IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 15 February 2024, accepted 20 March 2024, date of publication 29 March 2024, date of current version 9 April 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3382992

RESEARCH ARTICLE

PREVIR: Fortifying Vehicular Networks Against Denial of Service Attacks

AMANDEEP VERMA[1], RAHUL SAHA[2,3], (Member, IEEE), GULSHAN KUMAR[2,3], MAURO CONTI[2], (Fellow, IEEE), AND TAI-HOON KIM[4], (Member, IEEE)

[1]School of Computer Applications, Lovely Professional University, Phagwara, Punjab 144411, India
[2]Department of Mathematics, University of Padua, 35131 Padua, Italy
[3]School of Computer Science and Engineering, Lovely Professional University, Phagwara, Punjab 144411, India
[4]School of Electrical and Computer Engineering, Chonnam National University, Yeosu Campus, Yeosu-si, Jeollanam-do 59626, Republic of Korea

Corresponding authors: Rahul Saha (rsahaaot@gmail.com) and Tai-Hoon Kim (taihoonn@daum.net)

ABSTRACT Vehicular networks are expanding their applications for future sustainability. The reported increasing rate of data breaches through vehicular networks by Distributed Denial of Service (DDoS) type of intrusion creates concern for such networks. Existing security solutions focus only on intrusion detection. However, prevention solutions are more proactive and provide security by probabilistic analysis. Existing prevention models for vehicular networks have low accuracy and are unable to handle zero-day attacks and advanced persistent threats. In this paper, we solve the problems mentioned above and introduce *Predictive Risk Evaluation for Vehicular Infrastructure Resilience (PREVIR)*, the first amalgamated model of logit method (statistical analysis) and LogitBoost method (machine learning) to prevent DDoS attacks in vehicular networks. In PREVIR, the logit model predicts the packet probabilities for identifying maliciousness. The machine learning method improves PREVIR's performance through iterative refinement of the model's periodic updates based on new traffic parameters. We run a set of experiments on PREVIR. We use our NS3-generated dataset, NSL-KDD public dataset, and CIC-DDoS public dataset. PREVIR analyses multiple attack types, including UDP flood, TCP flood, mixed flooding, U2R, Probe, and R2L attacks. The results show that PREVIR classifies packets with accuracy up to 99.99%. Our proposed PREVIR model achieves a True Positive Ratio (TPR) up to 100% and an average False Positive Ratio (FPR) of 35%. The comparative analysis shows that PREVIR's efficiency is 20% better on average in the prevention of malicious packets as compared to the state-of-the-art models.

INDEX TERMS Vehicular, VANETs, DDoS, attack, prevention.

I. INTRODUCTION

The proliferation of technologies in the Internet of Things (IoT) has expanded its applicability across various domains. IoT applications have found their way into smart human wearable devices, smart homes and offices, smart industries, supply chains, and vehicular networks [1]. Vehicular Adhoc Networks (VANETs) consisting of connected vehicles with vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communication technology have gained significant momentum within the IoT ecosystem, particularly in the realm of Intelligent Transport Systems (ITS) [2]. However, the rapid growth of VANETs also brings inherent security challenges. Among the security concerns, Denial of Service (DoS) attacks and their distributed form, i.e., DDoS attacks, pose significant threats to VANETs. Thus, vehicular networks of VANETs need a prevention system that can calculate packet risk.

To address the research gap mentioned above, we propose a DDoS attack prevention system named "Predictive Risk Evaluation for Vehicular Infrastructure Resilience" (PREVIR). Our framework utilizes a collaborative approach integrating the statistical logit model with the machine learning method. By combining statistical and machine learning techniques, PREVIR aims to provide an effective preventive solution to mitigate DDoS attacks in VANETs.

The associate editor coordinating the review of this manuscript and approving it for publication was Hassan Omar.

The ML technique used in PREVIR enhances efficiency using iterative refinement of the model's periodic updates based on new traffic parameters. The rationale behind the iterative refinement process of PREVIR lies in its ability to adapt to evolving threats, improve performance through machine learning, and validate its effectiveness through comprehensive experimentation. By continuously updating the model based on new traffic parameters and analyzing its performance across diverse datasets and attack types, PREVIR aims to be a robust and adaptable early threat prevention system for identifying and mitigating malicious activities in network environments. Leveraging the essence of the logit model, PREVIR employs a probability-based approach to assess the likelihood of an attack event occurring for a given packet. Higher probability values indicate an increased risk of an attack. Through this research, we strive to contribute to developing robust prevention systems designed explicitly for VANETs. In this paper, we first delve into the understanding of DDoS attacks and their impact on VANETs, along with an overview of general defense mechanisms against such attacks. We also summarize the contribution and highlight the novelty of PREVIR.

A. DDOS IN VANETS

VANETs communicate using Dedicated Short Range Communication (DSRC) and the Internet. Due to a lack of standard protocols, DoS/DDoS attacks can easily disturb V2V and V2I communications. There are different types of attacks that disrupt service availability. Sleep deprivation attack keeps the victim node alive until all its battery power is consumed [3], [4]. Jamming attack jams radio signals [4], [5]. Jellyfish attack inserts time delays, and packets are dropped [3], [6]. Intelligent cheater attack changes the behavior of a node from normal to malicious [6]. Blackhole attacks [7] and gray hole attacks work by dropping packets [8]. In a greedy behavior attack, the attacker targets the MAC layer [9]. The target of a DoS/DDoS attack may be a vehicle, information, or any infrastructural unit. We show a general VANET architecture with an attack scenario in Figure 1.

VANETs are susceptible to Distributed Denial of Service (DDoS) attacks due to their unique characteristics and communication architecture. VANETs facilitate communication between vehicles and infrastructure to enhance road safety, and traffic efficiency, and provide various intelligent transportation services. The decentralized nature of VANETs, where vehicles communicate directly with each other or with roadside infrastructure, makes them vulnerable to DDoS attacks. These attacks can overwhelm the communication channels with a flood of malicious traffic, disrupting the normal functioning of the network.

One key vulnerability arises from the wireless communication protocols, such as IEEE 802.11p, which lack inherent security features. Attackers can exploit this by launching DDoS attacks that flood the communication channels with bogus messages, leading to network congestion and potentially blocking legitimate communication. Moreover, the highly dynamic nature of vehicular environments, with vehicles moving at high speeds and frequently joining or leaving the network, poses challenges for traditional security solutions.

Existing security solutions for VANETs often struggle to address these challenges effectively. Traditional intrusion detection systems may face difficulties in distinguishing between normal and malicious traffic in real time, especially when dealing with the intermittent connectivity and rapidly changing network topology of VANETs. Furthermore, resource constraints, such as limited processing power and memory in onboard vehicular units, pose challenges in implementing sophisticated security measures. Ensuring the authenticity of messages in a decentralized and dynamic environment is another hurdle, as existing cryptographic solutions may struggle to provide efficient and timely verification.

B. GENERAL DDOS DEFENSE SYSTEM

We classify the overall defense systems of VANETs into four categories: prevention, detection, response, and tolerance. These four categories provide various techniques to handle DDoS attacks.

1) PREVENTION

Prevention mechanisms identify the doubtful attacker and stop the traffic from probable attackers. Prevention takes place at the pre-attack stage. Various strategies are in use for attack prevention: Filtering [10], Ingress Filtering [11], route-based Distributed Packet Filtering (DPF) [12], History-based IP Filtering (HIF) [13], Hop Count Filter (HCF) [14], Secure Overlay Services (SOS) [15], and honeypots [16].

2) DETECTION

It is a process that detects the attacker by observing traffic patterns and various parameters. The researchers have proposed different models for attack detection. The techniques include statistical, soft computing, knowledge-based, and machine learning techniques.

3) RESPONSE

Attack response includes identifying the attacker using traceback techniques and reducing the attack's impact, called mitigation. Traceback methods include controlled flooding, deterministic packet marking (DPM) and link testing, naïve logging, hash-based logging, Flexible Deterministic Packet Marking (FDPM), and Probabilistic Packet Marking (PPM) and messaging techniques.

