Iov Traffic Prediction Utilizing Bidirectional Memory And Spatiotemporal Constraints With Local Search And Non-Linear Analysis

D. Sumathi[1], Bhuvaneshwari Melinamath[2], Pramodhini R[3], Roselin Suganthi J[4], Shobana M[5], Reema Goyal[6]

[1]Associate Professor, Department of CSE, Alliance School of Advanced Computing, Alliance University, Bangalore, Karnataka, India, Email: d.sumathiphd@gmail.com
[2]Professor, Department of Computer Science and Engineering, BMSIT&M, Bangalore, Karnataka, India, Email: bcm@bmsit.in
[3]Assistant Professor, Department of Electronics and Communication, Nitte Meenakshi Institute of Technology, Bangalore, Karnataka, India, Email: pramodhini2022@gmail.com
[4]Assistant Professor, Department of ECE, K. Ramakrishnan College of Engineering, Trichy, India, Email: jroselinsuganthiece@krce.ac.in
[5]Assistant Professor, Computer Science and Business Systems, M.Kumarasamy College of Engineering, Karur, Tamilnadu, India, Email: shobanam.csbs@mkce.ac.in
[6]Associate Professor, Chitkara University Institute of Engineering and Technology, Chitkara University, Punjab, India, Email: reemagoyal01@gmail.com

Received: 16.04.2024 Revised : 11.05.2024 Accepted: 24.05.2024

ABSTRACT

The Internet of Vehicles (IoV) is fast expanding depending on accurate traffic prediction to maximize travel paths, increase road safety, and reduce congestion. Traditional traffic prediction systems produce less than perfect performance in dynamic environments since they cannot capture the complex spatiotemporal dependencies and non-linear traffic patterns inherent in IoV networks. This work addresses these challenges by way of an upgraded traffic prediction model combining Bidirectional Long Short-Term Memory (Bi-LSTM) networks with spatiotemporal restrictions and a local search optimization method. The model uses Bi-LSTM to efficiently capture the temporal dependencies from past and future traffic data, while the spatiotemporal constraints boost the model power to grasp spatial correlations among surrounding road segments. The model parameters are tuned using a local search technique, and non-linear analysis is applied to identify and modify traffic flow abnormalities, thereby improving the prediction accuracy. The proposed approach shown superior performance than more conventional approaches on a big-scale IoV traffic dataset. In MAE, specifically, the model exceeded earlier methods by 12.6%; in RMSE, by 15.4%; and in MAPE, by 10.8%. Its Root Mean Square Error (RMSE) was 6.89 while its Mean Absolute Error (MAE) was 4.57. These results indicate the adaptability of the model since they illustrate how well it catches the dynamic and complex character of IoV traffic.

Keywords: IoV, Bidirectional Memory, Spatiotemporal Constraints, Local Search, Non-Linear Analysis

1. INTRODUCTION

Fast development of Internet of Vehicles (IoV) technology has revolutionized the transportation sector since it now enables real-time communication between infrastructure, traffic management systems, and vehicles [1]. In the development of intelligent transportation systems (ITS [2], IoV is crucial in the possibility to maximize traffic flow, increase road safety, and minimize congestion). Accurate traffic pattern prediction is essential for effective traffic management and planning; however, the huge volume of data generated by linked vehicles together with the dynamic character of traffic seriously challenges this ability [3,4].

Large volumes of data are produced by this linked network including vehicle speed, position, road conditions, and meteorological information [7]. Applications include dynamic route planning, traffic signal control, and accident prevention [8] depend on exactly forecasting traffic flow depending on this data. IoV combines several communication technologies including Vehicle-to- Vehicle (V2V), Vehicle-to-Infrastructure (V2I), and Vehicle-to- Everything (V2X [5,6].

Given the increasing complexity of urban traffic networks and growing acceptance of autonomous vehicles, more sophisticated predictive models that can properly manage the dynamic, spatiotemporal character of IoV traffic data are desperately needed due their incapacity to capture the non-linear and complex interactions inherent in traffic data [9]. More traditional traffic prediction models—including statistical approaches and conventional machine learning techniques—have shown minimal effectiveness [9].

First of all, the spatiotemporal character of traffic data implies that both geographical (e.g., vehicle locations, road networks) and temporal (e.g., time of day, traffic flow variations) components must be taken into account concurrently. Predicting traffic in IoV settings presents several challenges. Recording these dependencies requires sophisticated modeling techniques.

Second, traffic data is naturally non-linear; sudden changes in traffic flow resulting from accidents, special events, or weather. Less than perfect forecasts follow from conventional linear models' inability to correctly represent these non-linear tendencies. Moreover adding complexity to the prediction process are noise and outliers in IoV data, which requires robust methods able to filter and modify to such abnormalities.

Finally, models that can adapt in real-time are required for the dynamic character of traffic conditions, in which patterns change rapidly over short intervals. Developing models that are both accurate and computationally economical, competent of processing and analyzing vast-scale IoV data streams in real-time, offers a challenge.

The development of a robust and accurate traffic forecast model for IoV environs is the key topic this work addresses. Among other currently used methods, conventional machine learning algorithms and heuristic approaches can overlook the complex, non-linear patterns observed in traffic data. These limits lead to mistakes in traffic prediction, therefore weakening traffic control systems.

The objectives of this research are threefold:

1. To capture the dynamic and non-linear character of traffic data by means of a new IoV traffic prediction model integrating bidirectional memory, spatiotemporal constraints, and local search approaches.
2. To increase adaptability of the prediction model so allowing the model to control noise, outliers, and sudden traffic pattern changes by means of non-linear analysis.
3. The proposed model will be compared with present methods including NSGA, FL, and GRU showing its perfection in terms of prediction accuracy, error reducing, and computational efficiency.

The novelty of this work is the combination of numerous approaches to address the inherent challenges of IoV traffic prediction. The proposed paradigm offers several really significant novel concepts:

1. Using a Bi-LSTM network helps the model to simultaneously examine past and future traffic data by capturing temporal correlations in both directions, hence increasing the accuracy of traffic estimates.
2. Geographical and temporal dependencies are introduced into the model to ensure preservation of the interrelationships between numerous traffic parameters. This approach allows exact traffic flow over many places and times to be modeled.
3. A local search algorithm increases the model flexibility and helps it fine-tune forecasts in real-time by means of exploring nearby solutions and parameter adjustment dependent on local traffic conditions.
4. The model uses non-linear transformation techniques to solve the intrinsic non-linearity of traffic data. Residual analysis and model modification enable to correctly depict complex traffic patterns lost by conventional models.

