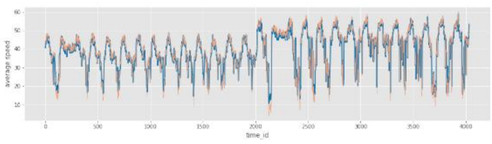


Figure 11. Average speed prediction using MLP

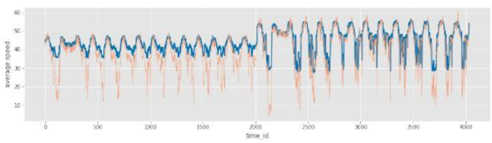


Figure 12. Average speed prediction using SVM

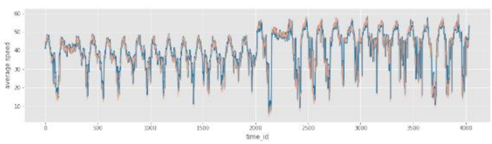


Figure 13. Average speed prediction using Linear SVR

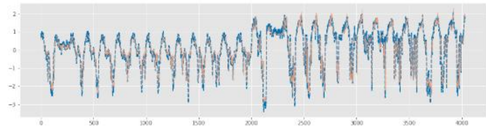


Figure 14. Average speed prediction using RNN

Table 1. Accuracies of different machine learning algorithms I used in predicting traffic

	score 1	score 2	score 3
Bagging Tree	**92.1%**	91.8%	**74.3%**
Ada Boost	**89.4%**	86.4%	**73.0%**
Random Forest	**84.6%**	83.5%	**57.7%**
Gradient Boosting	88.8%	**90.7%**	**69.0%**
MLP	**92.0%**	91.6%	**73.7%**
SVM	58.8%	**91.2%**	**49.6%**
Linear SVR	**92.2%**	91.7%	**74.2%**
RNN	90.4%	**91.0%**	**74.6%**

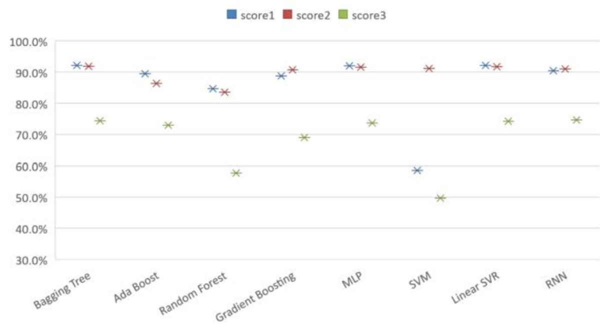


Figure 15. Accuracy of the algorithms

There are 12096 data in my data set including the average speed of 6 roads. Score 1 is obtained by using roads 3,4,5,6 as a training set, and roads 1 and 2 as testing sets. Score 2 is obtained by using roads 1,2,5 and 6 as training sets, and road 3 and 4 as testing sets. Score 3 is obtained by using roads 1,2,3 and 4 as a training set, and roads 5 and 6 as testing set. Accuracy is calculated based on mean square error.

Analysis and Evaluation

There is a common trend that score 1 and score 2 are significantly higher than score 3 for every algorithm. A potential reason for this is because from figure 5, the average speed for road 5 and 6 are lower than that of road 1, 2, 3. In score 3, the model is obtained from training from roads 1,2,3, and 4 which has a much higher average speed. As a result, these models lack the ability to provide a better performance in predicting the average 21 speed on road 5 and 6. Then, by analysing score 1, the model which obtains the accuracy from the highest to the lowest is Linear SVM (92.2%), Bagging Tree(92.1%), MLP (92.0%), RNN (90.4%), Ada Boost (89.4%), Gradient Boosting (88.8%), Random Forest (84.6%) and SVM (58.8). Noticeably, all the scores are in the range of 80% to 95%, except SVM which has an extremely low score of only 58.5%. Observing the prediction graph of SVM (figure 11), the main drawback of this model is that it has a poor performance on the range which has lower speed. For example, the average speed of road 1 is from 15 to 50, but the prediction range is from 35 to 50. However, SVM reaches 91.2% accuracy in score 2, but again, a really poor prediction in score 3. The second low prediction score is 84.6% from the random forest. In figure 8, the prediction result illustrates a rectangular shape on every extreme point of a day. This algorithm summarizes the main feature as straight lines instead of curved lines. That is the reason why it has a relatively low performance than the other models. Visualization is used to help analyse the result obtained. For example, in the SVM algorithms, fairly poor accuracy of 58.8% was obtained.

