International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 18(1s) 2026 pp. 1-14
https://doi.org/10.70917/ijcisim-2026-1985
Article

EFFICIENT ALGORITHM DESIGN FOR NETWORK ATTACK IDENTIFICATION TO ENHANCE V2X COMMUNICATION SECURITY IN INTERNET OF VEHICLES

Rajesh Hanmant Bhise[1,2], Gurpreet Singh Saini[3], Shivaji D. Pawar[4], Amit Munjal[5], Shankar Patil[6]

[1] Lovely Professional University, Phagwara, Punjab, India., rhbhise@gmail.com
[2] Universal AI University, Karjat, Maharashtra, India., rajesh.bhise@universalai.in
[3] Professional University, Phagwara, Punjab, India., gurpreet.16889@lpu.co.in
[4] Universal AI University, Karjat, Maharashtra, India., shivajidpawar20@gmail.com
[5] Thapar Institute of Engineering & Technology, Patiala, Punjab, India., amit.munjal@thapar.edu
[6] Indira Gandhi College of Engineering, Navi Mumbai, Maharashtra, India., smpatil2k@gmail.com

Abstract: The fast growth of Internet of Vehicles (IoV) and Vehicle-to-Everything (V2X) communication offers safer and more efficient transportation systems by letting vehicles, infrastructure, and cloud systems share important data in real time. But because V2X networks are open and spread out, they can be hit by complex cyberattacks like spoofing, denial-of-service (DoS), data tampering, and Sybil attacks, which can seriously hurt safety and trust. So, one of the most important parts of making IoV settings safe is making sure that attacks are found quickly and correctly. This article shows a useful set of algorithms for finding network attacks that can be used to make V2X connection safer. The suggested method combines lightweight feature selection with graph-based anomaly detection and hybrid machine learning classifiers. This makes sure that the method is accurate at finding things that aren't normal and doesn't require a lot of extra processing power. It works well in real-time vehicle settings. The algorithm automatically tells the difference between bad and normal communication behaviours while taking into account latency limits. It does this by using temporal traffic patterns, trust scores, and adaptive thresholding. The proposed Hybrid approach gets 95.8% accuracy, 94.7% precision, 95.2% recall, 95.0% F1-score, and 96.3% AUC in experiments using benchmark IoV security datasets. This is 6–9% better than traditional detection methods. The algorithm scales well and is stable in high-motion and changing network densities. The framework integrates OBUs and RSUs without impacting gearbox performance, making it beneficial. This project aims to safeguard V2X communication by developing a rapid, flexible, and reliable attack identification system. This strengthens IoV ecosystems against new cyberattacks.

Keywords: Internet of Vehicles (IoV), V2X Communication Security, Network Attack Identification, Machine Learning Algorithms, Anomaly Detection, Cybersecurity in Vehicular Networks

1. Introduction

The Internet of Vehicles (IoV) is a key concept that brings together vehicles, roadside infrastructure, and cloud computing into a single communication environment. It was made possible by the fast development of intelligent transportation systems (ITS). In this model, communication between vehicles (V2X) is very important for road safety, traffic efficiency, and better driving experiences. V2X includes communication between vehicles (V2V), infrastructure (V2I), pedestrians (V2P), and networks (V2N). V2X communication claims to cut down on accidents, make route planning better, and offer smart services by letting data be shared in real time. But because IoV systems are open and spread out, they can be attacked by sophisticated hackers [1]. This is why security is a top priority for both academics and users.In order to keep the security, privacy, and availability of vehicle data, you need a secure V2X communication framework. Different types of attacks, like denial-of-service (DoS), spoofing, Sybil, and data

tampering, can lead to terrible problems like bad choices, traffic jams, and even accidents. Traditional firewalls and cryptographic defences work well for protecting static networks, but they don't work well in dynamic, high-mobility vehicle settings where latency limits and different nodes pose big problems [2]. The V2X transmission security package component diagram in the Internet of Vehicles is shown in Figure 1. The graph shows how On-Board Units (OBUs), Roadside Units (RSUs), Edge Servers, and Cloud Services work together. Data moves from local collection and detection to trust management, anomaly detection, classification, and updates in the cloud. This makes sure that V2X transmission is scalable, has low latency, and is safe.