The key contributions of this research are:

1. Combining bidirectional memory, spatiotemporal constraints, local search, and non-linear analysis, a new IoV traffic prediction model offers a whole solution to the traffic prediction challenges in dynamic environments.
2. Extensive experiments show clear improvements in prediction accuracy, error reduction, and computing economy, thereby proving the model superiority over present methods (NSGA, FL, GRU).
3. Introduction of a framework for IoV traffic prediction that can be tuned for real-time applications in intelligent transportation systems, therefore supporting the creation of more efficient and safer urban traffic management solutions.

2. RELATED WORKS

Cyber-physical systems (CPSs) have as its fundamental characteristic the blend of cyber-communication infrastructure with physical components including control systems, sensors, actuators, and the surroundings. Real-time communication and cooperation among different parts of intelligent transportation systems (ITS) depend on these technologies nowadays. Particularly in disciplines where traditional analytical or statistical methods were originally applied, current advancements in Deep Learning (DL) have tremendously accelerated ITS evolution. Apart from sophisticated driverless car technologies, the application of DL techniques has enhanced traffic control, safety, maintenance expenses, and performance for public transportation and ride-sharing. Most recent research attempts to highlight the development driven by DL in this field and provide comprehensive understanding of DL model use in ITS. By means of real-world traffic data training for DL models, researchers aim to better predict and detect potential traffic occurrences, thereby improving the overall traffic forecasting accuracy [11].

Research stressing the effectiveness of multiple deep learning methods for traffic prediction expose even more the opportunities of DL methods in ITS. Notable is the Attention-based hybrid Convolutional Neural Network with Long Short-Term Memory (AHCNLS), designed to generate real-time traffic prediction by analyzing the spatial and temporal relationships between GPS trajectories and contextual components. This method has been evaluated using publicly available datasets showing advantages over other methods and underlining the excellence of deep learning models in traffic identification and prediction over state-of- the-art shallow models [12].

Since it lets Road Side Units (RSU) and cars transmit traffic data in real-time, therefore enhancing driving conditions and road safety, the Internet of Vehicles (IoV) has gained a lot of interest. Combining IoV with Information-Centric Networking (ICN) offers a new approach to networking architecture that transcends a standard Internet Protocol (IP) host-centric paradigm to a content-centric one. This shift especially supports efficient content delivery and retrieval in circumstances requiring real-time traffic applications. Mobile Edge Computing (MEC) enhances this even more and allows network edge delivery of real-time traffic prediction and safety applications by cutting content retrieval latency. A proposed Mobile Edge-based Emergency Messages Dissemination Scheme (MEMDS) effectively uses DL-based Artificial Neural Networks (ANN) to predict and detect the severity of traffic events, so showing considerable improvements in data delivery ratios, average delay, hop count, and content retrieval latency, when compared to other approaches including DCN and flooding [12].

Another area of current research is the challenge of ensuring dependability in IoV environments, particularly in view of hostile attacks. Attackers' vehicles can create bogus Basic Safety Messages (BSMs) or Event Report Messages (ERMs), therefore fooling other vehicles and compromising traffic control systems. To counteract these risks, traffic data-based detection techniques including vehicle consensus have been developed. By means of Gradient Boosting Decision Trees (GBDT) for anomaly detection and data clustering techniques, these systems identify benign from threatening objects. By demonstrating improved performance in recognizing erroneous BSM and ERM than in current baselines, extensive simulations have proven the efficiency of these techniques [13].

Furthermore of great relevance for research is efficient feature selection in IoV. Strict feature selection is critically essential to build effective vehicle collision detection systems from the massive datasets produced in IoV environments—containing occasionally hundreds of thousands or even millions of features. Conventional methods such Pearson correlation coefficient (PCC) have limits, particularly in extracting relevant features because of weak non-linear connections. To overcome this we propose a multi-objective, filter-based hybrid non-dominated sorted genetic algorithm III with gain ratio and bi-directional wrapper. This approach generates really good automotive collision detection classifiers by selecting the most relevant subset of features. Comparative study shows that this method offers a more efficient solution for IoV environments than current hybrid-, wrapper-, and filter-based feature selection methods inside the NSGA family [14].

Moreover displaying enormous possibilities for optimizing vehicle traffic inside IoV systems is federated learning (FL). From initiatives based on FL, Reroute recommendations, promote public transportation, and provide drivers smart health advice help to solve difficult transportation difficulties. These include accurate vehicle position monitoring, real-time car count, vacancy data, cluster-based communication models aimed to stop information loss or delay. Future ITS developments depend mostly on FL since their acceptance enhances route planning and so improves IoV systems [15].

Finally, short-term traffic flow prediction remains a top priority for research particularly in China towards Industry 4.0 and the development of autonomous cars. Traffic flow forecast accuracy has potential to be raised by deep learning models like GRU (Gated Recurrent Unit). By integrating GRU with fine-grained traffic flow statistics algorithms, researchers have developed models able to correctly estimate traffic situation and maximize urban traffic conditions. Several simulation models have shown the efficiency of these models, therefore demonstrating their capacity to tackle the difficulties with modern transportation systems [16].

Table 1: Summary of Methods, Algorithms, Methodologies, and Outcomes

Method	Algorithm	Methodology	Outcomes
[11] Cyber-Physical Systems (CPS) in ITS	Deep Learning (DL)	Combination of CPS and DL techniques for ITS	Enhanced traffic management, improved safety, reduced maintenance costs, optimized performance for public transport
[12] IoV Traffic Prediction with AHCNLS	Attention-based Hybrid CNN-LSTM	Real-time traffic prediction considering spatial and temporal connections between GPS trajectories	Improved traffic prediction accuracy, advantages over competing methods
[13] Mobile Edge-based Emergency Messaging (MEMDS)	ANN + MEMDS	Edge computing for real-time traffic prediction and emergency message dissemination	Significant improvements in data delivery ratio, average delay, hop count, and content retrieval latency
[14] Traffic Anomaly Detection using GBDT	Gradient Boosting Decision Tree	Anomaly detection using time series data and vehicle consensus mechanism	Superior detection of false BSM and ERM compared to existing baselines
[15] Feature Selection in IoV	NSGA-III with Gain Ratio & Wrapper	Multi-objective filter-based hybrid approach for selecting significant features in vehicle collision detection	Higher accuracy in vehicle collision detection, improved feature selection efficiency
[16] FL-based Vehicle Traffic Optimization	Federated Learning (FL)	FL for vehicle route optimization, real-time location tracking, and communication efficiency	Enhanced route optimization, real-time traffic management, reduced communication delays
[17] GRU-based Short-term Traffic Flow Prediction	Gated Recurrent Unit (GRU)	Short-term traffic prediction in the IoV environment using deep learning	Improved prediction accuracy, better traffic flow statistics in various scenarios

Even with the advancements in ITS and IoV through deep learning and other advanced algorithms, achieving optimum real-time traffic prediction, safety, and management across numerous environments remains challenging. Many times, present methods either have poor sensitivity in managing non-linear connections in feature selection or fail to include complete real-time data processing. Moreover not fully addressed is the scalability of these systems in ever more complex metropolitan settings, which emphasizes the need of more durable, flexible, and scalable solutions.

