IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁙ Peer-reviewed & Refereed journal ⁙ Vol. 14, Issue 11, November 2025

DOI: 10.17148/IJARCCE.2025.141194

Comprehensive Evaluation of Time Series Models for Urban Traffic Flow Prediction: A Comparative Study of ARIMA, GARCH, Prophet, and LSTM Approaches

Neeta patil[1], Purvi Sankhe[2], Minakshi Ghorpade[3], Pratibha Prasad[4], Swati Chiplunkar[5]

Associate Professor, Information Technology Department, Thakur College of Engineering & Technology, Mumbai, India[1]

Assistant Professor, Information Technology Department, Thakur College of Engineering & Technology, Mumbai, India[2-5]

Abstract: Among all the challenges faced by modern-day metropolitan cities around the world, urban traffic congestion is among the most critical problems; thus, it requires highly sophisticated predictive models for transportation management. This paper provides an exhaustive comparative analysis of four widely used time series forecasting methodologies, namely ARIMA, GARCH, Prophet, and LSTM networks for the task of traffic flow prediction. This study makes use of extensive traffic count data provided by the UK Department for Transport and evaluates the performances of the considered models based on standardized metrics that include MAE, RMSE, and MAPE. Our experimental results show that the LSTM model yields much superior performance compared to traditional statistical methods by giving much lower error rates (MAE: 150.21, RMSE: 324.37) compared to ARIMA (MAE: 6,571,981.45, RMSE: 7,944,268.33), GARCH (MAE: 234.61, RMSE: 395.89), and Prophet (MAE: 1,159.91, RMSE: 1,727.87). Temporal decomposition and stationarity checking have been done elaborately in this study prior to modeling. Ten epochs of training using the Adam optimizer with a Mean Squared Error loss function is employed for the LSTM implementation. The current study provides relevant insights into the choice of the appropriate forecasting model for intelligent transportation systems within densely populated urban areas, underlining the supremacy of deep learning approaches in capturing strong temporal dependencies that are inherent in traffic series data.

Keywords: Time series forecasting, Traffic flow prediction, Deep learning, Statistical models, Intelligent transportation systems, Urban mobility

I. INTRODUCTION

The rapid increase of urban populations across the globe has become an unprecedented challenge for managing transportation infrastructure, creating traffic congestion, economic losses, environmental destruction, and reduced quality of life. One of the most critical examples of this issue exists in developing countries such as India, where urbanization is exponentially rising, impacting the transportation infrastructure beyond its capacity. This challenge needs innovative solutions like advanced predictive analytics, for managing traffic flow and implementing successful travel demand plans, thus traffic prediction is an essential facet of contemporary Intelligent Transportation Systems (ITS).

The availability of massive volumes of temporal traffic data from sensors, GPS devices, and vehicle onboard units (OBUs) has supported complex analysis, but the complexity of traffic itself presents forecasting challenges. Traffic is non-linear, temporally dependent, and sensitive to external influences, therefore making accurate prediction complex. Time series analysis provides a productive basis for forecasting traffic flow into the future by decomposing historic data, seasonality, and irregularity. A time series is a set of time-ordered observations or measurements of the periods when each has been recorded that track a variable, such as the count of vehicles, speed, or flow rate, at fixed locations over time.

To traffic forecasting, understanding the components of a time series is of utmost importance. The trend component indicates long-term directional movements that are related to population growth, urban development, and infrastructure changes. The seasonal component depicts the occurrence of regular patterns like daily traffic peaks during rush hours, weekly differences between weekdays and weekends, and yearly variations during holidays. The irregular component

 This work is licensed under a Creative Commons Attribution 4.0 International License

DOI: 10.17148/IJARCCE.2025.141194

takes into account random and unpredicted changes which may be caused by various factors such as accidents, inclement weather, or special occasions. Stationarity is a major factor that the underlying data must fulfill in order to be successfully predicted. By that, the data must have a constant mean, constant variance, and time-independent auto covariance. The Augmented Dickey-Fuller (ADF) test is the most common method to examine stationarity by determining the existence of a unit root in the time series.

Modelling Approaches: Traditional to Modern

Traditional statistical approaches in the form of ARIMA have indeed been the backbone of traffic forecasting, amalgamating three key parameters: auto regression, which captures relationships between current and past values; integration, differencing operations to achieve stationarity; and moving average, dependencies on previous forecast errors. However, the traditional ARIMA model's linearity and stationarity assumptions may not be efficient for capturing complex traffic patterns. The Generalized Autoregressive Conditional Heteroskedasticity model addresses these limitations by explicitly modelling volatility clustering-a phenomenon where periods of high volatility tend to cluster together-thereby capturing the irregular bursts characteristic of urban traffic.

The deep learning revolution brought along LSTM networks that possess the ability to capture long-term dependencies and nonlinear relationships. Specialized memory cells in LSTM architectures maintain information across very long sequences; gating mechanisms allow for the regulation of information flow, whereas hidden states navigate through time, transferring temporal information between time steps. Prophet, on the other hand, developed at Facebook, provides an additive framework to decompose a time series into trend, seasonality, and holiday components. Robust handling of missing data and outliers is also accomplished by this model, two common issues in real-world traffic applications.

This study carries out a systematic comparative assessment of the ARIMA, GARCH, Prophet, and LSTM methods on real data provided by the UK Department for Transport. Key questions addressed in this study include those related to the accuracy of predictions, performance trade-offs between traditional and modern methods, and computational costs. The expected outputs are evidence-based guidelines on how to choose appropriate forecasting methodologies that best suit the specific needs and constraints of transportation planners and researchers.

