Street Level Travel Time Estimation Employing WBPST-SVR Model Using GTFS Features

Bhumika Patidar[1], Prof. Vinay W. Deulkar[2]
[1, 2]Dept of Civil Engineering
[1, 2] JIT, Borawan, India

Abstract- *Accurate estimation of street-level travel time is a fundamental requirement for intelligent transportation systems (ITS), enabling effective traffic management, route optimization, and real-time traveler information services. With the increasing availability of urban mobility data, particularly from public transport systems, data-driven approaches have gained prominence over traditional rule-based or simulation-based models. General Transit Feed Specification (GTFS) data provides a standardized and rich source of spatio-temporal information related to transit schedules, routes, stops, and frequencies, making it highly suitable for fine-grained travel time estimation at the street level. This paper presents a Wavelet Tree and Support Vector Regression (SVR) model for forecasting street level travel time employing GTFS features. The WBPST model has been used for filtration while the SVR model has been used for pattern recognition. The results show that the proposed model attains an MAPE of just 2.44% at 68 iterations. The model also predicts the level of congestion with an MAPE of 2.14%. The model when compared with existing benchmark models can be seen to improve upon the existing results.*

Keywords: General Transit Feed Specification (GTFS), Intelligent Traffic Systems (ITS), Statistical Models, Regression, Forecasting Accuracy.

I. INTRODUCTION

Intelligent Transportation Systems (ITS) aim to enhance mobility, safety, and efficiency in modern transportation networks through the integration of sensing, communication, and data analytics technologies [1]. A critical component of ITS is accurate traffic and travel time prediction, which supports applications such as route guidance, congestion management, and real-time traffic control [2]. However, conventional travel time prediction models often focus on primary traffic flows while overlooking parasitic traffic—unplanned, incidental, or secondary traffic movements that arise due to disruptions, roadside activities, transit operations, or urban interactions. The growing complexity of urban transport networks necessitates dedicated prediction mechanisms to account for such parasitic effects [3].

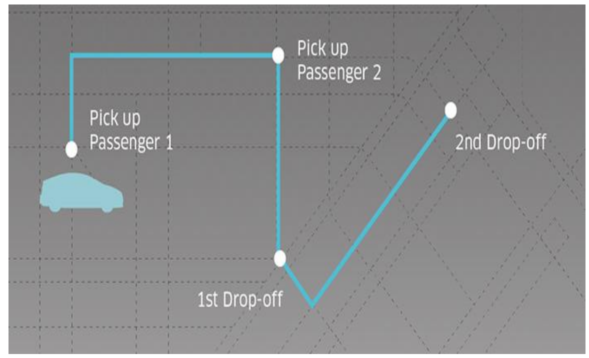


Fig.1 Route for parasitic time estimation

Figure 1 depicts the route between origin (O) and destination (D) pair.Data-driven mobility modeling and prediction are important aspects of modern urban planning [4]. With respect to travel forecasting, the two major areas of research are demand modeling and travel-time estimation, where demand modeling involves generating accurate statistics of the number of trips between origin– destination (O–D) pairs, and travel-time estimation involves predicting the travel times for trips between O–D pairs [5]. The focus of this work is on the latter, specifically, street-level travel-time estimation. In existing research on travel-time estimation, interstate link models have received disproportionate attention from the transportation research community, due primarily to the availability of large amounts of freeway sensor data. Although equally important, the same is not true for arterial models, where coverage is limited due to the costs related to installing probe sensors and associated infrastructure [6]
To forecast future travel time trends, it is necessary to model it in terms of a time series model as [7]:

$$Travel\ Time = f(time, other\ govering\ variables \quad (1)$$

Parasitic traffic refers to traffic behavior that indirectly impacts normal traffic flow, including curbside parking maneuvers, bus dwell times, ride-hailing pick-ups and drop-offs, pedestrian crossings, delivery vehicles, and spillbacks from adjacent intersections or corridors. These parasitic elements introduce localized disturbances that significantly alter travel time at the street and lane levels. In dense urban environments, the cumulative impact of parasitic traffic can lead to substantial travel time variability, making traditional macroscopic or average-based models inadequate for accurate prediction [8].

