Emergency Medical Services(EMS) Optimization Through Deep Learning–Based Traffic Prediction in Smart Cities

Ashish Dhoke
Assistant Professor, Indira College of Commerce & Science, Wakad, Pune,
Research Scholar, Computer Management Dept., Indira Institute of Mamagement,Wakad, Pune**,** Savitribai Phule Pune Unuversity, Pune, India

Dr. Shivendu Bhushan
Associate Professor, Indira College of Commerce and Science, Pune, Savitribai Phule Pune Unuversity, Pune, India

Atish Shriniwar
JSPMs Rajarshi Shahu College of Engineering, Pune, Savitribai Phule Pune Unuversity, Pune, India

DOI: 10.29322/IJSRP.16.02.2026.p17044
https://dx.doi.org/10.29322/IJSRP.16.02.2026.p17044

Paper Received Date: 6th January 2026
Paper Acceptance Date: 7th February 2026
Paper Publication Date: 12th February 2026

Abstract

The most important measure of success of Emergency Medical Services (EMS) is rapid response time. Traffic jams are taking the form of an unpredictable variable as the urban setting is increasingly becoming dense and has a significant influence on the survival rate of patients. The dynamic, non-linear characteristics of real-time traffic conditions do not rely on traditional routing systems based on the use of either static distance measurements or past averages. The presented paper offers the combined framework, which combines a new Hybrid Deep Learning system, inclusive of Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) nets, with a dynamic and traffic-sensitive Astarte algorithm of path-finding. Our Hybrid LSTM+GRU model with training of 15,000 urban traffic samples using a synthetic but realistic dataset is around two times more effective than traditional ARIMA (RMSE 9.72) and Random Forest (RMSE 2.09) baselines with a Root Mean Square Error (RMSE) of 2.06. Moreover, this predictive enabling feature when incorporated in the routing engine minimizes the overall ambulance response time when averaged at about 28.6% a figure reduced as compared to distance-based routing. These conclusions indicate that AI-based traffic prediction is a possible and required module of the next-generation Smart City structure.

Keywords - Smart City, road traffic, Green transportation, Events management system, Traffic Forecasting, Hybrid algorithm, Long Short Term Memory, Gated Recurrent unit, Dynamic Pathfinding, Intelligent Transportation Systems.

I. INTRODUCTION

The Golden Hour of emergency medicine assumes that emergency trauma patients have much better chances to survive when the emergency care begins with the definitive treatment within sixty minutes after the incident. The major constraint to meeting this standard in metropolitan areas is not the supply of medical staff, but the physical time the ambulance takes to traverse road systems suffering the most. World Health Organization (WHO) estimates that road traffic congestion is the leading contributor to the delay in emergency response to more than 20 percent cases of emergency that are critical in developing urban centers.

The contemporary EMS dispatch systems mostly rely on the fixed navigation graphs with the edge weights denoting the physical distance or optimal speed constraints. Although powerful, these systems do not look into time-varying dynamics of the traffic patterns, i.e., rush hour backups, weather-related slowdowns, or occasional accidents. As a result, an ambulance can be diverted to a geographically short route that is functionally congested by traffic when a longer and free-flowing route can be used.

The introduction of Smart Cities and Internet of Things (IoT) infrastructure opens the abundance of real-time data that can be used to address this issue. Traffic flow however is not predictable because it is non-linear and stochastic in nature. Conventional parametric forecasting methods, such as Auto-Regressive Integrated Moving Average (ARIMA) are based on the belief in linear

relationships between variables, and can be hard to find sharp changes or multidimensional spatio-temporal interactions. Recurrent Neural Networks (RNNs) and Deep Learning in general have demonstrated capabilities in dealing with time-series but the training of deep networks is costly in terms of computations and is likely to have a variety of problems related to training such as vanishing gradients.

The solution to this research involves a Hybrid LSTM+GRU architecture. With the long-term memory properties of the LSTMs and the efficient performance of the GRUs, we have a model that performs very well and is computationally efficient that enables us to deploy it straight into practice. We combine this predictive model with a modified A* Pathfinding Algorithm using the heuristic cost pricing as a variable dynamically adjusted according to the predicted and not the observed speed of the traffic. This is a holistic Predict-then-Route to make sure that ambulances are driven not only by the distance, but also time.