However, by only observing the test accuracy, a few conclusions can be made on the reason why this model is deficient. However, by observing the visualization of this model (figure 11), we can see that SVM does not have the ability in predicting lower speeds. By using a similar method, we can analyse directly why Random Forest has the second low performance and why the rest of the models are all fairly accurate. Finally, considering the rest of the algorithms (Linear SVM, Bagging Tree, MLP, RNN and Ada Boost), they demonstrate superior prediction results, showing great consistency with the real-world data.

Conclusion

This project focuses on traffic prediction using data collected from the city of Shenzhen, China, attempting to provide an approach to addressing traffic congestion problem in some cities. Several machine learning algorithms such Decision Trees, MLP, SVM and RNN are used to identify traffic patterns based on temporal features. The result shows an accuracy of over 90% in some superior models. We made the comparisons of different model using statistical evaluation and visualization with the fitting curve. We therefore compared different algorithms using the limited number of features but still having relatively acceptable results. In the regression problem, we found that the basic score does not give enough information in terms of how good the model could be. The information simply collapses too quickly, and especially in the regression problem, the score is too abstract for evaluating the overall performance of each model. Some models could behave well in certain conditions, while some models generally behave well enough. When two models give over 90% scores, we also need to see which one is better. Therefore, we also apply visualization to help get more information and therefore we are able to see at which segment the curve fits and also at which segment certain models do not have consistency or bad performance. We applied this simple fitting curve to compare the original data and the predicted curve. Future work can also be done to improve this visualization design to help produce more indicative information on individual model performance, which is, therefore, part of the VIS4ML workflow.

Conflict of Interests: the author has claimed that no conflict of interests exists.

References

1. Azzouni, A., & Pujolle, G. (2017). A long short-term memory recurrent neural network framework for network traffic matrix prediction. arXiv preprint arXiv:1705.05690 .
2. Basic ensemble learning (random forest, adaboost, gradient boosting)- step by step explained.(n.d.).https://towardsdatascience.com/basic-ensemble-learning-random-forest-adaboost-gradient-boosting-step-b

y-step-explained-95d49d1e2725. (Accessed: 2022/2/25)