Accurate prediction of parasitic traffic-induced delays is essential for real-time traffic management and operational decision-making within ITS [9]. Without explicitly modeling parasitic influences, travel time estimates may become overly optimistic, leading to inefficient routing decisions and increased congestion. Parasitic traffic time prediction enables traffic control centers to anticipate localized slowdowns, dynamically adjust signal timings, implement adaptive lane management strategies, and deploy targeted congestion mitigation measures. This capability is particularly important during peak hours, special events, and incident-prone periods.

From a traveler information perspective, parasitic traffic time prediction enhances the reliability of navigation and mobility services. Modern ITS applications, such as advanced traveler information systems (ATIS) and multimodal journey planners, require high-resolution and context-aware travel time estimates. Incorporating parasitic traffic effects allows these systems to provide more accurate estimated time of arrival (ETA), improve route recommendations, and reduce traveler uncertainty. This is especially valuable for public transport users, freight operators, and emergency services, where minor delays can have cascading consequences.Parasitic traffic time prediction also supports sustainable and equitable urban mobility goals [10].

By identifying and forecasting delay patterns caused by parasitic activities, urban planners and policymakers can design targeted interventions such as dedicated bus lanes, optimized curb management, and improved pedestrian infrastructure. These insights contribute to reduced fuel consumption, lower emissions, and improved roadway safety. Furthermore, understanding parasitic traffic dynamics enables better integration of emerging mobility services, including shared mobility and micro-mobility, into existing transport networks [11].

The increasing availability of high-resolution data from sensors, connected vehicles, public transit feeds, and roadside infrastructure has made parasitic traffic time prediction both feasible and necessary. Advanced machine learning and data-driven models within ITS can now capture the nonlinear, spatio-temporal interactions associated with parasitic traffic. Integrating these models into ITS architectures enhances system responsiveness and resilience, enabling proactive rather than reactive traffic management [12].

II. METHODOLOGY

Conventional travel time estimation techniques, such as historical averaging, regression models, and parametric traffic flow theories, often struggle to capture the nonlinear and dynamic nature of urban traffic conditions [13]. These methods are typically sensitive to noise, require strong assumptions, and perform poorly under congestion or irregular traffic patterns. Machine learning approaches, particularly Support Vector Regression (SVR), have demonstrated strong generalization capability and robustness against overfitting when dealing with high-dimensional and nonlinear transportation data. However, the performance of SVR is heavily dependent on effective feature representation and parameter tuning [14].

To address these challenges, this study proposes a Weighted Back-Propagation Spatio-Temporal Support Vector Regression (WBPST-SVR) model for street-level travel time estimation using GTFS features [15]. The WBPST component incorporates spatial and temporal weighting into a neural learning framework, enabling the extraction of complex relationships between traffic states, transit operations, and roadway characteristics [16]. By emphasizing critical features such as peak-hour indicators, stop density, route overlap, and headway variability, the weighted back-propagation mechanism enhances the relevance of spatio-temporal patterns that directly influence travel time variability [17].

The extracted spatio-temporal features are subsequently fed into an SVR model to perform precise travel time prediction. SVR is particularly well suited for this task due to its ability to model nonlinear relationships using kernel functions while maintaining strong theoretical guarantees on generalization [18]. In the proposed WBPST-SVR framework, the SVR model benefits from the refined and weighted feature space generated by the neural network, leading to improved prediction accuracy and stability. The integration of neural feature learning with kernel-based regression allows the model to effectively handle heterogeneous GTFS data and fluctuating traffic conditions [19].

GTFS features play a central role in the proposed approach, as they encapsulate both operational and temporal characteristics of public transit systems. Key features include scheduled and actual arrival times, dwell times at stops, route lengths, stop spacing, service frequency, and temporal indicators such as time-of-day and day-of-week. These features enable the model to capture interactions between transit operations and surrounding street traffic, providing a holistic view of urban mobility dynamics. The inclusion of

GTFS data also facilitates scalability and transferability of the model across different cities and transit networks [20].

Experimental evaluation of the WBPST-SVR model demonstrates its superiority over conventional SVR, standalone neural networks, and traditional regression-based methods. Performance metrics such as Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and Mean Absolute Percentage Error (MAPE) indicate significant reductions in prediction error, particularly during peak congestion periods. The results highlight the effectiveness of spatio-temporal weighting and hybrid learning in capturing complex urban traffic behaviors at the street level**.**

The proposed approach combines:

1. Wavelet Based Persistent Segment Tree (WBPST) for filtration.
2. Support Vector Regression (SVR) for pattern recognition.