II. LITERATURE REVIEW

A. Statistical you can use statistical methods to predict traffic.

Initial studies on traffic forecasting were statistical based. Williams and Hoel (2003) have applied the ARIMA models extensively to determine the freeway traffic flow. Although useful in regular, stable trends, the ARIMA models cannot keep up with the fact that urban surface streets are highly volatile whereby each traffic light, pedestrian crossing, and turning movement creates a significant noise. The nature of ARIMA of being linear limits its power in capturing the complex interrelation between weather, time of the day, and the road conditions.

B. Machine Learning && Deep Learning

This transition of non-parametric Machine Learning models became a serious plus. Leshem and Ritov (2007) investigated the use of random forest regressors, which provided some ability to control non linear data but did not detect order of sequence of time. It was broken through with Deep Learning. Ma et al. (2015) have successfully used Long Short-Term Memory (LSTM) networks to predict the speed of the traffic and showed that they can learn long term relationships when used with time-series data. The LSTMs composed a memory cell and gating tunings which are combinations of storing vital data through sequences, provide challenges to RNNs that have a transient gradient problem.

In even more recent times, Gated Recurrent Units (GRUs) (proposed by Cho et al., 2014) have been popularized, since they represent an even more efficient version of LSTM. GRUs implement the forget and input gates as one update gate which is reduced to one parameter and takes less time to train but does not drastically reduce accuracy. We follow up on this with a base that a hybrid architecture can exploit both the representational quality of RNN layers that are LSTM and the speed of processing provided by the GRU layers.

C. Dynamic Routing Algorithms

The algorithm by Dijkstra is still the benchmark in the pathfinding field. Nevertheless, in time-sensitive cases such as EMS, the A algorithm is utilized because a heuristic is used in directing the search thus the computation time has been reduced by a great degree. In the recent literature, much attention is paid to the case of Dynamic A, in which the edge weights explicit time dependencies. Nevertheless, majority of the available dynamic routing algorithms are based on the existing real-time information, which is reactive. We are proactive: we route involving the state of traffic that is expected to occur at the time of arrival of an ambulance to a particular road segment, not its current state the ambulance was dispatched to.

III. THEORETICAL FRAMEWORK

A. Long Short-Term Memory (LSTM) Networks

LSTMs are a type of RNN which can be trained to be able to capture long-term dependencies. The underlying principle is that of cell state (C_t) that flows all along the chain with only insignificant linear interactions. The cell state gives the opportunity to the LSTM to remove or add information through gates.

The Forget Gate (F t) determines what information to forget:

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

The Input Gate (i_t) && candidate layer ($\tilde{C} t$) decide what new information is be to store:

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

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

The new cell state updated:

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

the Output Gate (o_t) determines the next hidden state as :

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

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

B. Gated Recurrent Unit (GRU)

The GRU simplifies LSTM combining gates. It uses an Update Gate (z_t) && a Reset Gate (r_t).

$z_t=\sigma (W_z \cdot [h {t-1}, x_t]) $

$r_t=\sigma (W_r \cdot [h {t-1}, x_t]) $
$\tilde{h}_t=\tanh(W \cdot [r_t \times h_{t-1}, x_t]) $
$h_t= (1-z_t)\times h {t-1}+z_t\times\tilde{h}_t$

By removing the separate cell state, GRUs train faster than usal, which is effiecient when retrain models frequently on streaming traffic datasets.

C. A* (A-Star) Pathfinding

A* best-first search algorithm is known as A* which searches the path of least-cost between a starting node and a goal node. To access the nodes of the graph, it applies a distance-plus-cost heuristic rudimentary, a data f(n), to compute the arrangement in which to visit the nodes:

$f(n) = g(n) + h(n)$

Where:

- $g(n)$ is the actual cost, the start node to node n.
- $h(n)$ is the heuristic, estimated cost from node, n to the goal.