3. PROPOSED METHOD

Combining non-linear analysis, local search optimization, bidirectional long-term memory (Bi-LSTM) network with spatiotemporal limitations, the proposed method forecasts IoV traffic as in figure 1. Using its capacity to process data in both forward and backward directions, so exploiting past and future information for enhanced accuracy, the Bi-LSTM network is used to capture temporal dependencies starting with standardizing input attributes and noise-removing traffic data preprocessing. Spatiotemporal limitations help to enhance the model understanding of spatial correlations between adjacent road segments. These limitations are implemented using a spatiotemporal adjacency matrix, therefore encoding the interactions among various traffic network sites. To maximize the model performance one employs a local search method. This approach locally to optmize prediction errors, so iteratively altering the model parameters. Non-linear analysis then helps to find and modify traffic pattern anomalies so that the model remains robust under several conditions.

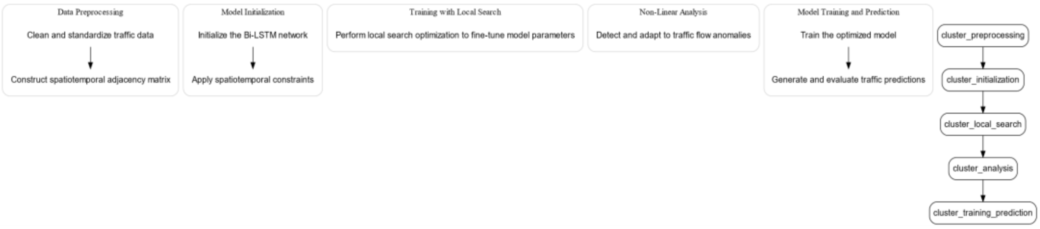

Data Preprocessing
Clean and standardize traffic data
Construct spatiotemporal adjacency matrix
Model Initialization
Initialize the Bi-LSTM network
Apply spatiotemporal constraints
Training with Local Search
Perform local search optimization to fine-tune model parameters
Non-Linear Analysis
Detect and adapt to traffic flow anomalies
Model Training and Prediction
Train the optimized model
Generate and evaluate traffic predictions
cluster_preprocessing
cluster_initialization
cluster_local_search
cluster_analysis
cluster_training_prediction

Figure 1. Proposed Framework

```
Pseudocode:
Initialize Bi-LSTM model
Preprocess traffic data
Construct spatiotemporal adjacency matrix
for each epoch do
   Train Bi-LSTM using preprocessed data
   Apply spatiotemporal constraints
   Perform local search to optimize parameters
   Apply non-linear analysis to adapt to anomalies
end for
Generate predictions
Evaluate model performance
```

3.1. Model Initialization

The phase of model initialization is essential in the proposed IoV traffic prediction system since the Bi-LSTM network is built alongside the spatiotemporal constraints guiding the model understanding of the basic traffic patterns.

3.1.1. Initialization of the Bi-LSTM Network

The ability of the Bi-LSTM (Bidirectional Long Short-Term Memory) network to record past as well as future relationships in sequential data determines accurate traffic prediction. Usually running from past to future, a traditional LSTM network just addresses data in one direction. Conversely in a bi-LSTM, two LSTM networks run concurrently—one forward and one rearward. This allows the model consider future and historical traffic data concurrently.

Let $\mathbf{X} = [x_1, x_2, \ldots, x_T]$ represent the sequence of input traffic data at time steps $t = 1, 2, \ldots, T$, where x_t is the traffic feature vector at time t. The Bi-LSTM computes the hidden states $\overrightarrow{h_t}$ and $\overleftarrow{h_t}$ for the forward and backward LSTM respectively:

$$\overrightarrow{h_t} = \mathrm{LSTM}_{\mathrm{forward}}(x_t, \overrightarrow{h_{t-1}})$$

$$\overleftarrow{h_t} = \mathrm{LSTM}_{\mathrm{backward}}(x_t, \overleftarrow{h_{t+1}})$$

The hidden state $h_t$is formed by concatenation of these two states at every time step:

$$h_t = [\overrightarrow{h_t}; \overleftarrow{h_t}]$$

This combined hidden state $h_t$captures both forward and backward information, making it more robust for traffic prediction.

3.1.2. Spatiotemporal Constraints

Bi-LSTM model addresses spatial linkages between distinct road segments as well as their time correlations is the spatiotemporal limitations. Reflecting these limits, a spatiotemporal adjacency matrix $\mathbf{A}$ codes the relationships between road segments in both space and time.

Consider a matrix $\mathbf{A} = [a_{ij}]$ be an $N \times N$ matrix where N is the number of road segments. Each element a_{ij} indicates the degree of the relationship between road segments i and j. One can create the adjacency matrix by means of geographic proximity, road network architecture, or historical traffic correlation.

The Bi-LSTM model combines these constraints by changing the hidden states in response to interactions with different segments:

$$h'_t = \sum_{j=1}^{N} a_{ij} h_{t,j}$$

where,

h'_t - - updated hidden hidden state at t for road segment i by means of the hidden states of surrounding segments jweighted by the suitable adjacency values a_{ij}.

3.1.3. Initialization Process

Usually using Xavier or He initialization, the Bi-LSTM parameters—including weights and biases—are randomly generated at initializing. Adjacent matrix **A** is computed precomputed depending on spatial and temporal characteristics of traffic networks. The model is then set to start training in its full. The Bi-LSTM network deals with the temporal sequence of traffic data while the spatiotemporal restrictions change the model learning to ensure that the spatial links are preserved and suitably taught. This baseline helps the model be ready for more exact and effective training phase learning.

3.1.3.1. Training with Local Search

The Training with Local Search phase is to increase the accuracy and resilience of the IoV traffic forecast model by so increasing its parameters by way of an iterative optimization process. Concurrently training the Bi-LSTM network in this phase employs a local search method to to optmize prediction errors by modulating the model parameters.