Each of the above two methods are presented in detail next:

WBPST Model:

Parasitic travel time data is extremely random in nature which makes pattern recognition difficult. Hence data preparation and pre-processing is fundamentally important for the prediction problem. Several metrics can be used to augment or bolster the pattern recognition process among which the persistent segment trees (PST) based data structure can be effective. This may help in partitioning the arrays to strings of user data to analyze some important features such as [21]:

1) Maximum clicks in a range
2) Least clicks in a range
3) Frequency of clicks in a range etc.

The concept of a persistent segment tree (PST) is depicted in figure 2.

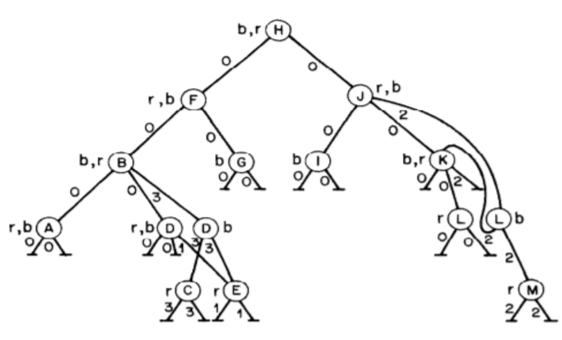


Fig.2A Persistent Segment Tree

The major problem with the PST based approach is the time complexity which is relatively high given by [6]:

$$T_{PST} = O(nlogn) \quad \textbf{(2)}$$

As the elements keep increasing, the steepness of the complexity increasing making the process computation heavy and slow.An alternative approach is the recursive binary partitioning using the Wavelet Tree. The complexity of such an approach is lesser and is given by:

$$T_{wavelet\ tree} = O(logn) \quad \textbf{(3)}$$

The recursive binary partition approach using the wavelet tree is depicted in figure 3.

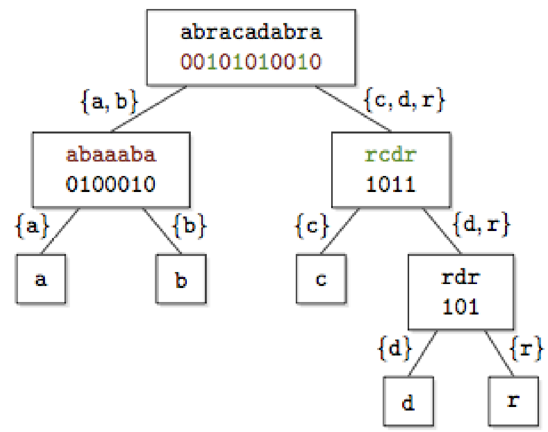


Fig.3 The wavelet Tree

The partitioning of a string or array 'S' is done in the following manner [22]:

1. Consider an array with 'n' elements denoted by S[n].
2. Find max(S) and min(S)
3. Compute the pivot point P as :

$$Pivot = \frac{lower+upper}{2} \quad \textbf{(4)}$$

4. Partition the unsorted string into two sub-strings based on the pivot values. The ones greater to the pivot go to one side of the partition and the ones smaller or equal go to the other side.
5. Recursively partition (without sorting) till you hit leaf node, (when all the elements are same in the decomposed array)

It is necessary to note that the pivot value P may or may not be an integer. The mean based partitioning is generally more common compared to the median based partitioning. The most common operations on the trees are the rank and the quantile. Rank of an element q is the frequency of the element q in the range (I,j) and is given by:

$$R_{q=}f_{q(i,j)} = f_{q(1,j)} - f_{q(1,i)} \quad (5)$$

Considering the first element to be 0 and considering the limit up to the value I,

$$R_{q=}f_{q(i)} = f_{q(j)} - f_{q(i-1)} \quad (6)$$

Quantile of k: kth largest element in the range (I,j), this helps us to avoid the persistent segment trees (PST) to solve m kth number problem.

To obtain the quantile, we can use the approach:

$$Q_{k\,(i,j)} \xrightarrow[partition]{} Q_{k(j)} : Q_{k(i)} \quad (7)$$

The quantile and the rank allow the additional features of the data set to be fed to the pattern recognition model so as to increase the accuracy of pattern analysis. The metrics often used are:

1) Initial tree
2) Best Tree
3) Best Level
4) Residuals
5) Denoised Tree

Support Vector Machine (SVM) model:

The SVM classifies based on the hyperplane.