4) TOLERANCE

We cannot entirely prevent DDoS assaults; hence it is essential to keep the services accessible to legitimate users even when an attack occurs. The two primary techniques,

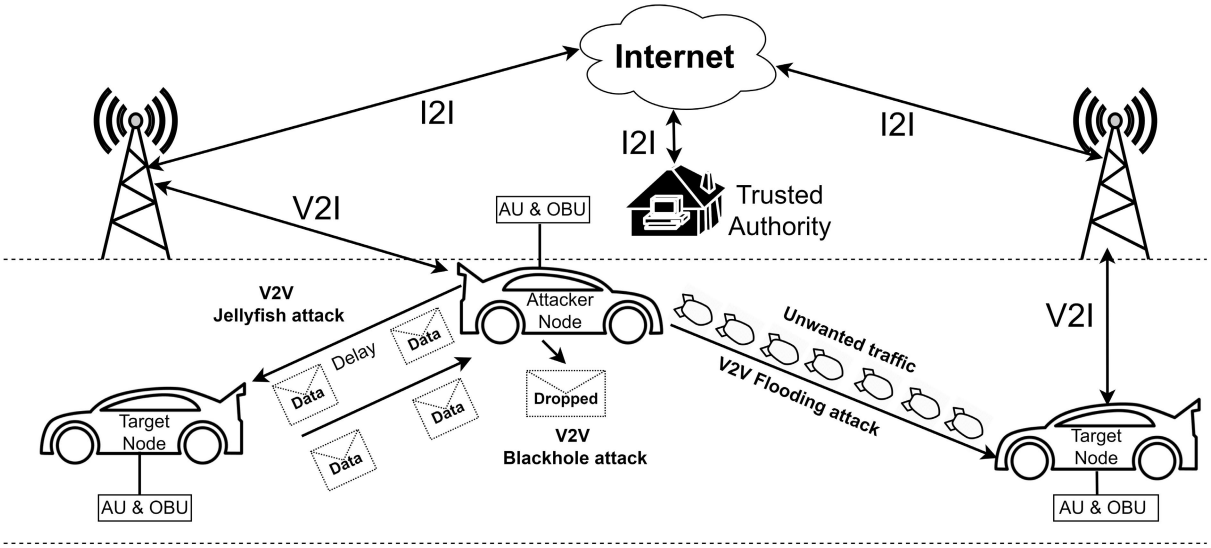


FIGURE 1. VANET architecture with different attacks.

i.e., quality of service and fault tolerance, can be used to establish attack tolerance and mitigation procedures. The other two strategies are throttling and pushback architecture. The network's promising capability is to continue operating as usual when attacked. Attack tolerance concerns preserving system quality of service by reducing jitters, delays, and packet loss. In a nutshell, DDoS handling is possible in the form of detection, prevention, tolerance, or mitigation [17].

C. PAPER CONTRIBUTION AND NOVELTY

Our present work shows a DDoS prevention technique using an integrated logit model. Adopting a combined approach of statistical analysis and machine learning holds significant importance for DDoS attack prevention in vehicular networks due to the dynamic and evolving nature of the threat landscape in these environments. Statistical analysis helps in identifying patterns and anomalies within the network traffic, allowing for the detection of abnormal behaviour that may indicate a potential DDoS attack. This method involves analyzing various network parameters, such as packet rates, traffic volumes, and communication patterns, to establish a baseline of normal behaviour. However, statistical methods alone may struggle to keep pace with the sophisticated and rapidly changing tactics employed by attackers. This is where machine learning comes into play. Machine learning algorithms can adapt and learn from historical data, continuously improving their ability to discern between normal and malicious network behaviour. By leveraging features such as traffic patterns, packet sizes, and source-destination relationships, machine-learning models can identify subtle deviations indicative of DDoS attacks.

The synergy between statistical analysis and machine learning is particularly powerful in the context of vehicular networks. Statistical analysis establishes a foundation for understanding typical network behaviour, while machine learning enhances the system's ability to detect novel and evolving attack patterns that statistical methods alone may struggle to identify. This combined approach enables a more robust and adaptive DDoS prevention system, capable of addressing the challenges posed by the decentralized and dynamic nature of vehicular networks. Moreover, as vehicular networks generate vast amounts of data in real-time, machine learning algorithms can efficiently process and analyze this data, providing timely and effective responses to emerging threats. The integration of statistical analysis and machine learning, therefore, enhances the overall security posture of vehicular networks by offering a comprehensive and dynamic defense mechanism against DDoS attacks. We provide the solution for vehicular networks. The proposed integrated logit model is novel in DDoS prevention in vehicular networks. It provides an accuracy of 100%, which is the best compared to other solutions. We experiment with PREVIR using three datasets and with different features. Our research aims to demonstrate the use of packet properties to prevent DDoS assaults in vehicular networks with high efficiency and accuracy. To this point, our significant contributions are as follows.

1) AMALGAMATION OF STATISTICAL AND ML METHODS

One of the key novelties of PREVIR is pioneering an amalgamation of the statistical logit model with machine learning methods for DDoS prevention in vehicular infrastructure. While logistic regression and ML methods, individually, have been widely used for attack detection, the proposed combination has never been tested in VANETs for DDoS prevention. By leveraging the unique capabilities of the logit model and ML techniques, PREVIR introduces a fresh perspective on DDoS prevention, opening up new avenues

for research and showcasing the untapped potential of the integrated logit model in combating DDoS attacks in VANET.

2) LOGIT MODEL

The combined logit model with statistical methods like the Z-Test, Goodness of Fit, and Hosmer-Lemenshow Goodness of Fit makes it efficient. This model contributes to the research domain by providing a robust and comprehensive analysis framework, enabling a deeper understanding of the relationships between variables and enhancing the validity and reliability of research findings.

3) OVERFITTING AND UNDERFITTING

Existing DDoS prevention systems suffer from the problem of Over and underfitting. To overcome this problem, PREVIR selects the best features through statistical methods and effectively addresses the overfitting and underfitting issues. Furthermore, PREVIR incorporates the LogitBoost ML classifier that deals with fitting problems by handling larger and smaller datasets.

4) ZERO-DAY ATTACKS

Existing DDoS prevention systems often rely on known signatures and patterns. Still, these techniques cannot deal with novel and evolving attack techniques that still need to be identified or documented. Our PREVIR uses the LogitBoost ML classifier to detect and mitigate zero-day attacks by leveraging its ability to learn and generalize patterns and behaviors in network traffic data.

5) EFFICIENT RESULTS

LogitBoost exhibits exceptional accuracy of 99.99%, indicating a high level of correctness in its predictions. The true positive rate (TP Rate) of 1 and false positive rate (FP Rate) of 0 further emphasize LogitBoost's ability to accurately identify positive instances while minimizing false positives. Precision, recall, and F1-score, all equal to 1, demonstrate LogitBoost's perfect performance in correctly identifying positive instances (precision), capturing all positive instances (recall), and maintaining a balance between precision and recall (F1-score).

D. ORGANIZATION OF PAPER

We organize the remaining part of the paper in the following sections. Section II discusses some contributory developments in DDoS prevention in VANET security. We categorize the existing solutions into conventional solutions and logit-probit solutions. Section III explains the proposed model and discusses various components and their functionalities. Section IV shows the results obtained after the implementation of PREVIR. Performance parameters used while calculating these results include accuracy, TPR, FPR, Precision, Recall, and F1-Score. Section V analyzes the implications of the proposed work followed by the strengths, weaknesses, and future suggestive works. Section VI concludes our work and notifies extension possibilities.

II. RELATED WORK

This section discusses the prevention solutions used to tackle DDoS attacks in different environments. VANETs are relatively new; therefore, the solutions available for VANETs lack optimal results, so we also consider other related networks to infer the current status of prevention techniques known in the literature. Section II-A shows some conventional solutions, and Section II-B discusses the preliminary understanding of the logit model. Section II-C discusses some solutions based on logit model.

A. CONVENTIONAL SOLUTIONS

We call the solutions conventional because we use them in computer networks, not new-generation networks such as Wireless Sensor Networks (WSNs).

Packet filtering [18] is one of the most popular and conventional solutions, which allows packets from legitimate sources and discards packets of malicious sources. Filtering allows only legitimate packets to enter the network or the system for processing and reduces the impact of DDoS attacks.

A honeypot may be a software, hardware unit, or a combination of both, which attracts the attacker and intentionally lures the attacker to launch an attack against the system. Types of honeypots include pure honeypots, high interaction honeypots, low interaction honeypots [19], [20] [21], Honeywall [22], Honeytokens [23], Honeynets [24], and Honeyfarms [25].

