IRAQI Academic Scientific Journals

Samarra Journal of Pure and Applied Science

SJPAS

www.sjpas.com

p ISSN: 2663-7405
e ISSN: 2789-6838

Study Estimating hourly traffic flow using Artificial Neural Network: A M25 motorway case

Ahmed Ibrahim Turki[1*] and Saad Talib Hasson[2]

1- Department of Physics, College of Education, University of Samarra, Iraq
2- Information Networks Department, College of information Technology, University of Babylon, Iraq

https://doi.org/10.54153/sjpas.2023.v5i1.448

Article Information

Received: 17/01/2023
Accepted: 20/02/2023

Keywords:

Regression, traffic flow estimation, machine learning, ANN, technical indicators

Corresponding Author

Email: ahmed.ibrahim@uosamarra.edu.iq
Mobile: 07704290098

Abstract

This paper examines the challenge of accurately computing highway performance measures by estimating traffic-flow between traffic sensors that are geographically dispersed. Consequently, predicting flows vehicle values is one of the most difficult issues in the field of traffic flow prediction. Therefore, there has been a rise in interest in combining machine learning (ML) methods with indicators from technical analysis. In this paper, we suggest a hybrid strategy for generating traffic to help with this issue. Our proposed method utilizes a technical indicator and an ANN technique to predict future flows. That this method can be applied to other technical indicators while still maintaining its simplicity and effectiveness thanks to the hybrid rules is what makes it novel. The performance of the proposed artificial neural network (ANN) was evaluated with a number of other machine learning techniques to help us choose the optimal ML approach. Daily traffic data from the Motorway Incident Detection and Automatic Signalling (MIDAS) system on the M25 highway was used to test the proposed method. The achieved results demonstrate that the predictive power of ML models is augmented when ML techniques are applied to technical analysis indicators.

Introduction

In order to determine state-wide performance metrics, transportation agencies need hourly data on traffic flow and speeds. Even though average vehicle speeds can be collected from the entire road network of a state with relative ease (via probe vehicles, for example), traffic flow is notoriously difficult to ascertain due to the fact that vehicle counts are typically only available at a small number of locations with fixed traffic sensors. For the purpose of making accurate hourly flow estimates, many transportation agencies today use speed profiles or other factors in addition to AADT [1]. These hourly profiles do not, however, take into account variations in traffic due to weather, incidents, or fluctuations in daily demand, and it is often impractical and costly to install additional sensors [1-2]. In order to conduct in-depth analyses that depend on accurate flow data, transportation agencies need to find ways to improve hourly flow estimates based on limited data sources.

While most of the research in this area is concerned with forecasting short-term flows, long-term flow forecasting is also a part of this field. Short-term flow prediction typically involves creating models or learning patterns from a series of historical flow measurements in order to forecast flows for a time period in the near future. In [3], the author categorizes the various methods of short-term traffic prediction into the categories of time series analysis, function approximation, optimization, pattern recognition, and clustering. Popular parametric approaches include the ARIMA model [4,5], non-ARIMA time series models [6,7], support vector regression [8,9], and neural networks [10,11]. As examples of non-parametric approaches, we can look at graphical models [4,12] and Bayesian inference [12,13]. Recent non-parametric studies in this field have made use of deep learning (see [9-10] for examples). In contrast, long-term traffic forecasts use historical information to make less precise predictions about future traffic volumes. In this work, we use robust regression to forecast traffic flows over the next several years; for details, please refer to [11].

The focus of this paper is on the challenge of estimating future traffic flow, which is essential for transportation agencies to have in order to calculate yearly variations in network-wide performance measures (e.g., user delay cost, traffic congestion and energy efficiency). Three main points are made in the paper:

1- We propose a new hybrid approach to flow estimation using machine learning and technical analysis by combining a traffic flow with an ANN model trained on a set of technical indicators derived from the MIDAS system in the United Kingdom that are specific to traffic flows (Average True Range (ATR), Simple Moving Average (SMA), Exponential Moving Average (EMA), Relative Strength Index (RSI), Rate of Change (ROC), and Momentum (MOM).

2- Evaluate how well the ANN-induced predictive models perform.

3- Comparing the proposed method to others that also rely on machine learning.

