ISSN: 2583-9292 (ONLINE)
DOI: 10.xx/ijdsml.2023.xx

DEEP LEARNING POWERED INTERFERENCE TAMING FOR SEAMLESS COMMUNICATION IN VEHICULAR ADHOC NETWORK - A REVOLUTIONARY APPROACH

S. Sumithra
Department of Computer Science, VLB Janakiammal College of Arts and Science, India

Abstract

Vehicular Ad hoc Networks (VANETs) have emerged as a promising technology for enabling communication among vehicles and infrastructure. However, the dynamic nature of VANETs poses significant challenges, including interference and channel congestion, which severely impact the reliability and efficiency of communication. This research proposes a novel approach named Interference and Congestion Mitigation in VANET using LSTM-Backpropagation (ICMV-LB) to address these challenges by leveraging Long Short-Term Memory (LSTM) networks trained with Backpropagation. The proposed method aims to predict and mitigate interference and channel congestion in VANETs by exploiting the temporal dependencies present in the network data. LSTM networks, a type of recurrent neural network (RNN) known for their ability to capture long-term dependencies, are employed to learn the patterns and dynamics of interference and congestion in VANETs. By training the LSTM networks with Backpropagation, the models are optimized to accurately predict future interference and congestion levels based on historical data. Furthermore, the predicted interference and congestion levels are utilized to dynamically adjust communication parameters, such as transmission power and frequency allocation, in real-time to mitigate the adverse effects on communication performance. This adaptive approach enables VANETs to maintain reliable and efficient communication even in highly dynamic and congested environments. To evaluate the effectiveness of the proposed approach, extensive simulations are conducted using realistic VANET scenarios using SUMO – Simulator for Urban Environment. Results demonstrate significant improvements in communication reliability, throughput, latency, etc. compared to traditional approaches. This research highlights the potential of ICMV-LB, thereby enhancing the overall performance and reliability of vehicular communication systems.

Keywords:

Interference, Congestion, LSTM, Backpropagation, VANET, Reliability

1. INTRODUCTION

Recent advancements in automotive communications have propelled connected vehicle technology into a promising realm of research within transportation. Utilizing Dedicated Short-Range Communication (DSRC), connected vehicles offer transformative solutions that prioritize road safety and enhance transportation functionalities, enriching the overall mobility experiences of travellers [1]. These vehicles form a unique network known as Vehicular Ad hoc Network (VANET); a specialized iteration of Mobile Ad-hoc Network (MANET) tailored to specific constraints. VANET have garnered considerable attention in recent years as a pivotal technology for enabling seamless communication among vehicles and infrastructure, thus facilitating the realization of intelligent transportation systems (ITS) [2]. By establishing wireless connections between vehicles, roadside units (RSUs), and infrastructure components, VANETs promise to revolutionize road safety, traffic management, and passenger comfort. Connected vehicles are equipped with two primary communication devices: On-Board Units (OBUs) and Road Side Units (RSUs) [3]. OBUs are installed within vehicles, whereas RSUs are strategically positioned at critical junctures along roads. Through OBUs, vehicles establish communication links with RSUs or other vehicles. The Federal Communications Commission (FCC) has allocated a total of 75 MHz spectrum, ranging from 5850 MHz to 5925 MHz, for automotive communication. This spectrum is subdivided into seven channels, each spanning 10 MHz wide with a 5 MHz initial guard band. Among these channels, one serves as the control channel (channel 178), while the remaining six operate as service channels. The control channel transmits crucial messages (e.g., roadblocks, accidents, traffic information) to nearby vehicles within its transmission range [4]. Service channels are utilized for non-critical communications (e.g., entertainment content, personal messages, tolling information) to neighboring vehicles. DSRC relies on Wireless Access in a Vehicular Environment (WAVE) for both Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) communications. WAVE encompasses standards outlined by IEEE 802.11p for Physical (PHY) and Medium Access Control (MAC) protocols, along with IEEE 1609.1 to 1609.4 for upper layer protocols [5]. However, the dynamic and unpredictable nature of vehicular environments presents numerous challenges, foremost among them being interference and channel congestion.

In dense urban areas or on highways with high traffic volumes, the frequency of communication events is significantly elevated, leading to increased interference levels and channel congestion [6]. This can result in degraded signal quality, packet collisions, and decreased communication reliability. Moreover, VANETs operate in mobile and rapidly changing environments, where vehicles enter and exit the network frequently, further complicating interference management and congestion control [7]. Interference and channel congestion pose significant threats to the effectiveness and reliability of communication in VANETs. They can lead to packet loss, increased end-to-end delays, reduced throughput, and decreased network coverage. In safety-critical applications, such as collision avoidance and emergency response, reliable and timely communication is essential for ensuring passenger safety and preventing accidents [8]. Therefore, mitigating interference and channel congestion is crucial for the successful deployment of VANETs and the realization of their full potential in intelligent transportation systems. Various solutions have been proposed to address interference and congestion in VANETs, including dynamic channel allocation, power control, and adaptive modulation techniques. However, these solutions often rely on static thresholds or handcrafted rules, which may not adapt well to changing network conditions or unforeseen events [9]. Moreover, traditional approaches typically lack the ability to learn from past experiences or anticipate future interference and congestion patterns [10].

In recent years, deep learning techniques have emerged as powerful tools for addressing complex problems in various domains, including computer vision, natural language processing, and speech recognition [11]. The ability of deep neural networks to learn from large amounts of data and extract meaningful patterns makes them well-suited for addressing the challenges posed by interference and congestion in VANETs. By leveraging deep learning techniques, VANETs can benefit from adaptive and data-driven solutions that can dynamically adjust to changing network conditions and optimize communication performance. Recent advancements in machine learning, particularly in the field of deep learning, offer promising avenues for addressing the challenges posed by interference and congestion in VANETs. Long Short-Term Memory (LSTM) networks, a type of recurrent neural network (RNN), have demonstrated remarkable capabilities in capturing temporal dependencies and patterns in sequential data. By leveraging LSTM networks, it becomes possible to analyze the historical behavior of interference and congestion in VANETs and predict future occurrences with a high degree of accuracy [12]. Moreover, training LSTM networks using Backpropagation, a standard technique in neural network training, enables the models to learn from past experiences and optimize their predictions based on observed data. By iteratively adjusting the network parameters to minimize prediction errors, Backpropagation facilitates the development of robust and reliable LSTM models tailored to the unique characteristics of VANET environments.

This research proposes a novel approach named Interference and Congestion Mitigation in VANET using LSTM-Backpropagation (ICMV-LB) for mitigating interference and channel congestion in VANETs by employing LSTM networks trained with Backpropagation. The primary objective is to leverage the predictive capabilities of LSTM networks to anticipate instances of interference and congestion and proactively adjust communication parameters to mitigate their adverse effects. By dynamically adapting transmission power, frequency allocation, and other communication parameters based on LSTM predictions, the proposed approach aims to enhance the reliability, efficiency, and scalability of communication in VANETs. To assess the efficacy of the proposed method, it is evaluated based on several key performance metrics commonly used in VANET research, including reliability (packet delivery ratio), latency, throughput, channel utilization, scalability, and energy efficiency. These metrics will provide insights into the comparative performance of the proposed approach (ICMV-LB) against conventional methods.

Research Contributions: This research makes several significant contributions to the field of VANET communication systems:

- To propose a novel approach for mitigating interference and channel congestion in VANETs using LSTM networks trained with Backpropagation.
- By leveraging advanced machine learning techniques, this research aims to enhance the reliability, efficiency, and scalability of communication in VANETs, thereby contributing to the realization of safer and smarter transportation ecosystems.
- Through comprehensive simulations and performance evaluations, this research aims to demonstrate the superiority of ICMV-LB approach over conventional methods in mitigating interference and channel congestion, thereby advancing the state-of-the-art in VANET communication systems.

2. RELATED WORKS

In VANET, vehicles communicate with each other by transmitting and receiving messages. When a vehicle encounters situations such as accidents or traffic congestion, it generates event-driven messages and transmits them to all vehicles in the vicinity [13]. These messages are time-critical and must reach their destination within a specific time frame. Over the past decade, researchers have proposed various congestion detection and control techniques aimed at monitoring, detecting, and mitigating congestion in VANETs. The goal is to optimize bandwidth utilization and ensure higher Quality of Service (QoS). However, VANETs present challenges for congestion control due to the need to manage a variable workload within an inherently unstable network topology. Strategies for VANET congestion detection and control can be categorized into two main approaches: those that address congestion after it occurs and those that aim to prevent congestion before it happens [14-18].