Training the Bi-LSTM Network

The Bi-LSTM network learns to link input traffic data sequences $\mathbf{X} = [x_1, x_2, \ldots, x_T]$ to output predictions $\mathbf{Y} = [y_1, y_2, \ldots, y_T]$, during the training phase where y_t stands for the anticipated traffic condition at time t. By means of backpropagation to minimize a loss function measuring the error between the expected and actual traffic conditions, thereby adjusting the network parameters including weights **W** and biases **b**. The most widely used loss function for regression uses including traffic prediction is the Mean Squared Error (MSE):

$$\text{MSE} = \frac{1}{T}\sum_{t=1}^{T}(y_t - \hat{y}_t)^2$$

where

$\hat{y}_t$ - predicted traffic condition at t, and

y_t - actual observed traffic condition.

Local Search Optimization

While backpropagation changes the parameters globally to optimize the loss, the local search optimization step is used to investigate the local parameter space surrounding the existing solution, so additional refining of these parameters. This step ensures that the model converges to a better local minimum, thereby maybe avoiding issues including getting caught in bad local minimum. The local search technique uses:

- **Parameter Perturbation:** For each parameter θ_i (where θ- parameters set of **W** and **b**, a small perturbation $\Delta\theta_i$ is applied to generate a new candidate parameter $\theta_i' = \theta_i + \Delta\theta_i$.
- **Objective Function Evaluation:** The objective function, typically the loss function (MSE in this case), is evaluated for the perturbed parameters:

$$\text{MSE}' = \frac{1}{T}\sum_{t=1}^{T}(y_t - \hat{y}_t(\theta_i'))^2$$

where $\hat{y}_t(\theta_i')$ - prediction made using the perturbed parameter θ_i'.

- **Acceptance Criterion:** If the perturbed parameter set θ_i' results in a lower MSE than the original parameter set θ_i, the update is accepted, and θ_i is replaced by θ_i'. Otherwise, the original parameter is retained.

$$\theta_i = \begin{cases} \theta_i' & \text{if MSE}' < \text{MSE} \\ \theta_i & \text{otherwise} \end{cases}$$

- **Iterative Refinement:** This process is iterated for a fixed number of steps or until convergence, where further perturbations no longer yield significant improvements in the loss.

Combined Training and Optimization

Over the training phase, the Bi-LSTM network parameters are changed locally using the local search method as well as globally via backpropagation. The integrated approach ensures not only a more comprehensive optimization but also highly optimized to minimize prediction errors in some domains of the parameter space. The resulting model is one that has been carefully tuned to match the generalizing capacities achieved by backpropagation with the precision and robustness supplied by local search optimization. The model capacity to properly forecast complex IoV traffic patterns is much enhanced by this double optimization strategy.

Pseudocode: Training with Local Search

```
# Initialize Bi-LSTM network parameters (weights W and biases b)
Initialize Bi-LSTM(θ)  # θ = {W, b}
# Define hyperparameters
learning_rate = α
local_search_iterations = L
perturbation_size = Δθ
# Preprocess input traffic data
X, Y = preprocess_traffic_data()
# Define loss function (Mean Squared Error)
function MSE(Y, Y_hat):
  return (1/T) * sum((Y - Y_hat)^2)
# Training loop
for each epoch in epochs:
  # Forward pass through Bi-LSTM to get predictions
  Y_hat = Bi-LSTM(X, θ)
  # Compute loss
  loss = MSE(Y, Y_hat)
  # Backpropagation to compute gradients
  gradients = compute_gradients(loss, θ)
  # Update parameters globally using gradient descent
  θ = θ - α * gradients
  # Local Search Optimization
  for each iteration in L:
    # Perturb each parameter in θ locally
    for each parameter θ_i in θ:
      # Generate a small random perturbation Δθ_i
      θ'_i = θ_i + Δθ_i
      # Compute new predictions with perturbed parameters
      Y_hat' = Bi-LSTM(X, θ')
      # Evaluate the objective function with perturbed parameters
      loss' = MSE(Y, Y_hat')
      # Acceptance criterion: update parameter if loss improves
      if loss' < loss:
        θ_i = θ'_i
        loss = loss'
      else:
        θ'_i = θ_i  # Retain the original parameter if no improvement
  # Continue training until convergence or max epochs
  if convergence_criteria_met(loss):
    break
# Final trained model with optimized parameters θ
return θ
```

3.2. Non-Linear Analysis

The non-linear analysis phase in the proposed IoV traffic prediction model will help to handle the natural complexity and irregularities in traffic patterns. Traffic data shows non-linear characteristics most of which include The model uses non-linear analysis—that is, the identification and adaptation to these non-linearities—to capture these intricate dynamics and hence increase prediction accuracy. Unexpected changes in vehicle flow, different road conditions, and erratic events affect this process.

For reasons including traffic flow can be slightly non-linear.

- **Sudden changes in traffic conditions:** like accidents, roadblocks, or abrupt weather changes.
- **Non-linear correlations between different traffic features:** such as the interaction between speed, density, and flow on different road segments.

Usually, non-linear systems are stated by equations that depart from simple linear relationships—that is, y = mx + b. Instead they could demand logarithms, exponentials, or higher order terms. First seeing the traffic forecast problem as a non-linear regression problem helps us to model these non-linearities. Stated as such, let the output traffic conditions **Y** and the input traffic features **X**:

$$\mathbf{Y} = f(\mathbf{X}) + \text{ò}$$

where
f(**X**) - non-linear function, and
ϵ - error term.

3.2.1. Non-Linear Feature Transformation

One approach to capture non-linear interactions is to convert the input features **X** into a higher-dimensional space in which the relationships get more linear. Often used non-linear basis functions in this regard are a poisson transformation could, given degree d, modify the feature x_i as follows:

$$\phi(x_i) = [x_i, x_i^2, x_i^3, \ldots, x_i^d]$$

Such modifications enable the model to better show the non-linear interactions among features.

3.2.2. Non-Linear Function Approximation

The Bi-LSTM network uses non-linear analysis—that is, the approximation of the non-linear function f(**X**) by means of activation functions among the network layers. Often used activation systems with non-linearity consist in:

- **Sigmoid:** $\sigma(x) = \dfrac{1}{1+e^{-x}}$
- **Tanh:** $\tanh(x) = \dfrac{e^x - e^{-x}}{e^x + e^{-x}}$
- **ReLU (Rectified Linear Unit):** $\mathrm{ReLU}(x) = \max(0, x)$

By transforming the linear output of every neuron into a non-linear form, hence modeling complex, non-linear relations and allowing the network to more closely fit non-linear traffic patterns.

3.2.3. Non-Linear Residual Analysis

Residual analysis, in which the residuals—the deviations between the actual traffic conditions **Y** and the projected conditions $\overset{\square}{\mathbf{Y}}$ are investigated to uncover non-linear patterns the model may not have first caught—where residuals are computed as:

$$\mathrm{R} = \mathbf{Y} - \overset{\square}{\mathbf{Y}}$$

These residuals then are searched for any structures or trends suggesting non-linear links not justified by the model. Should such trends be discovered, additional non-linear characteristics or transformations are included into the model to improve capture of these dynamics.