The remaining three sections of the paper elaborate on these themes. First, we'll go over how we calibrated and analyzed the proposed method. Conclusions are drawn and several suggestions for future work are made after the results have been discussed and the model's performance evaluated.

Materials and Methods:

Obtaining a traffic data time series is the initial step in the experiment. After the data is cleaned, the technical indicators can be extracted. With the aid of technical indicators, shifts in traffic patterns can be anticipated. These indicators are fed into the ANN learning process as features. Reconstructing past hourly flows using an artificial neural network (ANN) trained with dropout and activated with an Exponential Linear Unit (ELU). Below is the proposed framework as shown in fig.1:

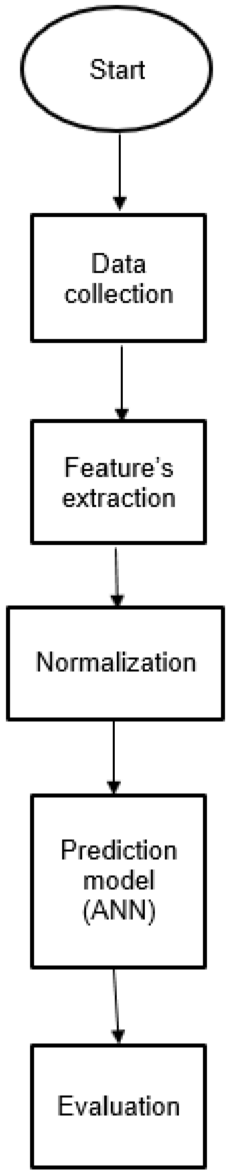

Fig. 1 *Proposed framework for artificial neural networks*

1. Data collection

MIDAS is constantly computing massive amounts of data. As a first step, storing MIDAS travel time and traffic data is necessary. Data on traffic flows and travel times were recorded every 15 minutes. If you're going to use raw data from the real world, you should expect to deal with issues like noise and missing values. Before being used, the data underwent a process of cleaning and checking for inaccuracies. We tried to get rid of or impute as many outliers and missing values as possible. The next step was to compile MIDAS traffic data to estimate hourly traffic volumes. To test the performance of the algorithm, we gathered MIDAS data from the busiest stretch of the M25 in the United Kingdom, between Junction (13-14) [12], as shown in Fig. 2.

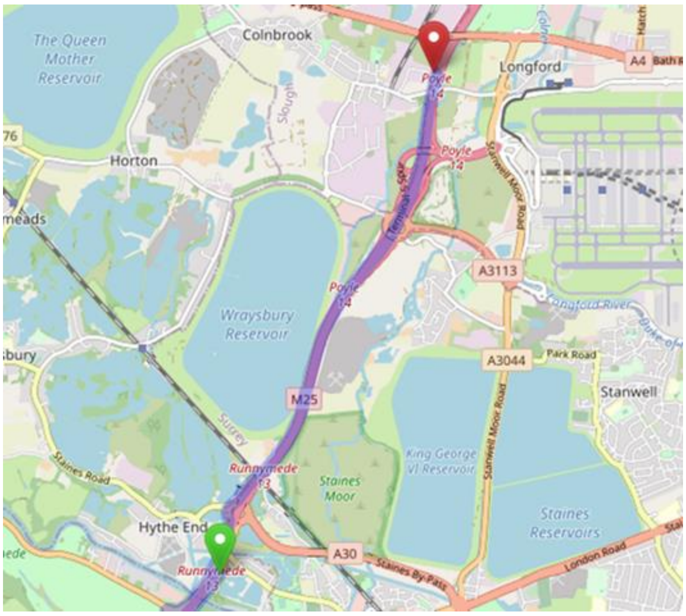

Fig. 2 *Part of (M25) highway chosen for the study from OpenStreetMap*

2. Feature extraction from data

Vehicle traffic flow and travel time over multiple days are the only variables taken into account. Thus, the format can be used to assess the quality of our input data (date, traffic flow). Data analysis yields these indicators:

2.1 Average True Range (ATR):

The ATR calculates the range and the standard deviation over a specific time [13]. It is constructed according to (1). The TR symbol denotes the actual true range.

$$ATR_n = \frac{1}{n}\sum_{i=1}^{n} TR_i \qquad (1)$$

Where:

$$TR_i = Max\{A_n, B_n, C_n\} \qquad (2)$$

$A_n = HighestFlow_n - LowestFlow_n$, $B_n = |HighestFlow_n - Flow_n|$

$C_n = |LowestFlow_n - Flow_n|$

To compute TR_i take the largest value between $A_n B_n and\ C_n$.