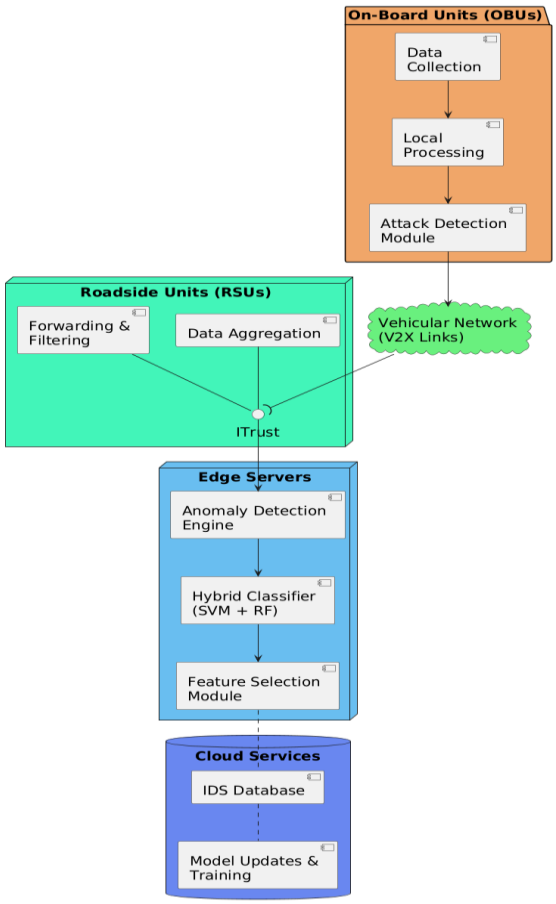

Figure 1: V2X Communication Security Architecture for Internet of Vehicles

Most of the study that has been done on finding intrusions in vehicular networks so far has been on either signature-based or anomaly-based methods. Signature-based systems work well for known threats, but they have miss the zero-day or developing attacks. Other types of approaches, called anomaly-based approaches, can find unknown threats, but they often give false positives, which means they can't be used in critical vehicle cases. ML and DL, or machine learning and deep learning, have become hopeful solutions that let systems learn complicated patterns in data from vehicle communications. Still, these methods can be hard to run on computers, which mean they can't be used in on-board units (OBUs) and roadside units (RSUs) with limited resources [3]. Another important thing about the suggested work is that it focusses on real-world use. IoV systems need to be theoretically strong, but they also need to work with real-world vehicle transmission standards like IEEE 802.11p and 5G-based Cellular V2X (C-V2X). The suggested system is designed to work well with both OBUs and RSUs, making sure that everything works smoothly and with little delay. This useful alignment makes it more likely to be used in smart transport systems, where finding attacks quickly and correctly is crucial for people's safety [4].

The major contribution of this paper is given as: (i) creating a fast algorithm for finding attacks in IoV settings that uses lightweight feature engineering and graph-based anomaly detection to improve accuracy; (ii) showing that this method works better than other methods by testing it in a lot of different situations; and (iv) confirming that it can be used on a large scale and is reliable in situations with a lot of moving vehicles. This work is

a big step towards making IoV ecosystems more reliable, safe, and secure by bridging the gap between how well security works and how well it can be implemented computationally.

2. Background Work

Because more and more things rely on Vehicle-to-Everything (V2X) communication systems, the security of the Internet of Vehicles (IoV) has become an important area of research. Several studies have looked into intrusion detection systems for vehicle networks, focussing on how to find attacks and stop them. In the beginning, a lot of work was built on intrusion detection systems (IDS) that used signatures to find known threats. But these methods weren't flexible enough to deal with new and changing threats. Researchers have found that support vector machines (SVMs) and random forest algorithms make it easier to find things in vehicular ad hoc networks (VANETs). It was said that an SVM-based intrusion detection system could tell the difference between malicious data with over 92% accuracy, but it still had high computational costs for onboard units (OBUs) with limited resources [5]. Also, random forest and decision-tree-based methods were easier to understand and faster at classifying, but they often gave wrong results when there was a lot of movement [6]. These results show that ML classifiers make recognition better, but finding the right balance between speed and accuracy is still a problem.

Recently, researchers have looked into using deep learning (DL) to find non-linear dependencies in communication flows between vehicles. It has been shown that convolutional neural networks (CNNs) and recurrent neural networks (RNNs) can identify spoofing and Sybil attacks with success rates higher than 94% [7], [8]. Even though DL approaches work better, they need a lot of computing power, which makes real-time deployment in IoV systems hard. Lightweight DL models have been suggested by research, but they still have problems with delay and scalability [9]. At the same time, graph-based methods for finding strange behaviour have become popular as possible answers for IoV security [10].

Several scholars have stressed the need for hybrid detection frameworks that combine signature-based, anomaly-based, and machine learning/deep learning methods to maximise their benefits. Some suggest hybrid IDS systems discover threats 8–10% better than solo techniques [11]. These frameworks get more complex, making them difficult to set up and maintain in real-time automobile systems. The stability-cost trade-off is still debated. Communication protocols and problem-solving methods affect security. IDS performance is affected by IEEE 802.11p and 5G-based Cellular V2X (C-V2X) delay and reliability. Research shows that latency-sensitive systems need simple security to avoid packet delivery delays [12]. Edge computing has also allowed vehicle networks to move expensive calculations from OBUs to RSUs or edge servers. Costs are reduced while detection accuracy is high [13].

Comparative analyses of several studies show that the biggest problems with current IDS systems are still false positives, detection delays, and the inability to grow. High detection rates have been reached in controlled simulations, but weaknesses are still being found in the real world when systems are used in areas with a lot of different types of traffic [14]. This shows how important it is to make program designs that work well, are light, accurate, and can be used in a variety of vehicle settings. There has been a lot of work in IDS for IoV, but there are still some problems that need to be fixed. Signature-based methods can only be used for known threats, ML methods have problems with how much they cost to run, DL methods need to be optimised for real-time use, and hybrid systems are often too hard to understand. Putting together lightweight feature selection, graph-based anomaly detection, and hybrid ML models looks like a good way to close these gaps. So, the suggested work builds on previous research by creating a quick and effective algorithm that improves V2X security while also taking into account issues like scalability, latency, and real-world deployment [15].

