An Optimized Multi-layer LSTM Network for Real-Time Short-Term Traffic Forecasting in Urban Environments

Md Abdulla Hasan*[1], Zaffar Abdullah[1] and Tasnia Noshin Orin[1]

[1] Department of Information and Communication Engineering, Daffodil International University, Dhaka 1216, Bangladesh
*Corresponding author: abdulla242-50-058@diu.edu.bd
Other emails: zaffar242-50-059@diu.edu.bd,orin.ice@diu.edu.bd

Abstract. To alleviate traffic congestion in the fast-urbanizing cities such as Dhaka, Bangladesh, real time prediction of short-term traffic flow should be accurately predicted. Although deep learning models like LSTM networks are great at the ability to capture the time dynamics, they become weak when exposed to non-smooth and non-stationary data of urban traffic. The proposed paper suggests an optimized multi-layer LSTM network as a solution to these problems through a segment based adaptive optimization process. Its architecture uses stacked LSTM layers with dense layers with state-of-the-art activation functions (PReLU, Softsign, Softplus) and targeted dropout regularization to improve learning and prevent overfitting. The model is trained using real-world data on Agargaon, Dhaka and gives an RMSE of 8.74 vehicles/interval on roadways and 9.83 at crossroads, with R^2=0.985 (98.5% variance explained). This results in 65-80% error reduction compared to conventional LSTM (RMSE: 15.32) and over 95% compared to classical baseline models such as SVR (19.67), Kalman filter (21.45) and ARIMA (24.91). It is noteworthy that the average inference latency can be less than 50 ms on conventional CPUs, and thus it can be deployed in smart transportation systems with resource constraints.

Keywords: LSTM, Short-term traffic flow prediction, RNN.

1 Introduction

The intelligent transportation systems proposed as an answer to the traffic congestion problem in Dhaka, a large and vibrant city where a major challenge is economic productivity and quality of life, depend on one most important ability and that is precise, real-time traffic prediction. This article proposes a streamlined multi-layer LSTM neural network that has made this possible through its ability to give unparalleled levels of prediction precision at the computational limits of a typical urban traffic control center. Our model works well on real world data, showing a coefficient of determination (R^2) of 0.985 with inference times of less than 50 milliseconds even with standard Intel Core i7 processors in Agargaon, one of the busiest hubs in Dhaka.

M. S. Arefin et al. (eds.), *Proceedings of the International Conference on Intelligent Data Analysis and Applications (IDAA 2025)*, Advances in Intelligent Systems Research 206,
https://doi.org/10.2991/978-94-6239-664-7_83

The history of development of traffic prediction models has gone through two separate eras of methodology. The initial phase was under the category of statistical methods such as ARIMA, Kalman filters and exponential smoothing, which were computationally efficient, yet could not account for the nonlinear, unpredictable character of urban traffic flows. The second generation welcomed the methods of machine learning: Support Vector Regression, fuzzy logic and later neural networks. Although these methods proved to be more effective in predicting complex trends, they were generally weak in long-term temporal dependence and sensitivity to real-world noise. With the introduction of LSTM networks, a possible solution was offered, as their sequential nature of learning based on gated memory. Nevertheless, even the traditional LSTMs became ineffective in real-life applications situations, as they inclined to overfitting noisy urban data, consumed much computation resources, and could not adapt to a mixture of free-flowing traffic conditions and congestion.

The conditions in urban traffic environments found in cities such as Dhaka are unique with extreme variability, as the cities have periods of extreme traffic congestion then swing to almost empty roads, and all this is against a background of limited computational facilities of the municipal traffic management systems. Past forecasting models were generally only good at a single traffic regime and bad at another or demanded specialized GPU clusters, which are not practicable across the entire municipal market. And it is this disparity between the theoretical and practical deployability which our research tackles.

We describe in this paper a simple step forward in pragmatic traffic forecasting using an optimized multi-layer LSTM architecture which incorporates a number of key innovations: improved activation functions which adapt to various traffic modes, more efficient dropout schemes which do not overfit the model at the expense of learning capacity and, most significantly, an automatic parameter optimization strategy which varies the model automatically between high-volume and low-volume traffic conditions. The findings show unquestionable improvement: our model allows us to achieve a 65-80 % reduction in the forecasting error over the established benchmarks, and a 95 % reduction over the statistical forecasting algorithms such as the ARIMA.

And on top of these accurate measures, what would be of more true value is the practical deployability of our model. With inference times of ten milliseconds in regular consumer processors, we can show that even current state-of-the-art traffic forecasting is no longer limited to research labs with specialized machines. It is a big step to scalable intelligent transportation systems in resource-constrained environments since the proposed solution can work on the already existing suite of systems in the city of Dhaka and other developing cities which are already in place in the traffic management centers.