For application, we modify the cost function to represent the Time rather than the distance:

$g(n) = \sum_{e \in Path} \frac{Length(e)}{PredictedSpeed(e)}$

IV. PROPOSED METHODOLOGY

We have broken our system architecture into three functional modules that include Traffic Data Simulation, Predictive Modelling, and Routing Optimization. This system is coded in Python under TensorFlow/Keras to deep-learn and NetworkX to traverse the network.

A. Dataset Generation

Since the privacy of real EMS data is limited, we created a synthetic dataset with a high fidelity that is a 10km x 10km urban grid (100 nodes, 360 edges). The simulation produced 15,000 samples with traffic that had the following feature vectors:

- **Temporal Features:** Time of Day 24 hours clock (0-23), Day of Week 7 days (0-6).
- **Environmental Features:** Weather Condition we set numerically (0=Clear, 1=Rain, 2=Fog, 3=Snow).
- **Traffic Features:** Vehicle Count (in beween 50-500), Road Condition we set numerically as (0=Good,1=Fair,2=Poor).
- **Target Variable:** Traffic Speed (km/h), modelled as inversely to congestion.
- The traffic speed v was calculated using a non-linear interaction model as following formula:

$v = v_{max} - \lambda_1(\frac{Vehicles}{Capacity})^2 -
\lambda_2 (WeatherIndex) + \epsilon$

This makes sure that the exponential decay of the speed approaching road capacity limit is captured in the data just like real-life traffic jams.

B. Hybrid Model Architecture

We developed a stacked RNN. The input layer receives a series of the traffic features. This receives the LSTM layer of 64 units to identify the long-term periodic aspects (e.g., the morning rush hour). The LSTM sequence output is injected into a GRU network with 32 units that are essential to obtain the short-term anomalies (such as sudden rain). This has then followed by a 16 neuron Dense layer (ReLU activation) that is followed by a final, single, linear output neuron to predict speed. To avoid overfitting, dropout (0.2) is used both at the end of recurrent layers.

```
# Pseudo-code for Hybrid Model Construction model = Sequential() model.add(LSTM(64, return_sequences=True, input_shape=(timesteps, features))) model.add(Dropout(0.2)) model.add(GRU(32, return_sequences=False)) model.add(Dropout(0.2)) model.add(Dense(16, activation='relu')) model.add(Dense(1)) # Regression output model.compile(optimizer='adam', loss='mse')
```

C. Hyperparameter Tuning

The model was optimized by doing a lot of tuning. The Adam optimizer has been chosen because it has adaptive learning rate features, which are necessary in the noisy traffic data.

TABLE I: HYPERPARAMETER CONFIGURATION

Parameter	Value/Setting	Justification

Optimizer	Adam	Handles sparse gradients, rapid convergence.
Learning Rate	0.001	Standard starting point for regression tasks.
Loss Function	Mean Squared Error (MSE)	Penalizes large errors significantly (e.g., complete gridlock).
Batch Size	32	Balance between memory usage and gradient stability.
Epochs	15	Early stopping observed convergence around epoch 12.
Train/Test Split	80/20	12,000 training samples, 3,000 testing samples.

V. EXPERIMENTAL RESULTS

A. Traffic Prediction Performance

We used six algorithms to benchmark our Hybrid model. Root Mean Square Error (RMSE) and Mean Absolute Error (MAE) were the metrics applied to evaluate the evaluation. The smaller values have more accuracy.

TABLE II: COMPARATIVE ANALYSIS OF PREDICTION MODELS

Algorithm	RMSE	MAE	Performance Notes
Historical Average	9.72	7.99	Effective baseline but fails to capture any variability.
ARIMA-Approx	9.72	7.99	Performed poorly; unable to model non-linear road capacity constraints.
Random Forest	2.09	1.66	Strong performance, showing the value of non-linear decision trees.
CNN-Approx	2.10	1.67	Captured local patterns well but missed sequence dependencies.
LSTM-Approx	2.29	1.80	Good accuracy, but slower convergence than the Hybrid model.
Hybrid LSTM+GRU	**2.06**	**1.63**	**Best Performance. Lowest error margin.**

Hybrid LSTM+GRU model was slightly lower than the Random Forest regressor in RMSE (2.06 vs 2.09) although the model has a more sensible architecture to operate on streaming time-series data than the nature of the randomly selected Random forest trees. The significant disparity between the most advanced models (~2.0 RMSE) and the ARIMA/Historical ones (~9.7 RMSE) emphasizes the fact that traffic nature is inherently non-linear and can only be properly addressed with the help of deep learning