Load Balancing is a technique used for DDoS defense systems. In this process, we distribute all incoming traffic among several servers, which a single server does not handle. It reduces the data processing burden on a single server [26].

SoS is a proactive methodology that uses secure overlay tunneling, filters, and consistent hashing for routing [15]. The Secure Overlay Access Point (SOAP) receives all packets. After verification, SOAP passes packets to the beacons using chord routing, and then the packets are forwarded to the secret servlets. Finally, these servlets send packets to their destination, which makes the overall system very secure.

System configurations and other techniques are also helpful in handling DDoS attacks. These are some simple actions that the system administrator may take to implement security [27]. These configurations are simple to implement and are beneficial in DDoS protection. We summarize the solutions in Table 1.

We also review some contemporary research contributions in the direction of attack prevention. The paper's authors present an attack prevention and mitigation system (DL-IDPS) that utilizes deep learning and attains 100% accuracy. It works in Software Defined Network (SDN) switch using packet length as the main parameter [28]. To provide security to IoT devices and SDNs, a three-tier IDPS uses user, packet, and flow validation [29]. The blockchain-based solutions also provide decentralized and

TABLE 1. Conventional attack prevention techniques.

Main Technique	Sub Category	Author(s) of the page	Functionality or Working
Egress Filtering	Default Allow and Default Deny Policy	Mahajan & Sachdeva (2013) [10]	For filtration, all outgoing going packets are examined for any abnormal patterns.
Ingress Filtering	Rule-Based ingress Filtering	Ferguson et al. (1998) [11]	Transmissions that fail to meet the filter's requirements are not permitted to access the network.
	Route Based Distributed Packet Filtering (DPF)	Park and Lee (2001) [12]	A proactive method of filtration works based on the path traveled by the packet.
	History Based IP Filtering (HIF)	Peng et al. (2003) [13]	Filters classify packets after reviewing the packet's past information stored in specific databases.
	Hop Count Filter scheme (HCF)	Jin et al. (2003) [14]	Packets are screened according to the number of hops (nodes) a packet has traveled.
	Secure Overlay Services	Keromytis et al. (2004) [15]	Nodes are given specific assignments and duties, and Hash functions are used to route legitimate transmissions.
Honeypot based Techniques	Simple Honeypots	Weiler N. (2002) [16]	Fictitious security mechanisms provide vulnerabilities and lure attackers.
	Interaction-based honeypots	Daimi, K. (2018) [19], Kim et al. (2011) [20], NG et al. (2018), [21]	The amount of activity permitted by the honeypot is determined by interaction.
	Honeytokens	L. Spitzner, (2003) [22]	An item or an expression attracts attackers.
	Honeynets	L. Spitzner, (2003) [23]	Multiple honeypots work together.
	Honeyfarms	Jiang et al. (2006) [24]	A centralised collection of honeypots.
Load Balancer	Benefit-based Load Balancing (BLB)	Anh et al. (2008) [26]	Traffic volume is regulated according to the need for processing and afterward forwarding.
Configuration Based Techniques	Disabling Unused Service	Fortunati et al. (2006) [27]	Options that are not necessary must be deactivated.
	Software Updates & Security patches	Fortunati et al. (2006) [27]	Patches and software updates must always be applied as soon as they are released.
	Account privileges and authentication	Fortunati et al. (2006) [27]	Maintaining user accounts up to date and providing rights is vital.
	Changing IP addresses	Fortunati et al. (2006) [27]	IP addresses need to be changed often.
	Disabling IP Broadcasting	Fortunati et al. (2006) [27]	IP broadcasts should not be allowed.
Current Trends	DL-IDPS	Lee et al. (2020) [28]	The most significant factor for threat identification and avoidance is packet length.
	3 tier IDPS	Ali et al. (2020) [29]	When patches become available, programs must be promptly upgraded and deployed.
	Blockchain	Rohit et al. (2019) [30], Jamder et al. (2019) [31]	Blockchain-based solutions for IoT devices are proposed
	Umbrella	Liu et al. (2019) [32]	This solution works with the help of ISPs.
	reCAPTCHA controller	Poongodi et al. (2019) [33]	It works with the help of information based metrics.
	Statistics based solution	Ahuja and Singhal (2018) [36]	Statistical solutions are used for DDoS.
	MAEC-X architecture	Dao et al. (2018) [37]	Special MAEC-X architecture is proposed.
	D-PIDs	Sreerekha et al. (2018) [38]	It works with the help of information based metrics.

transparent detection and prevention systems [30] [31]. With the aid of Internet Service Providers (ISPs), a unique solution known as "UMBRELLA" executes security [32]. The authors in [33] suggest an Intrusion Prevention System for DDoS assaults in vehicular environments using information and a reCAPTCHA controller. In a different research [34], the authors propose a DDoS remedy utilizing a VANET trust-based assessment mechanism. A virtual honeypot-based system has been provided by [35] for attack prevention. The solutions in [36], [37], [38],and [39] are based on statistics, MAEC-X architecture, Dynamic Path Identifiers (D-PIDs) for routing and information-centric networking environment.

B. LOGIT MODEL

A logit model is a type of statistical model used for binary outcome variables, which means the dependent variable can take on only two possible values, usually coded as 0 and 1. The logit model is a special case of the more general logistic regression model. The Cumulative Distribution Function (CDF) of random variable X is a probability that assumes values lower than or equivalent to X_0, where it is some observed exogenous value of X. We can formulate it as follows:

$$f(X = X_0) = p(X \leq X_0), \tag{1}$$

and the dependent variable is conditional, that is

$$P_i = E(Y_i = 1/X_i). \tag{2}$$

This implies that the non-linear probability function becomes as:

$$P_i = \frac{1}{1 + e^{-(\beta_1 + \beta_2 X_1 + \beta_3 X_2 ..... + \beta_6 X_5)}}, \tag{3}$$

thereby,

$$P_i = \frac{1}{1 + e^{-zi}} = \frac{e^{zi}}{e^{zi} + 1}. \tag{4}$$

This makes the following equation.

$$Z_i = \beta_1 + \beta_2(Flowpackets) + \ldots + \beta_n(Flowduration). \tag{5}$$

In Equation 5, the packet features are *Flowpackets*, *Flowduration*. We can have n number of packet features. β is the co-efficient for a corresponding feature. The equation makes it clear that Zi ranges between $-\infty$ and ∞, P_i ranges between 0 and 1 as the probability of malicious attack and legitimate traffic = 1. Therefore,

$$(P_i) + (1 - P_i) = 1. \tag{6}$$

In Equation 6,

$$(1 - P_i) = 1 - \frac{e^{zi}}{1 + e^{zi}} \tag{7}$$

and

$$(P_i = \frac{e^{zi}}{1 + e^{zi}}). \tag{8}$$

$$(1 - P_i) = \frac{1}{1 + e^{zi}}. \tag{9}$$

Linear transformation becomes as:

$$\frac{P_i}{(1 - P_i)} = \frac{1 + e^{-zi}}{1 + e^{zi}} = e^{zi}. \tag{10}$$

An odd ratio is considered in favor of malicious attacks. We then log to both sides and we obtain:

$$Li = In(\frac{P_i}{(1 - P_i)}) = Zi = (\beta_1 + \beta_2 X_1 + \ldots.. + \beta_6 X_5). \tag{11}$$

Parameter estimation: In the integrated logit model, we should minimize the log loss to obtain 'good fit' parametric values, which indicate the closeness of predicted probabilities to the corresponding actual values.

$$LogLoss = \frac{-1}{(N)} \sum_{i=1}^{n} Yi.Lu(Pi) + (1 - Yi)In(i - Pi). \tag{12}$$

As an alternative method, the logit model uses maximum likelihood estimation to compute the log of odds ratio as shown in Equation 11 [40].

C. EXISTING LOGIT MODEL-BASED SOLUTIONS

In this section, we show existing logit model-based solutions. These solutions benefit cyber security because they can efficiently prevent many security attacks using probability distributions. Similarly, a related study estimating the chances of bank failure proposed an early warning mechanism through detailed logit and discriminant analysis [41].