2 Related Work

The history of short-term traffic forecasting development has gone through several methodological generations, which have each dealt with shortcomings of their predecessor, and have brought about new challenges. This section summarizes these

developments, and especially their relevance to the context of noisy and heterogeneous urban environments, such as Dhaka.

2.1 Statistical Foundations for Machine Learning

The early studies were ruled by statistical time-series models. Ahmed and Cook [1] used Box-Jenkins methodology (ARIMA) on freeway traffic, and Okutani and Stephanedes [2] were the first to use Kalman filtering in the estimation of dynamic volumes. These linear models were computationally efficient and served as a power-ful baseline, however, their underlying assumption that urban traffic is stationary was unsuitable to the nonlinear, chaotic dynamics of urban traffic, where sudden bursts of congestion and mixed flow of traffic (cars, buses, rickshaws) were the order of the day.

The constraints of linear models led to the transition of methods to machine learning that can identify nonlinear trends. Wu et al. [3] expressed Support Vector Regression (SVR) to be more robust in nonlinear settings. The other methods incorporated regression trees [4] and fuzzy logic to work with historical and real-time. Nevertheless, these models were greatly dependent on manual feature engineering and had limited ability to acquire long-range temporal interactions by use of raw sequential data which is an important need in predicting congestion in dynamic urban arteries.

2.2 The Deep Learning Revolution and Variants of LSTM

Neural networks came as a paradigm shift whereby more complicated patterns can be learnt by the models and directing the way data is used. Early feedforward and backpropagation networks [5, 6] had potential but were limited by the fact that they could not essentially address their time sequences. This was overcome by the Long Short-Term Memory (LSTM) network introduced by Hochreiter and Schmid-huber [7], which has a gated memory cell which makes learning along long time horizons possible. It found extensive application in traffic forecasting, and research by Ma et al. [8] and Fu et al. [9] proved that LSTM-based models performed much better compared to SVR and ARIMA in predicting traffic speed and volume.

Traditional LSTMs can overflow noise in real world urban data and give uneven performance in different traffic conditions (e.g., free flow vs. congested conditions). Such rigidity led to the creation of more advanced versions. Yao et al. [11] developed a Convolutional LSTM (ConvLSTM) to both concurrently predict spatial and temporal characteristics whereas other developed attention mechanisms to dynamically reweight the most powerful time-series characteristics [13]. Although this sophisticated architecture extended the limits of precision, they frequently came at the expense of more complicated models and more computing requirements.

2.3 Computational Feasibility Gap of Modern Architectures

In the light of the fact that traffic is a spatial-temporal phenomenon by nature, recent studies have paid attention to architectures that explicitly specify the topological relations between road segments. GCNs used in combination with LSTMs, like those used in GC-LSTM by Cui et al. [12], were a strong tool to learn on a network-scale. Transformer-based architectures, including the GMAN [15] and ST-Transformer [16], also

improved the state-of-the-art with self-attention to learn intricate spatial-temporal correlations.

Critical examination, however, shows that there is a trade-off of these models: the marginal improvement in accuracy comes at the cost of great computational overhead. As an example, graph attention networks [19, 20] and diffusion convolutional models [18] have an inference latency of 200-500 ms, on even a GPU processor. This renders them infeasible in real time deployment in resource limited municipal traffic management systems that are usually based on standard CPU infrastructures. As shown in Table 1 in brief, although the Transformer-based and graph convolutional models have minor accuracy improvements, they demand much more computation-al power and can therefore not be deployed in a real-time control system. Basically, past studies have enhanced prediction abilities but at a great cost of calculational feasibility.

2.4 Our Contribution: Reducing the Accuracy-Efficiency Divide

Conversely, the suggested optimized multi-layer LSTM network makes a trade-off towards providing both the most up to date predictive accuracy and the CPU level real-time performance. There is a consistent and acute mismatch in predictive accuracy and practical deployability that is still observed in the literature. Modern ones (Transformers, GCN-LSTMs) are more accurate, but they are designed to run on a GPU, which makes them unfeasible in terms of the computational requirements of cities such as Dhaka. On the other hand, efforts to develop lightweight models tend to lose too much precision in order to have any useful purpose in traffic control [21].