The selection of the hyperplane H is done on the basis of the maximum value or separation in the Euclidean distance d given by:

$$d = \sqrt{x_1^2 + \cdots \dots \dots . x_n^2} \quad (8)$$

Here,
x represents the separation of a sample space variables or features of the data vector,
n is the total number of such variables
d is the Euclidean distance

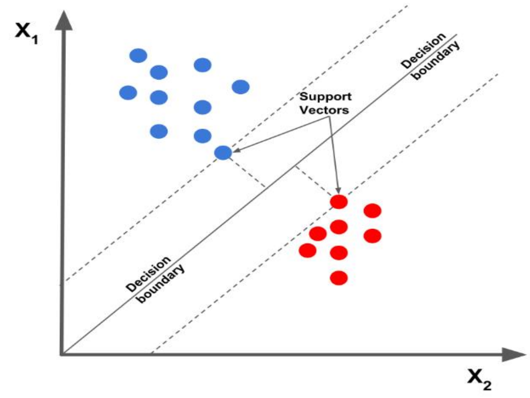


Fig.4 The SVM Model

Figure 4 depicts the SVM Model.

The (n-1) dimensional hyperplane classifies the data into categories based on the maximum separation. For a classification into one of **'m'** categories, the hyperplane lies at the maximum separation of the data vector **'X'.** The categorization of a new sample **'z'** is done based on the inequality [23]:

$$d_x^z = Min(d_{C1}^z, d_{C2}^z \dots d_{C2=m}^z) \quad (9)$$

Here,
d_x^z is the minimum separation of a new data sample from **'m'** separate categories
$d_{C1}^z, d_{C2}^z \dots d_{C2=m}^z$ are the Euclidean distances of the new data sample **'z'** from m separate data categories.

Support Vector Regression (SVR) model:

The proposed model uses the SVR model for estimating cloud spot instances. The support vector regression (SVR) model is a modified version of the support vector machine (SVM) with a modification in the objective or loss function. The advantages of support vector machines are:

1) Effective in high dimensional spaces.
2) Still effective in cases where number of dimensions is greater than the number of samples.
3) Uses a subset of training points in the decision function (called support vectors), so it is also memory efficient.
4) Versatile: different Kernel functions can be specified for the decision function. Common kernels are provided, but it is also possible to specify custom kernels.

SVMs do not directly provide probability estimates, these are calculated using an expensive five-fold cross-validation The support vector regression can be designed as a least squares optimization (LS optimization) as:

for (i=1 : n)
{
Update weights and bias
and

$$Minimize\{\frac{e_1^2+e_2^2+\cdots\ldots\ldots e_n^2}{n} \quad (1)$$

}

The SVR can be used for both linearly separable and non-linearly separable data.

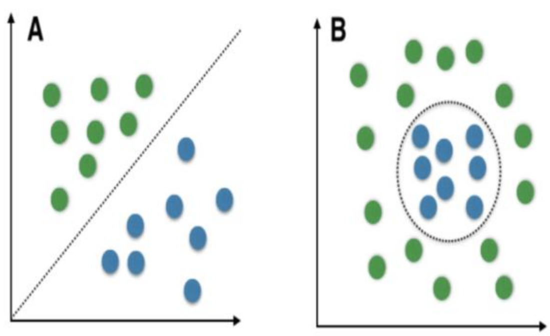


Fig.5 The linearly and non-linearly separable data

The least squares minimization approach is the fastest and most stable approach to convergence. The iterative update of the support vectors keeps changing the bias and weights to minimize the least squares objective function. The Kernalized SVR is modelled as:

To analyze non-linear data sets, linear SVR model is not applicable. Hence, Kernelized SVR is needed. The Kernel is typically a non-linear function. The Radial Basis Function Kernel (RBF) is the similarity between two points in the transformed feature space. Mathematically the SVR-RBF is defined as [24] :

$$K(X,X') = e^{-\gamma|X-X'|^2} \quad (10)$$

$$\gamma = \frac{1}{2\sigma} \quad (11)$$

Here,
γ is called the free parameter of RBF
σ is called the feature factor
K represents the RBF Kernel