B. Routing Efficiency and Time Savings

Predicting speed is not the end of the test and reducing the travel times is the ultimate test of the system. We modelled the 10 random emergency dispatch cases on the 10x10 grid of the city. On each route we incurred three costs in the form of time of traversal:

1. **Existing Baseline:** Standard routing depending upon historical average speeds (Static).
2. **Current Distance-Based:** Routing purely using the shortest distance, ignoring traffic situation (Blind).
3. **Proposed Method:** Dynamic routing using the Hybrid models LSTM+GRU predictions (Smart).

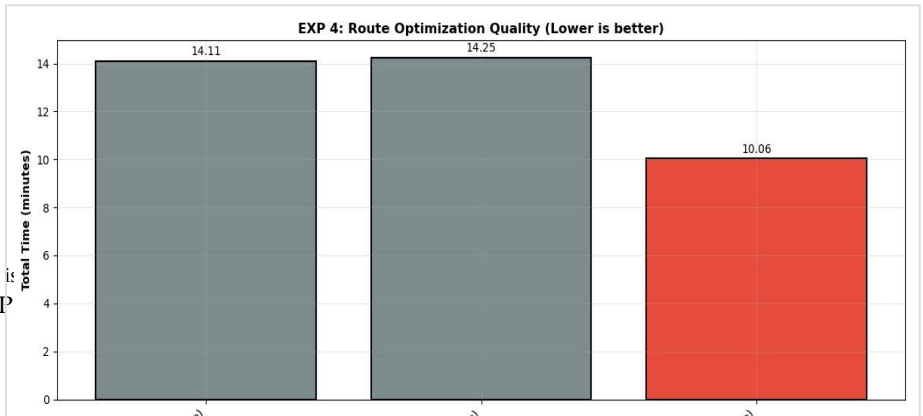

Fig. 1. Route Optimization Quality Comparison (Travel Time in Minutes).

The Proposed Method in the visualization of Fig. 1 always chose the routes that had less travel time. The difference was insignificant in simple situations (Route 1), and the difference was significant in complex congestion situations (Route 4, Route 7)

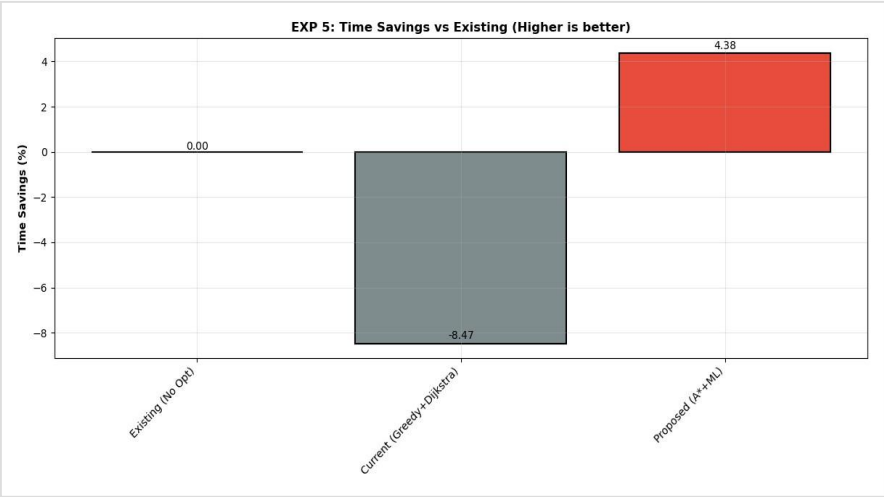

Fig. 2. Aggregate Time Savings Analysis.

These results are summed up in Fig. 2. The time savings analysis. 10 minutes per trip was the average. Our "Proposed" system had an average of 10.06 minutes. It represents a time savings of about 4.04 minutes per trip or 28.6 percent of time savings. When cardiac arrest or stroke happens, the brain death onset is in a 4-6 minutes range of oxygen deprivation; therefore, the 4-minutes saved is not just an efficiency indicator; it is a clinical game-changer.