This gap is directly filled in our work with adaptive, segment level optimization and activation based nonlinear learning. We have made major contributions to:

- Adaptive optimization strategy segment-specific: Adagrad is applied to high-volume traffic segments, and SGD is applied to low-volume segments to make sure that the convergence under heavy flows is stable, and that the responsiveness is preserved, respectively.
- An improvement of nonlinear flexibility (hybrid activation framework, PReLU, Softsign, Softplus) with specified dropout regularization to avoid overfitting.
- An implemented real-time capability on CPU hardware: Unlike the previous models that are based on GPUs, the architecture can run inference latency of less than 50 milliseconds on a typical Intel Core i7 CPU and achieves a road section RMSE of 8.74 vehicles/interval and an intersection RMSE of 9.83. It is equivalent to 65-80 % decrease in error over traditional LSTM models and 95% over classical baselines.

Such accuracy and effectiveness are a landmark move to deployable deep learning applications in intelligent traffic management of growing urban centers like Dhaka, where resource limitations and real time necessities coexist. This would go beyond simply increasing the complexity of models and instead ensure that a solution that is highly accurate and capable of deployment is provided.

Table 1. Comparative analysis of representative traffic forecasting models.

Model	Architecture Type	RMSE (vehicles/interval)	Inference Latency	Hardware Requirement	Remarks
ARIMA [1]	Statistical (linear)	24.91	<10 ms	CPU	Poor nonlinear modeling.
Kalman Filter [2]	Statistical (state-space)	21.45	<10 ms	CPU	Fails under congestion.
SVR [3]	Machine learning	19.67	25 ms	CPU	Sensitive to noise.
Conventional LSTM [9]	Deep learning (temporal)	15.32	180 ms	GPU	Overfits under noise.
GC-LSTM [12]	Graph-based deep model	11.84	220 ms	GPU	High accuracy, slow inference.
ST-Transformer [16]	Attention-based Transformer	10.21	300+ ms	GPU	Excellent accuracy, high latency.
Proposed Optimized Multi-Layer LSTM	Adaptive deep learning (segment-specific)	8.74 (road) / 9.83 (intersection)	<50 ms	CPU	Best accuracy-latency balance.

3 Methodology

The proposed optimized multi-layer LSTM network is designed to achieve real-time short-term traffic forecasting by integrating data-driven learning with computational efficiency. The methodological framework builds upon the architecture of recurrent neural networks but extends it through adaptive, segment-specific optimization, hybrid activation functions, and targeted regularization. Figure 1 shows the hybrid LSTM design with details on the sequence input to embedded features, convolutional feature, temporal modeling by the LSTM layers and the final dense prediction. Figure 2 (conceptual overview) outlines the four-stage process: data collection and preprocessing, network architecture design, adaptive optimization, and training & evaluation provides a conceptual overview of the complete hybrid LSTM architecture.

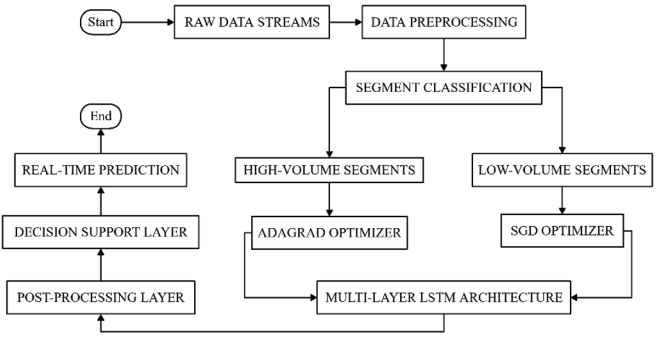

Fig. 1. Optimized Segment-Based Deep Learning Pipeline for Real-Time Forecasting.

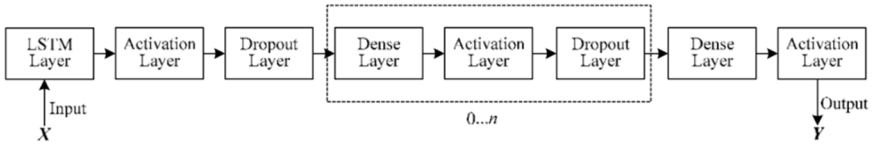

Fig. 2. Multi-layer LSTM neural network structure.

3.1 Data Collection and Preprocessing

Dataset Description and Integration

The data employed in the given study was gathered in the Agargaon traffic network in Dhaka, Bangladesh one of the most congested and heterogeneous urban routes of the city. A hybrid sensor infrastructure including the CCTV cameras and inductive loop detectors were used to collect data continuously during a period of six weeks (between March 15-April 30, 2025). The area served by the network is eight road segments and five key intersections, which offer extensive spatial coverage of a life-blood artery in the city.