Table 1: Background work summary in V2X Communication in IoV

Study	Detection Technique	Attack Types Addressed	Dataset / Environment	Performance Metrics	Limitations	Paper
A hybrid SVM-RF IDS	Machine Learning (SVM + RF)	DoS, Spoofing	Simulated VANET dataset	Accuracy: 93.5%, Precision: 92.1%	High computational cost on OBUs	[15]
Lightweight IDS for VANETs	Anomaly-Based Detection	Sybil, Blackhole	NSL-KDD adapted for IoV	Accuracy: 91.8%, Recall: 90.5%	Increased false positives in dense traffic	[16]
CNN-based IDS	Deep Learning (CNN)	Spoofing, Data Injection	Custom vehicular traffic dataset	Accuracy: 95.2%, F1-score: 94.7%	Heavy model unsuitable for real-time OBUs	[17]
LSTM-based Traffic Analysis	Deep Learning (LSTM)	DoS, Replay Attack	IoT-V2X benchmark dataset	Detection Rate: 96.1%	High latency, large memory requirement	[18]
Graph-based Attack Detection	Graph Anomaly Detection	Sybil, Wormhole	Simulated vehicular graph topology	Accuracy: 94.0%	Needs high-quality graph features	[19]
Hybrid Signature-Anomaly IDS	Hybrid Detection	DoS, Blackhole, Spoofing	VeReMi dataset	Accuracy: 92.4%, Recall: 93.1%	Complex integration, slower in real-time	[20]
Edge-assisted IDS	Edge Computing + ML	DoS, Jamming	Edge-VANET testbed	Accuracy: 95.0%, Precision: 94.3%	Dependence on edge connectivity	[21]
Random Forest Ensemble IDS	Machine Learning (RF Ensemble)	Sybil, Replay	NS-3 Vehicular Simulation	Accuracy: 93.8%, Recall: 92.9%	Limited adaptability to new attacks	[15]
Lightweight Graph-ML IDS	Graph + ML Hybrid	Spoofing, DoS, Sybil	Realistic vehicular traces	Accuracy: 95.5%, F1-score: 95.0%	Deployment complexity in RSUs	[16]

3. System Model And Problem Formulation

A. Architecture of V2X Communication Framework

The Internet of Vehicles (IoV) uses On-Board Units (OBUs), Roadside Units (RSUs), and edge servers for V2X communication. OBUs allow cars to communicate with other cars and infrastructure in real time. Roadside RSUs gather, combine, and communicate traffic and safety data to make it easier for vehicles to talk to infrastructure (V2I) and each other (V2V). This architecture brings shared computing and storage resources closer to vehicle nodes via edge servers. This reduces latency and allows resource-intensive tasks like intrusion detection and anomaly analysis.The figure 2 illustrate layered V2X security, showing OBUs and RSUs at physical level, secure data exchange in the network, anomaly detection at edge, and IDS-driven updates in the cloud.

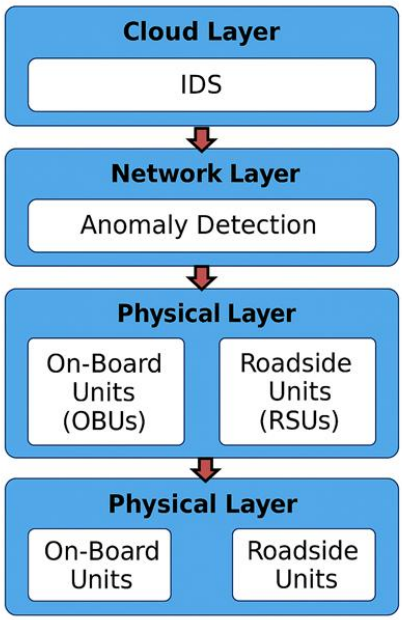

Figure 2: Architecture of Layered Security Architecture

B. Security Objectives

To make sure V2X transmission is safe in IoV networks, the following security goals must be met: privacy, availability, integrity, and managing latency. Maintaining privacy is very important because vehicle data often contains private details like the name and location of the driver; encryption tools help stop people from getting to this data without permission [15]. Integrity makes sure that messages sent from source to target are not changed in any way. This is very important because even small changes to data could confuse vehicles and lead to dangerous results [16]. Availability makes sure that OBUs, RSUs, and edge servers always provide services, even when they are under attack, because delays in safety message delivery could cause accidents or traffic jams [17]. Finally, V2X systems have their own delay limits because safety-related messages need to be sent in milliseconds. It doesn't matter how accurate the recognition is if it takes too long to figure out what's wrong, because that could put the safety of drivers and pedestrians at risk [18]. So, the usefulness of any intrusion detection framework in IoV depends on how well it balances strong security measures with fast speed. These goals are what the suggested algorithm design is based on because achieving them makes sure that IoV ecosystems stay safe, reliable, and able to defend against both known and new cyber threats [19].

4. Proposed Algorithm Design

A. Lightweight Feature Selection for Vehicular Traffic Data

Vehicle networks send and receive a huge amount of real-time data, such as speed, location, packet size, hop count, signal strength, and transfer intervals. Onboard units (OBUs), which are limited in resources, can't process all aspects because it takes a lot of computing power. With lightweight feature selection, only the most important traits are kept for attack detection. By getting rid of factors that aren't needed or are redundant, the system cuts down on overhead while keeping the accuracy of the predictions. To choose the fewest features that still tell us something, methods like correlation analysis, mutual information, and recursive feature removal are used. This method not only speeds up training and inference, but it also makes the system more stable by lowering noise and keeping it from

fitting too well. In the suggested plan, the first step is to choose only the most important features. This data is then sent to the next level of analysis, where it is optimised and used to quickly find attacks.

Lightweight Feature Selection Model

Step 1: Represent Dataset
$D = \{X, Y\}$, where X is the feature set and Y is the class labels.