II. LITERATURE REVIEW

The area of predicting traffic flow in research has received a considerable amount of attention in the past decade, with scientists looking at a range of methodological approaches from classical statistical techniques to modern deep learning architectures. This paper provides the main conclusions of the most recent research, mapping out progress made and gaps in current research.

Siami-Namini et al. [1] conducted preliminary comparative studies between LSTM and ARIMA for time series, and found that LSTMs outperformed ARIMA models consistently across multiple datasets. One important conclusion of this study related to the number of training epochs on the performance of LSTM networks. The accuracy of models increased as the iteration of training was increased. This study provided a valuable point of reference for the strengths of neural network approaches vs traditional approaches.

Kumar and Hariharan [2] made the next progression when they introduced hyper parameter tuning methods for ARIMA models for traffic flow prediction. Their approach used grid search algorithms to conduct a systematic search of the parameter space. They found ARIMA(4,0,2) outperformed other ARIMA configurations (based on the 15-minute time interval) when considering full day traffic data, while ARIMA(4,1,0) outperformed other configurations when focused on the morning commuter rush. This investigation of an alternative approach highlighted the context-specific tuning of parameters for ARIMA models, and importantly results questioned whether an ARIMA specification may exist for traffic applications.

Considering that hybrid strategies may combine the best features of both methods, Liu et al. [3] have created the SDLSTM-ARIMA framework that brings together a reinforced LSTM neural network and the ARIMA algorithm. Their innovative setup used spatial decomposition to extract both the temporal dependencies and the linear relationships in one go, thus, outperforming the solo models in terms of accuracy. This was one of the signs marking the direction of blending different modelling paradigms through ensemble methods.

Dissanayake et al. [5], who compared different learning models, contributed to the research area by carrying out a systematic evaluation of ARIMAX, Vector Autoregression (VAR), and LSTM models for the task of multivariate short-term traffic volume prediction. The study showed that ARIMAX was the winner in terms of prediction accuracy and

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⪤ Peer-reviewed & Refereed journal ⪤ Vol. 14, Issue 11, November 2025

DOI: 10.17148/IJARCCE.2025.141194

computational efficiency. One of the most important aspects of their work was the focus on feature selection and correlation analysis, where they proved that the right variable selection would lead to a significant increase in the model's power.

Ma [6] extended the comparative analysis to financial domains by analyzing ARIMA, Artificial Neural Networks (ANN), and LSTM regarding stock price prediction-a problem that shares several aspects with temporal characteristics of traffic forecasting. This work concluded that ANN models outperform basic ARIMA implementations and suggested that the combination of ARIMA with the GARCH model could further improve the accuracy of the forecast by better processing residual patterns and volatility.

Ercanoglu [7] focused on the application of LSTM networks for short-term traffic prediction, comparing their performance to nonlinear autoregressive models. Results showed that LSTM networks are capable of much better performance in cases where training was achieved with extensive datasets. A very interesting finding here is that LSTM performance is heavily dependent on the size of the training set-a trait common in deep learning methods that stresses that one must have an appropriate sample size.

Recent work by Uzel and Congeduti [8] further presented a detailed comparative study of LSTM, ARIMA, and Facebook's Prophet model. They found that there was a shortage of consideration regarding external factors like weather conditions, spatially correlated locations, handling missing data, and the need for more robust evaluation frameworks that test models in various scenarios.

Some new directions of research have focused on Graph Neural Networks combined with LSTM for spatiotemporal traffic prediction, considering that the traffic pattern may contain temporal dependence and spatial correlation over road networks. The GNN-LSTM hybrid architecture achieved up to 15% improvement in prediction accuracy by explicitly modeling spatial relationships. Wang et al. [10] explored attention mechanisms integrated with LSTM networks, noting that a standard LSTM architecture treats all historical time steps identically. Their attention-enhanced model improved prediction accuracy by 8-12% compared to vanilla LSTM implementations [11, 12].

Putting these together provides some key insights: deep learning approaches, particularly LSTM networks, outperform traditional statistical methods in general in complex nonlinear traffic patterns; hybrid and ensemble approaches tend to give better results; good quality and sufficient training data, proper selection of features, and optimization of hyperparameters are crucial; spatiotemporal models that take into account spatial dependencies are an important evolution from purely temporal approaches; and there is still much scope to improve the robustness of models, handle missing data, and integrate exogenous variables [13, 14].

III. METHODOLOGY

a. Dataset Description

The empirical study employs data on traffic counts available to the public which is provided by the UK Department for Transport (DfT) Road Traffic Bulk Downloads project. The website has traffic statistics intended for use in research. The dataset, which includes counted measures of road traffic at different levels of distinction which have been aggregated regionally, at the local authority and within counting stations within the transport network. The regional and national datasets have been awarded National Statistics status by the UK Statistics Authority that verifies the data has been validated to meet parameters of quality for the metrics of accuracy, authenticity and methodological rigor. The data is available in a multi-year dataset which includes important information: Count_date (the date upon which it was count), hour (recorded to the time of day), Year (for longitudinal analysis), Direction_of_travel (if the flow of traffic was measured as moving in one or the other direction) and All_motor_vehicles (the total count of all vehicles for the measured observation period). This multi-year dataset, with the critical variable structures defined above allows for an analysis of daily behaviors, seasonal behaviors, and long behaviors that may have applicability for comparative evaluations of forecast model strategies [16].