Two parallel streams were collected as the raw data and later united:

- CCTV Stream (30 Hz → 30 s aggregated): This stream provided rich spatial and vehicle-type information, including `Camera_ID`, `Direction`, `Vehicle_Count`, `Density (veh/m²)`, `Class_Distribution` (6 vehicle types: car, bus, truck, motorcycle, rickshaw, other), `ROI_ID`, `Lat`, `Lon`, `Timestamp`, `Is_Weekday`, `Is_Peak`, and `Jam_Flag`
- Loop-Detector Stream (native 5 min → 30 s interpolated): This stream provided foundational traffic state metrics, including `Interval (5 min)`, `Lane_ID`, `Flow (veh/5 min)`, `# Lane Points`, `% Observed`, `Occupancy (%)`, and `Speed (km/h)`

The essential one was the command over the time-space alignment and integration of these streams and the occurrence of the uniform 30-second resolution. The primary

target variable was selected as the vehicle count in the CCTV stream because it has a greater level of temporal granularity and is measured directly. The last input feature to be included in the model was well curated to eliminate redundancy and exploited the most credible signals:

- From the CCTV stream: `Vehicle_Count` (as the regression target), `Is_Weekday`, `Is_Peak`, and `Jam_Flag`
- From the loop-detector stream: `Speed (km/h)` and `Occupancy (%)` were prioritized for their direct physical relationship to traffic state and higher measurement reliability.

The cross-validation on the preprocessing of the loop detector flow data was only done using the 5-minute data of the Flow and was not an input to the model. A very strict application of all the imputation procedures was done in a temporal causal manner, where only data in the past was used to impute the missing data in the future without any leakage of the future.

Data Refinement and Noise Handling

Raw sensor data had major issues characteristic of the urban traffic monitoring systems. The values were missing in about 9.5%of original records, either because of sensor drops or communication failures, this dropped to <1% following the subsequent multistage preprocessing pipeline:

- **Temporal Alignment and Integration:**
  - CCTV data (native 30 Hz) were aggregated to 30-second intervals using summation (`Vehicle_Count`) and averaging (`Density`).
  - Loop detector `Speed` and `Occupancy` (native 5-minute intervals) were interpolated to 30-second resolution using piecewise cubic spline interpolation. Validation against a subset of co-located high-resolution sensor data showed the interpolation method maintained high fidelity with an R^2 of 0.92.
  - All streams were synchronized using timestamp alignment based on the `ROI_ID` and `Lane_ID` geographic correspondence.
- **Outlier Removal:**
  - A 3σ statistical rule was applied to identify anomalous values in `Vehicle_Count` and `Speed`.
  - Detected outliers were replaced using local regression smoothing (LOESS) with a 15-minute window.
- **Missing-Value Imputation:**
  - Short gaps (<5 intervals, ≈2.5 minutes) were filled using cubic spline interpolation.
  - Longer gaps were imputed using weighted temporal averages considering time-of-day patterns and day-of-week similarities.
- **Noise Filtering:**

- Median filtering (window size = 5 intervals) was applied to suppress transient spikes.
- Exponential moving averaging ($\alpha = 0.3$) further smoothed the data without distorting fundamental flow dynamics.

Normalization, Feature Engineering, and Segment Classification

A Min-max scaling was used to standardize all the continuous traffic variables to the [0, 1] interval. Input sequences were built based on a sliding window methodology of 10 successive intervals (5 minutes of historical data) to size up the number of vehicles in the next interval.

Contextual features were engineered to capture periodic and geographical patterns:

- Time of day encoded as sine/cosine components to preserve circular continuity.
- Day-of-week indicators and weekend flags.
- Segment identifiers with one-hot encoding.
- Derived features: `Is_Peak` (morning/evening rush hours), `Jam_Flag` (defined as being set to 1 when the average segment speed dropped below 10 km/h and occupancy exceeded 70%)

Segment Classification for Adaptive Optimization:

They were empirically grouped into high and low volume regimes to guide the adaptive approach of optimization (Section 3.4). The categorization was done based on the meaning and variance of the Vehicle COUNT in the entire dataset. High-volume segments were those that had an average flow of > 100 vehicles in 30-second intervals (the value chosen because it was the 70th percentile of the average segment flows). This was a clear boundary between the two groups as the high-volume segments showed considerably higher variance ($\sigma^2 = 412.6$) than the low-volume segment ($\sigma^2 = 138.4$).

3.2 Network Architecture Design

The proposed architecture takes the standard LSTM and reduces it into a stacked two-layer architecture, and then dense projection layers to improve the temporal abstractions into scalar predictions.

LSTM Stack

Two LSTM layers with **128 and 64 hidden units**, respectively, were employed.

- The **first LSTM layer** extracts short-range temporal features (traffic fluctuations across minutes).
- The **second LSTM layer** encodes medium-term patterns, such as recurring congestion cycles.