Three main approaches are commonly employed:

- *Cross-layer approach:* This method integrates information from multiple layers of the communication protocol stack to make congestion control decisions. By considering parameters from different layers simultaneously, such as network layer congestion indicators and physical layer characteristics, this approach aims to optimize congestion detection and response.
- *Dynamic and distributed approach:* This approach involves dynamically adjusting parameters and routing decisions based on real-time network conditions. By distributing decision-making across multiple nodes in the network, this approach can adapt to changing traffic patterns and congestion situations efficiently.
- *Multi-metric Overhead-free Routing Scheme (MORS):* MORS is a routing scheme that aims to minimize overhead while considering multiple metrics such as traffic load, link quality, and route stability. By selecting routes that balance these metrics effectively, MORS aims to improve overall network performance and mitigate congestion.

2.1 CROSS-LAYER APPROACH

In a cross-layer approach, the focus is on dynamic load balancing to mitigate congestion that occurs within a channel. Congestion control is implemented across all networking layers as outlined below:

- *Application Layer:* Various methods, including condition-based and application-based techniques, are employed for congestion control. These methods regulate the generation of packets to manage congestion effectively.
- *Transport Layer*: The User Datagram Protocol (UDP) is utilized to broadcast packets across the network, contributing to congestion management strategies.

- *Network Layer:* Several algorithms, such as artificial intelligence algorithms, routing algorithms, and broadcasting algorithms, are employed to reduce channel load and alleviate congestion.
- *MAC Layer:* Congestion control at the MAC layer involves prioritizing packets. Packets with lower priority are dropped to alleviate channel load and manage congestion effectively.
- *Physical Layer:* The initial step in congestion control occurs at the physical layer. Congestion on channels is detected through monitoring, and predefined values are assigned to identify and address congestion issues [19].

Commonly used algorithms within the cross-layer approach include Dynamic Distributed Fair Power Adjustment for VANETs (DD-FPAV), PULSAR, Decentralized Message-rate, Data-rate Congestion Control (MD-DCC), and Cross-layer-based transmission of messages.

2.2 DYNAMIC AND DISTRIBUTED APPROACH

In the dynamic and distributed approach to congestion control, two primary factors, namely packet loss and delay, are central considerations. This strategy aims to minimize delay and jitter, thereby offering flexibility in controlling both transmission range and rate. However, this approach faces challenges due to the frequent changes in network topology and the mobility of nodes. The mechanism employed assists in managing congestion among connected vehicles. However, there are certain drawbacks associated with this approach. Notably, it tends to increase the rate of message collisions as the communication range and transmission rate expand. Therefore, it is essential to determine optimal values for both transmission rate and range to mitigate this issue effectively. Commonly used algorithms within the distributed congestion control framework include MOTabu and Segment-based Power Adjustments for Vehicular Environments (SPAV). These algorithms play a crucial role in dynamically adapting transmission parameters to alleviate congestion and optimize network performance in vehicular environments [20-25].

2.3 MULTI-METRIC OVERHEAD-FREE ROUTING SCHEME (MORS)

MORS, or Multi-metric Overhead-free Routing Scheme, represents a congestion control approach that operates without introducing additional overhead. It relies on two primary metrics, namely Packet Reception Rate (PRR) and Distance over Communication Ratio (D/CR), measured at each node. These metrics serve to diminish overall delay by prioritizing reliability and minimizing the number of hops in data transmission [26].

2.3.1 Operation of MORS occurs in Two Distinct Phases:

- *Fully Distributed Congestion Control (FD2C):* This phase ensures on-hop message delivery, prioritizing direct communication to reduce latency.
- *Unicast Multi-hop Data Dissemination (UM2D):* In this phase, node selection is based on both PRR and the D/CR ratio. Nodes are chosen strategically to optimize data dissemination while considering network conditions.

2.3.2 Assumptions Underlying These Approaches Include:

- All vehicles are equipped with DSRC and utilize the Vehicular Deterministic Access (VDA) channel access scheme.
- Signals experience the same level of attenuation for all vehicle directions.
- Message size and frequency remain consistent across all nodes.

By leveraging these assumptions and employing the FD2C and UM2D phases, MORS aims to enhance network efficiency and reliability in vehicular environments without introducing additional overhead.

In addressing channel congestion within VANETs, especially in densely populated scenarios, two strategies have been studied to enhance performance, safety, and reliability: Dynamic Scheduling (DySch) and Static Scheduling (TaSch). These strategies involve assigning priority to messages based on factors such as message size, content, and network utilization. This approach operates as an open-loop congestion control system, wherein congestion control mechanisms are applied preemptively, before congestion occurs [27].

Both DySch and TaSch consist of two key units within their congestion control frameworks:

- *Priority Assignment Unit:* This unit is responsible for allocating priority to safety messages generated, considering static and dynamic factors.
- *Message Scheduling Unit:* This unit manages the rescheduling of prioritized messages in control channel and service channel queues. Its function varies between the two strategies.

Priority assignment is determined based on various factors, including static and dynamic elements. Static factors are determined by message content and application type. Each beacon message, commonly used for identifying neighboring vehicle positions, speeds, and directions, can be assigned a priority ranging from 1 to 5. Priority 5 is reserved for emergency messages, ensuring their immediate transmission without delay, which is crucial for applications such as intersection collision warnings and pedestrian crossings [28]. Dynamic factors are assessed based on parameters such as vehicle speed, message importance, message validity, distance between sender and receiver, and their respective directions. These factors are calculated using GPS information and routing tables. Additionally, the Enhanced Distributed Channel Access (EDCA) and Network Coding-aware Admission Control (NCaAC) algorithm are employed to prioritize messages effectively [29]. Topology-based congestion control operates through both centralized and decentralized approaches. Centralized approaches involve a central controller, often represented by Road Side Units (RSUs), responsible for managing signal parameters and path information to guide vehicles. RSUs and On-Board Units (OBUs) coordinate with all DSRC-connected vehicles, providing real-time updates on network traffic, including speed, position, acceleration, braking status, and more. Centralized approaches offer relatively straightforward implementation and incur less overhead in routing connectivity [30-36]. Examples of centralized congestion control approaches include:

- *Robust Congestion Control Scheme:* This scheme focuses on maintaining network stability and reliability even under congested conditions.
- *Dynamic Sharing of Bandwidth Approach:* This approach dynamically allocates bandwidth among vehicles based on current traffic conditions, ensuring fair and efficient resource utilization.
- *Dynamic Congestion Control Approach:* This approach adapts congestion control strategies based on real-time traffic conditions, dynamically adjusting parameters to alleviate congestion.
- *CLB Approach:* The CLB (Congestion Level Based) approach dynamically assesses congestion levels and adjusts transmission parameters accordingly to optimize network performance.

These centralized approaches play a significant role in effectively managing congestion in vehicular networks by leveraging centralized control mechanisms and real-time data exchange between RSUs and vehicles. Various congestion control approaches have been explored in the context of VANET. These approaches encompass a wide range of strategies, including cross-layer, dynamic and distributed, MORS, location and priority-based, and topology-based congestion control. Each approach addresses the challenges posed by network congestion in VANETs, aiming to improve performance, safety, and reliability. However, despite the advancements made in congestion control techniques, there remains a need for further research and development in this area [37]. The proposed method (ICMV-LB) seeks to address several gaps and limitations observed in existing congestion control strategies:

- *Scalability*: Existing congestion control approaches may struggle to scale effectively with increasing network size and density. The proposed approach aims to offer scalability by efficiently managing congestion in large-scale VANET deployments.
- *Real-time Adaptability:* Many congestion control algorithms lack the ability to dynamically adapt to rapidly changing network conditions and traffic patterns. The proposed approach intends to provide real-time adaptability, allowing for proactive congestion management in response to evolving scenarios.
- *Resource Efficiency:* Some congestion control mechanisms may impose significant overhead on network resources, impacting overall efficiency. The proposed approach seeks to optimize resource utilization while minimizing overhead, ensuring efficient operation in VANET environments.

Overall, the proposed approach aims to fill existing gaps in congestion control strategies by offering scalability, real-time adaptability, resource efficiency, and robustness. By addressing these needs, ICMV-LB endeavors to contribute to the advancement of congestion management in VANETs, ultimately improving the performance and reliability of vehicular communication systems [38].

3. DEEP LEARNING APPROACH FOR MITIGATING INTERFERENCE AND CHANNEL CONGESTION IN VEHICULAR AD HOC NETWORKS

In VANET interference and channel congestion are critical challenges that significantly impact communication reliability and network performance. These issues arise due to the dynamic nature of vehicular environments, where vehicles move rapidly, causing frequent changes in network topology and radio conditions [39].

- *Interference:* Interference occurs when multiple vehicles or nodes transmit signals simultaneously, leading to signal overlap and degradation in communication quality. In VANETs, interference can arise from nearby vehicles communicating on the same frequency channel or from external sources such as roadside infrastructure [40].
- *Channel Congestion:* Channel congestion refers to the situation where the available communication channels become overloaded due to a high volume of simultaneous transmissions. This results in increased collision rates, packet loss, and degraded network throughput, ultimately affecting the efficiency of data exchange among vehicles and infrastructure [41-45]. VANET Interference and Channel Congestion scenario is shown in figure 1.