X and X' are the samples in an input feature space
$|X - X'|$ is termed as the Euclidean Distance

The loss function is computed as:
The support vector regression can be designed as a least squares optimization (LS optimization) as:
Minimize $\{\frac{1}{n}\Sigma(predicted\ value - actual\ value)^2\}$
Or
Minimize$\{\frac{1}{n}\Sigma(error)^2\}$

The parameters which can be used to evaluate the performance of the SVR design for time series models is given by:

1) Mean Absolute Error (MAE)
2) Mean Absolute Percentage Error (MAPE) and
3) Mean square error (MSE)

The above mentioned errors are mathematically expressed as:

$$MAE = \frac{1}{N}\sum_{t=1}^{N} |V_t - \hat{V}_t| \quad (12)$$

Or

$$MAE = \frac{1}{N}\sum_{t=1}^{N} |e_t| \quad (13)$$

$$MAPE = \frac{100}{N}\sum_{t=1}^{N} \frac{|V_t-\hat{V}_t|}{V_t} \quad (14)$$

The mean square error (MSE) is given by:

$$MSE = \frac{1}{N}\sum_{t=1}^{N} {e_t}^2 \quad (15)$$

Here,
N is the number of predicted samples
V is the predicted value
$\hat{V}_t$ is the actual value
e is the error value

The accuracy of prediction is computed as:

$$Ac = 100 - \frac{100}{M}\sum_{i=1}^{N}\frac{E-E_i|}{i}\ \% \quad (16)$$

Here,
n is the number of errors
i is the iteration number
E is the actual value
E_i is the predicted value

III. EXPERIMENTAL RESULTS

The experimental results are presented in this section. The dataset has been taken from Kaggle:

https://www.kaggle.com/datasets/charvibannur/gtfs-traffic-prediction-dataset

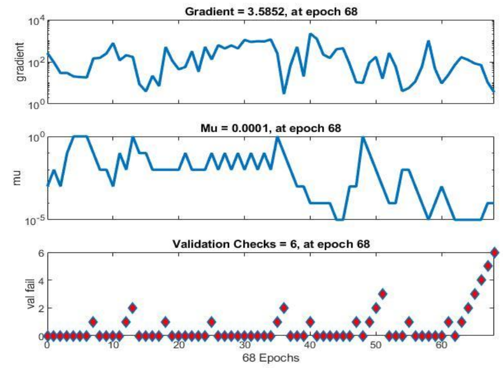
The dataset is curated by the authors from the Pune GTFS dataset by explicitly deriving multiple parameters and attributes that distinguish it from others. The data attained was categorised into 4 distinct classes using SRI (Speed Reduction Index). As a result, the massive dataset with nine csv files was condensed into a single file with minute detail on the most key facets.

• stop id from is the unique identification number for a stop or a station from where the bus departs.
• stop id to is the unique identification number for a stop or a station from where the bus is headed towards.
• trip id is a number or a set of characters that identifies a specific trip/bus.
• arrival time The time at which the bus arrives at a particular stop. Also used as the departure time in cases - where it is not explicitly mentioned.
• time Duration of trip between two consecutive stops. Calculated as the difference between arrival times between the two adjacent stops as defined in Eq. 1
Time = Arrival_time_stop_id_to − Arrival_time_stop_id_from (1)
• speed The ratio of the distance between two adjacent pauses and the time between them was used to calculate speed. The distance between the stops is determined using
the Haversine formula, which maps the distance between the two stops using the spatial attributes of each stop, such as the latitude and longitude coordinates as shown in Eq. 2.
$\text{haversin}(dr) = \text{haversin}(\phi1-\phi2)+\cos\phi1 \cos\phi2\,\text{haversin}(\lambda2-\lambda1)$ (2)

Haversine formula used to determine the great circle distance between two points given their latitude and longitude coordinates. The distances calculated using this formula are straight paths, curves are not considered. Degree of congestion labels which classify each data entry into very smooth, smooth, mild congestion and heavy congestion. Each value is classified into:

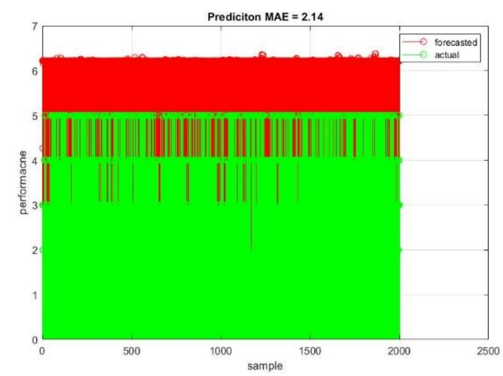
1. very smooth: **5**
2. smooth: **4**
3. moderate congestion: **3**
4. high congestion: **2**

The iterations, regression, and prediction MAE are computed next:



Fig,6 Training States

Figure 6 shows that the model trains in 68 iterations and the gradient as well as the value of learning rate can be observed.



Fig,7 Congestion Estimate

Figure 7 presents the congestion estimate with an MAPE value of 2.14%.

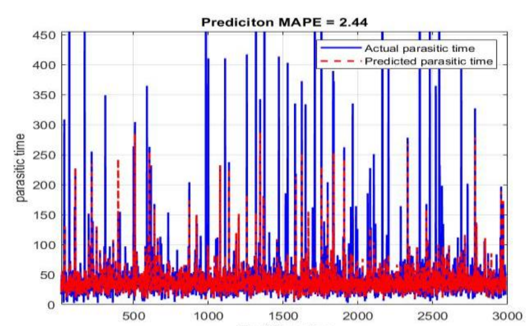


Fig.8 Parasitic time prediction

Figure 8 depicts the estimation of parasitictime with an MAPE of 2.44%.

The results are summarized in Table.1.

Table 1: Summary of Parameters

S.No.	Parameter	Value
1	GTFS Parameters	9
2	Pre-Processing Model	WBPST
3	Pattern Recognition Model	SVR
4	Iterations	68
5	Gradient	3.5852
6	Learning Rate	0.0001
7	Congestion MAPE	2.14
8	**Parasitic Time MAPE (Proposed Work)**	**2.44%**
9	**Parasitic Time MAPE (Previous Work, ANN Ukam et al.)**	**14%**

It can be observed that the proposed work attains an MAPE value of only 2.44% compared to the value of 14% of previous work. The improvement can be thought of the due to the amalgamation of the WBPST and the SVR models.

VI. CONCLUSION

It can be concluded that the proposed WBPST-SVR model presents a robust and accurate framework for street-level travel time estimation using GTFS features. By combining weighted spatio-temporal data Support Vector Regression, the approach effectively addresses the nonlinear, dynamic, and data-intensive nature of urban traffic systems. The approach combines the WBPST and the SVR models for data filtration and pattern recognition respectively. The model attains convergence in 68 iterations with an MAPE value of just 2.44% for parasitic travel time estimation clearly outperforming existing work in the domain of research.

REFERENCES

[1] S. Vlahogianni, M. Karlaftis, and J. Golias, "Short-term traffic forecasting: Where we are and where we're going," Transportation Research Part C: Emerging Technologies, vol. 43, pp. 3–19, Jun. 2018.

[2] K. Tang et al., "Lane-level short-term travel speed prediction for urban expressways: An attentive spatio-temporal deep learning approach," *IET Intelligent Transport Systems*, vol. 18, no. 4, pp. 709–722, Apr. 2024.

[3] B. Yao et al., "A deep-learning framework considering multiple motifs for traffic travel time prediction," *Proceedings of the Institution of Civil Engineers - Transport*, May 2023.

[4] M. Afandizadeh et al., "Deep Learning Algorithms for Traffic Forecasting: A Comprehensive Review and Comparison with Classical Ones," *Journal of Advanced Transportation*, vol. 2024, Article ID 9981657, 2024.

[5] M. Li, D. Zhou, and S. Zhang, "The Application of Reinforcement Learning in Traffic Flow Prediction: Advantages, Problems, and Prospects," *ITM Web of Conferences*, vol. 70, 2025.

[6] Y. Lin et al., "Long-term Traffic Flow Prediction using Stochastic Configuration Networks for Smart Cities," *IECE Transactions on Intelligent Systematics*, vol. 1, no. 2, pp. 79–90, Sep. 2024.

[7] H. Yuan and G. Li, "A Survey of Traffic Prediction: from Spatio-Temporal Data to Intelligent Transportation," *Data Science and Engineering*, Springer, vol. 6, pp. 63–85, Mar. 2021.