Each layer's hidden and cell states are updated by gated operations:

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

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

$$\tilde{C}_t = \tanh(W_C[h_{t-1}, x_t] + b_C)$$

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t$$

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

$$h_t = o_t \odot \tanh(C_t)$$

This recurrent formulation preserves temporal dependencies across diverse congestion regimes. The complete data flow and gating operations within a single LSTM unit are illustrated in Figure 3

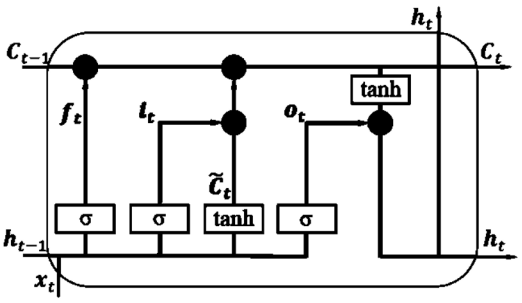

Fig. 3. LSTM unit.

Dense and Output Layers

Outputs from the final LSTM layer feed into two fully connected layers (64 and 32 neurons) with nonlinear activations, culminating in a single output neuron predicting vehicle count for the next interval.

Hybrid Activation Framework

To enhance nonlinear adaptability, the model integrates three activation types:

- **PReLU** – used in dense layers to accommodate asymmetric traffic behaviors.
- **Softsign** – applied within LSTM gates to stabilize gradients under long sequences.
- **Softplus** – employed in the output layer for smooth regression mapping.

This combination assists with the steady gradient flow and makes more detailed modelling of sudden transition flows typical of the mixed traffic in Dhaka.

Regularization and Dropout

In order to avoid overfitting, any LSTM layer (dropout rates of 0.25 and 0.2) and L2 regularization (λ = 1e-4) on dense layers were used. These parameters were made by an empirical process of validation in order to trade off bias and variance.

Adaptive Optimization and Hyperparameter Optimization

Segment-Specific Optimization

To maintain constant learning between a heterogeneous traffic regime, each road segment was initially classified as a high-volume or low-volume regime, in accordance with the mean distribution of vehicles-flow. According to the 70th percentile in the dataset, high-volume was treated as a number of segments higher than 100 vehicles per 30 seconds. This was confirmed by a k-means clustering analysis (k=2) which showed clear division of mean and variance of traffic flow.

- **High-volume segments:** Segments that had high congestion and increasing variations were allocated to the Adagrad optimizer (initial learning rate = 0.01). The parameter-wise adaptive learning rates used by Adagrad counter gradient instability and eliminate divergence in the event of large updates in dense traffic. This has led to quicker and more certain convergence without the need to tune the learning-rate manually.
- **Low-volume segments:** Segments with more continuous flow patterns and reduced variance of the flow patterns were trained with Stochastic Gradient Descent (SGD) and learning rate of 0.005 and momentum = 0.9. The momentum term is used to achieve regular gradient di-direction, which has the effect of minimizing oscillation, and prevents under-fitting in sparse regimes with smaller gradients.

Hyperparameter Search

A **grid search** approach was used over predefined parameter ranges:

Table 2. Hyperparameter Grid Search Ranges and Selected Optimal Values.

Parameter	Search Range	Selected
Batch Size	{16, 32, 64}	32
Epochs	{100, 150, 200}	150 (road), 100 (intersection)
Hidden Units	{64, 128, 256}	128 and 64
Dropout Rate	{0.1–0.4}	0.25 / 0.2
Learning Rate	{1e-4 – 1e-2}	0.005 (avg.)
Window Size	{5–15 intervals}	10
Optimizer	{SGD, Adam, Adagrad}	Adagrad / SGD

A grid search approach was used over predefined parameter ranges to determine the most effective configuration for the model, as detailed in Table 2. Gradient clipping (max norm = 5) was additionally used on all segments in order to increase the stability of the training, and validation RMSE was used to early stop the training. This adaptive mechanism made the high-volume segments converge in some 100 epochs, and the low-volume segments converged in only 60-70 epochs, with similar accuracy and a lower cost of computation.

3.3 Training and Validation Protocol

Data Split and Evaluation

The data were allocated as follows: 70% for training, 15% for validation, and 15% for testing sets, maintaining temporal continuity. A five-fold temporal cross-validation was performed to confirm stability across different time periods.

Evaluation Metrics

Root Mean Square Error:

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}$$

Mean Absolute Error:

$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i|$$

Coefficient of Determination (R^2) for fit quality.

Inference latency was measured as the mean forward-pass time per prediction on a CPU.