2.2 Simple Moving Average (SMA):

The SMA [14] is calculated by adding the traffic flow of a fleet of vehicles over a range of times, then dividing it by the range of times.

$$SMA_n = \frac{1}{n}\sum_{i=1}^{n-1} Flow_{t-i} \qquad (3)$$

2.3 Exponential Moving Average (EMA):

The EMA [15] is a type of moving average that averages flow data from the past in an exponential way. Through its weighting, the EMA can give more weight to more recent changes in flow. This weighting makes the EMA different from the SMA and lets the EMA respond more quickly to changes in traffic flow. When volatility is high, it can be helpful to put more weight on recent changes in flow.

$$EMA = \frac{d_t + \alpha * d_{t-1} + \cdots \alpha^t * d_0}{1 + \alpha + \cdots \alpha^t} \qquad (4)$$

Where:

$\alpha = \frac{s-1}{s+1}$ is the weighting term,

d_t flow at time t, α can be tailored to give more or less importance to the recent past.

2.4 Relative Strength Index (RSI):

To express normalized current flow as a percentage between (0-100). Despite its misleading moniker, this oscillator does not make comparisons between instruments but instead shows the current flow relative to output from the last lookback window length [16]. The formula for the RSI is: (5).

$$RSI_n = 100 - \left[\frac{100}{D_n}\right] \qquad (5)$$

Where:

$$D_n = \left[1 - \frac{\frac{1}{n}\sum_{i=1}^{n} Flowup[flow_i - flow_n]}{\frac{1}{n}\sum_{i=1}^{n} flowdown[flow_i - flow_n]}\right] \qquad (6)$$

2.5 Rate of Change (ROC):

a technical indicator that compares the two flows is the ratio of the present flow to the average flow across the window used to measure the time under observation [17]. The ROC may be calculated using the following formula:

$$\text{ROC} = (\text{Current flow / flow of n bars ago}) - 1.0) * 100 \qquad (7)$$

2.6 Momentum (MOM):

The MOM indicator calculates an evaluation of the current flow based on a comparison to a historical sample. Like the "Rate of Change" indicator, the MOM does not normalize the flow, so its value will vary across instruments based on their point values [18]. MOM is calculated using (8):

$$\text{MOM} = \text{Flow} - \text{Flow of n periods ago} \qquad (8)$$

3. Data preparation (Normalization)

The normalization strategy is there to transform the raw data records into a specified interval. This is because most NN model activation functions, like the Tanh function or the Sigmoid function, map the input into a fixed range. The range of the original records is, however, unclear. This is why the normalization function is so useful; it allows us to use the original data with the activation functions and speeds up the model's convergence. To standardize the following representation of traffic-flow data, which we use in our work, we employ the Z-score function:

$$v' = \frac{v - \mu_A}{\sigma_A} \qquad (9)$$

Where μ: is the mean value of the feature, σ: is the standard deviation of the feature.

4. Machine learning (Fully-connected feedforward multi-layer ANN)

Layers of neurons in a fully connected feedforward multi-layer ANN represent input variables and output predictions, respectively. Each neuron in a fully connected feedforward multi-layer ANN is linked to its counterparts in all the layers below it (i.e., no loops). Within this framework, forward propagation is used to generate model estimates, with the computation of neuron outputs occurring as follows:

$$a_i^{(l+1)} = f(w_i^{(l+1)} a^{(1)} + b_i^{(l+1)}) \qquad (10)$$

where $a_i^{(l+1)}$ is the output of the i-th neuron in layer $l + 1$, $a^{(l)}$ is the output vector from all the neurons in layer l, $b_i^{(l+1)}$ is a vector of weights between the $i - th$ neuron in layer $l + 1$ and all the neurons in layer l, $b_i^{(l+1)}$ is the bias unit associated with the $i - th$ neuron in layer In contrast to "shallow" networks, which only have a single hidden layer, the model in question has several, hence the "multi-layer" moniker. For more complex relationships, it is often more efficient to use a network with many hidden layers (a "deep network") than to simply add the neurons to the same hidden layer. Problems like overfitting and inefficient computation are real-world challenges for deep networks [1]. Overfitting occurs when a model has too many parameters and can only learn the patterns in the training dataset, rather than successfully generalizing the results to other datasets. Thus, the model's estimation accuracy is best on the training data and suffers on the cross-validation data and the testing data. The standard approaches of L1 and L2 regularization are widely used to deal with this problem, but they are often insufficient for deep networks with many parameters, reducing the model's flexibility and its ability to capture subtle dependencies. In order to prevent the ANN from overfitting the data, we use a dropout procedure recently proposed [19] to keep the network complex enough to detect subtle relations.

Dropout is a technique in which a certain percentage of neurons are temporarily ignored at each training step. Ignored neurons do not contribute to the activation of neurons further down the network, and their weights are not updated during the forward or backward pass. The feedforward operations are characterized by dropout as:

$$r^{(1)} \sim Bern(p), \qquad (11)$$

$$\tilde{a}^{(1)} = r^{(l)} a^{(l)}, \qquad (12)$$

$$a_i^{(l+1)} = f\left(w_i^{(l+1)} \tilde{a}^{(l)} + b_i^{(l+1)}\right), \qquad (13)$$

and the Bernoulli distribution parameter is denoted by p. In this paper, we use cross-validation in both the training and testing phases to verify dropout effectiveness. Particularly, there was a small difference between the error metrics on the training and testing datasets, indicating that the overfitting problem had been solved. The computational efficiency of the training procedure is a second issue with deep networks. The sigmoid function is the most frequently used activation function in shallow networks.

$$f(x) = (1 + e^{-\lambda x})^{-1}, \qquad (14)$$

that has many uses and advantages (e.g., its range is between 0 and 1, and its derivative is easy to compute). Since the derivative of the sigmoid function is between -0.25 and 0.25, increasing the number of layers in a neural network does not speed up the backpropagation process, which can lead to increased error. In order to solve this issue, a new ELU activation function [20] was created.

$$F(x) = \begin{cases} x, & x > 0 \\ \alpha(e^x - 1), & x \leq 0 \end{cases} \qquad (15)$$

is actively involved in this task. The derivative of ELU is 1 if and only if $x > 0$, while $x < 0, F'(x) = F(x) + \alpha$ for all x below 0. When compared to more traditional activation functions (like sigmoid and tanh) and deep learning functions, it accelerates the learning process.

To conclude, the ANN is trained using the Adaptive Moment Estimation (Adam) algorithm [21], a recent stochastic gradient-based optimization method well-suited to high-dimensional parameter spaces in non-convex optimization problems. New studies [22] also show that Adam's built-in bias-correction allows it to outperform other stochastic gradient-based optimization methods.

Remark 1. Recent improvements in training algorithms for large datasets [21], overfitting techniques [19], and best practices for combining the two are all part of the ANN described above, as are their respective discussions in [23].

2.4.1 Hyperparameters

The ANN's hyperparameters, such as its number of layers and the number of neurons in the hidden layers, were determined with the help of preliminary experiments. There are three layers total in this model structure: input (made up of 40 neurons each), hidden (two hidden layer made up of 60 neurons each), and output (made up of one neuron) (Fig. 3). Here is a table showing the hyperparameters we used during model training.