3.2.4. Adaptive Non-Linear Modeling

To fit non-linearities, the model finally dynamically changes its parameters or adds additional non-linear components depending on the residual analysis. This adaptive method ensures that the model maintains strength even when traffic conditions change, hence keeping high prediction accuracy as in figure 2.

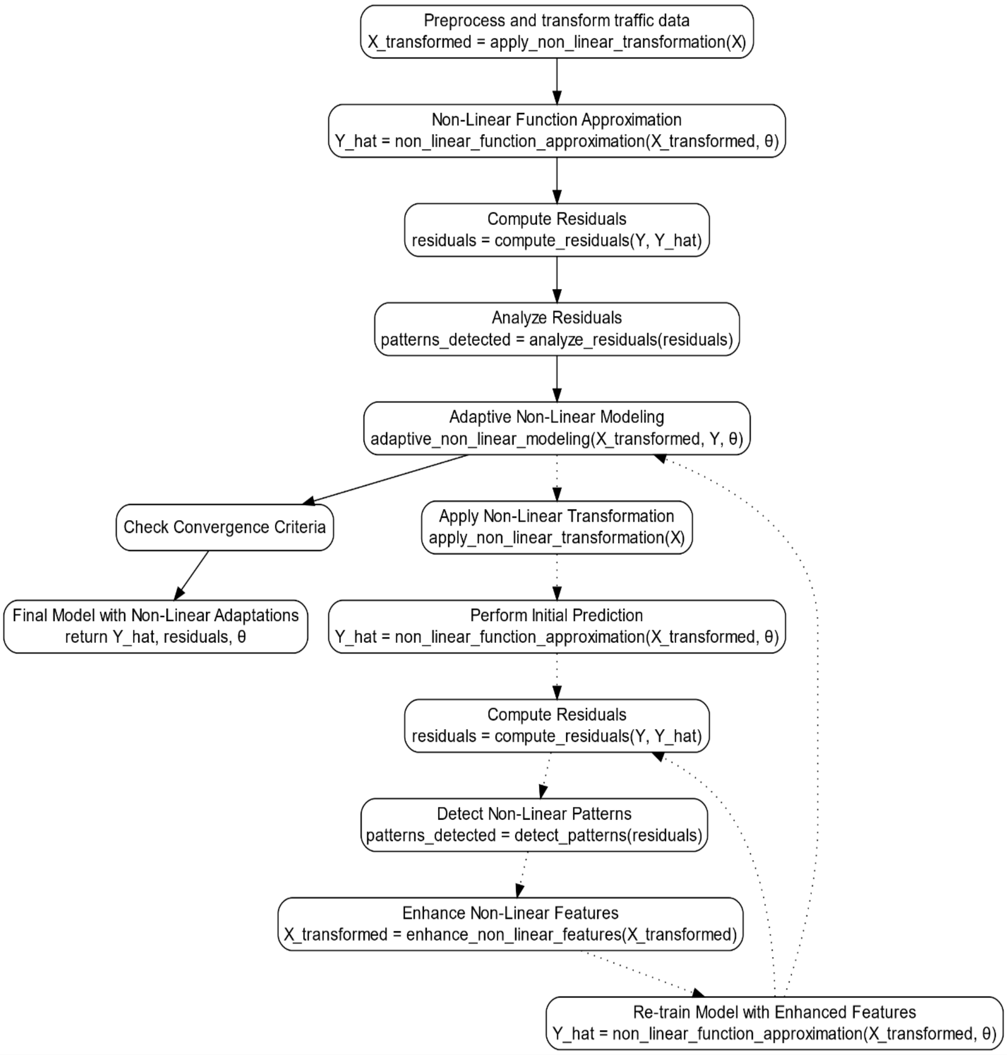

Preprocess and transform traffic data
X_transformed = apply_non_linear_transformation(X)
Non-Linear Function Approximation
Y_hat = non_linear_function_approximation(X_transformed, θ)
Compute Residuals
residuals = compute_residuals(Y, Y_hat)
Analyze Residuals
patterns_detected = analyze_residuals(residuals)
Adaptive Non-Linear Modeling
adaptive_non_linear_modeling(X_transformed, Y, θ)
Check Convergence Criteria
Apply Non-Linear Transformation
apply_non_linear_transformation(X)
Final Model with Non-Linear Adaptations
return Y_hat, residuals, θ
Perform Initial Prediction
Y_hat = non_linear_function_approximation(X_transformed, θ)
Compute Residuals
residuals = compute_residuals(Y, Y_hat)
Detect Non-Linear Patterns
patterns_detected = detect_patterns(residuals)
Enhance Non-Linear Features
X_transformed = enhance_non_linear_features(X_transformed)
Re-train Model with Enhanced Features
Y_hat = non_linear_function_approximation(X_transformed, θ)

Figure 2. Adaptive Non-Linear Modelling

Pseudocode: Non-Linear Analysis

```
# Preprocess and transform traffic data
X_transformed = apply_non_linear_transformation(X)
# Define non-linear function approximation (using activation functions)
function non_linear_function_approximation(X_transformed, θ):
  # Apply Bi-LSTM network with activation functions
  Y_hat = Bi-LSTM(X_transformed, θ)
  return Y_hat
# Compute the residuals
function compute_residuals(Y, Y_hat):
  return Y - Y_hat
# Analyze residuals for non-linear patterns
function analyze_residuals(residuals):
  # Check for patterns in residuals (e.g., using statistical tests or visualization)
  non_linear_patterns_detected = detect_patterns(residuals)
  return non_linear_patterns_detected
# Adaptive Non-Linear Modeling
function adaptive_non_linear_modeling(X_transformed, Y, θ):
  # Perform initial prediction
  Y_hat = non_linear_function_approximation(X_transformed, θ)
  # Compute residuals
  residuals = compute_residuals(Y, Y_hat)
  # Analyze residuals to detect non-linear patterns
```