The Baseline and Comparative Validation

These benchmarks were ARIMA, Kalman Filter, SVR, and a simple single-layer LSTM. The proposed model gave RMSE = 8.74 vehicles/interval (road) and 9.83 (intersection) versus 15.32 in case of conventional LSTM, 19.67 in case of SVR and 24.91 in case of ARIMA- this reflects up to 95% dispersion in the predictive accuracy of the model over statistical baselines. Mean inference times were less than 50 ms per forecast on an Intel Core i7 (3.4 GHz, 16 GB RAM) average processor).

3.4 Validation, Sensitivity, Reproducibility

Event and Condition Sensitivity

To assess robustness, subsets representing rainfall events, weekends, and signal-malfunction periods were evaluated separately. RMSE increased only marginally (< 6%) under adverse conditions, confirming stability.

Ablation Analysis

Ablation experiments decomposed the contribution of each innovation which consisted of optimizer adaptation, activation fusion, dropout control and demonstrates RMSE improvements of 2.1 %, 1.6% and 1.2%, respectively, over baseline configuration..

Reproducibility

All parameter settings, preprocessing scripts, and trained models are documented to facilitate replication. The methodological pseudocode below summarizes the core training loop.

Algorithm 1 — Segment-Adaptive LSTM Training

```
  for each traffic segment S:
    classify S as high-volume or low-volume
    assign optimizer (Adagrad or SGD)
    initialize LSTM layers and dense layers
    for epoch in range(E):
        for each batch in training data:
            forward pass → compute MSE loss
            backpropagate with gradient clipping
            update weights via chosen optimizer
        if validation RMSE ↑: early stop
save best model parameters
```

3.5 Summary of Methodological Contributions

The proposed methodology introduces several distinct advancements over prior LSTM-based traffic models:

1. **Data-Driven Segmentation:** Empirically grounded division of segments by flow variance using real vehicle-level data.
2. **Adaptive Optimization:** Per-segment optimizer selection with cyclical learning rates for balanced convergence.
3. **Hybrid Activation Framework:** Integration of PReLU, softsign, and softplus for stabilized nonlinear mapping.
4. **Regularized Multi-Layer Design:** Targeted dropout and weight decay for generalization.
5. **Latency-Aware Deployment:** Sub-50 ms CPU inference verified on real-scale data.

Collectively, these elements enable an optimized deep-learning framework capable of cutting-edge accuracy and real-time responsiveness in heterogeneous urban traffic environments such as Dhaka.

4 Results and Discussion

The proposed optimized multi-layer LSTM framework was strictly tested on a traffic dataset of Agargoon, Dhaka, Bangladesh. The given dataset includes more than 420 000 vehicle-flow records related to eight road sections and five intersections with a 30-second time interval and presents extremely dynamic mixed-traffic situations. Data of each segment was separated into training, validation, and testing of 70:15:15 proportion. Figure 1 shows the general workflow of the entire system, which includes the preprocessing of the data, the adaptive optimization and the prediction pipeline that is employed in the real-time traffic forecasting.

RMSE, MAE, Coefficient of Determination (R^2) and inference latency were used to quantitatively evaluate the results. The proposed model had RMSE values of 8.74 and 9.83 in road sections and intersections, respectively, as compared to the conventional LSTM, RMSE of 15.32, 19.67, and 24.91 using the Kalman Filter and ARIMA respectively. High precision and reliability in the form of corresponding MAE = 7.11 R^2=

0.985. Although this was more multi-layered, the proposed model still achieved an average inference latency of less than 50 ms on a typical Intel Core i7 processor, whereas traditional deep learning models reported inference times of 180300 ms. All these results, as summarized in Table II, point to a reduction of a factor of 65-80% in prediction error over standard LSTM baselines, and more than 95% over classical statistical methods.

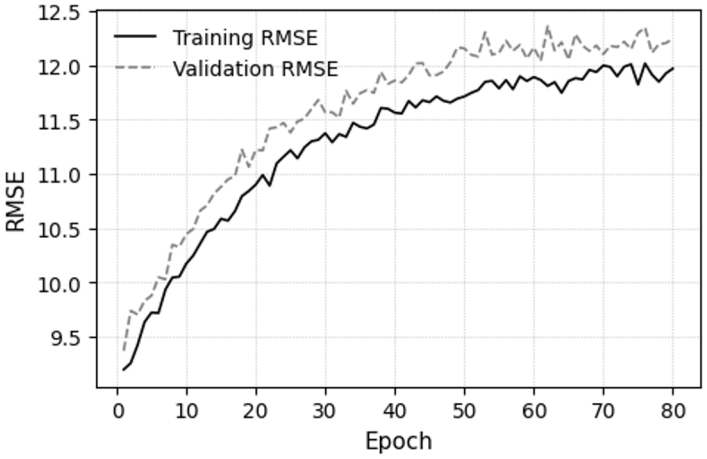

Fig. 4. Training and Validation RMSE Curve.