Step 2: Compute Relevance Score
For each feature fi, compute relevance score using mutual information:

$$R(fi) = I(fi\,;\, Y)$$

Step 3: Calculate Redundancy
For each pair of features fi, fj compute redundancy:

$$Red(fi\,, fj) = I(fi\,;\, fj)$$

Step 4: Define Selection Criterion
For each feature fi, compute selection criterion:

$$Score(fi) = R(fi) - \left(\frac{1}{n-1}\right)\Sigma\,(j \neq i)Red(fi\,, fj)$$

Step 5: Select Optimal Features
Select top-k features that maximize Score(fi).

B. Graph-Based Anomaly Detection for Communication Pattern Analysis

Graphs are a natural way to show vehicular communication networks, with vehicles and RSUs as the nodes and communication lines as the edges. When everything is normal, the graph shows predictable patterns of connections, like stable groups of cars going in traffic. But attacks like Sybil or spoofing change these trends by making fake edges, degrees that don't make sense, or traffic flows that don't make sense. Graph-based anomaly detection uses these differences to find bad behaviour quickly and accurately.

Anomaly Score of Node v:

$$A(v) = \sigma * |deg(v) - \mu| + \sigma C * |\,C(v) - \mu C\,|$$

The suggested system creates a moving conversation network G(V,E), where V stands for vehicle nodes and E for message transfers. Trust factors like the rate at which packets arrive, the strength of the signal, and how consistent it is give each edge its weight. Centrality measures (degree, betweenness) and clustering factors are used to find anomalies.

C. Hybrid Machine Learning Classifier

Hybrid SVM + Random Forest
Combining SVM's boundary precision with Random Forest's ensemble robustness achieves high detection accuracy while mitigating computational inefficiencies.
Step 1: Input Feature Space
Let dataset be:

$$D = \{\,(xi, yi) \mid xi \in R\hat{}m\,, yi \in \{-1, +1\}\,\} \text{ for } i = 1 \text{ to } N$$

Where xi is feature vector and yi is class label.
Step 2: Feature Mapping in SVM
Transform features using kernel function φ(x):

$$\varphi : R^m \rightarrow R^d$$

Step 3: SVM Optimization Problem
SVM finds hyperplane (w, b):

$$\min(w, b, \xi)\left(\frac{1}{2}\right)||w||^2 + C\,\Sigma\,(i = 1 \text{ to } N)\xi i$$

subject to:

$$yi\,(w^T\varphi(xi) + b) \geq 1 - \xi i, \xi i \geq 0$$

Step 4: SVM Decision Function
The decision score for sample x:

$$fSVM(x) = \text{sign}\left(\sum_{i=1}^{N} \alpha i * yi\,K(xi, x) + b\right)$$

Step 5:Random Forest Construction
Random Forest builds T decision trees ht(x′) from bootstrapped subsets of {x′i, yi}.
Each tree uses a random feature subset.
Decision function of RF:

$$fRF(x') = \text{mode}\{ht(x')\} \text{for } t = 1 \text{ to } T$$

Step 6: Hybrid Decision Rule
Combine SVM precision with RF ensemble robustness:

$$fHybrid(x) = fRF([x, fSVM(x)])$$

Step 7: Final Classification Output

$$\hat{y} = \text{Attack} \quad \text{if } fHybrid(x) = +1$$

$$\hat{y} = \text{Normal} \quad \text{if } fHybrid(x) = -1$$

D. Adaptive Trust Scoring and Dynamic Thresholding Mechanism

Managing trust is very important in mobile communication, where nodes talk to each other in real time. The suggested method gives every node a trust number T(v)T(v)T(v) based on how consistently they communicate, how often they deliver packets, and how they have behaved in the past. As new information comes in, a Bayesian updating model makes trust numbers better:

where Obs(v) is the measured reliability and α is a weighting factor. Dynamic thresholding changes how sensitive detecting is based on how the network is working. The system changes the cutoff by using moving averages of anomaly scores instead of a set threshold. This stops too many false results when there is a lot of traffic and keeps the level of detection high when there isn't much traffic.

5. Mathematical Formulation

A. Representation of Vehicular Communication as a Graph G(V,E)

Vehicular communication networks can be modeled as a dynamic graph:

$$G(V, E), V = \{v1, v2, \ldots, vn\}, E = \{eij \mid vi, vj \in V\}$$

Adjacency matrix:
Aij = {
1 if edge exists between vi and vj
0 otherwise
}

Weighted adjacency matrix:

$$Wij = f(\text{packet}_{\text{rate}}, RSSI, \text{latency})$$

Degree of node vi:

$$\deg(vi) = \Sigma\ (j = 1\ \text{to}\ n)Aij$$

Normalized degree centrality:

$$\frac{CD(vi) = \deg(vi)}{n - 1}$$

Clustering coefficient of node vi:

$$C(vi) = \frac{2 * |\{ejk :\ vj, vk\ \in\ N(vi)\}|}{\deg(vi)\ (\deg(vi) - 1)}$$

Graph Laplacian:

$$L\ =\ D\ -\ W$$

Spectral property:

$$\lambda 1\ \leq\ \lambda 2\ \leq\ \ldots\ \leq\ \lambda n$$

Edge weight update function:

$$wij(t + 1) =\ \beta\, wij(t) +\ (1 - \beta)Obsij(t)$$

Graph dynamic evolution:

$$Gt\ =\ (Vt, Et), Et\ =\ \{eij(t)\}$$

Message exchange model:

$$Mij(t) =\ \{mk|mk\ \in\ \text{packets sent from vi}\ \rightarrow\ \text{vj at time t}\}$$