Online transactions have become susceptible to e-threats more than ever, jeopardizing organizations' market value. In this direction, [42] provides the study that uses preventive measures based on quantification of the probability of eight types of malicious attacks through generalized linear models such as logit and probit. Multinomial logit and probit regression are applied using time parameters and types of attacks as dependent attributes. Expected probability curves quantify the frequency of attacks for each variant type over the years. Although this model helps prevent eight different kinds of attacks, the model results are not available for comparison purposes. This model has findings like the time when the attack increases and the location from where more attacks occur.

Probit regression predicts vulnerability for online threat prevention. It inputs the probability of accumulative Vulnerability Loss (VL) concerning time as a binary dependent variable. Predicting the capability of a model by comparing the observed and actual VL is highly accurate [43]. For blocking real-time e-hostility, data retrieved from specialized software, notably an active intrusion prevention system, is analyzed through logit regression. In this context, We base our research on binary grouped logit regression, which uses categorical parameters named protocol, geographical origin of the event, service accessed & method of access. We characterize the dichotomous dependent variable by the nature of the attack, i.e., malicious or non-malicious. We use the model's results as risk prevention and reduction tools [44]. Several research methodologies compare various ML algorithms for their accuracy in malware analysis [45].

The cyber risk assessment and mitigation model for DDoS (CRAM-D model) attacks runs logit and probit models for risk prevention and mitigation. It applies the process to a dataset having 10329 records for the period (2012-18) in the online gaming industry. Bits per second in GB terms and duration of attack in hours are the parameters used in this model. It predicts the probability of each type of DDoS attack (taken six in total); it applies a generalization of the probabilities to obtain the mean for probability values for each attack. Thus, output in both assists in computing expected loss amounts and expected premiums to provide e-risk insurance for highly susceptible categories [47]. CRAM-D can detect six types of attacks, but it is mainly used in the gaming industry, so its applicability in VANETs is pending.

A logistic regression-based solution is proposed by Cvitić et al. [48]. This study tackles the persistent challenge of mitigating Distributed Denial-of-Service (DDoS) attacks within diverse Internet of Things (IoT) devices, especially in home networks. Although this approach employs boosted logistic model trees, it works only for distinct IoT devices. Nayak et al. suggested a machine learning-based misbehavior detection system (ML-MDS) [49]. The study focuses on analyzing the trust value of vehicles using 49 features such as source IP, destination IP, source port, destination port, transaction protocol, etc with various classifiers like logistic regression. The machine learning-based misbehavior detection system achieved a high detection accuracy of 98.4%. A similar lightweight IDS for IoT is provided by Sadhwani et al. [50]. The proposed solution incorporates an LR-based classifier to make an IDS.

TABLE 2. Logit and Probit Model-based prevention techniques.

Author(s)	Strategy Used	Advantage	Parameters	Results
Martin, d.(1977) [41]	logit & discriminant analysis	Fund supply differs as per premium.	asset risk, liquidity, capital adequacy,	log odds ratios, marginal effect coefficients
Green et al. (2007) [42]	Binary grouped logit regression	data analysis of intrusion prevention systems to see hackers' behavior	protocol, the origin of the event, service accessed, and method of access.	false and true negative alerts report from the model.
Mukhopadhyay et al. (2009) [43] [45]	Multinomial logit & probit regression	Propose e-risk insurance	time	quantify expected risk, attack trend projection
Jinkun et al.(2016) [44].	Probit regression	provided novel way to construct algorithm of VL model	time	predictions and validity for accumulative vulnerability loss
Prakash et al.(2020) [46]	i) support vector ii) random forest iii) logistic which includes newton-raphson,	evaluation through accuracy scores.	flow duration, forward packets, backward packets, total length of the packet	confusion matrix (performance analysis), precision results
Sharma et al. (2020) [47]	CRAM-D model-based logit-probit	Targeted risk mitigation through heat matrix (probabilities*loss)	Bits per second (in gb) & duration (hrs)	generalized probabilities, expected losses, risk-severity matrix.
Cvitić et al. (2021) [48]	Boosted logistic model trees	Multi-model and multi-phased approach	Packet size, packet interim times, protocols used, and changes in the number of destination Internet protocol (IP)	High detection accuracy of 99.9%
Nayak et al. (2022) [49]	A machine learning-based misbehavior detection system (ML-MDS)	By analyzing trust value of vehicles	49 features like as source IP, destination IP, source port, destination port, transaction protocol, etc.	High detection accuracy of 98.4%
Sadhwani et al. (2023) [50]	Lightweight IDS with ML	Combination of multiple classifiers is used	Timestamps, IP addresses, ports, protocols, service details, duration, etc.	Both Binary and multiple-class classification of NaïveBayes obtains an accuracy of 100%

We summarize the existing techniques as shown in Table 2.

III. PROPOSED MODEL: PREVIR

We use the NS3 tool to generate three attack scenarios. One scenario contains benign traffic, the second has only attack traffic, and the third scenario includes benign traffic along attack traffic. During the simulation process, we use a random mobility model to generate a realistic environment with uncertain behavior of the vehicles. We capture the total traffic in a.pcap file. We observe this file in Wireshark and create a dataset in the.csv format. The attack scenario is based on a peer-to-peer-based reflection DDoS attack. The generated/ synthesized dataset contains all characteristics applicable in any generalized network scenario; however, the generation of the dataset uses a VANET architecture in NS3 with RSUs and vehicles (nodes). We apply the proposed system PREVIR to this synthesized and public dataset to check PREVIR's efficacy. The model selects only those attributes which may give the highest accuracy. PREVIR tests prevention efficacy in a vehicular environment. In the future, we will test PREVIR in realistic VANET.

In Galician, PREVIR denotes prevention. PREVIR predicts the probability of occurrence of malicious attacks through a logit model based on a simulated and public dataset to develop an early threat prevention system. The logit model offers both classification and computations of probabilities. We get the best outcomes using its mathematical capabilities with little training and without over-fitting. The learned weights (predicted parameters) provide information on the different variables' weight distribution. Additionally, it indicates if the connection is optimistic or adverse. We use an integrated logit model to analyze the correlation among these variables. It is a highly adaptable model that uses regularization, which lowers model error using regularization parameters. In this section, we discuss the overall functioning of PREVIR and the detailed process of model selection, dataset, parameter selection, goodness of fit, and probability calculation. We show the complete process in a flowchart through Figure 2.

The PREVIR model is developed to predict the probability of malicious attacks in vehicular environments. It utilizes a logit model based on a simulated and public dataset to create an early threat prevention system. The model selects attributes that provide the highest accuracy and tests prevention efficacy in a vehicular environment. The logit method in the PREVIR model contributes to predicting packet probabilities by transforming non-linear equations into linear equations. This transformation simplifies the task as coefficients become linear in log odds. The logit model offers both classification and computation of probabilities, allowing for effective analysis of individual packet probabilities to differentiate between malicious and benign packets. The LogitBoost method plays a crucial role in refining the PREVIR model. It is used to analyze

the correlation among variables and refine the model's predictive capabilities. By using a highly adaptable model that incorporates regularization, the LogitBoost method helps in lowering model error and improving the overall efficiency of the model.

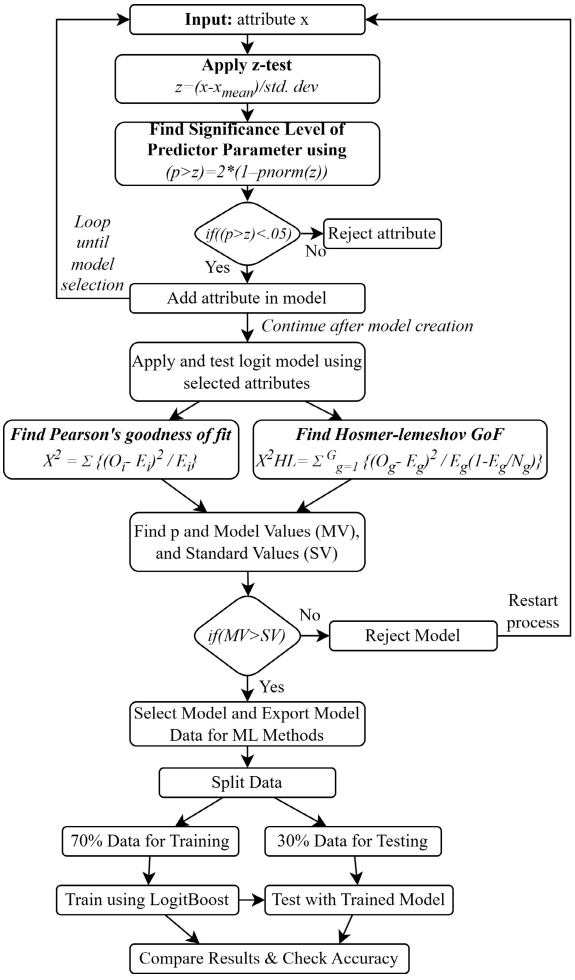


FIGURE 2. Flowchart of PREVIR.