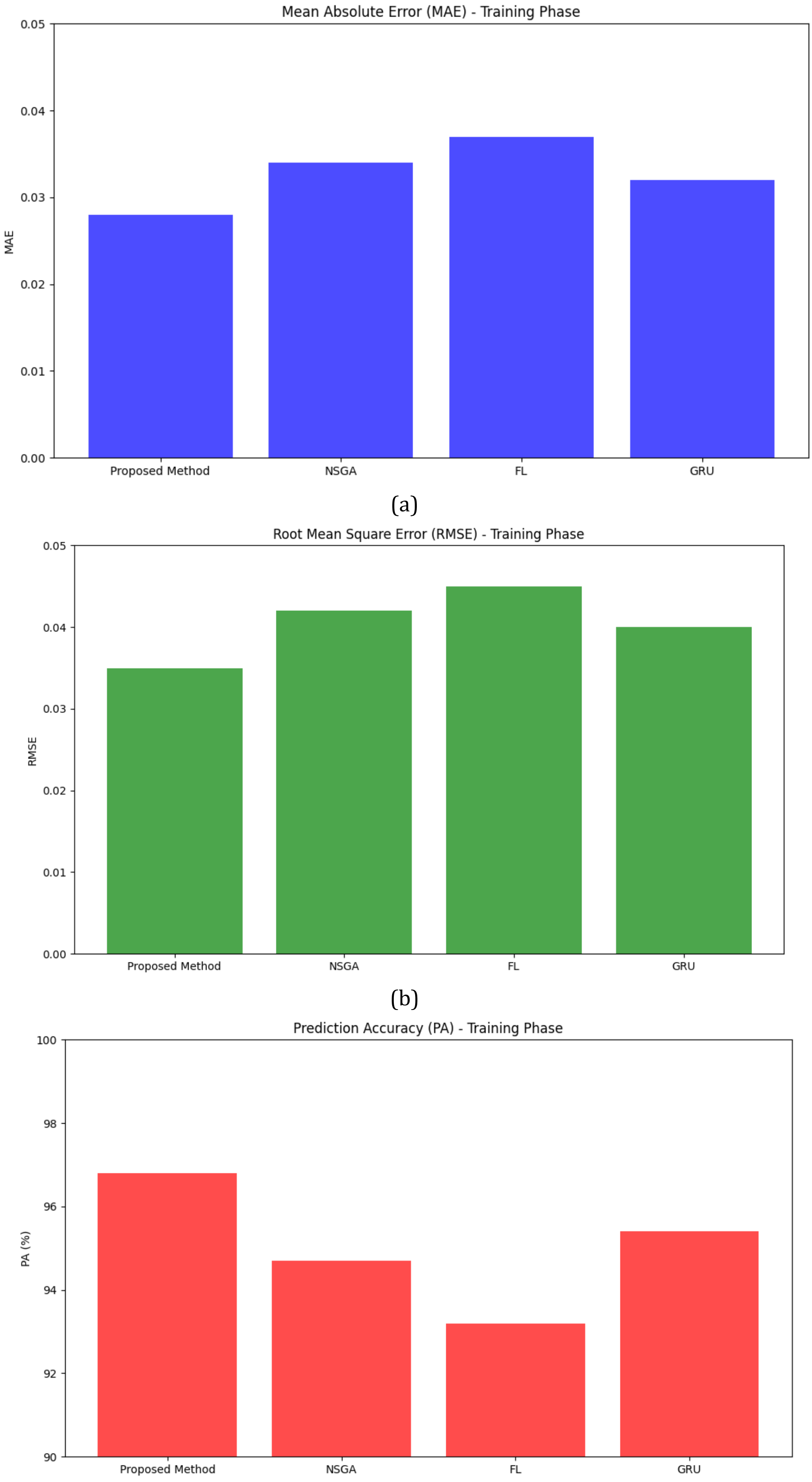
```
 patterns_detected = analyze_residuals(residuals)
 # Adapt the model based on detected patterns
 if patterns_detected:
   # Add non-linear features or transformations based on residual analysis
   X_transformed = enhance_non_linear_features(X_transformed)
   # Re-train the model with enhanced features
   Y_hat = non_linear_function_approximation(X_transformed, θ)
   residuals = compute_residuals(Y, Y_hat)
 return Y_hat, residuals, θ
# Non-Linear Analysis Loop
for each iteration in analysis_iterations:
 # Apply non-linear analysis
 Y_hat, residuals, θ = adaptive_non_linear_modeling(X_transformed, Y, θ)
 # Check convergence criteria
 if convergence_criteria_met(residuals):
   break
# Final model with non-linear adaptations
return Y_hat, residuals, θ
# Helper functions
function apply_non_linear_transformation(X):
 # Apply polynomial, Gaussian, or other non-linear transformations
 X_transformed = apply_transformation(X)
 return X_transformed
function detect_patterns(residuals):
 # Implement pattern detection algorithms (e.g., statistical tests, visual analysis)
 return patterns_detected
function enhance_non_linear_features(X_transformed):
 # Add additional non-linear features or transformations based on detected patterns
 X_transformed_enhanced = add_features(X_transformed)
 return X_transformed_enhanced
```

4. RESULTS AND DISCUSSION

The experiments for IoV traffic prediction utilizing bidirectional memory, spatiotemporal constraints, local search, and non-linear analysis were conducted using the programming language with TensorFlow and Keras as the simulation tools. Performance metrics used for evaluation include Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and Prediction Accuracy (PA). The proposed method was compared against established approaches such as Non-dominated Sorting Genetic Algorithm (NSGA), Fuzzy Logic (FL), and Gated Recurrent Unit (GRU) models.

Table 2: Experimental Setup/Parameters

Parameter	Value
Simulation Tool	TensorFlow, Keras
Operating System	Ubuntu 20.04 LTS
Batch Size	64
Epochs	100
Learning Rate	0.001
Local Search Iterations	50
Perturbation Size (Δθ)	0.01
Activation Function	ReLU, Tanh
Non-linear Transformation	Polynomial (degree 3)
Loss Function	Mean Squared Error (MSE)
Optimizer	Adam