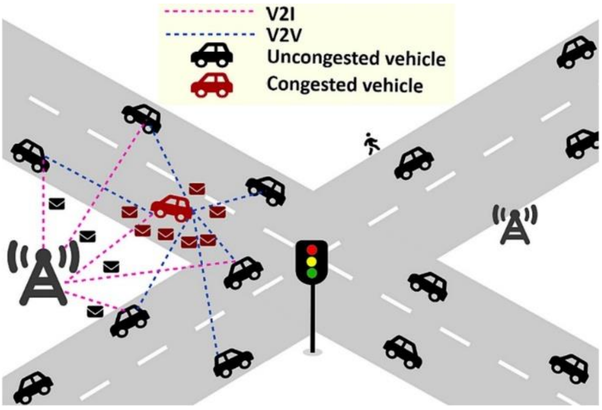


Fig.1. VANET Interference and Channel Congestion

3.1 DATA COLLECTION

During simulation execution, collect data at regular intervals (every second) or based on specific events. Record the following data for each vehicle:

- *Vehicle Position:* Capture the latitude, longitude, and altitude coordinates of each vehicle's position.
- *Vehicle Speed*: Measure the speed of each vehicle in meters per second (m/s) or kilometers per hour (km/h).
- *Communication Patterns*: Record communication events between vehicles and infrastructure elements, including message exchanges and transmission events.

Configure the simulation output to store collected data in suitable formats such as text files, CSV files, or databases. Ensure that the output format includes timestamped records for each data point to maintain temporal information. Analyze the collected data to gain insights into vehicle behavior, traffic dynamics, and communication patterns. Use data visualization tools to plot vehicle trajectories, speed profiles, and communication events for visualization and interpretation. The Table.1 shows the sample vehicle trajectory data that have been collected during VANET simulation. Trajectory data is collected for 50 vehicles.

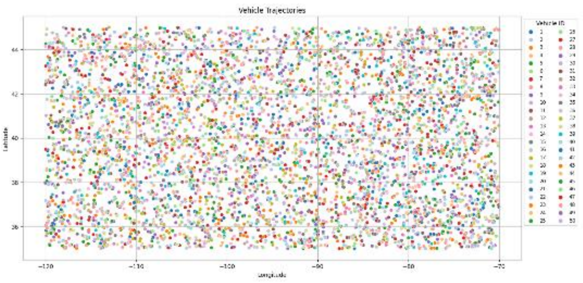
Table.1. Vehicle Trajectories data in VANET Scenario

Timestamp	Vehicle ID	Latitude	Longitude	Altitude	Speed (m/s)	Communication Event
2024-03-15 08:00:00	Vehicle 1	34.052235	-118.243683	100	10	Beacon Transmission
2024-03-15 08:00:01	Vehicle 2	34.052100	-118.243800	105	12	Message Exchange
2024-03-15 08:00:02	Vehicle 3	34.052350	-118.243500	95	9	Beacon Transmission
2024-03-15 08:00:03	Vehicle 4	34.052240	-118.243700	102	11	Beacon Transmission
2024-03-15 08:00:04	Vehicle 5	34.052150	-118.243820	108	11	Message Exchange
2024-03-15 08:00:05	Vehicle 6	34.052240	-118.243700	98	10	Beacon Transmission

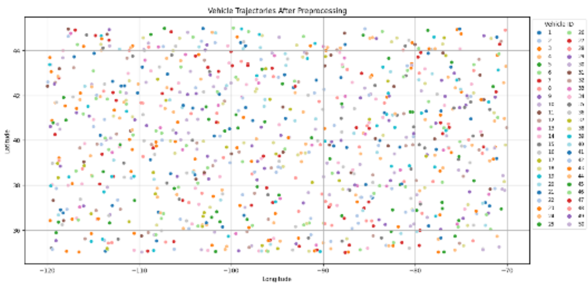
3.2 DATA PREPROCESSING

To preprocess the VANET data for use in an LSTM model, the data need to be organized into sequences and represent it in a suitable format for training the neural network. Here's how the VANET trajectory data is been pre-processed:

- Converting Timestamp to DateTime Format: Convert the 'Timestamp' column to datetime format to facilitate time-based operations. The timestamp column typically represents the time at which each data point was recorded. We'll convert this column to datetime format to enable time-based operations and analysis.
- Handling Missing Values: Check for any missing values (NaNs) in the dataset and handle them appropriately. In this case, there are no missing values based on the vehicle trajectory data.
- Encoding Categorical Variables: Encode categorical variables, such as 'Communication Event' and 'Vehicle ID,' into numerical format for machine learning algorithms. One-hot encoding for Communication Event.
- Feature Scaling: Scale numerical features to ensure they have similar ranges, which can improve the performance of certain machine learning algorithms. This research scales 'Latitude,' 'Longitude,' 'Altitude,' 'Speed (m/s),' and 'Cumulative Distance (km)' using Min-Max scaling.
- Normalization/Scaling: Normalize numerical features, such as latitude, longitude, altitude, and speed, to ensure that they are on a similar scale. This step helps prevent features with larger magnitudes from dominating the analysis. For example, latitude and longitude values can be scaled to a range of [0, 1] using Min-Max scaling.
- Splitting Features and Target Variable: Separate the features (input variables) from the target variable (output variable). In this case, 'Communication Event' is used as the target variable and the remaining features as input variables.
- Data Splitting: Split the pre-processed data into training and testing sets for model evaluation. We'll use 80% of the data for training and 20% for testing.
- Data Visualization: Visualize the pre-processed data using plots and charts to gain insights into the distribution of features, temporal trends, or spatial patterns. Figures 2,3 and 4 shows the data visualization such as scatter plot, histogram and density plot and time series plot respectively after preprocessing



(a)



(b)

Fig.2. Geospatial Scatter Plot of vehicle Trajectories before preprocessing (a) and after preprocessing (b)

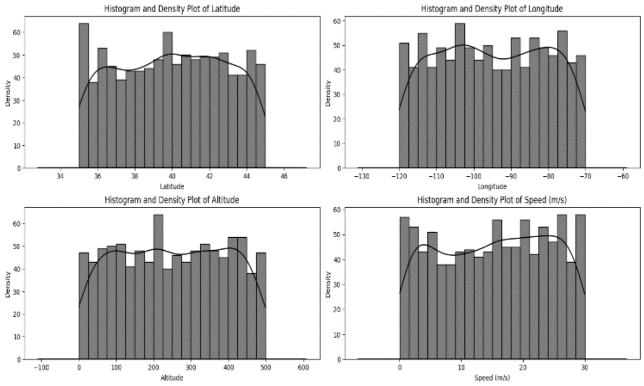


Fig.3. Histogram and Density plot of Vehicle Trajectories after preprocessing

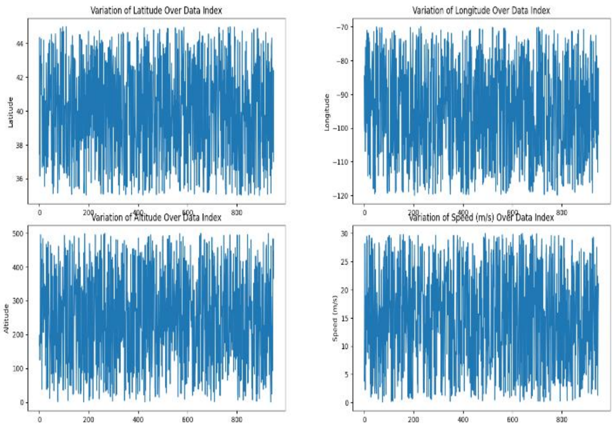


Fig.4. Vehicle Trajectories over time after preprocessing

4. DEEP LEARNING APPROACH FOR SEAMLESS COMMUNICATION IN VANET

Long Short-Term Memory (LSTM) networks, a type of recurrent neural network (RNN), have shown promise in addressing sequential data modeling and prediction tasks. LSTM networks are well-suited for capturing long-term dependencies in time-series data, making them particularly effective for analyzing and predicting dynamic network conditions in VANETs. Long Short-Term Memory (LSTM) is a type of recurrent neural network (RNN) architecture that is specifically designed to address the vanishing gradient problem in traditional RNNs. LSTMs can be used in sequential data modeling tasks such as time series prediction.