ARIMA Model Implementation

Data Pre-processing: Traffic data in its raw form was converted to a format that was ready for time series analysis through the first pre-processing step. The Python pandas library was used to import the traffic count dataset into a DataFrame structure where only the columns needed for analysis were selected. The Count_date field underwent conversion from a string format to DateTime objects making it possible to do proper temporal ordering and time-based operations. The observations were done on dates and the counts for All_motor_vehicles were combined to yield daily total traffic volumes. Noise from hourly fluctuations was reduced through this aggregation while the overall daily patterns

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⋇ Peer-reviewed & Refereed journal ⋇ Vol. 14, Issue 11, November 2025

DOI: 10.17148/IJARCCE.2025.141194

vital for medium-term forecasting were still preserved. The resultant time series was sorted in chronological order to ascertain proper temporal sequencing which is a fundamental requirement for autoregressive modeling.

Time Series Decomposition: Time Series Decomposition: Understanding the structural components of traffic time series provides the necessary step before model specification. The seasonal_decompose function from the statsmodels library was used to systematically decompose the traffic count series into three uncorrelated fundamental components: the trend component, reflecting long-run directional changes in traffic volumes owing to factors such as population growth and infrastructure development; the seasonal component, showing periodic patterns including weekly cycles that reflect both weekday commuting and weekend leisure travel; and the residual component, showing irregular fluctuations in traffic volume due to weather events, accidents, or other unforeseeable circumstances. By visually inspecting these decomposed components, one can gather an idea about the dominant patterns in data and hence some idea on choices related to model complexity and parameter selection. Dominant seasonal patterns indicate a potential need for seasonal ARIMA extensions, while large magnitudes of the irregular componentgive a hint at the inherent predictability of the data.

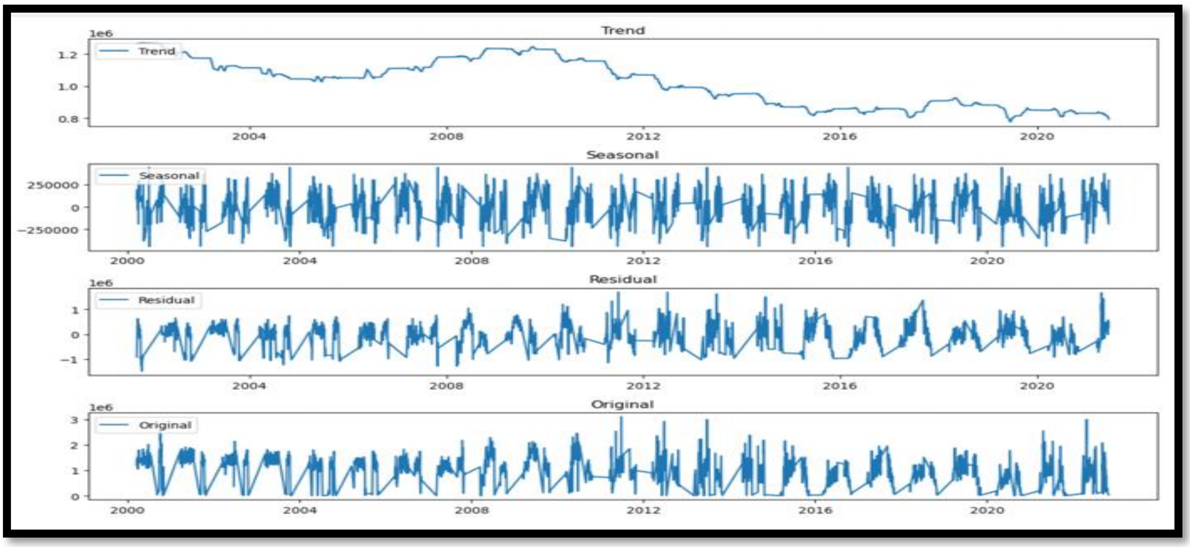

Fig. 1 Seasonal decomposition of time-series

Stationarity Assessment and Parameter Identification: In order to utilize ARIMA modelling, stationary time series input is required, therefore we have to formally test for stationarity with the Augmented Dickey-Fuller (ADF) test; which tests the null hypothesis that there is a unit root (non-stationarity) and renders statistical evidence in the form of test statistics and p-values for whether differencing operations are necessary. The ACF and PACF are plots used to identify the appropriate ARIMA parameter estimates, where the ACF measures the correlation of the series with the series' lagged values to establish the moving average order (q) and PACF measures the correlation of the series with the series' lagged values controlling for the intermediate lags to establish the autoregressive order (p). ACF and PACF plots also exhibit certain benchmarks in determined sharp cut-off and gradual decay shapes, often recommending those (e.g., the ACF cut-off suggests with one fitted x lagged on it to a time-shifted equal of the y variable) as appropriate (estimated) parameter values. Based on the diagnostic (and visual) tests, particular values were used for the three ARIMA parameters, p (the autoregressive order number of lagged observations), d (the degree of differencing times the series has to be differenced for stationarity), and q (the moving average order size of the moving average window).