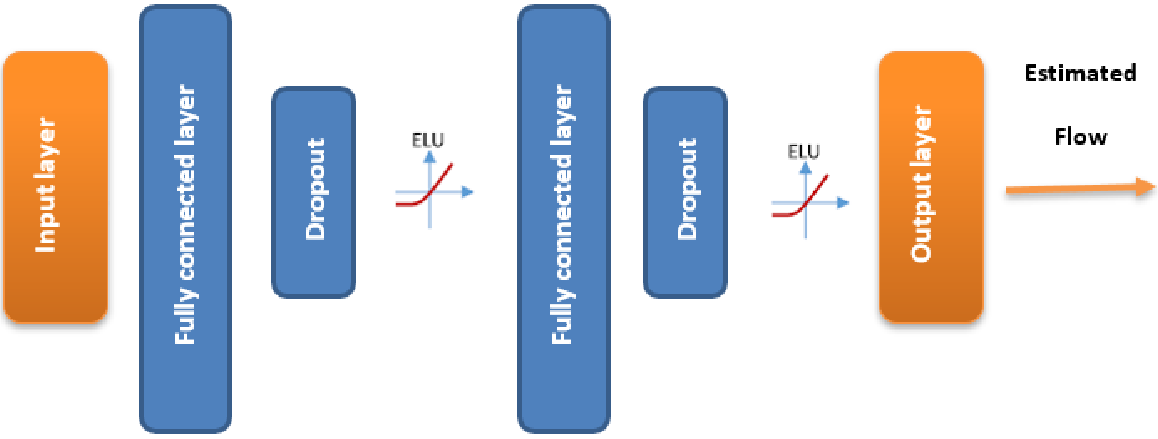

Fig. 3. *Traffic Flow is predicted using an ELU-activated fully-connected ANN.*

Table 1: Learning scheme parameters.

Learning scheme	Parameter	Description
ANN	Batch_size = 32	No. of samples/gradient update
	EPOCHS = 1000	Repeatability of the learning algorithm in terms of iterations
	Neuron1= 60, neuron2= 60	The total number of neurons in two hidden layers
	learning_rate = 0.001	How often the weights are adjusted during training

5. Model evaluation:

For the 25M highway between Junction 13-14 shown in Figure 1, the aforementioned ANN model is trained on four days' worth of data points, and then tested on a full day's worth of data. Selected technical indicators computed from 5 days of real-time data serve as the input features. To assess the efficacy of the models, the predicted flow is compared with the observed flow at this location. The steps outlined here are carried out a thousand times, and the accuracy of the models used in the tests is summarized with the two metrics we'll look at in a moment. For the M25 highway between Junction 13-14 shown in Figure 1, the aforementioned ANN model is trained on four days' worth of data points, and then tested on a full day's worth of data. After that, the predicted flow is compared to the observed flow at this location to gauge the accuracy of the model. Following a thousand iterations of the outlined process, we summarize the model accuracy for test locations using the two measures we'll go over now.

5.1 Root Mean Squared Error (RMSE):

Indicates the typical size of the mistakes and is defined as:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}(predicted_i - actual_i)^2}{N}} \qquad (16)$$

Recalls that the criterion in question uses a quadratic scoring system. In other words, it quantifies the typical size of the deviation. This metric is thus more sensitive to extreme values [24].

5.2 Mean absolute error (MAE):

Finds the typical range of absolute discrepancies between a predicted and observed value for a given sample size. The following formula describes it:

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|predicted_i - actual_i| \qquad (17)$$

MAE is a linear metric. It means that every individual difference is weighted equally in the mean. Larger errors will, therefore, add linearly to the overall error. One of the main problems with this metric is that it can be skewed by extreme cases. One benefit of using MAE is that it is a straightforward metric to understand [25].

Results and Discussion:

In this subsection, we use actual traffic data to further illustrate the efficacy of the proposed short-term traffic prediction frameworks. Typical weekday commuter traffic on a UK urban highway is used to test the reliability of the forecasts. In order to test how well the proposed framework works in other contexts, the MIDAS system collected real-world traffic data, such as link traffic flow and travel time. MAE and RMSE are used for quantitative evaluation of accuracy.

1. The ANN vs. other machine learning models

In this section, we evaluate the proposed ANN in comparison to several popular models. We evaluate it with the gradient boosting (GB), the random forest (RF) and the linear regression (LR). All of the models' overall test data performance are summarized in Table 2. Due to the linearity of the data, the LR model achieves the best results in terms of median RMSE. Although the proposed model's average performance is nearly as good as LR's, we still favour it. The findings indicate that the performance of the 2-stage and 3-stage proposed frameworks is superior to that of the 1-stage framework when both normal traffic conditions are present. This is the case regardless of the traffic variables that are used as the input for a prediction model. Because the proposed method (ANN) can detect changes in patterns relatively quickly and has the flexibility to match the most effective patterns from historical datasets.

Also, we take into account a pair of ensembles, each of which is an average of two different models: (a) the Gradient Boosting model and (b) the Random Forest model. They have the lowest median accuracy, continue to have subpar worst-case performance, and require more processing power to train multiple models on a sizable dataset. Since the ANN's overall performance and easier implementation are both highly relevant for real world application and deployment, it was chosen as the most reasonable option.