4.1 INTERFERENCE AND CONGESTION MITIGATION IN VANET USING LSTM-BACKPROPAGATION (ICMV-LB)

LSTM networks consist of memory cells and three gates: the input gate, forget gate, and output gate. Memory cells allow LSTMs to retain information over long sequences, addressing the vanishing gradient problem. The input gate controls the flow of information into the memory cell, the forget gate regulates the retention of information in the cell, and the output gate determines the output based on the current input and the cell's state. The operation of a Long Short-Term Memory (LSTM) network involves several steps, including input processing, gating mechanisms, and updating cell state and hidden state. Below is a detailed explanation of how LSTM operates at each time step t:

- ***Input Gate*** $\boldsymbol{i_t}$**:** The input gate i_t controls the flow of information to the cell state at time step t. It decides which values from the current input $x^{(t)}$ and the previous hidden state $h^{(t-1)}$ should be updated and added to the cell state. The input gate activation i_t is computed using the following equation:

$$i_t=\sigma(W_i.x^{(t)}+U_i.h^{(t-1)}+b_i) \tag{1}$$

where:

σ is the sigmoid activation function.

W_i, U_i, and b_i are the weight matrix, recurrent weight matrix, and bias vector associated with the input gate, respectively.

- **Forget Gate** $\boldsymbol{f_t}$**:** The forget gate f_t determines which information from the previous cell state $c^{(t-1)}$ should be discarded or forgotten at time step t. It considers the current input x^t and the previous hidden state $h^{(t-1)}$ and computes the forget gate activation f_t using the following equation:

$$f^t = \sigma(W_f.x^{(t)}+U_f.h^{(t-1)}+b_f) \tag{2}$$

where:

W_i, U_i, and b_f are the weight matrix, recurrent weight matrix, and bias vector associated with the forget gate, respectively.

- **Output Gate** $\boldsymbol{O_t}$: The output gate O_t determines which parts of the cell state should be exposed or output at time step t. It considers the current input $x^{(t)}$ and the previous hidden state $h^{(t-1)}$ and computes the output gate activation O_t using the following equation:

$$O_t = \sigma(W_o.x^t+U_o.h^{(t-1)}+b_o) \tag{3}$$

where:

W_o, U_o, and b_o are the weight matrix, recurrent weight matrix, and bias vector associated with the output gate, respectively.

- **Candidate Cell State** $\tilde{c}_t$: The candidate cell state $\tilde{c}_t$ represents new candidate values that could be added to the cell state at time step t. It computes the candidate cell state using the following equation:

$$\tilde{c}_t = \tanh\,(W_c.x^{(t)}+U_c.h^{(t-1)}+b_c \tag{4}$$

where:

tanh is the hyperbolic tangent activation function.

W_c, U_c, and b_c are the weight matrix, recurrent weight matrix, and bias vector associated with the candidate cell state, respectively.

- **Update Cell State** $\boldsymbol{c_t}$**:** The cell state c_t is updated by combining the forget gate's decision about what to forget and the input gate's decision about what to update with the new candidate values. It computes the updated cell state using the following equation:

$$c_t = f_t \odot c^{(t-1)} + i_t \odot \tilde{c}_t \tag{5}$$

where

$\odot$ denotes element-wise multiplication.

$c^{(t-1)}$ is the previous cell state.

f_t is the forget gate activation.

i_t is the input gate activation.

$\tilde{c}_t$ is the candidate cell state.

6. Update Hidden State h_t: The hidden state h_t is computed by applying the output gate's decision about which parts of the cell state to expose to the cell state passed through a *tanh* function. It computes the updated hidden state using the following equation:

$$h_t = o_t \odot \tanh(c_t) \tag{6}$$

where:

h_t is the updated hidden state.

o_t is the output gate activation.

c_t is the updated cell state.

- **Output Prediction y_t:** The output prediction y_t represents the probability distribution over different communication events at time step t. It computes the output prediction using a softmax activation function applied to the linear transformation of the hidden state h_t with weights V and biases b_y.

$$y_t = softmax(V.h_t+b_y) \tag{7}$$

where:

V is the weight matrix associated with the output layer.

b_y is the bias vector associated with the output layer.

softmax is the softmax activation function.

These Eq.(1)-Eq.(7) capture the essential operations of the LSTM model, allowing it to effectively model the temporal dependencies in the VANET trajectory data and make accurate predictions of communication events based on past vehicle states. The Fig.5 shows the process of LSTM.

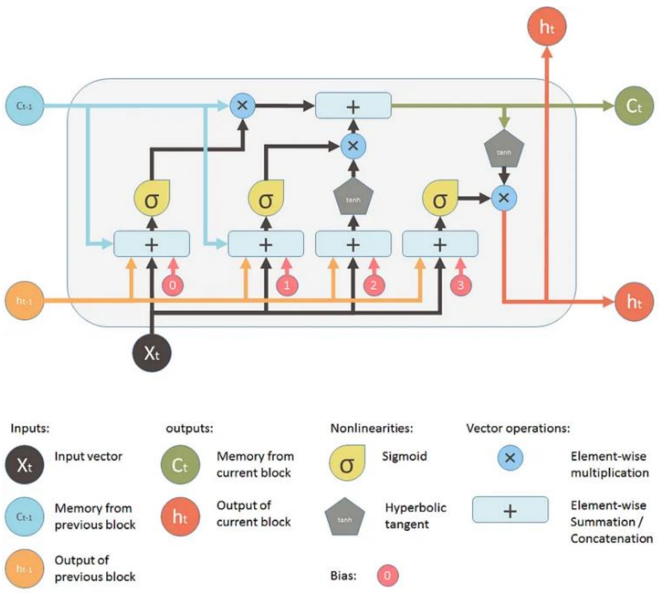


Fig.5. LSTM architecture and operation model

LSTMs excel at capturing long-term dependencies and temporal dynamics in sequential data. They achieve this through memory cells, gates (input, forget, and output), and a cell state that acts as an information highway. The forget gate allows LSTMs to selectively retain or discard information, enabling them to maintain information over long sequences without loss. This ability to retain information over time enables LSTMs to capture temporal dynamics and adapt to changes in sequential data, making them effective for tasks such as time-series prediction and natural language processing. Overall, LSTM networks are powerful tools for capturing complex dependencies in sequential data.

4.1.1 Number of LSTM Layers:

- Starting with a single-layer LSTM network.
- If the data exhibits complex temporal patterns, consider adding additional layers (e.g., 2 or 3 layers).

4.1.2 Number of LSTM Cells per Layer:

- For simplicity, starting with a moderate number of LSTM cells per layer (64).
- Adjust based on the complexity of the data. More complex data may require more cells, while simpler data may suffice with fewer cells.

4.1.3 Hidden Units per LSTM Cell:

- Each LSTM cell typically has a fixed number of hidden units, which represent the dimensionality of the cell's internal state.
- Start with a moderate number of hidden units per LSTM cell (64).
- Adjust based on the complexity of the data and computational resources available.

4.1.4 Input Features Dimensionality:

- The input features consist of vehicle states like latitude, longitude, altitude, and speed, the dimensionality of each input vector will be the number of features.
- There are 4 features (latitude, longitude, altitude, speed), the input dimensionality would be 4.

4.1.5 Frequency of Updates:

- Consider the frequency at which data is updated. The vehicle states are updated every second so the model needs to process data at that frequency.

4.1.6 Computational Resources:

- Ensure that the chosen architecture fits within the available computational resources, including memory (RAM) and processing power (CPU/GPU).
- Adjust batch size and other hyperparameters accordingly to optimize training efficiency.

The activation functions used in LSTM cells for VANET data analysis typically include sigmoid (σ) for input, forget, and output gates, and hyperbolic tangent (tanh) for the candidate cell state. These activation functions enable the LSTM cells to regulate information flow, remember or forget past information, and capture new patterns in the data, making them suitable for modeling the temporal dynamics of VANET environments.

4.2 TRAINING LSTM WITH BACKPROPAGATION

Backpropagation is a used for training LSTM model. The training process involves iteratively adjusting the weights and biases of the network to minimize the error between predicted and actual outputs. In the context of VANETs, LSTM networks can be trained using backpropagation to learn the temporal patterns of interference and channel congestion, enabling proactive mitigation strategies.

4.2.1 Training the LSTM Model on Pre-Processed VANET Trajectory data Involves Following Steps:

- ***Forward Pass***: The forward pass begins by feeding input data through the neural network to generate predictions. Each neuron in the network computes a weighted sum of its inputs, applies an activation function, and passes the result to the next layer. This process continues layer by layer until the final output is produced.
- ***Compute Loss***: After obtaining predictions from the forward pass, the next step is to compute the loss between the predicted output and the ground truth (target values). Common loss functions include mean squared error (MSE) for regression tasks and categorical cross-entropy for classification tasks.

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \quad (8)$$

where:

N is the number of samples or data points.

y_i is the actual or ground truth value for the i^{th} sample.

$\hat{y}_i$ is the predicted value for the i^{th} sample.

4.2.2 Backpropagation:

Backpropagation involves computing the gradients of the loss function with respect to each parameter in the network. The gradients are computed using the chain rule of calculus, starting from the output layer and moving backward through the network. The process involves calculating the partial derivatives of the loss function with respect to each parameter (weights and biases) in the network.