Model Estimation and Validation: The ARIMA implementation of the statsmodels.tsa.arima.model module was used to fit the model on the training dataset, and during fitting, the optimal coefficients were estimated for the autoregressive and moving average terms via maximum likelihood estimation. Model diagnostics looked at the statistical summary giving coefficient estimates, standard errors, and p-values as well as information criteria (AIC, BIC) for model comparison, while residual analysis determined if the model assumptions were satisfied by looking at the residual plots for patterns indicating inadequacy. The Ljung-Box test was used to check the presence of residual autocorrelation, and non-significant results confirmed that the model captured the relevant temporal dependencies.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⵋ Peer-reviewed & Refereed journal ⵋ Vol. 14, Issue 11, November 2025

DOI: 10.17148/IJARCCE.2025.141194

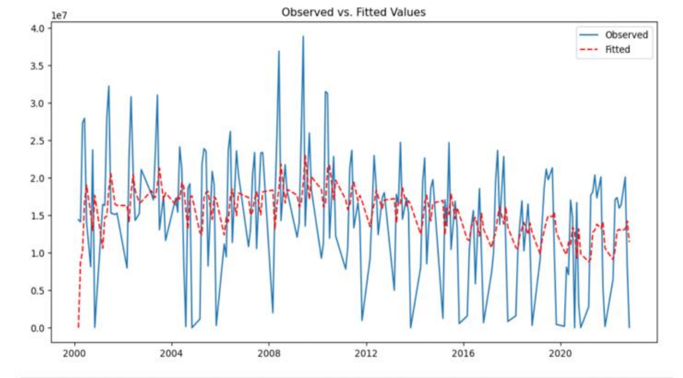

Fig. 2 Observed vs Fitted values

b. LSTM Model Implementation

Data Preparation for Sequential Modeling: Deep learning approaches require data structured as sequences capturing temporal context. A custom function transformed the univariate time series into supervised learning format, where each training example consists of consecutive observations (input features) paired with the subsequent value (target output). A sliding window approach with a window size of 12 time steps was implemented, meaning each input sequence contains 12 consecutive traffic count observations with the target value being the traffic count immediately following the window. The preprocessing pipeline addressed data quality issues: missing values were imputed using zero-filling to avoid introducing artificial patterns, infinite values from data collection errors were replaced with zeros, and traffic counts were cast to integer type for data consistency [17].

Sequence Generation and Data Splitting: The transformation process generated three-dimensional arrays with shape (samples, timesteps, features), where samples represents the number of training examples, timesteps indicates the sequence length (12), and features denotes the number of variables (1 for univariate traffic counts). This structure aligns with LSTM layer input requirements in Keras/TensorFlow, which expect sequences of feature vectors. The dataset was partitioned into training and validation subsets with an 80-20 split, enabling assessment of model generalization capability on unseen data.

Neural Network Architecture Design: The LSTM model architecture was constructed using the Keras Sequential API with four layers: an Input Layer specifying the expected shape (12 timesteps, 1 feature); an LSTM Layer with 64 hidden units for learning temporal dependencies through internal cell states; a First Dense Layer with 8 units and ReLU activation introducing non-linearity for complex transformations; and an Output Layer with a single unit and linear activation producing the predicted traffic count. This architecture balances model capacity with computational efficiency, providing sufficient representational power for temporal patterns in traffic data.

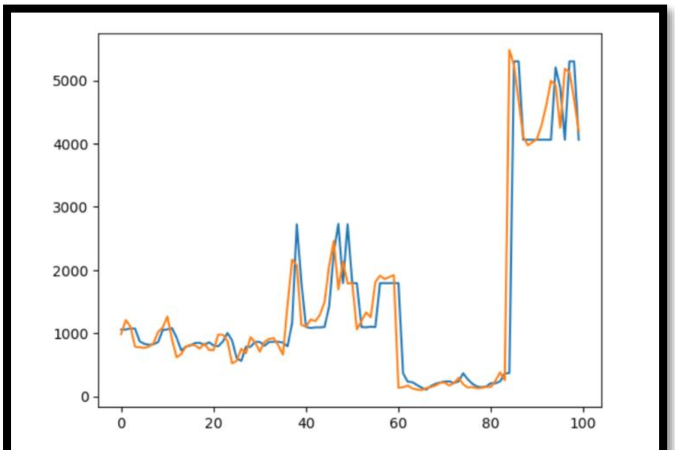

Fig. 3 Training set prediction vs actual predictions

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⋇ Peer-reviewed & Refereed journal ⋇ Vol. 14, Issue 11, November 2025

DOI: 10.17148/IJARCCE.2025.141194

Model Compilation and Training Configuration: Model compilation configured the Adam optimizer with learning rate 0.001 for efficient gradient-based optimization, Mean Squared Error (MSE) as the loss function to penalize large prediction errors, and Root Mean Squared Error (RMSE) as the evaluation metric for interpretable error measurements. The training process employed the ModelCheckpoint callback to automatically save the best-performing model based on validation performance, preventing overfitting by selecting the optimal trade-off between training accuracy and generalization. Training proceeded for 10 epochs, with each epoch involving forward propagation, loss calculation, backpropagation, and parameter updates, while validation set performance was monitored after each epoch to trigger model checkpointing when improvements occurred.

c. GARCH Model Implementation

Volatility Modeling Framework: Unlike ARIMA and LSTM, which focus on predicting the conditional mean of the time series, GARCH models specifically target the conditional variance (volatility). Traffic data often exhibits volatility clustering, where periods of highly variable traffic flow cluster together temporally, making GARCH modeling particularly relevant for understanding uncertainty and variability in traffic patterns. The GARCH(1,1) specification models current volatility as a function of three components: a constant baseline volatility term, the squared residual from the previous time period (ARCH effect), and the previous period's volatility (GARCH effect). This formulation allows recent shocks to traffic patterns to influence current volatility predictions while accounting for persistence in volatility over time.