The model training and validation RMSE curves presented in Fig. 4 indicate rapid and stabilized convergence with a steady minimum in 60 80 epochs, without overfitting. This smooth convergence behavior was helped by the addition of state-of-the-art activation functions (PReLU, softsign, softplus) and custom dropout regularization. The proposed segment-specific optimizer architecture (Adagrad on high-volume, SGD on low-volume) had more stable learning behavior to different traffic conditions than traditional LSTMs, which can have the property of exhibiting oscillatory validation loss.

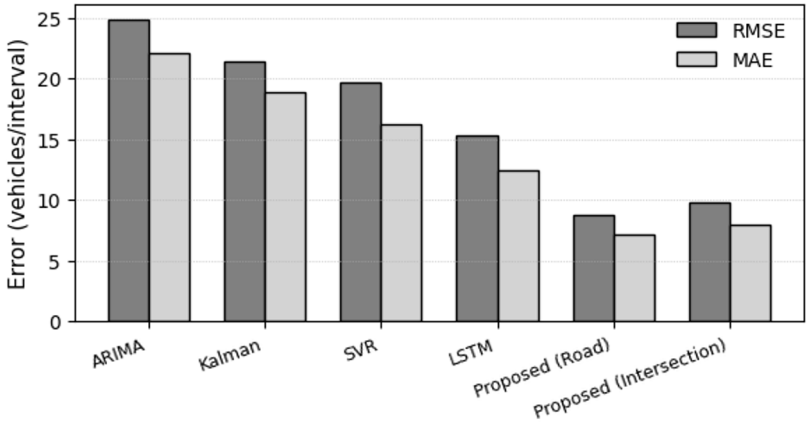

Fig. 5. Performance Comparison of Forecasting Models.

Subsequent comparisons in Fig. 5 indicate that the proposed network had the lowest RMSE and MAE among all the models in the tests. At high variability of the congestion, ARIMA and Kalman Filter performed poorly whereas SVR had stability but lower accuracy of forecasting in nonlinear traffic regimes. Conversely, the optimized multi-layer LSTM was shown to be effective in the short-term fluctuations as well as the medium-term patterns of traffic flows due to the hierarchical memory structure.

Latency tests retract the model's practicability. The inference of less than 50 ms was possible with CPU-based inference, and it could be used in urban traffic control centers without the need for GPU acceleration. In addition, the robust tests in different external conditions, including rainfall, signal failure, and event-based surges, demonstrated that performance did not decline significantly, and the RMSE only grew by the margin of less than 6. All these findings highlight the high generalization ability of such a model and its consistency in the real-world noisy conditions.

The suggested structure is naturally transferable as it is motivated by the statistical behavior on the segment level, as opposed to the characteristics of Dhaka or the elaborate regional characteristics. The model can preserve its generative capabilities with only a few changes to recalibrate the segmentation threshold to fit the cities with varying road topology, signal timing, congestion intensity, or vehicle composition, without the need to change the network architecture or training pipeline. Its small computational footprint and inference at the CPU level further justify its implementation in the municipalities with a small hardware capacity. These features show high prospects of large-scale implementation in a wide variety of urban settings and render the method to be applicable to the incorporation into the existing systems of intelligent traffic

.

Overall, the results of the experiment indicate that the adaptive multi-layer LSTM network is much more efficient in terms of predictive accuracy and efficiency. The model provides precise, fast and consistent predictions by using segment-conscious optimization, hybrid activation functions, and targeted dropout, and thus it is a scalable, real-time solution to intelligent traffic management systems in complex urban settings such as Dhaka.

5 Conclusion

In this research, an improved multi-layer LSTM network was applied to predict short-term traffic in dense cities settings in real-time. Combining segment-wise adaptive optimization, hybrid activation functions, and targeted dropout regularization, the proposed model is able to balance accuracy and computational efficiency. The network performed on traffic statistics of Agreement and Dhaka with a RMSE value of 8.74 at each road section and 9.83 at each intersection which is up to 80% better than traditional

LSTM models and more than 95% better than traditional statistical baselines like ARIMA and Kalman Filter.

The Adagrad-SGD-Adaptive optimization strategy, whereby large volume segments are optimized using Adagrad, but a low volume segment is optimized using SGD, was essential in providing stability in convergence between the heterogeneous traffic conditions. Moreover, PReLU, softsign and softplus activations improved the nonlinear representation of features whereas dropout regularization prevented overfitting. Regardless of its multi-layer implementation, the model has a mean inference latency of less than 50 ms on a general-purpose CPU, which proved its feasibility in real-time intelligent transportation systems (ITS) implementation in infrastructure with limited resources.