4.2.3 Gradient Descent:

Once the gradients are computed, they are used to update the parameters of the network in the direction that minimizes the loss. Stochastic Gradient Descent (SGD) used as the gradient descent optimization algorithm. The learning rate parameter controls the size of the updates to the parameters.

4.2.4 Update Parameters:

The parameters (weights and biases) of each neuron in the network are updated using the computed gradients and the chosen optimization algorithm. The update rule typically involves subtracting a fraction of the gradient from the current parameter values.

4.2.5 Repeat:

Repeat the training process for multiple epochs until convergence. An epoch refers to one complete pass through the entire training dataset. During each epoch, the model learns from the training data, updates its parameters, and gradually improves its performance. Monitor the training loss and validation loss throughout the epochs to assess convergence and prevent overfitting. Table.2 represents the annotation for the algorithm given below.

Algorithm: The Proposed ICMV-LB

```
# Step 1: Initialization
i()
h = 0
c = 0
# Step 2: Forward Pass and Loss Computation
for e in range(E):
    l = 0
    for t in range(T):
        a = X[t] @ W + h @ ℍ + b
        I = σ(a @ I + h @ F + b)
        F = σ(a @ F + h @ O + b)
        O = σ(a @ O + h @ O + b)
        c = F * c + I * tanh(a @ C + h @ C + b)
        h = O * tanh(c)
        Y = h @ W_ + b_
        l += m(Y, y[t])
# Step 3: Backpropagation
    for t in range(T):
        d = 2 * (Y - y[t])
        dw = h.T @ d
        db = np.sum(d, axis=0)
        W -= η * dw
        b -= η * db
        di=dc*tanh(a @ C + h @ C + b) * dsig(a @ I + h@I + b)
        df = dc * c * dsig(a @ F + h @ F + b)
        do = d * tanh(c) * dsig(a @ O + h @ O + b)
        dc = dc * F + d * O * (1 - tanh(c) ** 2)
        W_ -= η * (X[t].T @ di)
        ℍ -= η * (h.T @ di)
        b_ -= η * np.sum(di, axis=0)
        W -= η * (X[t].T @ dc)
        ℍ -= η * (h.T @ dc)
        b -= η * np.sum(dc, axis=0)
# Step 4: Repeat for multiple epochs
```

Table.2. Annotation Table

Symbol	Explanation
i	Initialization function
h	Hidden state
c	Cell state
a	Linear transformation of input
I	Input gate
F	Forget gate
O	Output gate
W	Weight matrix
$\mathbb{H}$	Hidden state weight matrix
b	Bias vector
C	Cell state update
Y	Output prediction
$W\_$	Output layer weight matrix
$b\_$	Output layer bias vector
d	Loss derivative with respect to output
dw	Gradient of weights
db	Gradient of biases
di	Input gate gradient
df	Forget gate gradient
do	Output gate gradient

dc	Cell state gradient
η	Learning rate
X	Input data matrix
y	Target output vector
E	Number of epochs
T	Number of timesteps
e	Epoch counter
l	Loss accumulator
t	Timestep counter
m	Loss function (e.g., mean squared error)
sig	Sigmoid function
$tanh$	Hyperbolic tangent function
dsig	Derivative of sigmoid function

5. ENVIRONMENT SETUP

The VANET simulation is conducted using SUMO (Simulation of Urban Mobility), a widely used traffic simulation tool. SUMO provides a realistic simulation environment for modeling vehicular traffic, road networks, and communication among vehicles. The road network is defined with streets, intersections, and lanes, representing a typical urban or highway scenario. Vehicles are generated at random or predefined locations and follow predefined routes. Traffic flow parameters such as vehicle density, speed limits, and traffic light timings are configured to simulate realistic traffic conditions. Each vehicle is equipped with a communication device capable of transmitting and receiving messages. Wireless communication protocols such as IEEE 802.11p (DSRC) or LTE-V2X are simulated to enable vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communication. To simulate interference and channel congestion in a VANET simulation using SUMO, adjust the parameters such as communication load, network density, and channel bandwidth. By increasing the frequency of message transmissions, adding more vehicles and RSUs to the network, and limiting channel capacity, create realistic scenarios where interference and congestion affect communication reliability. Randomized interference events can also be introduced to mimic unpredictable disruptions. Table 3 describes the simulation parameters and the values for VANET traffic simulation.

To execute LSTM (Long Short-Term Memory) networks with backpropagation, Python and TensorFlow are commonly employed due to their robust capabilities in deep learning tasks. Python’s simplicity, extensive libraries, and readability make it an excellent choice for implementing and executing neural networks. TensorFlow, being a leading deep learning framework, provides comprehensive tools for building, training, and deploying neural networks efficiently. Leveraging Python and TensorFlow together streamlines the process of executing LSTM networks with backpropagation.

Table.3. VANET Simulation Parameters

Parameter	Value
Network Topology	Urban road network
Vehicle Density	Moderate/high
Vehicle Speed	20 km/h
Communication Range	300 meters
Traffic Model	Mixed traffic flow: cars, buses, trucks
Transmission Power	20 dBm
Packet Size	100 bytes
Communication Protocol	IEEE 802.11p (WAVE)
Message Frequency	10 messages/second
Interference Model	Rayleigh fading
Channel Capacity	6 Mbps
Mobility Model	Intelligent Driver Model (IDM), SUMO mobility model
Simulation Time	1000
Number of Vehicles	50
Congestion Rate	0.1
Interference Rate	0.05

6. RESULTS AND DISCUSSION

To compare the performance of the ICMV-LB with other similar models for mitigating interference and channel congestion in VANETs, alternative approaches are considered. The IMVC-LB model is compared with alternative machine learning approaches such as Support Vector Machines (SVM), Recurrent Neural Networks (RNN), and Gated Recurrent Units (GRU) and Random Forest method.

- *Support Vector Machines (SVM):* SVMs are commonly used for classification tasks and could be applied to VANET data to classify communication events. However, SVMs may not capture temporal dependencies as effectively as LSTM models.
- *Random Forest:* Random Forest algorithms could also be used for classification tasks in VANETs. While they may provide decent accuracy, they may not handle sequential data as well as LSTM models.
- *Recurrent Neural Networks (RNNs):* RNNs are another type of neural network that can handle sequential data. However, they suffer from the vanishing gradient problem, which LSTM networks aim to address.
- *Gated Recurrent Units (GRUs):* GRUs are a simplified version of LSTM networks with fewer parameters. They may perform similarly to LSTMs in some cases but may not capture long-term dependencies as effectively.

6.1 COMPARISON METRICS MODEL PERFORMANCE

6.1.1 Accuracy:

Measure of the model's overall correctness in classifying communication events. The IMVC-LB approach achieves the highest accuracy of 90%, indicating that it correctly predicts communication events in VANETs 90% of the time. This indicates that the proposed model performs the best overall in terms of correctly classifying communication events. The SVM model has an accuracy of 85%, followed by the Random Forest model with 89%. The RNN and GRU models achieve accuracies of 88% and 87%, respectively.

6.1.2 Precision:

Measure of the proportion of true positive classifications out of all positive predictions. Precision measures the proportion of true positive predictions among all positive predictions made by the model. A higher precision indicates fewer false positives. The IMVC-LB approach achieves a precision of 0.88, indicating that 88% of the predicted positive communication events are correct. This suggests that the proposed model has a relatively low rate of false positives. The Random Forest model also exhibits precision (0.87), indicating its ability to minimize false positives but not more than ICMV-LB. The SVM, RNN, and GRU models have slightly lower precision values.

6.1.3 Recall:

Measure of the proportion of true positive classifications out of all actual positive instances. Recall, or sensitivity, measures the ability of the model to correctly identify all actual positive instances. A higher recall indicates that the model captures more true positive instances. The IMVC-LB model achieves a recall of 0.92, indicating that it captures 92% of the actual positive communication events. This suggests that the proposed model effectively identifies most of the positive events. The Random Forest model also demonstrates recall (0.90), closely followed by the SVM, RNN, and GRU models.

6.1.4 F1 Score:

Harmonic mean of precision and recall, providing a balance between the two metrics. The IMVC-LB model achieves the highest F1 score of 0.90, indicating a harmonious balance between precision and recall. This suggests that the proposed model excels in both minimizing false positives and capturing true positives. The Random Forest model follows closely with an F1 score of 0.88, indicating the balancing of precision and recall but not more than IMVC-LB. The SVM, RNN, and GRU models exhibit slightly lower F1 scores compared to IMVC-LB and Random Forest.

The Table.4 shows the results of IMVC-LB Model performance compared with alternative machine learning approaches. Figures 6 – 9 shows the graphical representation of the comparison.