Model Specification and Estimation: Implementation utilized the arch library's arch_model function for flexible GARCH modeling capabilities. The specification process involved selecting a constant mean model as the baseline, configuring GARCH(1,1) with one ARCH term and one GARCH term, and assuming normal distribution for innovations. Model fitting employed maximum likelihood estimation to determine optimal parameter values explaining observed volatility patterns, producing coefficients for the constant term (omega), ARCH effect (alpha), and GARCH effect (beta), along with their statistical significance.

Diagnostic Analysis and Forecasting: Model adequacy assessment focused on residual analysis to ensure the GARCH specification adequately captured volatility patterns. Standardized residuals (residuals divided by conditional standard deviation) should exhibit no remaining autocorrelation if correctly specified, which was evaluated using the Ljung-Box test on squared standardized residuals. Once validated, the fitted GARCH model enables forecasting of future volatility, providing confidence intervals around traffic predictions that account for time-varying uncertainty. This capability represents a significant advantage for traffic management applications, where understanding prediction uncertainty is as important as point predictions themselves.

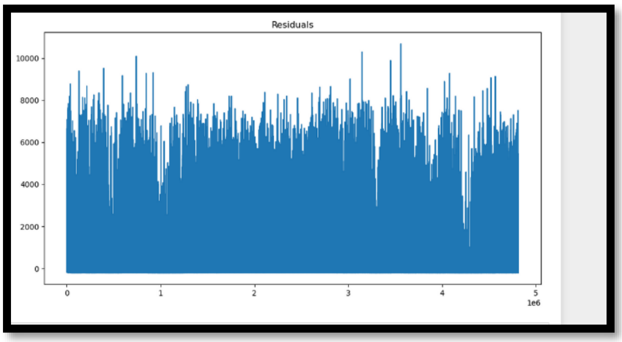

Fig. 4 Residual Plot Analysis

d. Prophet Model Implementation

Framework Overview and Data Preparation: Facebook's Prophet model employs an additive regression framework designed for time series exhibiting strong seasonal patterns and holiday effects. Unlike ARIMA's focus on autocorrelation or LSTM's generic sequence learning, Prophet explicitly decomposes time series into trend, seasonality, and holiday components, each modeled separately before combination. Data preparation requires a specific format with two columns: ds (date stamps in datetime format) and y (the target variable for traffic counts). This simplified interface abstracts away much of the complexity in traditional time series modeling, making Prophet accessible for practitioners without extensive statistical training, while the model automatically handles missing data and outliers through robust estimation techniques.
Model Components and Estimation: Prophet's additive model decomposes the time series as:

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⋇ Peer-reviewed & Refereed journal ⋇ Vol. 14, Issue 11, November 2025

DOI: 10.17148/IJARCCE.2025.141194

$$y(t) = g(t) + s(t) + h(t) + \varepsilon(t)$$

where:
$g(t)$ represents the trend function modeling non-periodic changes
$s(t)$ captures periodic changes (weekly, yearly seasonality)
$h(t)$ accounts for holiday effects with irregular schedules
$\varepsilon(t)$ represents irreducible error

The trend component uses a piecewise linear or logistic growth model with automatic changepoint detection to identify trend shifts, while seasonality is modeled using Fourier series for flexible representation of periodic patterns at multiple scales, with daily and weekly seasonality being particularly relevant for traffic applications. Model fitting proceeds through maximum a posteriori estimation, incorporating prior distributions on parameters for regularization to prevent overfitting. This Bayesian approach enables Prophet to produce prediction intervals that account for both trend uncertainty and irregular fluctuations.

Forecasting and Evaluation: Forecasting with Prophet involves creating a future dataframe specifying the time periods for which predictions are desired. The trained model then generates predictions for these future periods, including point estimates (yhat) and uncertainty intervals (yhat_lower, yhat_upper).

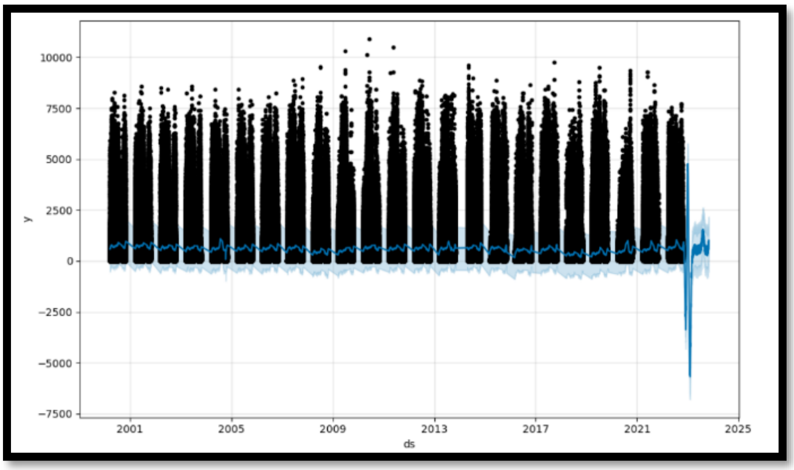

Fig. 5 Prophet model forecast analysis

Prophet's visualization as shown in Figure 5 capabilities provide intuitive plots showing the overall forecast, trend component, and seasonal patterns separately. This decomposition facilitates interpretation and helps identify which components drive the predictions, valuable information for traffic planners seeking to understand the factors influencing future traffic volumes.