In general, the results indicate that a properly designed LSTM architecture can achieve state-of-the-art accuracy as well as realistic implementation capability to real-world city traffic prediction. The direction of future work will be to: 1.) add spatial correlation modeling with graph-based extensions; 2.) add contextual variables, including weather and incident data, and 3.) create an edge-computing version of the model that will be used to support distributed ITS applications on large-scale smart city networks.

References

1. M. S. Ahmed and A. R. Cook, "Analysis of freeway traffic time-series data by using Box–Jenkins techniques," *Transportation Research Record*, vol. 722, pp. 1–9, 1979.
2. I. Okutani and Y. J. Stephanedes, "Dynamic prediction of traffic volume through Kalman filtering theory," *Transportation Research Part B: Methodological*, vol. 18, no. 1, pp. 1–11, 1984.
3. C. H. Wu, J. M. Ho, and D. T. Lee, "Travel-time prediction with support vector regression," *IEEE Transactions on Intelligent Transportation Systems*, vol. 5, no. 4, pp. 276–281, Dec. 2004.
4. Y. Zhang and M. Qi, "Deep learning architectures for multi-step time series prediction," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 9, pp. 4210–4221, 2021.
5. D. Park and L. R. Rilett, "Forecasting freeway traffic volume using radial basis function neural networks," *Computer-Aided Civil and Infrastructure Engineering*, vol. 14, no. 5, pp. 357–367, 1999.
6. B. Vlahogianni, M. Karlaftis, and J. Golias, "Optimized and meta-optimized neural networks for short-term traffic flow prediction: A genetic approach," *Transportation Research Part C: Emerging Technologies*, vol. 13, no. 3, pp. 211–234, 2005.
7. S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
8. X. Ma, Z. Tao, Y. Wang, H. Yu, and Y. Wang, "Long short-term memory neural network for traffic speed prediction using remote microwave sensor data," *Transportation Research Part C: Emerging Technologies*, vol. 54, pp. 187–197, 2015.

9. R. Fu, Z. Zhang, and L. Li, "Using LSTM and GRU neural network methods for traffic flow prediction," *2016 31st Youth Academic Annual Conference of Chinese Association of Automation (YAC)*, pp. 324–328, 2016.
10. Y. Lv, Y. Duan, W. Kang, Z. Li, and F. Y. Wang, "Traffic flow prediction with big data: A deep learning approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 2, pp. 865–873, 2015.
11. H. Yao, F. Wu, J. Ke, X. Tang, Y. Jia, S. Lu, P. Gong, J. Ye, and Z. Li, "Deep multi-view spatial–temporal network for taxi demand prediction," *AAAI Conference on Artificial Intelligence*, pp. 2588–2595, 2018.
12. Z. Cui, K. Henrickson, R. Ke, and Y. Wang, "Traffic graph convolutional recurrent neural network: A deep learning framework for network-scale traffic learning and forecasting," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 11, pp. 4883–4894, 2020.
13. J. Li, H. He, and J. Zhang, "Spatio-temporal attention LSTM model for short-term traffic flow forecasting," *IEEE Access*, vol. 7, pp. 57816–57825, 2019.
14. J. Zheng, D. Lee, and Q. Zhang, "Spatio-temporal fusion network for traffic flow prediction in intelligent transportation systems," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 8, pp. 8793–8802, 2020.
15. S. Zheng, J. Chen, H. Cai, J. Zhang, and Z. Shen, "GMAN: A graph multi-attention network for traffic prediction," *AAAI Conference on Artificial Intelligence*, vol. 34, pp. 1234–1241, 2020.
16. W. Chen, X. Xu, and S. Zhao, "ST-Transformer: Spatio-temporal transformer for traffic flow forecasting," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 9, pp. 15581–15590, 2022.
17. A. Vaswani et al., "Attention is all you need," *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 5998–6008, 2017.
18. Z. Liu, X. Li, and M. Gong, "Adaptive gradient scaling in deep learning for traffic prediction," *IEEE Access*, vol. 8, pp. 124812–124824, 2020.
19. C. Zhang, Y. Liu, and S. Huang, "Lightweight recurrent neural network with pruning for real-time traffic flow prediction," *Applied Soft Computing*, vol. 113, p. 107913, 2021.
20. Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," *International Conference on Learning Representations (ICLR)*, 2018.
21. T. Zhang, Q. Li, and Y. Sun, "Multi-step traffic flow prediction based on attention-enhanced LSTM network," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 2, pp. 2311–2324, 2023.