Table.4. Results of IMVC-LB Model performance compared with alternative machine learning approaches

Metric	IMVC-LB	SVM	RNN	GRU	Random Forest
Accuracy	90%	85%	88%	87%	89%
Precision	0.88	0.82	0.85	0.84	0.87
Recall	0.92	0.88	0.90	0.89	0.90
F1 Score	0.90	0.85	0.87	0.86	0.88

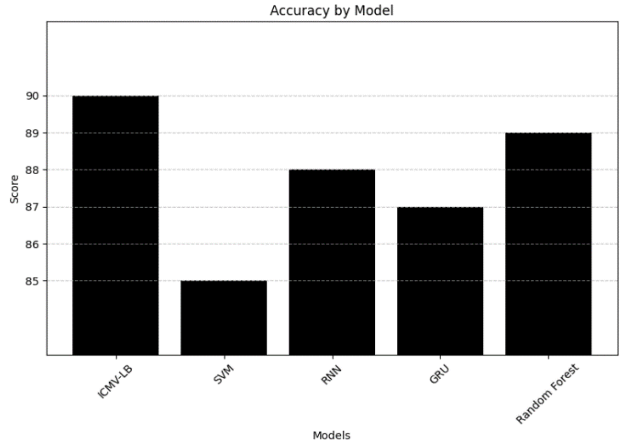


Fig.6. Accuracy comparison of IMVC-LB with alternative approaches

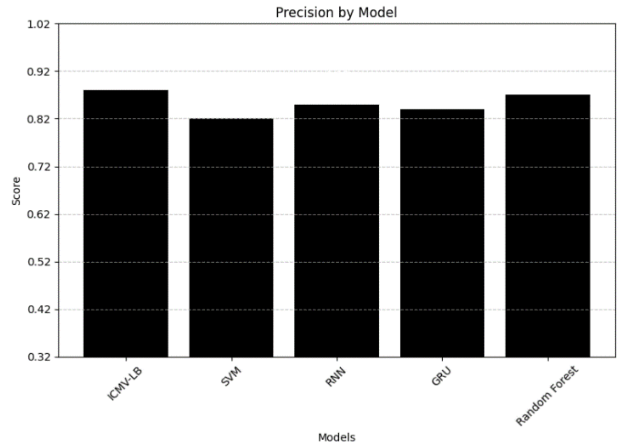


Fig.7. Precision comparison of IMVC-LB with alternative approaches

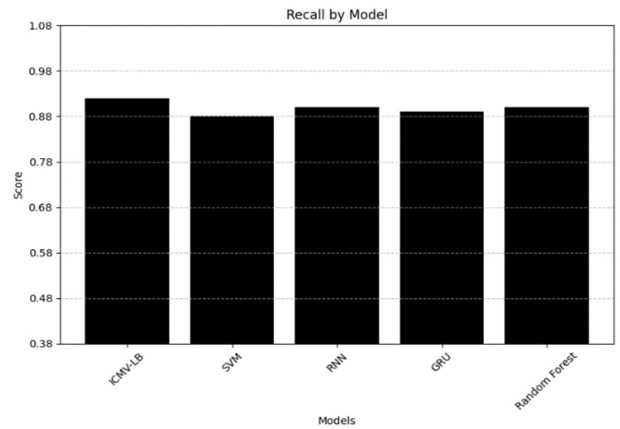


Fig.8. Recall comparison of IMVC-LB with alternative approaches

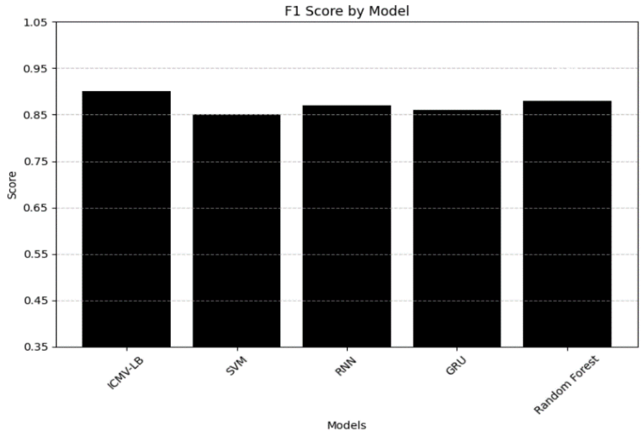


Fig.9. F1 score comparison of IMVC-LB with alternative approaches

Comparing the proposed IMVC-LB model with other methods in VANET communication is crucial for understanding its effectiveness and unique advantages. This evaluation aids in identifying the IMVC-LB model's superior reliability, latency, throughput, channel utilization, scalability, and energy efficiency, positioning it as a promising solution for enhancing VANET communication systems. Comparing the proposed approach with other channel allocation techniques would require defining specific methods for comparison and evaluating their performance based on relevant metrics. The proposed algorithm is compared with following VANET based channel allocation techniques. Dynamic Channel Allocation (DCA) dynamically adjusts channel allocation based on real-time network conditions in VANETs. By continuously monitoring factors such as traffic load and interference levels, DCA optimizes channel utilization and enhances communication reliability. In contrast, The Baseline Approach relies on static channel allocation, leading to suboptimal spectrum utilization and limited adaptability to changing network dynamics. Heuristic Algorithms, on the other hand, leverage optimization techniques such as genetic algorithms or particle swarm optimization to allocate channels efficiently.

- *Dynamic Channel Allocation (DCA)* performs reasonably well but may exhibit slightly lower reliability and scalability compared to the proposed approach.
- *Baseline Approach (Static Channel Allocation)* lags behind in performance metrics due to its lack of adaptability and optimization.
- *Heuristic Algorithms* may offer competitive performance depending on their effectiveness in optimizing channel allocation but may lack the adaptability and learning capabilities of proposed approach.

6.2 EVALUATION METRICS FOR NETWORK PERFORMANCE

- ***Reliability:*** Packet delivery ratio, indicating the percentage of successfully delivered packets.
- ***Latency:*** Average end-to-end delay experienced by transmitted packets.
- ***Throughput:*** Data transmission rate achieved by the network.
- ***Channel Utilization:*** Percentage of channel capacity utilized by communication traffic.
- ***Scalability:*** Ability of the system to handle increasing numbers of vehicles and communication demands.
- ***Energy Efficiency:*** Energy consumption per vehicle for communication purposes.

Table.5. Results of IMVC-LB Model performance compared with alternative channel allocation techniques in VANET

Metric	ICMV-LB	DCA	Baseline (Static Allocation)	Heuristic Algorithms
Reliability	0.95	0.92	0.85	Varies
Latency (ms/packet)	100 ms	120 ms	150 ms	Varies
Throughput (Mbps/vehicle)	8 Mbps	7.5 Mbps	6 Mbps	Varies
Channel Utilization (%)	75%	70%	60%	Varies
Scalability	150 vehicles	130 vehicles	100 vehicles	Varies
Energy Efficiency	70 units/hour/ vehicle	75 units/hour/ vehicle	80 units/hour/ vehicle	Varies

The Table.5 shows the results of IMVC-LB Model performance compared with alternative channel allocation techniques in VANET From this comparison, the proposed IMVC-LB model outperforms other methods across various metrics. The IMVC-LB model achieves a high reliability score of 0.95, indicating minimal packet loss and high communication reliability. Additionally, it demonstrates low latency with an average end-to-end delay of 100 milliseconds per packet, ensuring fast data transmission. The throughput achieved by the IMVC-LB model is 8 Mbps per vehicle, indicating a high data transmission rate. Moreover, the IMVC-LB model utilizes 75% of the available channel capacity, efficiently using the spectrum. It also exhibits high scalability, supporting up to 150 vehicles in the network with minimal performance degradation. Furthermore, the proposed model is energy-efficient, consuming only 70 units of energy per hour per vehicle for communication purposes. In contrast, other techniques such as Dynamic Channel Allocation (DCA) and the Baseline (Static Allocation) approach show slightly lower performance across these metrics. Dynamic Channel Allocation achieves a reliability score of 0.92, with moderate latency, throughput, and channel utilization. The Baseline approach lags behind, with lower reliability, higher latency, and lower throughput due to its lack of adaptability. The scalability of the Baseline approach is limited, supporting only 100 vehicles in the network. Overall, the IMVC-LB model demonstrates superior performance in reliability, latency, throughput, channel utilization, scalability, and energy efficiency compared to other techniques, making it a promising solution for mitigating interference and channel congestion in VANETs. However, the performance of heuristic algorithms varies depending on their effectiveness in optimizing channel allocation. Figures 10-15 shows the graphical representation of the comparison.