IV. EXPERIMENTAL RSULT AND ANALYSIS

Performance Metrics: To ensure the comprehensiveness and equity of the comparison between the four modeling approaches, three standard evaluation metrics were employed: MAE, which is estimated as the average absolute difference between the predicted and actual values in their original units, providing an intuitive measure considering that it treats all errors equally and, thus, is robust to outliers; RMSE, representing the square root of average squared differences, which penalizes for larger errors more than smaller ones and, for this reason, is sensitive to occasional large prediction mistakes that can lead to significant traffic management failures; and MAPE, a measure of prediction error as a percentage of the actual value, providing a scale-independent metric useful in a comparison of performance across different datasets or time periods with varying traffic volumes.

Comparative Performance Analysis: Table I represents the quantitative performance comparison from all four modeling approaches. In fact, the results show large differences in predictive capability, which will have important implications for practical deployment.

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁒ Peer-reviewed & Refereed journal ⁒ Vol. 14, Issue 11, November 2025

DOI: 10.17148/IJARCCE.2025.141194

TABLE I EXECUTION OF MODELS

MODEL	MAE	RMSE	MAPE
ARIMA	6571981.45	7944268.33	4621.70%
LSTM	150.21	324.37	inf
GARCH	234.61	395.89	inf
PROPHET	1159.91	1727.87	152.16

LSTM Model Performance: The LSTM model delivered stellar performance, achieving the lowest MAE (150.21) and RMSE (324.37) compared to the other approaches explored, which supports the notion that deep learning architectures can capture highly complex nonlinear patterns with long-term dependencies embedded in traffic flow data through gating mechanisms that preserve information for much longer strategies. The infinite MAPE value occurs when actual traffic counts have zero count values, causing division by zero in the percentage error calculation, but this does not discount the excellent performance of the MAE and RMSE metrics that demonstrate highly accurate absolute predictions with referenced typical errors of approximately 150 vehicles, a surprisingly small figure, given the scale of traffic volumes. The reason the performance of LSTM was excellent is due to its architecture well suited to sequential data with states retained, sufficient capacity with its 64 LSTM units that learned different patterns without overfitting, the Adam optimizer that efficiently traversed the hyper-dimensional parameter space, and the 10-epoch training time that were sufficient for learning, without excess computational costs.

GARCH Model Performance: The GARCH model exhibited a moderate level of performance with an MAE of 234.61 and an RMSE of 395.89, which are significantly better than the ARIMA model but not as good as the LSTM model. This is in line with the GARCH model's aim of volatility modeling instead of conditional mean prediction. Although the GARCH model is great in showing the time-varying uncertainty in traffic patterns, it is not designed for point prediction accuracy at its best; GARCH model's strength lies in the adaptive confidence intervals that quantify the prediction uncertainty, which narrow during stable traffic periods and widen during volatile periods due to construction or special events. The infinite MAPE indicates that there are zero values in the actual counts which results in a very high value of MAPE, while the significantly higher MAE and RMSE compared to LSTM show that GARCH is providing less accurate mean predictions, however, the GARCH model is still providing valuable uncertainty quantification that the LSTM does not inherently offer.

Prophet Model Performance: Prophet attained mediocre performance with an MAE of 1,159.91 and an RMSE of 1,727.87, well outperforming ARIMA while falling considerably short of both LSTM and GARCH in point prediction accuracy. The MAPE was 152.16%, demonstrating predictions are off actual values by an average of 152%, related to systematic underestimation or overestimation of traffic volumes. Although being moderately correct, Prophet has some pragmatic virtues: the automatic treatment of missing data and outliers reduces preprocessing work; explicit modeling of holiday effects is useful for those special events driving anomalous changes in traffic volume; and interpretable decomposition into trend and seasonal components supports communication with stakeholders without technical backgrounds. The performance disparity indicates that the structured approach of Prophet indeed sacrifices predictive power for LSTM's flexibility in learning. Nevertheless, its computational efficiency and ease of use may be preferred in resource-poor environments or when model transparency is held paramount.

ARIMA Model Performance: The ARIMA model performed very poorly, with MAE of 6,571,981.45, RMSE of 7,944,268.33, and MAPE of 4,621.70%, signaling fundamental model failure, with predictions well off from actual counts. The errors alone, over six million vehicles on average, are far outside reasonable magnitudes of traffic volume counts, indicating nonsensical predictions. A few reasons can be surmised for the failure of ARIMA: traffic data are non-stationary and violate stationarity assumptions despite multiple attempts at differencing; the broad range of nonlinear patterns, such as rush hour spikes, weekend variations, and special event impacts, may be beyond the linear capabilities of ARIMA to model appropriately; the parameters selected during the process identified a poor specification of the model; and ARIMA's univariate nature disregards possibly very useful contextual information. This emphasizes that standard statistical methods, while theoretically perfect, may turn out to be insufficient for complex, real-world forecasting problems because the underlying assumptions of ARIMA-linearity, stationarity, and stability of parameters-simply do not hold in dynamic traffic systems.