[8] C. Wu et al., "Spatiotemporal scenario generation of traffic flow based on LSTM-GAN," *IEEE Access*, vol. 8, pp. 186191–186198, 2020.

[9] L. Romo et al., "Short-Term Traffic Speed Prediction via Machine Learning," in *Green, Pervasive, and Cloud Computing – GPC 2020 Workshops*, Springer, Singapore, 2020, pp. 25–34.

[10] M. Cao, V. O. Li, and V. W. Chan, "A CNN-LSTM model for traffic speed prediction," in *2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*, 2020, pp. 1–5.

[11] H. Zhu et al., "Online Spatio-Temporal Correlation-Based Federated Learning for Traffic Flow Forecasting," *IEEE Transactions on Intelligent Transportation Systems*, vol. 25, no. 3, pp. 1234–1245, Mar. 2024.

[12] C. Zheng et al., "Adaptive and Interactive Multi-Level Spatio-Temporal Network for Traffic Forecasting," *IEEE Transactions on Intelligent Transportation Systems*, vol. 25, no. 2, pp. 987–998, Feb. 2024.

[13] A. M. Philip, G. Ramadurai, and L. Vanajakshi, "Urban Arterial Travel Time Prediction Using Support Vector Regression," Transportation in Developing Economies, vol. 4, no. 7, Apr. 2018.

[14] B. Priambodo and Y. Jumaryadi, "Time Series Traffic Speed Prediction Using k-Nearest Neighbour Based on Similar Traffic Data," in MATEC Web of Conferences, vol. 218, 2018.

[15] J. Tang et al., "Statistical and Machine-Learning Methods for Clearance Time Prediction of Road Incidents: A Methodology Review," Analytic Methods in Accident Research, vol. 27, pp. 1–16, 2020.

[16] K Maass, AV Sathanur, A Khan, R Rallo, "Street-level Travel-time Estimation via Aggregated Uber Data", 203 Proceedings of the SIAM Workshop on Combinatorial Scientific Computing", pp. 76-84

[17] M Akhtar, S Moridpour, "A review of traffic congestion prediction using artificial intelligence", Journal of Advanced Transportation, Wiley Online Library 2021, pp.1-18.

[18] P Chen, W Zeng, M Chen, G Yu, "Modeling arterial travel time distribution by accounting for link correlations: a copula-based approach", Journal of Intelligent Systems, Taylor and Francis, 2020, vol-6, issue-3, pp. 118-141.

[19] X. Sun, S. Wandelt, and E. Stumpf, "Competitiveness of on-demand air taxis regarding door-to-door travel time: A race through Europe," Transportation Research Part E: Logistics and Transportation Review, vol. 119, pp. 1–18, Nov. 2018.

[20] Y. Zhang and Y. Liu, "Short-term traffic flow forecasting using fuzzy logic system methods," *Journal of Intelligent Transportation Systems*, vol. 22, no. 1, pp. 1–9, Jan. 2018.

[21] M. Castro-Neto, Y. Jeong, M. Jeong, and L. Han, "Online-SVR for short-term traffic flow prediction under typical and atypical traffic conditions," *Expert Systems with Applications*, vol. 36, no. 3, pp. 6164–6173, Apr. 2018.

[22] Y. Zheng, L. Capra, O. Wolfson, and H. Yang, "Urban Computing: Concepts, Methodologies, and Applications," ACM Transactions on Intelligent Systems and Technology, vol. 5, no. 3, pp. 1–55, 2014.

[23] H. Ma, E. -P. Li, A. C. Cangellaris and X. Chen, "Support Vector Regression-Based Active Subspace (SVR-AS) Modeling of High-Speed Links for Fast and Accurate Sensitivity Analysis," in IEEE Access, vol. 8, pp. 74339-74348, 2020.

[24] Z. Tan, J. Zhang, Y. He, Y. Zhang, G. Xiong and Y. Liu, "Short-Term Load Forecasting Based on Integration of SVR and Stacking," in IEEE Access, vol. 8, pp. 227719-227728, 2020

[25] G Ukam, C Adams, A Adebanji, W Ackaah, "An artificial neural network-based model for predicting paratransit travel time", Journal of Civil And Environmental Engineering, Taylor and Francis, 2024 vol.11, no.1. pp.1-16.