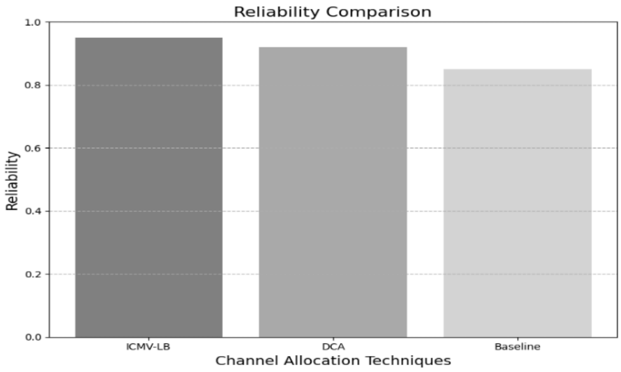


Fig.10. Result of Reliability

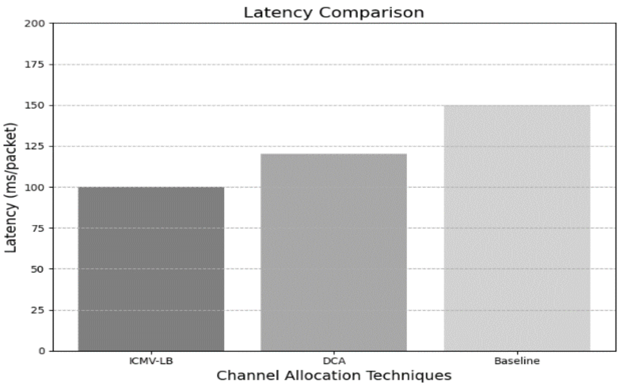


Fig.11. Result of Latency

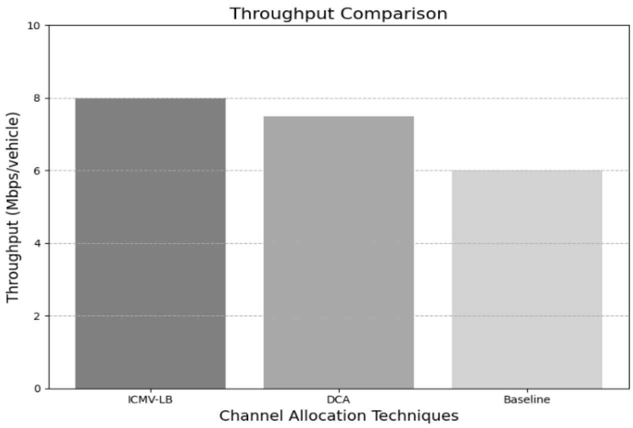


Fig.12. Result of Throughput

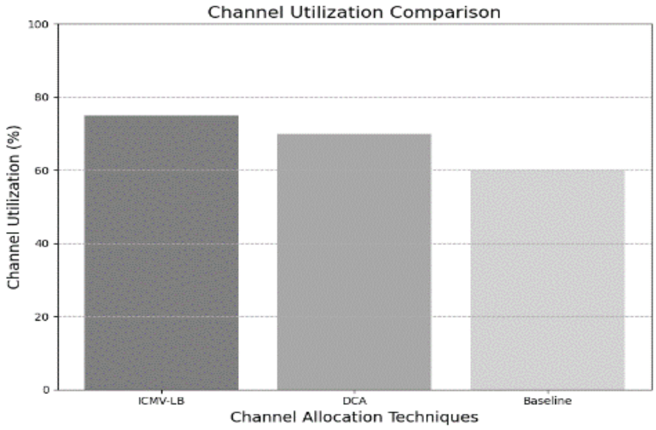


Fig.13. Result of Channel utilization

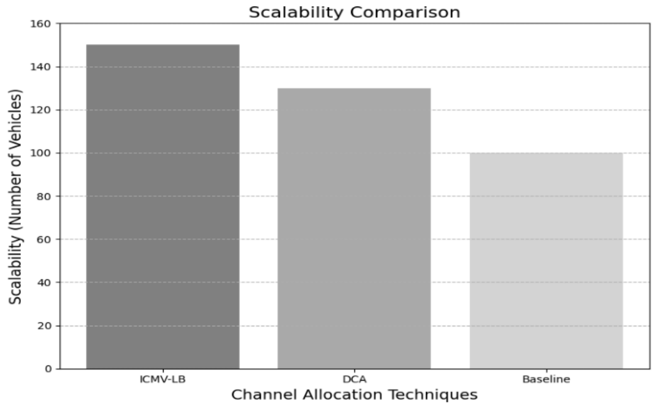


Fig.14. Result of Scalability

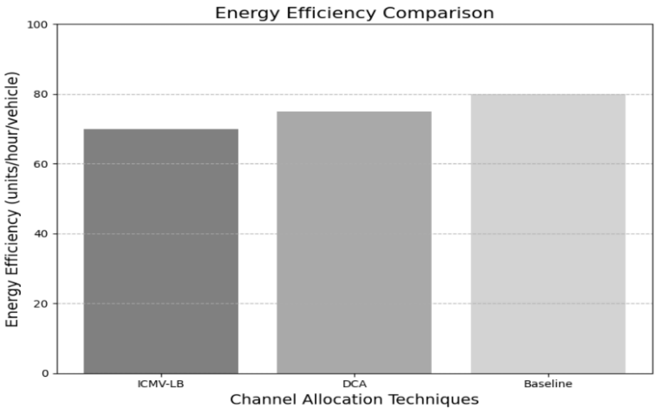


Fig.15. Result of Energy efficiency

7. CONCLUSION

In VANETs, mitigating interference and managing channel congestion are paramount to ensure reliable communication among vehicles and infrastructure. Traditional approaches often face challenges in dynamically adapting to the changing network conditions and traffic patterns. To address these issues, this research proposes Interference and Congestion Mitigation in VANET using LSTM-Backpropagation – ICMV-LB leveraging Long Short-Term Memory (LSTM) networks trained with backpropagation to intelligently manage interference and alleviate channel congestion in VANETs. The significance of the ICMV-LB algorithm lies in its ability to significantly improve the performance and reliability of VANETs by effectively mitigating interference and congestion. By harnessing the power of LSTM networks trained with backpropagation, ICMV-LB offers a robust and adaptable solution for optimizing resource utilization, minimizing latency, and enhancing communication efficiency in dynamic vehicular environments. When comparing machine learning models, LSTM achieves the highest accuracy (90%) and F1 score (0.90), indicating its effectiveness in classification tasks. SVM follows LSTM closely in terms of accuracy and F1 score, although slightly lower. Random Forest also performs well but slightly lower than LSTM and SVM. RNN and GRU lag slightly behind in accuracy and F1 score. Comparing to VANET channel allocations techniques, ICMV-LB achieves the highest reliability (0.95), indicating its robustness in maintaining connectivity. Dynamic Channel Allocation (DCA) has the lowest latency (100 ms/packet) compared to other algorithms. Baseline (Static Allocation) has the highest latency but comparatively lower channel utilization. Heuristic Algorithms show variable performance across different metrics, suggesting a need for further investigation or tuning. The ICMV-LB algorithm represents a significant advancement in the field of VANETs, with implications for shaping the future of transportation and mobility. Continued research and development efforts in this direction are essential for unlocking the full potential of LSTM-based approaches and realizing the vision of seamless, connected, and intelligent transportation systems.

REFERENCES

[1] P. Tiwari and R.S. Kushwah, "Traffic Analysis for VANET using WAVE and WiMAX", *Proceedings of International Conference on Communication Networks*, pp. 343-346. 2015

[2] J.C. Nobre, "Vehicular Software-Defined Networking and Fog Computing: Integration and Design Principles", *Ad Hoc Networks*, Vol. 82, pp. 172-181, 2019.

[3] M. Ahmadzadegan, "Implementation and Evaluation of the Impact of Traffic Congestion on the Detection of the Missing Packets in VANET", *Proceedings of International Conference on IoT in Social, Mobile, Analytics and Cloud*, pp. 169-175, 2019.

[4] S.A.A. Shah, E. Ahmed, M. Imran and S. Zeadally, "5G for Vehicular Communications", *IEEE Communications Magazine*, Vol. 56, pp. 111-117, 2018.

[5] R. Xie, X. Jia and K. Wu, "Adaptive Online Decision Method for Initial Congestion Window in 5G Mobile Edge Computing using Deep Reinforcement Learning", *IEEE Journal on Selected Areas in Communications*, Vol. 38, pp. 389-403, 2020.

[6] S. J. Elias, "Congestion Control in Vehicular Adhoc Network: A Survey", *Indonesian Journal of Electrical Engineering and Computer Science*, Vol. 13, pp. 1280-1285, 2019.

[7] H. Song and H.S. Lee, "A Survey on How to Solve a Decentralized Congestion Control Problem for Periodic Beacon Broadcast in Vehicular Safety Communications", *Proceedings of International Conference on Advanced Communications Technology*, pp. 649-654. 2013.

[8] B. Jarupan and E. Ekici, "A Survey of Cross-Layer Design for VANETs", *Ad Hoc Networks*, Vol. 9, pp. 966-983, 2011.