Table 2: Methods other than the one proposed by artificial neural networks were compared using MAPE and MAE.

Model	RMSE	MAE
LR	1.01	8.73
RF	95.9	64.9
GB	54.1	32.2
ANN	1.3	1.09

2. Visualize Model Training History

The loss plot demonstrates that the model achieves nearly identical results on training and testing data (labelled test). As shown in fig. 3, if these parallel plots begin to deviate consistently, it may be time to end training.

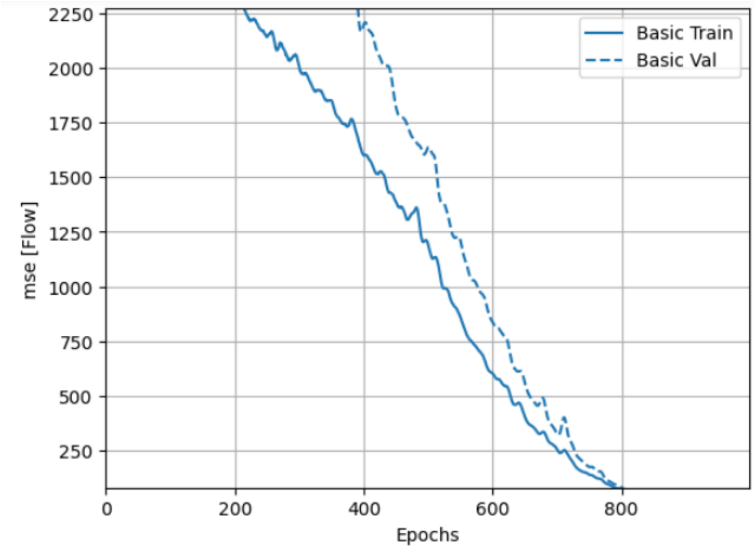

Fig. 3 *Plotting the loss/mse across epochs*

3. Sensitivity analysis

We performed a sensitivity analysis on the ANN model to determine how crucial various inputs are. In order to identify the input parameters that have the greatest impact on robustness and model performance, a sensitivity analysis is conducted [26]. In this analysis, we eliminated one technical indicator at a time from the model inputs and measured the impact on the model's performance with the two-summary metrics. Because the parameters are changed with each iteration, this method is called a parametric bootstrap [27]. Table 3 shows a summary of the findings for each eliminated technical indicator. In light of these findings, SMA emerges as the most crucial technical indicator, its exclusion resulting in a marked decline in model accuracy (RMSE=74.45, MAE=28.98). Once RSI was taken out of the model, the accuracy was still very close to the original (MAE=0.38, RMSE=0.4), making it the least important parameter. The overall profile of the results is still not quite as good as using all the technical indicators, and the ATR and ROC aren't much better.

Table 3: ANN model sensitivity analysis.

Parameter removed	RMSE	MAE
None	1.13	0.78
ATR	1.78	1.05
SMA	74.45	28.98
EMA	66.57	46.5
RSI	0.4	0.38
ROC	0.94	0.77
MOM	13.06	8.99

Conclusion

This paper provides a solution to the challenge of estimating hourly traffic flows, which is necessary for transportation agencies in need of state-wide and planning performance measurement (e.g., traffic congestion, user delay cost and energy efficiency). This proposal utilizes a method for estimating traffic flows that combine the precision of technical analysis rules with that of machine learning models. Both Linear regression and Artificial Neural Network outperformed the other machine learning methods considered. ANN's strong performance in this area has led to its widespread use in the past, particularly in attempting to foresee the occurrence of traffic jams. The success of LR in making predictions may indicate that linear models are effective for making forecasts over shorter time horizons.

We've used some carefully chosen technical indicators to fine-tune our proposal, and the results show that it's helped us get better outcomes. As a result of the positive outcomes, the proposed strategy can be applied in the future by incorporating additional parameters like density, speed, weather conditions, and others to further enhance the precision of forecasts.

Acknowledgments: The authors offer thanks and gratitude to Samarra University and Babylon University, College of Information Technology, Information Networks Department, for their endless support and complete this research.