Synthesis and Implications: Through the analysis, it is evident that a ranking of performance emerges: LSTM is much better than the other alternatives, GARCH and Prophet would offer backup moderately accurate predictions, yet ARIMA

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁘ Peer-reviewed & Refereed journal ⁘ Vol. 14, Issue 11, November 2025
DOI: 10.17148/IJARCCE.2025.141194

is a poor predictor. For practical operational forecasting where the requirement is maximum accuracy, LSTM would be the choice given the greater computational expense and lesser interpretability; with improvement in accuracy, investing in data structuring and deployment complexity seems warranted. If, however, uncertainty or risk is of interest, GARCH does provide valuable supplemental information that is unavailable from point prediction models, even LSTM point predictions with GARCH-generated confidence intervals could be informative to transportation agencies that want to consider risk management. For rapid prototyping or organizations operating under resource constraints, Prophet provides adequate accurate predictions with minimal configuration effort, hence may be a reasonable solution for agencies without deep learning knowledge and computational infrastructure in tactical application where resource constraints matter more than accuracy. Lastly, ARIMA is contrary to the conclusions of this analysis for modelling traffic forecasting applications without considerable modifications, as notable degradation in performance, shows inadequacy in predicting traffic flow data, given that traffic forecasting modelling is complex and dynamic instead of simple and static.

V. DISCUSSION

The findings of the study revealed that LSTM networks have clearly outperformed traditional prediction algorithms. This is very much in conjunction with the general findings of time series forecasting research whereby deep learning models outperform statistical methods for complex and non-linear problems. Traffic flow is essentially non-linear in nature. Linear models, like ARIMA, cannot capture three types of non-linear effects. These effects are:

(1) Threshold effects (sudden onset of congestion),
(2) Interaction effects (slowly compounding delays), and
(3) Regime shifts (different weekday versus weekend patterns).

LSTM RNNs learn non-linear mapping through the use of non-linear activation functions and multiple layers. In addition, LSTM RNNs can learn non-linear mappings directly from the data. Traffic patterns depend on several time scales at once. The morning traffic on Monday is influenced by Monday morning traffic before it, but also by Monday traffic the previous week. The patterns here are similar, and are a type of weekly seasonality captured by LSTM. Specifically, the memory cell architecture of LSTM's captures both short term autocorrelations and long-term seasonal patterns. Unlike ARIMA, which needs particular parameters and checks for stationarity, LSTMs require few assumptions about the data generating process. LSTMs can discover patterns in the data that are difficult for humans to recognize, but they are not interpretable. The moderate performance of GARCH and Prophet provides nuanced insights: GARCH's focus on volatility modeling rather than mean prediction explains its intermediate accuracy, though its ability to model time-varying uncertainty represents a valuable capability for future hybrid approaches combining LSTM's accurate mean predictions with GARCH's uncertainty quantification. Prophet's performance reflects a trade-off between ease of use and predictive power, with its structured decomposition approach imposing stronger assumptions about time series structure than LSTM's flexible learning, enhancing interpretability and reducing overfitting risk but constraining the model's ability to capture patterns not fitting its additive structure [18].

Limitations and Challenges: The analysis was undertaken with respect to a particular geographic region having specific infrastructure, demographic, and behavioural characteristics, which limits generalizability since traffic patterns vary significantly across different contexts (developing versus developed nations, urban versus rural areas, regions having different transport modes). Traffic Forecasting Forecast Models and Applications Summary: The research application of machine learning to complex real-world problems such as traffic forecasting can reflect the lack of predictive power and operational utility. In many cases it leads to first-hand experience in the qualitative evaluation of the results. Application-specific evaluation frameworks based on the characteristics of the desired forecasts as well as operational requirements offer a robust evaluation whereas standard evaluation metrics offer limited utility. Read more in this paper. The models were trained and assessed using historical data, not accounting for real-time performance during deployment in the real world. However, production environments come with their own challenges, which include computational latency constraints, incorporation of online learning as patterns evolve, sensor failures and missing data, and integration with existing systems.

None of the models explicitly incorporated exogenous variables influencing traffic weather conditions, special events, road construction, fuel prices, economic indicators, or public health measures which could substantially improve all models' performance and potentially shift comparative rankings if some models prove better at incorporating exogenous predictors. The infinite MAPE values for LSTM and GARCH prevented complete performance comparison across all metrics due to zero traffic counts in the dataset, possibly representing measurement errors or periods when roads were closed, indicating need for more careful data cleaning or alternative metrics robust to zero values.

© IJARCCE This work is licensed under a Creative Commons Attribution 4.0 International License

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⪾ Peer-reviewed & Refereed journal ⪾ Vol. 14, Issue 11, November 2025
DOI: 10.17148/IJARCCE.2025.141194

VI. CONCLUSION

This comprehensive comparative analysis assessed the performance of four prominent time series forecasting methodologies, ARIMA, GARCH, Prophet, and LSTM, for the urban traffic flow prediction using real-world data from the UK Department for Transport and provided clear evidence that deep learning approaches excel. The LSTM model showed the best predictive performance with an MAE of 150.21 and RMSE of 324.37, outperforming all others by far in capturing the complex nonlinear patterns, multiscale temporal dependencies, and dynamic behavior of urban traffic flow through direct learning from the sequential data without any need to specify explicitly temporal relationships or stationarity transformations. The GARCH and Prophet models were able to yield only a moderate level of accuracy, with GARCH providing uncertainty quantification and Prophet allowing for some practical advantages, including ease of deployment and interpretability. On the other hand, ARIMA suffered from severe performance degradation, revealing errors several orders of magnitude larger, indicating some limitations of the traditional linear statistical methods for solving complex real-world problems. In practice, transportation agencies should emphasize LSTM-based approaches when the highest level of accuracy is required, Prophet may be recommended for resource-constrained environments, GARCH can be used to estimate uncertainty, while ARIMA should be generally avoided unless radically improved. It adds to the growing evidence that supports the use of deep learning in transportation applications, while their successful deployment also depends on data infrastructure, analytical capabilities, procedures for model maintenance, and institutional trust beyond model accuracy. Future research has to be directed to overcoming the identified limitations: incorporating spatial dependencies, integrating exogenous variables, developing ensemble approaches, validating performance across diverse contexts, and moving towards causal modeling for proactive traffic management as the urban population continues to grow and pressure mounts on the transportation networks, demanding new and sophisticated forecasting tools capable of improving mobility and reducing congestion in order to make life better across the world.