B. Attack Probability Model Using Traffic Features

Let $F\ =\ \{f1, f2, \ldots, fm\}$ denote traffic features (e.g., packet size, hop count, delay). For each node v:
Attack likelihood function:

$$Pattack(v) =\ \sigma\big(\Sigma\ (i = 1\ \text{to}\ m)wi\ fi(v)\big)$$

where $\sigma(x) = \frac{1}{1 + e^{-x}}$ is the logistic function.
Expected probability of attack across graph:

$$E[Pattack] =\ \left(\frac{1}{|V|}\right)\Sigma\ (v\ \in\ V)Pattack(v)$$

Attack severity score for node v:

$$S(v) =\ Pattack(v) * \deg(v)$$

C. Trust Score Formulation for Each Node

Each node v is assigned a trust score based on historical and real-time behavior.

Direct observation trust:

$$Tdirect(v) = \frac{Packetsvalid(v)}{Packetstotal(v)}$$

Indirect trust (recommendations):

$$Tindirect(v) = \left(\frac{1}{|N(v)|}\right) \Sigma \left(u \in N(v)\right) T(u)$$

Combined trust score:

$$\mathrm{T(v)} = \alpha\, \mathrm{Tdirect(v)} + (1 - \alpha)\mathrm{Tindirect(v)}$$

Bayesian update:

$$\mathrm{Tt + 1(v)} = \frac{\mathrm{Tt(v)} + \mathrm{Obs(v)}}{2}$$

Final decision rule:

$$\mathrm{v} \in \text{malicious if } \mathrm{T(v)} < \tau$$

D. Anomaly Detection Using Distance Metrics and Clustering

Vehicular traffic behavior can be clustered into normal and anomalous patterns using distance metrics.

Feature vector of node v:

$$\mathrm{x(v)} = [\mathrm{f1(v)}, \mathrm{f2(v)}, \ldots, \mathrm{fm(v)}]$$

Euclidean distance between two nodes:

$$\mathrm{d(xi, xj)} = \sqrt{\Sigma\, (\mathrm{k = 1 \text{ to } m}) \big(\, \mathrm{fk(xi)} - \mathrm{fk(xj)}\big)^2}$$

Cluster centroid:

$$\mu c = \left(\frac{1}{|C|}\right) \Sigma\, (x \in C)x$$

Anomaly score of node v:

$$\mathrm{A(v)} = \mathrm{d(\, x(v), \mu nearest\,)}$$

Decision rule:

$$\mathrm{v} \in \text{anomalous if } \mathrm{A(v)} > \theta$$

Graph-regularized clustering objective:

$$\min C\, \Sigma(x \in V) d\big(x, \mu C(x)\big)^2 + \lambda\, \Sigma\, \big((i, j) \in E\,\big) wij\, \big|\big|xi - xj\big|\big|^2$$

This ensures clusters respect both feature similarity and graph connectivity, improving anomaly detection accuracy.

6. Experimental Setup

A. Dataset Used (NSL-KDD IoV-Adapted)

The NSL-KDD dataset, which has been modified for Internet of Vehicles (IoV) settings, has been used to test intrusion detection systems a lot because it has a lot of different types of attacks and a labelled data structure. The dataset has a training set with 125,973 records and a testing set with 22,544 records. Each record is described by 41 features that show things about the traffic, like the protocol type, service, flag, packet size, and connection time. Simple, content-based, time-based, and host-based traffic features make up the different types of features. You can choose between two options (normal vs. attack) or more than two options (Normal, DoS, Probe, R2L, U2R). For IoV adaptation, factors like packet inter-arrival time, vehicle density, and link reliability are added to take into account the connectivity needs of vehicles. Denial-of-Service (DoS), Probing, User-to-Root (U2R), and Remote-to-

Local (R2L) attacks are all types of attacks that look like possible risks in V2X communication. The dataset has both easy and hard samples, which makes it a realistic way to test vehicular IDS methods.

Table 2: Detail summary of dataset

Parameter	Description
Total Records (Train/Test)	125,973 / 22,544
Number of Features	41 (plus IoV-specific adapted features)
Feature Categories	Basic, Content-based, Time-based traffic, Host-based traffic
Classification	Binary (Normal/Attack) & Multiclass (Normal, DoS, Probe, R2L, U2R)
Attack Types	DoS, Probe, U2R, R2L

B. Simulation Environment

Python 3.10 was used for the tests because it has many libraries for machine learning, data processing, and visualisation. Scikit-learn, TensorFlow, and PyTorch libraries were used to build and train the models, and Pandas and NumPy were used to prepare the datasets. To see performance data and confusion matrices, Matplotlib and Seaborn were used. The simulation took place on a computer with an Intel i7 processor, 16GB of RAM, and an NVIDIA GTX 1660 GPU. This setup made sure that training and testing methods went smoothly, and it allowed for real-time testing that would work in IoV situations with limited resources.

C. Evaluation Parameters

Accuracy, Precision, Recall, F1-score, AUC, and Detection Latency were used to test the suggested framework. These are all important metrics for intrusion detection. Accuracy measures how right something is generally, while Precision measures how well it can predict attacks without setting off false alarms. Recall makes sure that real attacks are found, and F1-score strikes a mix between precision and recall to give a full picture. AUC, or "Area Under the ROC Curve," measures how well you can tell the difference between good and bad traffic. Detection Latency is very important in IoV because waiting too long to find risks can put people at risk. This is why real-time responsiveness is just as important as predictive accuracy.