3. Benevolo, C., Dameri, R. P., & D'auria, B. (2016). Smart mobility in smart city. In Empowering organizations (pp. 13–28). Springer.
4. Bhatia, N., et al. (2010). Survey of nearest neighbour techniques. arXiv preprint arXiv:1007.0085 .
5. Cai, P., Wang, Y., Lu, G., Chen, P., Ding, C., & Sun, J. (2016). A spatiotemporal correlative k-nearest neighbour model for short-term traffic multistep forecasting. Transportation Research Part C: Emerging Technologies, 62, 21–34.
6. Cervantes, J., Garcia-Lamont, F., Rodríguez-Mazahua, L., & Lopez, A. (2020). A comprehensive survey on support vector machine classification: Applications, challenges and trends. Neurocomputing, 408, 189–215.
7. Chen, M., Mao, S., & Liu, Y. (2014). Big data: A survey. Mobile networks and applications, 19 (2), 171–209.
8. Chen, M.-Y. (2011). Predicting corporate financial distress based on the integration of decision tree classification and logistic regression. Expert systems with applications, 38 (9), 11261–11272.
9. Cortes, C., & Vapnik, V. (1995). Support-vector networks. Machine learning, 20 (3), 273–297.
10. Cover, T., & Hart, P. (1967). Nearest neighbour pattern classification. IEEE transactions on information theory, 13 (1), 21–27.
11. Data source. (n.d.). https://download.csdn.net/download/weixin_50918736/44388374?ops_request_misc=%257B%2522request. (Accessed: 2022/1/15)
12. Entropy, information gain, and Gini index; the crux of a decision tree. (n.d.). https://blog.clairvoyantsoft.com/entropy-information-gain-and-gini-index-the-crux-of-a-decision-tree-99d0cdc699f4#:~:text=Gini. (Accessed:2022/2/24)
13. Feng, X., Ling, X., Zheng, H., Chen, Z., & Xu, Y. (2018). Adaptive multi-kernel svm with spatial–temporal correlation for short-term traffic flow prediction. IEEE Transactions on Intelligent Transportation Systems, 20 (6), 2001–2013.
14. Finding value in the information explosion. (n.d.). https://sloanreview.mit.edu/article/finding-value-in-the-information-explosion/. (Accessed: 2022/2/14)
15. Graves, A. (2012). Supervised sequence labelling. In Supervised sequence labelling with recurrent neural networks (pp. 5–13). Springer.
16. The hidden cost of congestion. (n.d.). https://www.economist.com/graphic-detail/2018/02/28/the-hidden-cost-of-congestion. (Accessed: 2022/2/14)
17. Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural computation, 9 (8), 1735–1780.
18. Jordan, M. I., & Mitchell, T. M. (2015). Machine learning: Trends, perspectives, and prospects. Science, 349 (6245), 255–260.
19. Katz, B., & Bradley, J. (2013). The metropolitan revolution: How cities and metros are fixing our broken politics and fragile economy. Brookings Institution Press.
20. Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems,
21. Liang, X., Zhu, L., & Huang, D.-S. (2017). Multi-task ranking svm for image cosegmentation. Neurocomputing, 247, 126–136.
22. Multilayer perceptron. (n.d.). https://www.sciencedirect.com/topics/computer-science/multilayer-perceptron. (Accessed: 2022/2/25)
23. Nagy, A. M., & Simon, V. (2018a). Survey on traffic prediction in smart cities. Pervasive and Mobile Computing, 50, 148-163. Retrieved from https://www.sciencedirect.com/science/article/pii/S1574119217306521 doi: https://doi.org/10.1016/j.pmcj.2018.07.004