A. FUNCTIONALITY OVERVIEW

PREVIR uses log functions to transform the probability functions. We change Non-linear equations (Equation 3) to linear equations (Equation 11). Equation 11 gives the probability in log odds form, where log odds depict the chances in favor of the outcome. Hence, it simplifies the task as coefficients such as b_1, b_2, b_3, and b_4 are linear in log odds. We apply Wald's Z-test and individual probabilities using these coefficients. We also apply Hosmer-lemeshov's Goodness of fit to check the overall efficiency of the model. In computing results, we use p-values to see the effectiveness where 1.00 signifies a perfect fit model. We check individual packet probabilities using the coefficients to differentiate between malicious and benign packets. As we use logic and probit model with linear regression to prevent attacks in VANETs, we call our solution as *Predictive Risk Evaluation for Vehicular Infrastructure Resilience (PREVIR)*.

B. EXPERIMENTAL SETUP

The experimental setup for the research involving the PREVIR model encompassed the following key components:

1) SYSTEM CONFIGURATION

The primary operating system was Windows 11, while Ubuntu 20.04.2 LTS was employed as the virtual operating system for simulations. Other essential software tools utilized included Network Simulator 3 (NS3), Weka, Python, and Wireshark. The hardware configuration comprised a specialized 4GB graphic card, Intel i7 9th generation processor, 8 GB of RAM, 256 GB SSD, and 1 TB HDD, providing the necessary computational power and storage capacity for conducting realistic simulations.

2) EXPERIMENTAL IMPACT ON PERFORMANCE

The potential impact of virtualization on performance was acknowledged and deemed acceptable based on experiment isolation, resource allocation, and preliminary experiment validation.

3) NETWORK TOPOLOGY

The simulation involved a distinct attack topology consisting of 12 vehicles and one RSU for communication. Specific details included the roles of attacker, victim, and bot nodes among the 12 vehicles, the utilization of routing protocols such as UDP, TCP, and ICMP, data rates for benign and DDoS traffic, simulation duration, communication area, and the application of a random mobility model for all nodes.

The effectiveness of the PREVIR model is evaluated based on specific criteria and metrics, including model performance metrics, network performance metrics, experimental validation, model monitoring and output, and model evaluation metrics. These criteria and metrics were utilized to comprehensively assess the model's predictive capabilities and prevention efficacy in a vehicular environment, ensuring a thorough evaluation of its performance in the experimental setup.

C. DATASET

The three datasets that we used are listed below. We find different DDoS attack types in such datasets. If the proposed system or classifier successfully recognizes these attacks, we may use real vehicular networks to test and implement them. PREVIR is a statistical model that works accurately if appropriate data is available. We implement PREVIR on three datasets; two are pre-built (publicly available), and another is a synthesized dataset. The NS3-generated dataset was chosen for its customizability, allowing tailored creation of a dataset reflecting the experimental conditions relevant to vehicular environments. The NSL-KDD and CIC-DDoS datasets were selected due to their real-world relevance, encompassing

diverse attack types and providing benchmark data for evaluating the model's efficacy in identifying and preventing known attack patterns. The combination of these datasets facilitated a comprehensive evaluation of the PREVIR model's performance, from controlled vehicular network environments to real-world attack scenarios, ensuring its adaptability and effectiveness across a spectrum of network security contexts.

1) GENERATED DATASET

It is our own synthesized dataset that we made through simulation in NS3. We developed three different simulation scenarios to collect packets. In the first scenario, there is only benign traffic; in the second, there is only attack traffic; and in the third scenario, there are both benign and attack traffic. This scenario directs traffic from all intermediate nodes toward the victim, making it a deadly assault scenario. In NS3, we record all packet movement data in a.pcap file. We capture a total of 27 parameters with 697792 records. We also use an additional attribute for classification. We generate the dataset using the VANET characteristics shown in Table 3.

TABLE 3. Attributes of Topology.

Parameter	Value
Simulation Platform	NS3.2.7
No. of Vehicles	12
Attacker Nodes	1
Bot Nodes	10
Victim	1
No. of RSUs	1
Routing Protocol	UDP, TCP, ICMP
Visualisation Tool	NetAnim
DDoS Rate	20480 kbps
Normal data rate	512 kbps
Maximum Bulk Bytes	100000 kbps
Simulation Time	40 Seconds
Data Transmission Rate	100 Mbps
Communication Range	100 m X 100 m
Mobility Model	Random mobility

2) PUBLIC NSL-KDD DATASET

The KDD99 dataset is an improved dataset produced by the NSL-KDD. The University of New Brunswick developed this NSL-KDD dataset to fix several problems with the KDD99 dataset. A few instances of the assault types of DoS, Probe, U2R, and R2L, with their respective eleven, six, seven, and fifteen assault sub-classes, are included in this compilation. They enhanced NSL-KDD by removing redundancy and precisely splitting data for training and testing. Therefore, testing this dataset on a vehicular network proves to be feasible.

3) CIC-DDOS DATASET

The CIC-DDoS dataset, specifically the CIC-DDoS-2019, is typical actual network traffic, including benign and frequent DDoS assaults. The collection consists of labeled flows in CSV files and PCAP files that record network traffic data. Various parameters, including timestamps, source, and destination IP addresses, source and destination ports, protocols, and attack kinds, are used to categorize the flows. The researchers prioritized producing background traffic that simulates human interactions to provide a realistic environment for the dataset. To profile the abstract behavior of interpersonal encounters, they used the B-Profile approach, which was put out in a paper by Sharafaldin et al. in 2016. This system simulates the behavior of 25 people over various protocols.

D. PARAMETER SELECTION

For model selection, we identify variables and check which can likely show an impact on the model. For example, in the first experiment with the generated dataset, we select five variables: flow packets, flow duration, total forward packets, total backward packets, and forward header length.

We apply Wald's Z-test to see which variables are important and contribute to DDoS prevention. Wrong variable selection may lead to wrong model selection and lead to failure.

$$z = (x - x_{mean})/std.dev \tag{13}$$

We observe and analyze the individual values of variables obtained by the Z-test. Results show that all the variables are statistically significant at a 5% confidence level. For instance, interpreting the variable TotalFwdPackets, we say that holding other variables constant, if 'TotalFwdPackets' increases by one, the average Logit value goes up by 0.04. That is, the log of odds in favor of the attack goes up by 0.04. We interpret all the variables in the same way.

E. MODEL CONSTRUCTION AND TESTING

After attribute selection, we construct the logistic regression model using the selected attributes. The logit model is a statistical model that predicts binary or multi-class outcomes. We then create the logit model and test using the chosen characteristics. Logit and Probit models have pros and cons; in our proposed PREVIR, we decided on the Logit model. The logit model predicts the probabilities of happening or non-happening of an event. These probabilities assist in attack prevention. Another idea behind the selection of the Logit model is that it uses the Cumulative Distribution Function (CDF) of the logistic distribution, whereas the Probit model uses standard normal distribution. We can use single or multi-class classification depending upon the attack packets. We can adapt the model if attack packets contain multiple packet types. It uses a regularization parameter that we use to fine-tune the model to reduce model errors. The Logit and Probit models yield the same results. If the number of observations is high in the distribution tail and values of Probit estimates become comparable with logistic estimates, we multiply logistic estimates by 0.625-factor value.

F. OVERALL GOODNESS OF FIT

We apply LR Chi (Likelihood - Ratio Statistics) to check the aggregate significance of the model over the no variables (only intercept model). The results show that a p-value for LR chi is less than 5%, showing a significant impact of the selected parameters compared to the intercept model with no variables. We implemented the chi-square goodness of fit test using Equation 14.

$$X^2 = \sum \frac{(O_i - E_i)^2}{E_i}, \tag{14}$$

where X^2 is chi-square value, O_i is the observed frequency, and E_i is the expected frequency.