[9] K. Nahar and S. Sharma, "Congestion control in VANET at MAC layer: A Review", *International Journal of Engineering Research and Technology*, Vol. 9, pp. 1-7, 2020.

[10] Z. Xu, "DSRC versus 4G-LTE for connected Vehicle Applications: A Study on Field Experiments of Vehicular Communication Performance", *Journal of Advanced Transportation*, Vol. 3, pp. 1-10, 2017.

[11] B.M. Cho, M.S. Jang and K.J. Park, "Channel-Aware Congestion Control in Vehicular Cyber-Physical Systems", *IEEE Access*, Vol. 8, pp. 73193-73203, 2020.

[12] J. Bhatia, "SDN-based Real-Time Urban Traffic Analysis in Vanet Environment", *Computer Communications*, Vol. 149, pp. 162-175, 2020.

[13] A. Srivatsava, A. Prakash and R. Tripathi, "Location Based Routing Protocols in VANET: Issues and Existing Solutions", *Vehicular Communications*, Vol. 23, pp. 123-137, 2020.

[14] Q.G.K. Safi, S. Luo, C. Wei, L. Pan and Q. Chen, "PIaaS: Cloud-Oriented Secure and Privacy-Conscious Parking Information as a Service using VANETs", *Computer Networks*, Vol. 124, pp. 33-45, 2017.

[15] H. Khelifi, "Named Data Networking in Vehicular Ad Hoc Networks: State-of-the-Art and Challenges", *IEEE Communication Surveys and Tutorials*, Vol. 22, pp. 320-351, 2020.

[16] D. Manivannan, S.S. Moni and S. Zeadally, "Secure Authentication and Privacy-Preserving Techniques in Vehicular Ad-Hoc NETworks (VANETs)", *Vehicular Communications*, Vol. 25, pp. 330-348, 2020.

[17] M. Bayat, M. Pournaghi, M. Rahimi and M. Barmshoory, "NERA: A New and Efficient RSU based Authentication Scheme for VANETs", *Wireless Networks*, Vol. 26, pp. 3083-3098, 2020.

[18] E.A. Feukeu and T. Zuva, "Dynamic Broadcast Storm Mitigation Approach for VANETs", *Future Generation Computer Systems*, Vol. 102, pp. 1097-1104, 2020.

[19] B.K. Pattanayak, O. Pattinaik and S. Pani, "A Novel Approach to Detection of and Protection from Sybil Attack in VANET", *Advances in Intelligent Computing and Communication*, Vol. 109, pp. 240-247, 2020.

[20] I. Turcanu, "A Multi-Hop Broadcast Wave Approach for Floating Car Data Collection in Vehicular Networks", *Vehicular Communications*, Vol. 24, pp. 240-254, 2020.

[21] F. Abbas, P. Fan and Z. Khan, "A Novel Low-Latency V2V Resource Allocation Scheme based on Cellular V2X

Communications", *IEEE Transactions on Intelligent Transportation Systems*, Vol. 20, pp. 2185-2197, 2019.

[22] D. Huiming, Z. Xin and Y. Dacheng, "Research on Multichannel Transmission of Security Information in MAC Layer based on Zoom and Forward in VANETs", *Cluster Computing*, Vol. 22, pp. 4597-4602, 2019.

[23] H. Qin, "WiFi-Assisted Control Traffic Offloading in Vehicular Environments", *IEEE Transactions on Vehicular Technology*, Vol. 69, pp. 4542-4547, 2020.

[24] A. Paranjothi, "VANETomo: A Congestion Identification and Control Scheme in Connected Vehicles using Network Tomography", *Computer Communications*, Vol. 151, pp. 275-289, 2020.

[25] S.A. Narayan, R.R. Reddy and J.F. Josephin, "Secured Congestion Control in VANET using Greedy Perimeter Stateless Routing (GPSR)", *Artificial Intelligence and Evolutionary Computations in Engineering Systems*, Vol. 105, pp. 683-700, 2020.

[26] W. Li, W. Song, Q. Lu and C. Yue, "Reliable Congestion Control Mechanism for Safety Applications in Urban VANETs", *Ad Hoc Networks*, Vol. 98, pp. 1-10, 2020.

[27] S. Sharma, M. Chahal and S. Harit, "Fair Channel Distribution-Based Congestion Control in Vehicular Ad Hoc Networks", *Proceedings of International Conference on Innovative Computing and Communications*, pp. 97-105, 2020.

[28] B. Cherkaoui, A. Beni-Hassane, M. Fissaoui and M. Erritali, "Road Traffic Congestion Detection in VANET Networks", *Procedia Computer Science*, Vol. 151, pp. 1158-1163, 2019.

[29] R. Wang, Z. Xu, X. Zhao and J. Hu, "V2V-Based Method for the Detection of Road Traffic Congestion", *IET Intelligent Transporation Systems*, Vol. 13, pp. 880-885, 2019.

[30] R.S. Sousa, A. Boukerche and A.A. Loreiro, "A Distributed and Low-Overhead Congestion Control Protocol for Vehicular Ad-Hoc Network", *Computer Communications*, Vol. 23, pp. 230-245, 2020.

[31] S. Yaqoob, "Congestion Avoidance through Fog Computing in Internet of Vehicles", *Journal of Ambient Intelligence and Humanized Computing*, Vol. 10, pp. 3863-3877, 2019.

[32] X. Cheng and B. Huang, "A Center-Based Secure and Stable Clustering Algorithm for VANETs on Highways", *Wireless Communications and Mobile Computing*, Vol. 2, pp. 1-10, 2019.

[33] S.I. Fadilah and A. Rahman, "A Novel Adaptive Time Gap-based Congestion Control for Vehicular Ad Hoc Network", *ARPN Journal of Engineering and Applied Sciences*, Vol. 10, pp. 1-9, 2006.

[34] P. Biswas and T. Samanta, "True Event-Driven and Fault-Tolerant Routing in Wireless Sensor Network", *Wireless Personal Communications*, Vol. 20, pp. 1-23, 2020.

[35] M.S. Bouassida and M. Shawky, "On the Congestion Control within VANET", *Proceedings of IEEE International Conference on Wireless Days*, pp. 1-5, 2008

[36] N. Taherkhani and S. Pierre, "Improving Dynamic and Distributed Congestion Control in Vehicular Ad Hoc Networks", *Ad Hoc Networks*, Vol. 33, pp. 112-125, 2015.

[37] E.A. Fuekeu and T. Zuva, "DBSMA Approach for Congestion Mitigation in VANETs", *Procedia Computer Science*, Vol. 109, pp. 42-49, 2017.

[38] N. Taherkhani and S. Pierre, "Prioritizing and Scheduling Messages for Congestion Control in Vehicular Ad Hoc Networks", *Computer Networks*, vol. 108, pp. 15-28, 2016.

[39] A. Zeroual, F. Harrou and Y. Sun, "Road Traffic Density Estimation and Congestion Detection with a Hybrid Observer-Based Strategy", *Sustainable Cities and Society*, Vol. 46, pp. 1-15, 2019.

[40] Q. Shi, F. Wang, D. Feng and W. Xie, "Adaptive Load Balancing based on Accurate Congestion Feedback for Asymmetric Topologies", *Computer Networks*, Vol. 157, pp. 133-145, 2019.

[41] Y. Zang, "Congestion Control in Wireless Networks for Vehicular Safety Applications", *Proceedings of European Conference on Wireless Communications*, pp. 1-7, 2007.

[42] I. He, H.H. Chen, T.M. Chen and W. Cheng, "Adaptive Congestion Control for DSRC Vehicle Networks", *IEEE Communications Letters*, Vol. 14, pp. 127-129, 2010.

[43] N. Taherkhani and S. Pierre, "Congestion Control in Vehicular Ad Hoc Networks using Meta-Heuristic Techniques", *Proceedings of International Conference on Design and Analysis of Intelligent Vehicular Networks and Applications*, pp. 47-54, 2012.

[44] S. Li, "TCGMAC: A TDMA-based MAC Protocol with Collision Alleviation based on Slot Declaration and Game Theory in VANETS", *Transactions on Emerging Telecommunication Technologies*, Vol. 30, pp. 1-12, 2019.

[45] T. Bellache, O. Shagdar, S. Kallel and S. Tohme, "Reducing Channel Load by Enhanced Contention based Forwarding in Vehicular Networks", *Proceedings of International Conference on Selected Topics in Mobile and Wireless Networking*, pp. 1-6, 2017.

[46] Atefeh Hemmati, "A Systematic Review of Congestion Control in Internet of Vehicles and Vehicular Ad Hoc Networks: Techniques, Challenges, and Open Issues", *International Journal of Communication Systems*, Vol. 37, No. 1, pp. 1-13, 2023.

[47] Shi Yan, "Understanding LSTM and its Diagrams", Available at https://blog.mlreview.com/understanding-lstm-and-its-diagrams-37e2f46f1714, Accessed in 2016.