24. Nagy, A. M., & Simon, V. (2018b). Survey on traffic prediction in smart cities. Pervasive and Mobile Computing, 50, 148–163.
25. Naik, V. A., & Desai, A. A. (2017). Online handwritten gujarati character recognition using svm, mlp, and k-nn. In 2017 8th international conference on computing, communication and networking technologies (icccnt) (pp. 1–6).
26. Nikravesh, A. Y., Ajila, S. A., Lung, C.-H., & Ding, W. (2016). Mobile network traffic prediction using mlp, mlpwd, and svm. In 2016 ieee international congress on big data (bigdata congress) (pp. 402–409).
27. Number of cars sold worldwide between 2010 and 2021. (n.d.). https://www.statista.com/statistics/200002/international-car-sales-since-1990/. (Accessed: 2022/2/14)
28. Pinkus, A. (1999). Approximation theory of the mlp model in neural networks. Acta numerica, 8, 143–195.
29. Pradhan, A. (2012). Support vector machine-a survey. International Journal of Emerging Technology and Advanced Engineering, 2 (8), 82–85.
30. Qin, H., Li, H., & Zhao, X. (2010). Development status of domestic and foreign smart city. Global Presence, 9, 50–52.
31. Rokach, L., & Maimon, O. (2005). Top-down induction of decision trees classifiers-a survey. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 35 (4), 476–487.
32. Schrank, D., Lomax, T., & Eisele, B. (2012). 2012 urban mobility report. Texas Transportation Institute,[ONLINE]. Available: http://mobility. tamu.edu/ums/report.
33. Shaikhina, T., Lowe, D., Daga, S., Briggs, D., Higgins, R., & Khovanova, N. (2019). Decision tree and random forest models for outcome prediction in antibody incompatible kidney transplantation. Biomedical Signal Processing and Control, 52,456–462.
34. Sharma, S., & Gidde, M. (2014). Estimated outcome of application of split cycle offset optimization technique (scoot) & reduction of frequency of buses. International Journal of Engineering Research and Management Technology, 1 (5), 82–90.
35. Singh, A., Thakur, N., & Sharma, A. (2016). A review of supervised machine learning algorithms. In 2016 3rd international conference on computing for sustainable global development (indiacom) (pp. 1310–1315).
36. Song, Y.-Y., & Ying, L. (2015). Decision tree methods: applications for classification and prediction. Shanghai archives of psychiatry, 27 (2), 130.
37. Support vector machine — introduction to machine learning algorithms. (n.d.). https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47.(Accessed: 2022/2/25)
38. Townsend, A. M. (2013). Smart cities: Big data, civic hackers, and the quest for a new utopia. WW Norton & Company.
39. Understanding lstm networks. (n.d.). https://colah.github.io/posts/2015-08-Understanding-LSTMs/. (Accessed:2022/2/25)
40. Vinayakumar, R., Soman, K., & Poornachandran, P. (2017). Applying deep learning approaches for network traffic prediction. In 2017 international conference on advances in computing, communications and informatics (icacci) (pp. 2353–2358).
41. The world bank: Urban population. (n.d.). https://data.worldbank.org/indicator/SP.URB.TOTL.IN.ZS.(Accessed:2022/2/14)
42. Xhemali, D., J HINDE, C., & G STONE, R. (2009). Naïve bayes vs. decision trees vs. neural networks in the classification of training web pages. D. XHEMALI, CJ HINDE and Roger G. STONE," Naive Bayes vs. Decision Trees vs. Neural Networks in the Classification of Training Web Pages", International

Journal of Computer Science Issues, IJCSI, Volume 4, Issue 1, pp16-23, September 2009, 4 (1).
43. Yao, H., Tang, X., Wei, H., Zheng, G., & Li, Z. (2019). Revisiting spatial-temporal similarity: A deep learning framework for traffic prediction. In Proceedings of the aaai conference on artificial intelligence (Vol. 33, pp. 5668–5675).
44. Yin, H., Wang, W., Wang, H., Chen, L., & Zhou, X. (2017). Spatial-aware hierarchical collaborative deep learning for poi recommendation. IEEE Transactions on Knowledge and Data Engineering, 29 (11), 2537–2551.
45. Yu, B., Yin, H., & Zhu, Z. (2017). Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. arXiv preprint arXiv:1709.04875 .
46. Yu, R., Li, Y., Shahabi, C., Demiryurek, U., & Liu, Y. (2017). Deep learning: A generic approach for extreme condition traffic forecasting. In Proceedings of the 2017 siam international conference on data mining (pp. 777–785).
47. Zhang, L., Liu, Q., Yang, W., Wei, N., & Dong, D. (2013). An improved k-nearest neighbour model for short-term traffic flow prediction. Procedia-Social and Behavioral Sciences, 96, 653–662.
48. Zhao, Z., Chen, W., Wu, X., Chen, P. C., & Liu, J. (2017). Lstm network: a deep learning approach for short-term traffic forecast. IET Intelligent Transport Systems, 11 (2), 68–75.
49. Zheng, C., Fan, X., Wang, C., & Qi, J. (2020a). Gman: A graph multi-attention network for traffic prediction. In Proceedings of the aaai conference on artificial intelligence (Vol. 34, pp. 1234–1241).
50. Zheng, C., Fan, X., Wang, C., & Qi, J. (2020b, Apr.). Gman: A graph multi-attention network for traffic prediction. Proceedings of the AAAI Conference on Artificial Intelligence, 34 (01), 1234-1241. Retrieved from https://ojs.aaai.org/index.php/AAAI/article/view/5477.