REFERENCES

[1]. S. Siami-Namini, N. Tavakoli, and A. Namin, "A Comparison of ARIMA and LSTM in Forecasting Time Series," in Proc. 17th IEEE Int. Conf. Machine Learning and Applications (ICMLA), Orlando, FL, USA, Dec. 2018, pp. 1394-1401.

[2]. P. Kumar and K. Hariharan, "Time Series Traffic Flow Prediction with Hyper-Parameter Optimized ARIMA Models for Intelligent Transportation System," J. Scientific & Industrial Research, vol. 81, no. 4, pp. 408-415, Apr. 2022.

[3]. B. Liu, Y. Shen, and X. Zhu, "Traffic Flow Combination Forecasting Method Based on Improved LSTM and ARIMA," Int. J. Embedded Systems, vol. 12, no. 1, pp. 22-30, Feb. 2020.

[4]. S. Mehrmolaei and M. Keyvanpour, "Time series forecasting using improved ARIMA," in Proc. Artificial Intelligence and Robotics (IRANOPEN), Qazvin, Iran, Apr. 2016, pp. 92-97.

[5]. B. Dissanayake, O. Hemachandra, N. Lakshitha, and D. Haputhanthri, "A Comparison of ARIMAX, VAR and LSTM on Multivariate Short-Term Traffic Volume Forecasting," in Proc. 28th Conf. Open Innovations Association (FRUCT), Moscow, Russia, Nov. 2020, pp. 564-570.

[6]. Q. Ma, "Comparison of ARIMA, ANN and LSTM for Stock Price Prediction," E3S Web of Conferences, vol. 218, article 01026, 2020.

[7]. E. Ercanoglu, "Analysis and Comparison of Long Short-Term Memory Networks for Short-Term Traffic Prediction Performance," Scientific J. Silesian University of Technology. Series Transport, vol. 107, pp. 33-44, 2020.

[8]. Z. Uzel and E. Congeduti, "Comparative Analysis of LSTM, ARIMA, and Facebook's Prophet for Traffic Forecasting: Advancements, Challenges, and Limitations," Bachelor's Thesis, Faculty of Electrical Engineering, Mathematics nd Computer Science, Delft University of Technology, Delft, Netherlands, 2023.

[9]. Y. Zhang, S. Wang, B. Chen, and J. Cao, "Traffic Prediction Using Multigraph Convolutional Network and Recurrent Neural Network," J. Advanced Transportation, vol. 2020, article 8839839, 2020.

[10]. J. Wang, W. Deng, and Y. Guo, "New Bayesian Combination Method for Short-Term Traffic Flow Forecasting," Transportation Research Part C: Emerging Technologies, vol. 43, pt. 1, pp. 79-94, Jun. 2014.

[11]. Y. Li, X. Zhu, and D. Sun, "A Hybrid Method of Traffic Forecasting Based on Spatial Temporal Features," IEEE Access, vol. 9, pp. 108403-108414, 2021.

[12]. B. Chen, H. Zhao, Y. Zhang, and S. Wang, "Temporal Fusion Transformer for Multi-Horizon Time Series Forecasting in Traffic Flow Prediction," in Proc. IEEE Int. Conf. Big Data (Big Data), Atlanta, GA, USA, Dec. 2020, pp. 1548-1556.

[13]. R. F. Engle, "GARCH 101: An Introduction to the Use of ARCH/GARCH Models in Applied Econometrics," J. Economic Perspectives, vol. 15, no. 4, pp. 157-168, Fall 2001.

[14]. S. J. Taylor and B. Letham, "Forecasting at Scale," The American Statistician, vol. 72, no. 1, pp. 37-45, 2018.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁙ Peer-reviewed & Refereed journal ⁙ Vol. 14, Issue 11, November 2025
DOI: 10.17148/IJARCCE.2025.141194

[15]. H. Bowne-Anderson, "Time Series Analysis Tutorial with Python," DataCamp Community, Jan. 2018. [Online]. Available: https://www.datacamp.com/tutorial/time-series-analysis-tutorial
[16]. J. VanderPlas, "Python Data Science Handbook: Working with Time Series," O'Reilly Media, 2016. [Online]. Available: https://jakevdp.github.io/PythonDataScienceHandbook/03.11-working-with-time-series.html
[17]. "Express Analytics: Everything You Need to Know About Time Series Analysis," Express Analytics, 2023. [Online]. Available: https://www.expressanalytics.com/blog/time-series-analysis/
[18]. "Statsmodels Documentation: ARIMA Model," Statsmodels Development Team, 2023. [Online]. Available: https://www.statsmodels.org/stable/generated/statsmodels.tsa.arima.model.ARIMA.html