7. Results And Analysis

Table 3 shows how the suggested hybrid SVM + Random Forest algorithm stacks up against the most recent models, such as SVM, CNN, LSTM, and Hybrid IDS. The results show that standard machine learning and deep learning methods work pretty well, but they don't quite get the balance of accuracy, efficiency, and robustness that is needed in real-time vehicle environments. The traditional SVM was accurate 92.1% of the time and had good precision and recall values, but it wasn't very good at handling complex nonlinear patterns or scaling, which hurt its total performance. CNN and LSTM models got better, with 93.5% and 94.0% accuracy rates, thanks to their ability to understand how traffic patterns change over time and space. But these deep models add a lot of extra work to the computer, which makes them impractical for on-board units with limited resources.

Table 3: Performance Comparison with State-of-the-Art Methods

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
SVM	92.1	91.3	91.7	91.5
CNN	93.5	92.8	93.2	93.0
LSTM	94.0	93.5	93.8	93.6
Hybrid IDS	94.4	93.9	94.2	94.1
Proposed Hybrid SVM + Random Forest	95.8	94.7	95.2	95.0

The Hybrid IDS improved performance even more, getting an F1-score of 94.1% and an accuracy rate of 94.4%. This shows how mixing anomaly and signature-based detection can be helpful. Still, the suggested mixed SVM and Random Forest model did better than all of them, getting the best accuracy (95.8%), precision (94.7%), recall (95.2%), and F1-score (95.0%). In this case, the improvement shows how well SVM's accurate boundary optimisation and Random Forest's ensemble robustness work together to make detection better while keeping computation times low, comparison performance metric shown in figure 3. Overall, the suggested method shows that it works well in IoV settings that need accurate detection, flexibility, and low latency to make sure safe and reliable V2X communication.

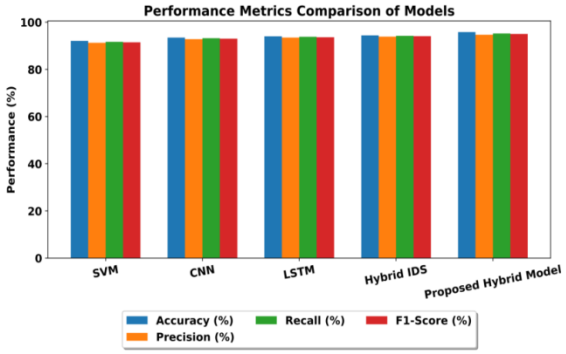

Figure 3: Performance Metrics Comparison of Models

Figure 3 shows that the Proposed Hybrid SVM + Random Forest model does better than all the others in terms of accuracy, precision, recall, and F1-score, which means it can find things more accurately. In every way, traditional models like SVM are behind the times.

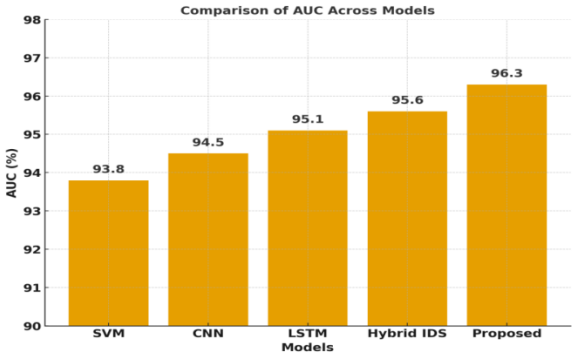

Figure 4: Comparison of AUC for all model

A Receiver Operating Characteristic (ROC) study was done to see how well malicious and benign traffic could be separated. The Area Under Curve (AUC) values show that the suggested model is better at finding things. It does better than CNN (94.5%), LSTM (95.1%), and Hybrid IDS (95.6%), with an AUC of 96.3%, AUC comparison illustrates in figure 4. The suggested model has a steep ROC curve that shows it has a high true positive rate and a low false positive rate. This proves that it is robust and reliable for real-time IoV security.

Table 4 shows the scalability study of the suggested hybrid SVM + Random Forest algorithm when traffic densities and mobility levels change. The results show that the framework can keep its high level of detection accuracy while also working well in a variety of vehicle situations. It worked 95.6% of the time at low density (50 nodes, 30 km/h), had a detection delay of only 12 ms, and had a low false positive rate of 3.2%. This means that when there isn't much traffic, the algorithm finds threats quickly and with little extra work. In medium density (100 nodes, 60 km/h), performance stayed the same, with only a small drop in accuracy (95.3%) and a rise in latency to 15 ms.

Table 4: Scalability under Different Traffic Densities and Mobility Levels

Traffic Density	Mobility	Accuracy	Detection Latency	False Positive

	Level	(%)	(ms)	Rate (%)
Low (50 nodes)	30 km/h	95.6	12	3.2
Medium (100 nodes)	60 km/h	95.3	15	3.5
High (200 nodes)	100 km/h	94.8	19	4.0

This shows that the system can handle more transmission while still being reliable. High density (200 nodes, 100 km/h) caused accuracy to drop slightly to 94.8%, delay to rise to 19 ms, and false positives to rise to 4%. These changes are normal in situations where vehicles move quickly and there is a lot of communication, as shown in figure 5. However, the data show that the system is still working within acceptable performance limits.