Abbreviations

The following abbreviations are used in this manuscript:

AADT - Annual Average Daily Traffic
ARIMA - Autoregressive Integrated Moving Average
ATR - Average True Range
ANN- Artificial Neural Network
ELU - Exponential Linear Unit
EMA - Exponential Moving Average
GB - Gradient Boosting
LR – Linear Regression
ML – Machine learning
MIDAS - Motorway Incident Detection and Automatic Signaling
MOM – Momentum
MAE - Mean absolute error
Non-ARIMA – non- Autoregressive Integrated Moving Average

RSI - Relative Strength Index
ROC - Rate of Change
RMSE - Root Mean Squared Error
RF - Random Forest
SMA - Simple Moving Average

Reference

1. Sekuła, P., Marković, N., Vander Laan, Z., & Sadabadi, K. F. (2018). Estimating historical hourly traffic volumes via machine learning and vehicle probe data: A Maryland case study. Transportation Research Part C: Emerging Technologies, 97, 147-158.
2. Hou, Y., Young, S. E., Sadabadi, K., SekuBa, P., & Markow, D. (2018). Estimating highway volumes using vehicle probe data-proof of concept (No. NREL/CP-5400-70938). National Renewable Energy Lab. (NREL), Golden, CO (United States).
3. Vlahogianni, E. I., Karlaftis, M. G., & Golias, J. C. (2014). Short-term traffic forecasting: Where we are and where we're going. Transportation Research Part C: Emerging Technologies, 43, 3-19.
4. Thanoon, S. R. (2020). Modeling Time Series for Prediction of Thalassemia in Nineveh Governorate. Samarra Journal of Pure and Applied Science, 2(3), 120-131.
5. Ghosh, B., Basu, B., & O'Mahony, M. (2007). Bayesian time-series model for short-term traffic flow forecasting. Journal of transportation engineering, 133(3), 180-189.
6. Stathopoulos, A., & Karlaftis, M. G. (2003). A multivariate state space approach for urban traffic flow modeling and prediction. Transportation Research Part C: Emerging Technologies, 11(2), 121-135.
7. Zhang, Y., & Liu, Y. (2009). Traffic forecasting using least squares support vector machines. Transportmetrica, 5(3), 193-213.
8. Vlahogianni, E. I., Karlaftis, M. G., & Golias, J. C. (2005). Optimized and meta-optimized neural networks for short-term traffic flow prediction: A genetic approach. Transportation Research Part C: Emerging Technologies, 13(3), 211-234.
9. Lv, Y., Duan, Y., Kang, W., Li, Z., & Wang, F. Y. (2014). Traffic flow prediction with big data: a deep learning approach. IEEE Transactions on Intelligent Transportation Systems, 16(2), 865-873.
10. Polson, N. G., & Sokolov, V. O. (2017). Deep learning for short-term traffic flow prediction. Transportation Research Part C: Emerging Technologies, 79, 1-17.
11. Kingan, R. J., & Westhuis, T. B. (2006). Robust regression methods for traffic growth forecasting. Transportation research record, 1957(1), 51-55.
12. Highways England, Highways England network journey time and traffic flow data, 2015, [Online]. Available: https://data.gov.uk/dataset/highways-englandnetwork-journey-time-and-traffic-flow-data, (Accessed Sept. 2022).
13. Tanaka-Yamawaki, M., & Tokuoka, S. (2007). Adaptive use of technical indicators for the prediction of intra-day stock prices. Physica A: Statistical Mechanics and its Applications, 383(1), 125-133.
14. Pring, M. Technical Analysis Explained: The Successful Investor's Guide to Spotting Investment Trends and Turning Points; McGraw-Hill Education: New York, NY, USA, 2014.
15. Ayala, J., García-Torres, M., Noguera, J. L. V., Gómez-Vela, F., & Divina, F. (2021). Technical analysis strategy optimization using a machine learning approach in stock market indices. Knowledge-Based Systems, 225, 107119.