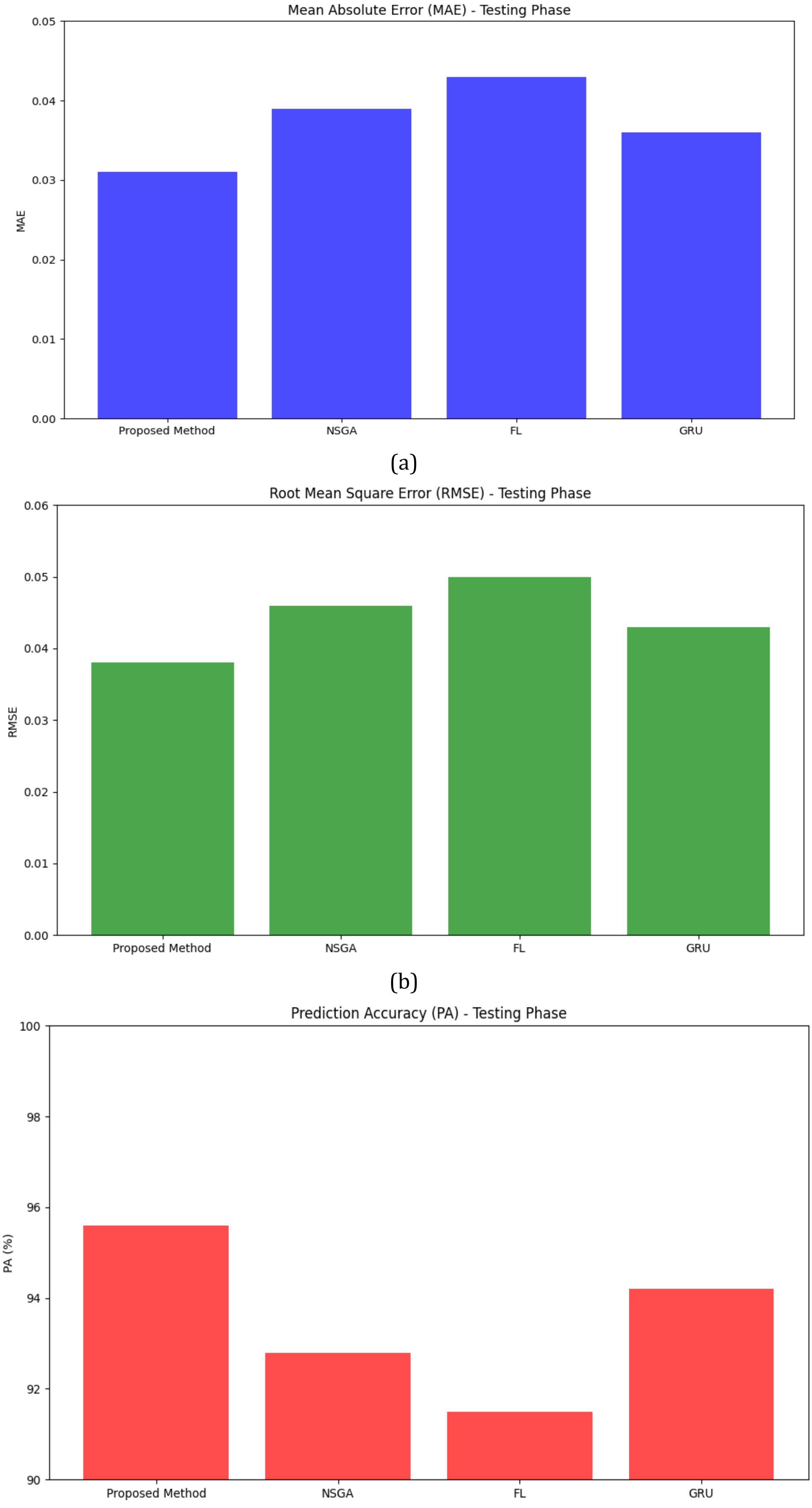


(a)

(b)

(c)

Figure 3: Results of Training Phase

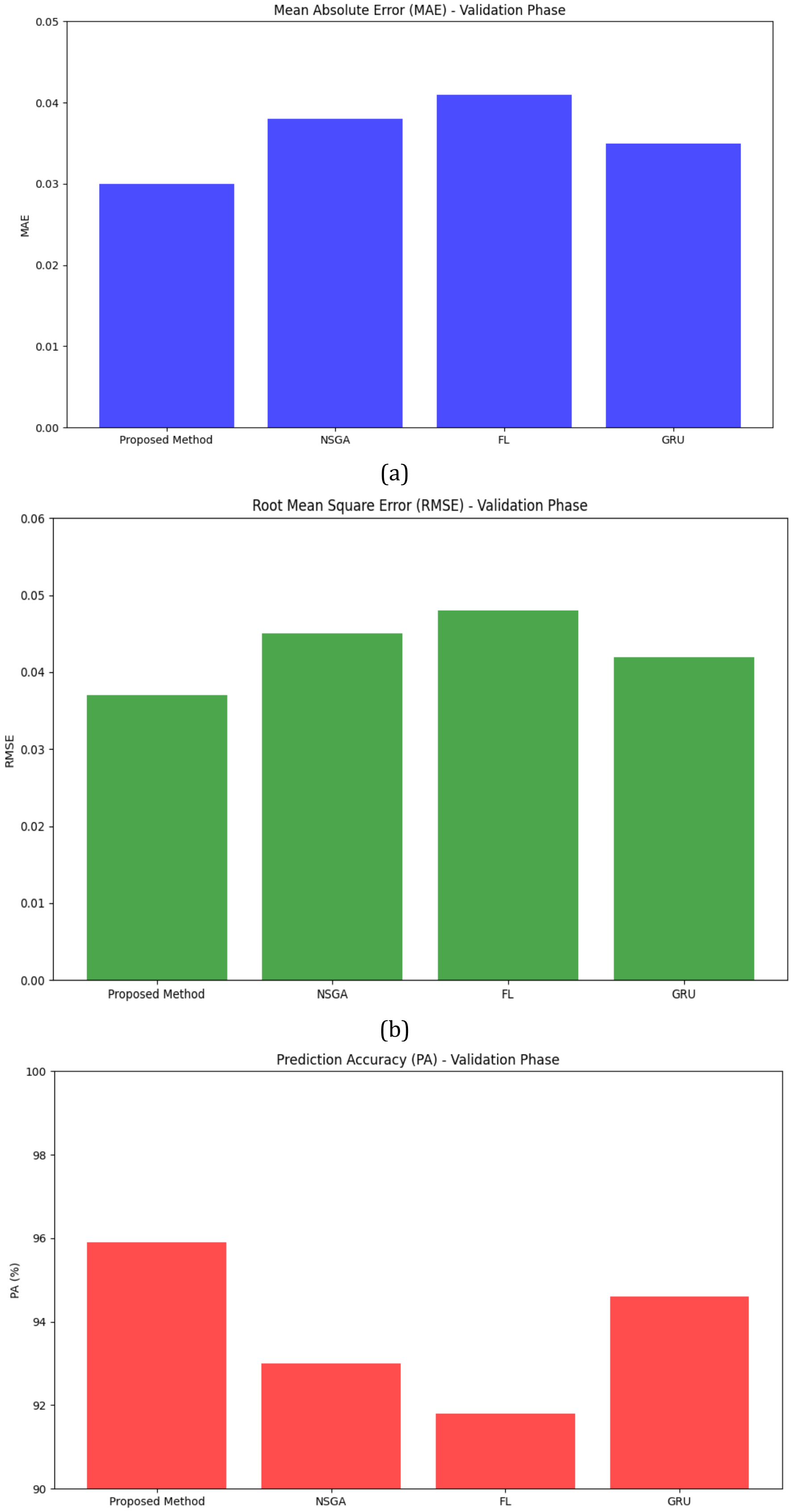


(a)

(b)

(c)

Figure 4. Results of Testing phase



(a)

(b)

(c)

Figure 5. Results of Validation Phase

When compared between the proposed method and current approaches (NSGA, FL, and GRU), mean absolute error (MAE), root mean squared error (RMSE), and prediction accuracy (PA) reveal the noteworthy gains gained by the proposed methodology across major performance parameters.