We apply Hosmer-lemeshov's Goodness of fit to measure the overall model significance. It assumes the null hypothesis of the model is a good fit, and a p-value of more than 5 percent significance level shows the good fit model. Computed results show that a p-value of 1.00 signifies a perfect fit model. We calculate the Hosmer-lemeshov Goodness of fit is computed using Equation 15.

$$X^2HL = \sum_{g=1}^{G} \frac{(O_g - E_g)^2}{E_g(1 - E_g/N_g)}, \tag{15}$$

where O_g signifies the observed events, E_g signifies the expected events and N_g signifies the number of observations for the gth group. G is the number of groups and $\sum$ is the summation notation. The test statistic follows a chi-squared distribution with $G-2$ degrees of freedom. The output returns a chi-square value (a Hosmer-Lemeshow chi-squared) and a p-value (e.g., $Pr > ChiSq$). Small p-values mean that the model is a poor fit.

G. COMPUTING PROBABILITIES

We can also compute probabilities besides computing coefficients in the Logit model. We use the given values of the explanatory variables to calculate the probabilities of attack. We use the "predict" command in Stata to generate these probability values. We indicate values of probabilities using p values, and these values range between 0 and 1. Values equal to 0 or near 0 indicate that the packet is benign, and values equal to 1 or near 1 indicate the packet is an attack packet. The predict command determines the predicted probability (packet probabilities) for each observation in the dataset using a logit model. The logistic function, sometimes called the sigmoid function, is the foundation of the internal equation to determine these probabilities.

$$p = 1/(1 + exp(-z)) \tag{16}$$

where:

p represents the predicted probability (packet probability) for an observation.

z represents the linear combination of the predictors in the logistic regression model, weighted by their respective coefficients.

H. EXPORT MODEL DATA FOR ML APPLICATION

The next step is to export the model data for use in a machine learning (ML) environment once the model is trained and ready for use. We save the essential data so that the machine learning application can easily use the same for the model training.

I. TRAINING

We often split the available dataset into two subsets: a training set and a testing set, to evaluate the performance of the trained model. We prepare the model using the training set and assess its performance using the testing set. A 70:30 data split is used, with 70% going to training and 30% to testing. Once we divide the data into training and testing sets, the following step is to load the 70% training dataset. The accuracy achieved by various ML algorithms at different data splits is shown below in Table 4.

TABLE 4. Accuracy results at various data splits.

Split Ratio	LB	ABM1	MLP	NBM	Vote	IMC
40-60	99.9967	99.3284	66.9534	66.2951	66.2951	66.2951
50-50	99.9974	99.3247	85.9213	66.3467	66.3467	66.3467
60-40	99.9979	99.3279	87.9839	66.3754	66.3754	66.3754
70-30	99.9981	99.3317	70.2443	66.3955	66.3955	66.3955
80-20	99.9979	99.3257	85.6698	66.3409	66.3409	66.3409

In Table 4, it is evident that the accuracy kept on rising till 70-30 split and after then started declining. MultilayerPerceptron is one exception in which the highest accuracy is maximum at 60-40 data split. As the majority(84%) of algorithms give the highest accuracy at 70-30 data split, we decided to keep it for other datasets. The trained data model finds links and patterns in the data that make predictions. A popular machine-learning technique for classification problems is LogitBoost. We use the DecisionStump classifier to initialize the LogitBoost algorithm in this stage. Simple decision tree algorithms like DecisionStump work as weak learners within boosting algorithms like LogitBoost. LogitBoost with DecisionStump works in the following manner.

1) Initialize the training set:
   - Let N be the number of training instances.
   - Let N be the number of training instances. Initialize the instance weights w_i for $i = 1$ to N such that $1/N$ is assigned to each instance initially.
2) For each iteration $t = 1$ to T, where T is the maximum number of iterations:
   - Train a decision stump:
     - Let $f_t(x)$ represent the prediction of the t^{th} decision stump.
     - Select a feature j and find the best-split point s that minimizes the weighted impurity.
     - The decision stump's prediction for an instance x is:

- Calculate the weighted error (ε) of the decision stump by summing the weights of misclassified instances.
- Compute the stump's weight (α) using the equation: $\alpha = 0.5 * ln((1-\varepsilon)/\varepsilon)$
- Update the instance weights using the equation: $w_i = w_i * exp(\alpha)$ if instance i is misclassified $w_i = w_i * exp(-\alpha)$ if instance i is correctly classified

3) Combine the decision stumps weighted by their respective α values to form the final ensemble model.
4) To make predictions for new instances:
   - Each decision stump predicts either class 0 or class 1.
   - The final prediction is obtained by summing the predictions weighted by their α values and applying the logistic function (sigmoid) to the sum: $y = sigmoid(\Sigma(\alpha_j * h_j(x)))$ where $h_j(x)$ is the prediction of the j^{th} decision stump and α_j is its corresponding weight.

J. TESTING

Each observation in the testing dataset undergoes exposure to the trained model in this stage. The model predicts each observation's output label (classification) or value (regression) based on the discovered patterns and correlations from the training process. The system saves the predicted output label once the prediction is completed for each observation in the testing dataset. These projected labels are compared with the actual labels in the testing dataset to evaluate the model's effectiveness.

K. EVALUATION

It is crucial to consider various performance measures when assessing a trained machine learning model's performance. Metrics like accuracy, precision, recall, and F1-score are significant in this situation. Accuracy indicates how accurately the model predicts outcomes by computing the ratio of cases predicted correctly to all instances. Precision measures the percentage of accurately detected positive cases among all positive predictions to assess the quality of positive predictions by evaluating the ratio of genuine positive predictions to the total number of positive cases. Recall, often called sensitivity, evaluates how well a model can spot positive events. We show this complete process in Algorithm 1.

The PREVIR algorithm begins by taking an attribute x as input and proceeds with a repeat-until loop, iterating until a model is selected. Within this loop, it conducts a Z-test on each attribute in the dataset, computing the Z-score based on mean and standard deviation. A significance level ($p > z$) is calculated using the Z-score and the cumulative distribution function. If this significance level is less than 0.05, indicating statistical significance, the attribute is selected for the model; otherwise, it is rejected. The selected attributes are then used in a logistic regression model. The goodness of fit is assessed using the Chi-squared statistic (X^2), comparing observed and expected values, and the Hosmer-Lemeshow GoF (X^2HL), considering group size and count. The expected values are the probabilities of observing a specific outcome (such as 1 or Yes) for each case in the dataset, as predicted by the logistic regression model. These expected probabilities are then used in the computation of the Chi-squared statistic (X^2), which measures the difference between the observed outcomes and the outcomes expected by the model. The model's variance and structural variance are compared, and if the model variance (MV) surpasses structural variance (SV), the model is selected; otherwise, it is rejected. The chosen model's data is exported for use in machine learning applications. The dataset is split into a 70-30 ratio for training and testing, respectively. LogitBoost with Decision Stump is initialized and trained with the training features. Predictions are made for each observation in the testing dataset, and the predicted labels are stored. Finally, the algorithm evaluates the model's performance metrics, including accuracy and precision, providing a comprehensive framework for feature selection, model selection, and predictive modeling.

Algorithm 1 Proposed Algorithm: PREVIR

Input: Attribute x
Output: Prediction and classification
repeat
  for each $x \in \mathcal{D}ataset$ **do**
    $z = (x - x_{mean})/std.dev$ ▷ Z-test on attribute
    $(p > z) = 2 * (1 - pnorm(z))$ ▷ Significance level of predictor attribute
    if $((p > z) < 0.05)$ **then**
      select attribute
    else
      reject attribute
until Model is selected
Apply logit model using selected attribute set
$X^2 = \sum \frac{(O_i - E_i)^2}{E_i}$ ▷ Goodness of Fit
$X^2HL = \sum_{g=1}^{G} \frac{(O_g - E_g)^2}{E_g(1-E_g/N_g)}$ ▷ Hosmer-lemeshov GoF
compute p, MV and SV
if $(MV > SV)$ **then**
  select model
else
  reject model
Export *Model Data for ML application*
Split *Data in 70-30 Ratio*
Load *70% training dataset*
Initialize *LogitBoost with DecisionStump*
Train *model with training features*
for each *Observation* $\in \mathcal{T}estingDataset$ **do**
  Predict *the output label using the trained model*
  Store *the predicted label*
Evaluate *Performance metrics: Accuracy, precision*

IV. RESULTS AND DISCUSSION

A logit model is an existing approach, but researchers have never used it in the proposed combined form for DDoS prevention. Current solutions do not focus on the early prevention of attacks in VANETs. However, this logit model has the potential for machine learning approaches. Therefore, we explore the logit model for DDoS prevention in VANET architecture. In addition to prevention, existing solutions do not focus on sensitivity, specificity, or accuracy. Our model has shown accuracy up to 100%, which is the maximum possible accuracy. We have also compared our proposed model with state-of-the-art models to prove its efficacy. It has also outperformed in terms of accuracy compared to existing prevention solutions. This technique has never been tested before for DDoS prevention in a vehicular environment. We manifest innovation by implementing it to prevent a particular type of DDoS attack with UDP and TCP flooding and generate a novel dataset.