16. Ayala, J., García-Torres, M., Noguera, J. L. V., Gómez-Vela, F., & Divina, F. (2021). Technical analysis strategy optimization using a machine learning approach in stock market indices. Knowledge-Based Systems, 225, 107119.
17. Alhashel, B. S., Almudhaf, F. W., & Hansz, J. A. (2018). Can technical analysis generate superior returns in securitized property markets? Evidence from East Asia markets. Pacific-Basin Finance Journal, 47, 92-108.
18. Weng, B., Ahmed, M. A., & Megahed, F. M. (2017). Stock market one-day ahead movement prediction using disparate data sources. Expert Systems with Applications, 79, 153-163.
19. Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. R. (2012). Improving neural networks by preventing co-adaptation of feature detectors. arXiv preprint arXiv:1207.0580.
20. Clevert, D. A., Unterthiner, T., & Hochreiter, S. (2015). Fast and accurate deep network learning by exponential linear units (elus). arXiv preprint arXiv:1511.07289.
21. Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.
22. Ruder, S. (2016). An overview of gradient descent optimization algorithms. arXiv preprint arXiv:1609.04747.
23. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, *15*(1), 1929-1958.
24. Ameen, M. G., Najim, Z. A., & Ahmed, N. G. (2021). Study the Effect of the molecular modeling of the Aza Michael compounds on the cyclooxygenase-1. Samarra Journal of Pure and Applied Science, 3(4), 26-40.
25. Weng, B., Ahmed, M. A., & Megahed, F. M. (2017). Stock market one-day ahead movement prediction using disparate data sources. Expert Systems with Applications, 79, 153-163.
26. Kim, M. K., Kim, Y. S., & Srebric, J. (2020). Predictions of electricity consumption in a campus building using occupant rates and weather elements with sensitivity analysis: Artificial neural network vs. linear regression. Sustainable Cities and Society, 62, 102385.
27. Saltelli, A. (2002). Sensitivity analysis for importance assessment. Risk analysis, 22(3), 579-590.

IRAQI Academic Scientific Journals
Samarra Journal of Pure and Applied Science
SJPAS
مجلة سامراء للعلوم الصرفة والتطبيقية
www.sjpas.com
p ISSN: 2663-7405
e ISSN: 2789-6838

تقدير تدفق حركة المرور كل ساعة باستخدام الشبكة العصبية الاصطناعية: دراسة حالة الطريق السريع M25

أحمد ابراهيم تركي[1]*، سعد طالب حسون[2]
1- قسم الفيزياء، كلية التربية، جامعة سامراء، العراق
2- قسم شبكات المعلومات، كلية تكنولوجيا المعلومات، جامعة بابل، العراق

معلومات البحث:
تأريخ الاستلام: 2023/01/17
تأريخ القبـــول: 2023/02/20
الكلمات المفتاحية:
الانحدار، تقدير تدفق المرور، التعليم الالي، الشبكات العصبية الاصطناعية، المؤشرات الفنية.

معلومات المؤلف
الايميل: ahmed.ibrahim@uosamarra.edu.iq
الموبايل: 009647704290098

الخلاصة:
تبحث هذه الورقة في التحدي المتمثل في حساب مقاييس أداء الطرق السريعة بدقة من خلال تقدير تدفق حركة المرور بين مستشعرات المرور المنتشرة جغرافيًا. وبالتالي، فإن التنبؤ بقيم تدفقات المركبات هو أحد أصعب القضايا في مجال التنبؤ بتدفق حركة المرور. لذلك، كان هناك ارتفاع في الاهتمام بدمج أساليب التعلم الآلي (ML) مع مؤشرات من التحليل الفني. في هذه الورقة، نقترح استراتيجية مختلطة لتوليد حركة المرور للمساعدة في هذه المشكلة. تستخدم طريقتنا المقترحة مؤشرًا تقنيًا وتقنية ANN للتنبؤ بالتدفقات المستقبلية. إن إمكانية تطبيق هذه الطريقة على المؤشرات الفنية الأخرى مع الحفاظ على بساطتها وفعاليتها بفضل القواعد المختلطة هو ما يجعلها جديدة. تم تقييم أداء الشبكة العصبية الاصطناعية المقترحة (ANN) بعدد من تقنيات التعلم الآلي الأخرى لمساعدتنا في اختيار نهج ML الأمثل. تم استخدام بيانات حركة المرور اليومية من نظام اكتشاف حوادث الطرق السريعة والتشوير التلقائي (MIDAS) على الطريق السريع M25 لاختبار الطريقة المقترحة. توضح النتائج المحققة أن القوة التنبؤية لنماذج ML تزداد عند تطبيق تقنيات ML على مؤشرات التحليل الفني.