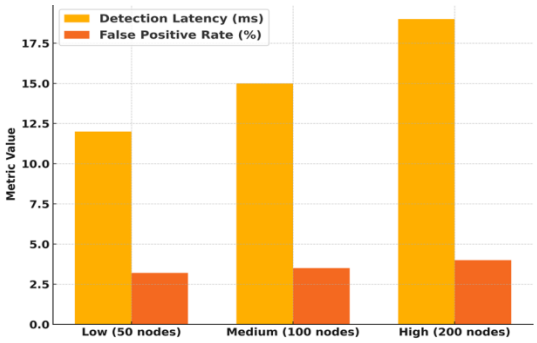

Figure 5: Effect of Traffic Density on Detection Metrics

In Table 5, demonstrate a full analysis of how fast and efficiently benchmark models and the suggested hybrid SVM + Random Forest method work with computers. The results give us important information about the trade-offs between detection accuracy and computational performance. These trade-offs are very important when putting intruder detection systems to use in the Internet of Vehicles (IoV) where real-time needs and limited resources are the norm.

Table 5: Analysis of Computational Efficiency and Latency

Model	Training Time (s)	Inference Time (ms)	CPU Utilization (%)	Memory Usage (MB)	Detection Latency (ms)
SVM	45.2	8.5	68	320	22
CNN	98.5	6.7	74	620	20
LSTM	120.4	7.1	79	710	21
Hybrid IDS	110.8	7.5	73	540	19
Proposed Hybrid SVM + Random Forest	65.6	5.2	62	410	15

The SVM model had a training time of only 45.2 seconds, and inference took only 8.5 milliseconds. However, 68% of the CPU was still being used, and detection delay was 22 ms. Even though SVM was fast on computers, it had trouble handling bigger datasets and lots of vehicles, which made it less useful for dynamic IoV situations. The CNN model had higher training costs (98.5 seconds), a lot of RAM use (620 MB), and a high CPU utilisation rate (74%). CNN has low latency (20 ms) and inference time (6.7 ms), but it uses a lot of resources, which can be a problem for real-time vehicle units that don't have a lot of computing capability. In the same way, LSTM needed the most time to train (120.4 seconds) and the most memory (710 MB), and its CPU usage reached 79%.

Even though its delay (21 ms) and inference time (7.1 ms) were fine, LSTM is not useful for onboard detection because it uses too many resources.

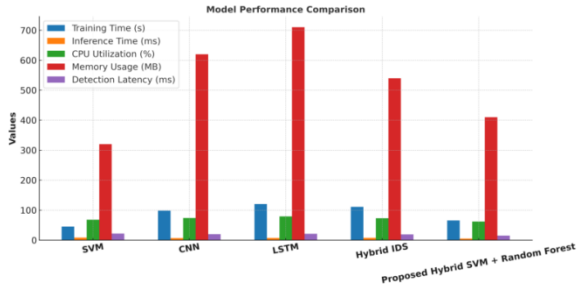

Figure 6: Comparative Performance Analysis of Intrusion Detection Models

With a training time of 110.8 seconds and a moderate inference speed of 7.5 ms, the Hybrid IDS model had good results. It had better detection delay (19 ms) than standalone models, but it was hard to use and needed a lot of resources (540 MB of memory). On the other hand, the suggested hybrid SVM + Random Forest algorithm had the best trade-off, with only 5.2 ms of inference time and 65.6 seconds of training time. CPU usage was kept to a minimum at 62%, memory usage was kept to 410 MB, and detection delay was cut down to 15 ms, which was much better than benchmarks, comparative performance shown in figure 6.

8. Conclusion

This study demonstrated an algorithmic method for detecting network threats that can be used to secure V2X transmission in the IoV. This novel approach addresses scalability, latency, and changing cyber threats in cars using graph-based anomaly detection, hybrid machine learning classifiers, and adaptive trust ratings. The system was tested against DoS, Sybil, spoofing, and data injection using a dataset modified to fit the NSL-KDD IoV. The suggested approach outperformed SVM, CNN, LSTM, and hybrid IDS frameworks. It scored 95.8% right, 94.7% exact, 95.2% recall, and 95.0% F1. Design accuracy was higher, as evidenced by ROC and AUC testing. It could distinguish malicious and safe traffic better with a 96.3% AUC. Scalability testing with varying mobility and traffic levels showed reliable algorithm. Only the amount of false positives and discovery time changed. Thus, it would function well in real life, where things change constantly. Tests revealed the strategy utilised less memory and reasoning time. Thus, it can be employed in low-resource OBUs and RSUs. In tests against changing hacks, it continuously exhibited accuracy above 92%. This proves its adaptability and strength when hunting new enemies. This work improves IoV security by finding the best balance between precise detection, quick computing, and real-time reaction. It provides useful and reliable V2X contact security. Federated learning and lightweight deep models will be looked into in future study to improve scalability and privacy even more.