We pass the packet through the model with the selected variables to identify maliciousness for a new packet that we have not evaluated in the past. Suppose the probability estimated by the model is greater than the threshold. In that case, we say it is a case of a potential attack, and if the probability is less than the threshold, we declare it a benign packet. Ultimately, we validate the calculated probabilities against actual observations and obtain the results.

A. RESULTS OF Z-TEST

We employ the Z-test in the earlier phase to determine whether the parameters are significant. Similarly, we use the Z-Test to construct the logit model based on a specific collection of parameters from both datasets. We test 235,364 records in the Generated dataset, 125973 in the NSL-KDD dataset, and 1048575 records in the CIC-DDoS 2019 dataset. These datasets are evaluated and only relevant features are selected for inclusion in the model.

1) GENERATED DATASET

We chose four variables for the first test with the generated dataset, i.e., time-to-live, time-since-previous-frame, and time-since-first-frame. This test evaluates p, z, and $p > z$ values to suggest which parameters suit PREVIR. The values of $p > z$ are most important in this test. As all the $p > z$ are 0 for all 4 attributes, this attribute combination is selected for the model. We show other resulting parameters obtained from the generated dataset in Table 5.

TABLE 5. Results of Wald's Z-test on Generated dataset.

Class/ Feature	Co-eff	Std. Err.	Z	P > Z	95% Conf	Interval
Time	-0.869	0.170	-5.110	0	-1.2	-0.535
TTL	-8.138	1.782	-4.570	0	-11.6	-4.644
TSPF	-900.144	202.140	-4.450	0	-1296.3	-503.950
TSFF	1.229	0.248	4.950	0	0.7	1.716

Abbreviations used in Table 5 are the Time To Live for TTL, Time Since the Previous Frame for TSPF, and Time Since the First Frame for TSFF.

2) NSL-KDD DATASET

Similarly, Table 6 presents the results of the Z-test performed on the NSL-KDD dataset. The table shows that all the $p > z$ values are less than the 5% significance level. Consequently, we consider variables with these values appropriate for the parameter selection procedure. To find the ideal combination, we analyzed a variety of attribute combinations. Forty-two characteristics and 125,973 occurrences make up the NSL-KDD Dataset. We have chosen the following five parameters for the model: duration, number of file creations, hot, number of failed logins, and count. Table 6 contains the findings we acquired using the NSL-KDD dataset.

TABLE 6. Results of Wald's Z-test on NSL-KDD Dataset.

Class/ Feature	Co-eff	Std. Err.	Z	P > Z	95% Conf	Interval
DUR	-0.000	3.33e-06	-32.270	0	-0.000	-0.000
NFC	0.240	0.045	5.340	0	0.152	0.329
Hot	-0.041	0.002	-15.350	0	-.046	-0.035
NFL	-0.529	0.133	-3.960	0	-0.791	-0.267
Count	-.018	0.000	-166.860	0	-0.018	-0.018

Table 6 uses the abbreviations for variables: DUR represents Duration, NFC represents the Number of files created, and NFL represents the Number of Failed Logins.

3) CIC-DDOS 2019 DATASET

The CIC-DDoS 2019 dataset comprises multiple subsets of this CIC-DDoS dataset. We employed the Syn Dataset in our research. It has 1048575 records and 88 characteristics with the class labels Syn and Benign. While building the model, we select 5 parameters having the most relevance. These parameters are appropriate for the suggested Logit Model since they all generate a value of 0 for $p > z$.

TABLE 7. Results of Wald's Z-test on CIC-DDoS 2019 Dataset.

Class/ Feature	Co-eff	Std. Err.	Z	P > Z	95% Conf	Interval
C1	-.08020	.00371	-21.59	0	-.08748	-.07291
C2	2.37e-1	2.39e-1	9.95	0	1.90e-1	2.84e-1
C3	5.88e-1	2.84e-1	20.72	0	5.33e-1	6.44e-1
C4	-4.4733	0.31192	-14.34	0	-5.08468	-3.86198
C5	-1.37e-1	7.40e-1	-18.57	0	-1.52e-1	-1.23e-1

In Table 7, class label C1 stands for protocol, C2 for flowbytess, C3 for flow packets, C4 for flag count, C5 for flow duration

B. RESULTS OF PEARSON'S GOODNESS OF FIT

After observing variables, we analyze the overall goodness of fit to see whether our proposed model is working fine or not. In Table 8, $Prob > chi2$ is 0 in both datasets; therefore, we claim that the PREVIR model is significant.

C. RESULTS OF HOSMER-LEMESHOV GOODNESS OF FIT

Table 9 shows the results of probabilities and co-variate. This test produces $Prob > chi2$ as 1.000; therefore, we accept the

TABLE 8. Results of goodness of fit.

Parameter	Generated	NSl-KDD	CIC-DDoS 2019
Number of obs	253364	125973	1048575
LR chi2	46298.38	36.46	72.84
Prob >chi2	0.0000	0.0000	0.0000
Pseudo R2	0.992	0.295	0.412
Log likelihood	-17.942503	- 43.559954	- 28.54

null hypothesis is accepted. The tests for both datasets results show the probability value of 1.

TABLE 9. Results of Hosmer-Lemeshov goodness of fit.

Parameter	Generated	NSL-KDD	CIC-DDoS 2019
Total observations	235364	125973	1048575
Pearson chi2	146.09	44693.63	78.45
Prob >chi2	1	1	1

D. INDIVIDUAL PACKET PROBABILITIES

PREVIR shows that the probability of "attack packets" is 0.9 or more; it signifies that our model's prediction for the attack is accurate. Therefore, we can claim that PREVIR is usable to prevent future attacks by using these probability values. Values near 0 or precisely 0 indicate a benign packet the system can process. We show the results in Table 10.

TABLE 10. Individual packet probabilities.

No	Time	Source	Dest.	EpochTime	p
1	0	10.1.1.1	10.1.2.2	0.002009	0.999997
2	0	10.1.2.2	10.1.1.1	0.002009	0.989718
3	0.000277	10.0.0.1	10.1.2.2	0.002286	0.999997
4	0.000321	10.0.0.5	10.1.2.2	0.002330	0.999997
5	0.000364	10.0.0.9	10.1.2.2	0.002373	0.999997
6	0.000407	10.0.0.13	10.1.2.2	0.002416	0.999997
7	0.000451	10.0.0.17	10.1.2.2	0.002460	0.999997
8	0.000494	10.0.0.21	10.1.2.2	0.002503	0.999997
9	0.000537	10.0.0.25	10.1.2.2	0.002546	0.999997
10	0.000581	10.0.0.29	10.1.2.2	0.002590	0.999997

E. CLASSIFICATION RESULTS

In this part, we first discuss the metrics followed by the results for individual metrics.

1) METRICS

The performance of PREVIR model is evaluated based on accuracy, True Positive Rate (TPR), False Positive Rate (FPR), precision, recall, and F1-score.