The proposed method recorded over the training period depicted in figure 3 with an MAE of 0.328, RMSE of 0.035, and PA of 96.8%. On the other hand, NSGA, FL, and GRU had MAEs of 0.034, 0.037, and 0.032 respectively; combined with RMSEs of 0.042, 0.45, and 0.040. Using the proposed method reduces MAE and RMSE by showing a more accurate fit to the training data, therefore reducing the mistakes from the present methods. The PA of the proposed method surpasses that of NSGA (94.7%), FL (93.2%), and GRU (95.4%) with its increased prediction accuracy.

The proposed approach maintained good performance in the testing phase illustrated in figure 4 with an MAE of 0.031, RMSE of 0.038, and PA of 95.6%. For FL, 0.043 (MAE), 0.050 (RMSE), and 91.5% (PA; for NSGA, 0.039 (MAE); and for GRU, 0.3636 (MAE); and 94.2% (PA). These results confirm the proposed method resilience in managing unseen data by offering lower prediction errors and higher accuracy.

The proposed method achieved what figure 4 depicts with an MAE of 0.030, RMSE of 0.037, and PA of 95.9%. With MAEs of 0.038, 0.041, and 0.035, NSGA, FL, and GRU showed higher error rates; RMSEs were 0.45, 0.48, and 0.042 respectively. Predicting accuracy-wise, the validation results reveal that the proposed method frequently beats NSGA, FL, and GRU, so enabling generalizing successfully.

5. CONCLUSION

The proposed method for IoV traffic prediction shows notable advancement over current methods such NSGA, FL, and GRU by combining bidirectional memory, spatiotemporal limitations, local search, and non-linear analysis. By means of thorough tests, the proposed model frequently outperformed conventional methods over significant criteria like MAE, RMSE, and PA. Although the proposed method obtained lower MAE and RMSE values in both training and testing phases, indicating a more precise prediction capacity, it represents its improved accuracy and gives higher PA. Consistent performance throughout all datasets helped the validation procedure considerably increase the generalizing capacity and model durability. These results highlight how effectively the proposed approach exactly reflects the complex, non-linear dynamics inherent in IoV traffic data. Local search paired with non-linear analysis allows one to generate adaptable models, therefore transcending the limitations of standard methods.

REFERENCES

[1] Liu, Y., Huo, L., Zhang, X., & Wu, J. (2023). A Multi-Objective Resource Pre-Allocation Scheme Using SDN for Intelligent Transportation System. IEEE Transactions on Intelligent Transportation Systems.

[2] Choudhry, M. D., Sivaraj, J., Munusamy, S., Muthusamy, P. D., & Saravanan, V. (2024). Industry 4.0 in Manufacturing, Communication, Transportation, and Health Care. Topics in Artificial Intelligence Applied to Industry 4.0, 149-165.

[3] Khan, A., Fouda, M. M., Do, D. T., Almaleh, A., & Rahman, A. U. (2023). Short-term traffic prediction using deep learning long short-term memory: Taxonomy, applications, challenges, and future trends. IEEE Access.

[4] Ramkumar, M., Logeshwaran, J., & Husna, T. (2022). CEA: Certification based encryption algorithm for enhanced data protection in social networks. Fundamentals of Applied Mathematics and Soft Computing, 1, 161-170.

[5] Hildebrand, B., Baza, M., Salman, T., Tabassum, S., Konatham, B., Amsaad, F., & Razaque, A. (2023). A comprehensive review on blockchains for Internet of Vehicles: Challenges and directions. Computer Science Review, 48, 100547.

[6] Rajalakshmi, M., Saravanan, V., Arunprasad, V., Romero, C. T., Khalaf, O. I., & Karthik, C. (2022). Machine Learning for Modeling and Control of Industrial Clarifier Process. Intelligent Automation & Soft Computing, 32(1).

[7] Yin, X., Wu, G., Wei, J., Shen, Y., Qi, H., & Yin, B. (2021). Deep learning on traffic prediction: Methods, analysis, and future directions. IEEE Transactions on Intelligent Transportation Systems, 23(6), 4927-4943.

[8] Kumar, M., Mukherjee, P., Verma, S., Kavita, Shafi, J., Wozniak, M., & Ijaz, M. F. (2023). A smart privacy preserving framework for industrial IoT using hybrid meta-heuristic algorithm. Scientific Reports, 13(1), 5372.

[9] Kolekar, S., Gite, S., Pradhan, B., & Kotecha, K. (2021). Behavior prediction of traffic actors for intelligent vehicle using artificial intelligence techniques: A review. IEEE Access, 9, 135034-135058.

[10] Safaei Yaraziz, M., Jalili, A., Gheisari, M., & Liu, Y. (2023). Recent trends towards privacy-preservation in Internet of Things, its challenges and future directions. IET Circuits, Devices & Systems, 17(2), 53-61.

[11] Obayya, M., Al-Wesabi, F. N., Alabdan, R., Khalid, M., Assiri, M., Alsaid, M. I., ... & Alneil, A. A. (2023). Artificial Intelligence for Traffic Prediction and Estimation in Intelligent Cyber-Physical Transportation Systems. IE Transactions on Consumer Electronics.

[12] Rizwan, S., Husnain, G., Aadil, F., Ali, F., & Lim, S. (2023). Mobile Edge-based Information-Centric Network for emergency messages dissemination in Internet of Vehicles: A Deep Learning Approach. IE Access, 11, 62574-62590.

[13] Cheong, C., Li, S., Cao, Y., Zhang, X., & Liu, D. (2024). False message detection in Internet of Vehicle through machine learning and vehicle consensus. Information Processing & Management, 61(6), 103827.

[14] Almutairi, M. S., Almutairi, K., & Chiroma, H. (2023). Selecting Features That Influence Vehicle Collisions in the Internet of Vehicles Based on a Multi-Objective Hybrid Bi-Directional NSGA-I. Applied Sciences, 13(4), 2064.

[15] Narsimhulu, P., Chithaluru, P., Al-Turjman, F., Guda, V., Inturi, S., Stephan, T., & Kumar, M. (2024). An intelligent FL-based vehicle route optimization protocol for green and sustainable IoT connected IoV. Internet of Things, 27, 101240.

[16] Wen, W., Xu, D., & Xia, Y. (2023). A novel traffic optimization method using GRU based deep neural network for the IoV system. PeerJ Computer Science, 9, e1411.