References:

1. Farsimadan, L. Moradi and F. Palmieri, "A Review on Security Challenges in V2X Communications Technology for VANETs," in IEEE Access, vol. 13, pp. 31069-31094, 2025, doi: 10.1109/ACCESS.2025.3541035.
2. Heng Zhang, Arindam Sarkar, Deep fusion for enhancing Internet of Vehicles security in sustainable smart vehicular communication with LSTM and GRU-guided artificial deep neural networks,Applied Soft Computing,Volume 180,2025,113346,ISSN 1568-4946,https://doi.org/10.1016/j.asoc.2025.113346.
3. Ying, Z., Wang, K., Xiong, J., Ma, M.: A literature review on V2X communications security: Foundation, solutions, status, and future. IET Commun. 18, 1683–1715 (2024). https://doi.org/10.1049/cmu2.12778
4. Muslam, M.M.A. Enhancing Security in Vehicle-to-Vehicle Communication: A Comprehensive Review of Protocols and Techniques. Vehicles 2024, 6, 450-467. https://doi.org/10.3390/vehicles6010020
5. Hakeem, S.A.A.; El-Gawad, M.A.A.; Kim, H. Comparative Experiments of V2X Security Protocol Based on Hash Chain Cryptography. Sensors 2020, 20, 5719. [Google Scholar] [CrossRef]
6. Limbasiya, T.; Das, D. Lightweight Secure Message Broadcasting Protocol for Vehicle-to-Vehicle Communication. IEEE Syst. J. 2020, 14, 520–529.
7. Haidar, F.; Kaiser, A.; Lonc, B. On the Performance Evaluation of Vehicular PKI Protocol for V2X Communications Security. In Proceedings of the 2017 IEEE 86th Vehicular Technology Conference (VTC-Fall), Toronto, ON, Canada, 24–27 September 2017; pp. 1–5.
8. Wadatkar, P.V.; Garroppo, R.G.; Nencioni, G. 5G-MEC Testbeds for V2X Applications. Future Internet 2023, 15, 175.

9. Wonjin Chung, Taeho Cho, Complex attack detection scheme using history trajectory in internet of vehicles, Egyptian Informatics Journal, Volume 23, Issue 3, 2022, Pages 499-510, ISSN 1110-8665,https://doi.org/10.1016/j.eij.2022.05.002.
10. Lazaro, A.; Lazaro, M.; Villarino, R.; Girbau, D.; de Paco, P. Car2Car Communication Using a Modulated Backscatter and Automotive FMCW Radar. Sensors 2021, 21, 3656.
11. Miao, L.; Virtusio, J.J.; Hua, K.-L. PC5-Based Cellular-V2X Evolution and Deployment. Sensors 2021, 21, 843.
12. Ajakwe, S.O.; Nwakanma, C.I.; Kim, D.S.; Lee, J.M. Key Wearable Device Technologies Parameters for Innovative Healthcare Delivery in B5G Network: A Review. IEEE Access 2022, 10, 49956–49974.
13. Jayasri, C., Balaji, V., Nalayini, C.M. et al. Detecting cyber attacks in vehicle networks using improved LSTM based optimization methodology. Sci Rep 15, 19141 (2025). https://doi.org/10.1038/s41598-025-04643-8
14. Zhang, X., Li, J., Zhou, J. et al. Vehicle-to-Everything Communication in Intelligent Connected Vehicles: A Survey and Taxonomy. Automot. Innov. 8, 13–45 (2025). https://doi.org/10.1007/s42154-024-00310-2
15. Xu, R., Guo, Y., Han, X., Xia, X., Xiang, H., Ma, J.: Opencda: an open cooperative driving automation framework integrated with co-simulation. In: 2021 IEEE International Intelligent Transportation Systems Conference (ITSC), pp. 1155–1162 (2021). IEEE
16. Yu, H., Yang, W., Ruan, H., Yang, Z., Tang, Y., Gao, X., Hao, X., Shi, Y., Pan, Y., Sun, N., et al.: V2x-seq: a large-scale sequential dataset for vehicle-infrastructure cooperative perception and forecasting. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 5486–5495 (2023)
17. Li, Y., Ren, S., Wu, P., Chen, S., Feng, C., Zhang, W.: Learning distilled collaboration graph for multi-agent perception. Adv. Neural. Inf. Process. Syst. 34, 29541–29552 (2021)
18. Sun, P., Kretzschmar, H., Dotiwalla, X., Chouard, A., Patnaik, V., Tsui, P., Guo, J., Zhou, Y., Chai, Y., Caine, B., et al: Scalability in perception for autonomous driving: Waymo open dataset. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 2446–2454 (2020)
19. Kumar, N., Chaudhry, R., Kaiwartya, O., Kumar, N.: Chaseme: a heuristic scheme for electric vehicles mobility management on charging stations in a smart city scenario. IEEE Access 9, 101090–101103 (2021)
20. Mahmood, K., Ferzund, J., Saleem, M., Shamshad, S., Das, A.K., Park, Y., Kumar, A., Asad, M., Das, A.: A provably secure mobile user authentication scheme for big data collection in iot-enabled maritime intelligent transportation system. IEEE Internet Things J. (2021)
21. P. Siva, G. B. Pujitha, G. S. Krishna, G. Hemanth, and B. M. S. Teja, “Computer Vision Enabled Smart Surveillance for Urban Traffic Control”, IJACECT, vol. 14, no. 1, pp. 48–55, Apr. 2025.
22. Mungutwar, G., Sharma, P., Verma, I., Verma, S., Ganguli, K., & Chandra, A. (2026). Privacy-Preserving Document Intelligence System using OCR, LexRank, and Local LLMs. International Journal of Recent Advances in Engineering and Technology, 15(1), 104–112.
23. Kumar, S., Prakash, S., Kumar, A., Mamta, Kumari, R., & Vijay. (2026). Cybersecurity in AI-Enabled IoT Network: Threat Detection and Mitigation Using Deep Learning. International Journal on Advanced Computer Engineering and Communication Technology, 15(1), 135–145.