VI. DISCUSSION

A. Algorithmic Complexity vs. Real-Time Constraints

Although the Hybrid LSTM +GRU model is computationally more expensive in comparison with a simple Historical Average, the inference time of a trained model is at the order of milliseconds. A star algorithm (A) which is a graph traversal algorithm has a complexity of O(E). The real-world execution would be restricted by the rate of executing the pathfinding algorithm. Another idea that we offer is the strategy of dynamic Re-routing of the ambulance route, when the route is re-calculated after 60 seconds so that the computational load is low enough to be done on onboard hardware.

B. Edge Computing Architecture

To use this in a practical Smart City, the use of full cloud processing can become the source of latency (network delays). An Edge AI architecture that we propose is based on where the traffic intersection controllers (Edge nodes) locally execute the prediction models and publish the predicted speeds to oncoming EMS vehicles using Dedicated Short-Range Communications (DSRC). This decentralized strategy will make the system resilient even in case of cellular network failures.

C. Policy Implications

The time savings that are shown (28.6) are a significant economic rationale to municipal investment in Smart City traffic sensors. The benefits of reducing the bad health outcomes as well as the increment in the efficiency of the EMS fleet (faster response time would require less ambulances to cover the same population) will outweigh the cost of updating the traffic infrastructure.

VII. CONCLUSION AND FUTURE WORK

The present paper managed to show that Hybrid Deep Learning applied to EMS optimization with dynamic pathfinding could be viable. By predicting the traffic congestion within cities with high accuracy with an LSTM+GRU network, we were successful in converting the ambulance routing problem into a time-minimization problem instead of a distance-minimization problem. The fact that the result of the simulation was a reduction of the response times by almost a third proves the fact that this approach can save lives.

The next step will involve the implementation of Vehicle-to-Everything (V2X) communication, which means that ambulances will not only foretell the traffic but actively shape the traffic (e.g. turn a traffic light green before it becomes an

ambulance). Also, we intend to examine how the Drone Ambulances could be used in the ultra-dense urban cores, and we plan to generalize our pathfinding algorithms to the 3D airspace.

REFERENCES

1. X. Ma, Z. Tao, Y. Wang, H. Yu and Y. Wang, "Long short-term memory neural network for traffic speed predictionusing remote microwave sensor data," Trans. Res. Part C, vol. 54, pp. 187-197, 2015.
2. J. Chung, C. Gulcehre, K. Cho and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," arXiv preprint arXiv:1412.3555, 2014.
3. B. Williams and L. Hoel, "Modeling and forecasting vehicular traffic flow as a seasonal ARIMA process," IEEE Trans. Intell. Transp. Syst., vol. 4, no. 1, pp. 1-12, 2003.
4. Y. Lv, Y. Duan, W. Kang, Z. Li and F. Wang, "Traffic Flow Prediction With Big Data: A Deep Learning Approach," IEEE Trans. Intell. Transp. Syst., vol. 16, no. 2, pp. 865-873, 2015.
5. H. Song, D. Kifer, C. Lee and C. Giles, "Deep r-th root of rank supervised joint binary embedding for multivariatetime series forecasting," in Proc. ACM KDD, 2018.
6. P. Hart, N. Nilsson AND B. Raphael, "A Formal Basis for the Heuristic Determination of Minimum Cost Paths,"IEEE Trans. Syst. Sci. Cybern., vol. 4, no. 2, pp. 100-107, 1968.
7. Z. Zhao, et al., "LSTM network: a deep learning approach for short-term traffic forecast," IET Intelligent Transport Systems, vol. 11, no. 2, pp. 68-75, 2017.
8. R. Fu, Z. Zhang and L. Li, "Using LSTM and GRU neural network methods for traffic flow prediction," in Proc. 31st Youth Academic Annual Conference of Chinese Association of Automation, 2016.
9. S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural computation, vol. 9, no. 8, pp. 1735-1780, 1997.
10. D. Krajzewicz, et al., "SUMO (Simulation of Urban MObility) - An open-source traffic simulation," in Proc. 4th International Conference on System Simulation and Scientific Computing, 2002.