- **Accuracy:** In the study, the performance of various machine learning algorithms, including LogitBoost, AdaBoostM1, MultilayerPerceptron, NaiveBayesMultinomial, Vote, and InputMappedClassifier, was evaluated in terms of their accuracy across different datasets. LogitBoost achieved the highest accuracy, with a perfect score of 99.99% in the Generated Dataset. In the NSL-KDD Dataset, LogitBoost performed best with an accuracy of 83%. The CIC-DDoS 2019 Dataset showed high accuracy for all algorithms, with MultilayerPerceptron, AdaBoostM1, and LogitBoost performing exceptionally well, with accuracy scores above 99%.
- **TPR & FPR:** The TP and FP rates were also analyzed for the algorithms. LogitBoost achieved a perfect TP Rate of 1 and an FP Rate of 0 in the Generated Dataset, indicating excellent performance in correctly identifying positive instances and avoiding false positives. However, in the NSL-KDD public dataset, LogitBoost, AdaBoostM1, and MultilayerPerceptron exhibited TP Rates of 0.83 and FP Rates of 0.17 or 0.18, suggesting a relatively higher misclassification rate of negative instances.
- **Precision, Recall, and F1-Score:** LogitBoost achieved perfect precision, recall, and F1-score (all equal to 1) in the Generated Dataset. However, the values for other algorithms were available, making it difficult to assess their performance accurately. In the CIC-DDoS 2019 Dataset, Multilayer Perceptron (MLP) achieved the highest precision and F1-score among the algorithms.

2) ACCURACY

In our study, we evaluated the performance of various machine learning algorithms, namely LogitBoost, AdaBoostM1, MultilayerPerceptron, NaiveBayesMultinomial, Vote, and InputMappedClassifier, in terms of their accuracy. The LogitBoost achieves the highest accuracy in the Generated Dataset with a perfect score of 99.99%. AdaBoostM1 follows closely with an accuracy of 99.33%. The remaining algorithms, including MultilayerPerceptron, NaiveBayesMultinomial, Vote, and InputMappedClassifier, have lower accuracy scores of 70.24%, indicating relatively less accurate predictions. The NSL-KDD Dataset's accuracy score ranges from 49.01% to 83%. LogitBoost performs best with an accuracy of 83%, while AdaBoostM1 and MultilayerPerceptron achieve accuracy scores above 82%. The CIC-DDoS 2019 Dataset shows high accuracy for all algorithms, with MultilayerPerceptron, AdaBoostM1, and LogitBoost performing exceptionally well, with accuracy scores above 99%.

3) MODEL CONSTRUCTION TIME

The model construction time varies across the datasets and algorithms. In the Generated Dataset, LogitBoost takes the longest time to construct, with a duration of 27.12 seconds. AdaBoostM1 and NaiveBayesMultinomial have shorter construction times of 12.36 and 0.06 seconds, respectively. On the other hand, MultilayerPerceptron exhibits the longest construction time of 252.41 seconds, indicating a more time-consuming process. In the NSL-KDD Dataset, the construction times are lower overall, with NaiveBayesMultinomial, Vote, and InputMappedClassifier having the shortest construction times of 0.01, 0.02, and 0.03 seconds,

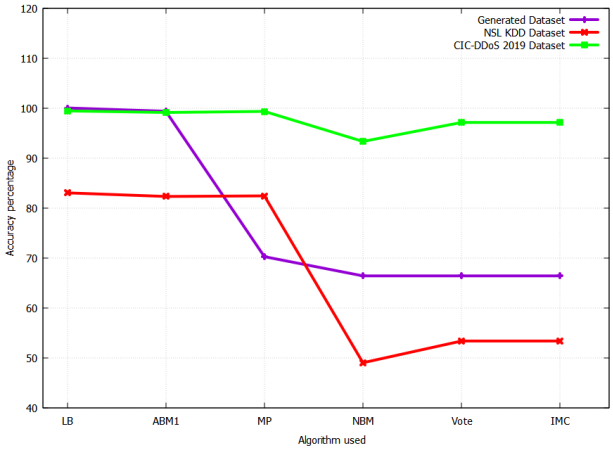


FIGURE 3. Accuracy of the algorithms.

respectively. MultilayerPerceptron takes the longest time to construct among these algorithms, with a duration of 31.88 seconds. The construction times in the CIC-DDoS 2019 Dataset are relatively higher, with NaiveBayesMultinomial, Vote, and InputMappedClassifier exhibiting the shortest construction times. However, even the longest construction time in this dataset, observed for MultilayerPerceptron with 748.27 seconds, is considerably lower than the Generated Dataset.

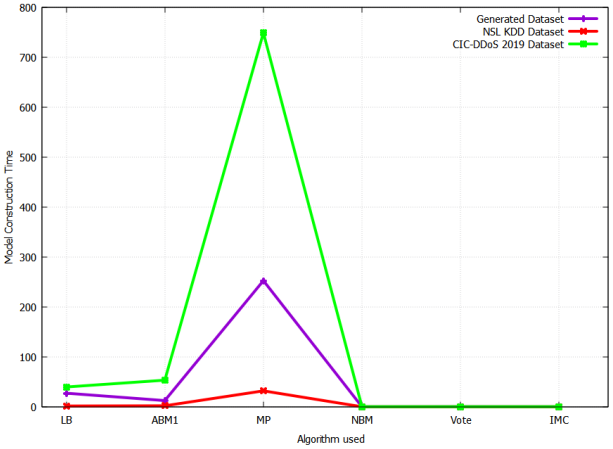


FIGURE 4. Model construction time.

4) TESTING TIME

AdaBoostM1 has the shortest testing time of 0.14 seconds in the Generated Dataset, indicating its prediction efficiency. Conversely, MultilayerPerceptron and Vote require more time for testing, with respective durations of 0.56 and 0.27 seconds. In the NSL-KDD Dataset, the testing times are relatively low across all algorithms, with Multilayer-Perceptron, NaiveBayesMultinomial, and Vote having the shortest testing times of 0.03 seconds. The CIC-DDoS 2019 Dataset shows slightly higher testing times, with AdaBoostM1 having the shortest duration of 0.28 seconds, while InputMappedClassifier requires the longest time of 0.72 seconds.

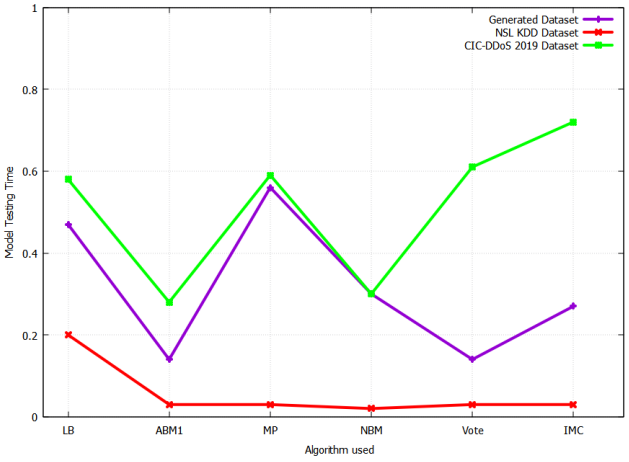


FIGURE 5. Model testing time.

5) TP AND FP RATE

In our generated dataset, LogitBoost achieves a perfect TP Rate of 1 and an FP Rate of 0, indicating excellent performance in correctly identifying positive instances and avoiding false positives. AdaBoostM1 exhibits a high TP Rate of 0.99 and a low FP Rate of 0.01. MultilayerPerceptron, NaiveBayesMultinomial, Vote, and InputMappedClassifier have TP and FP Rates of 0.7 and 0.14, respectively. In the NSL-KDD public dataset, LogitBoost, AdaBoostM1, and MultilayerPerceptron have TP Rates of 0.83 and FP Rates of 0.17 or 0.18. NaiveBayesMultinomial, Vote, and InputMappedClassifier have TP and FP Rates of 0.49 and 0.45 or 0.53, respectively. Similarly, in the CIC-DDoS 2019 Dataset, FP rates are consistently high for all algorithms, suggesting a relatively high misclassification rate of negative instances.

6) PRECISION, RECALL, AND F1-SCORE

In our generated dataset, LogitBoost achieves perfect precision, recall, and F1-score (all are equal to 1). However, the values for AdaBoostM1, MultilayerPerceptron, NaiveBayesMultinomial, Vote, and InputMappedClassifier are available, making it difficult to assess their performance in terms of precision, recall, and F1-score accurately. In the NSL-KDD public dataset, LogitBoost, AdaBoostM1, MultilayerPerceptron, and NaiveBayesMultinomial exhibit similar precision, recall, and F1 scores. However, we use the precision values for Vote and InputMappedClassifier provide must be provided, making it difficult to assess their precision, recall, and F1-scores accurately. In the CIC-DDoS 2019 Dataset, Multi-Layer Perceptron (MLP) achieves the highest precision and F1-score among the algorithms.

The analysis of the datasets reveals variations in model construction time, testing time, accuracy, TP Rate, FP Rate,

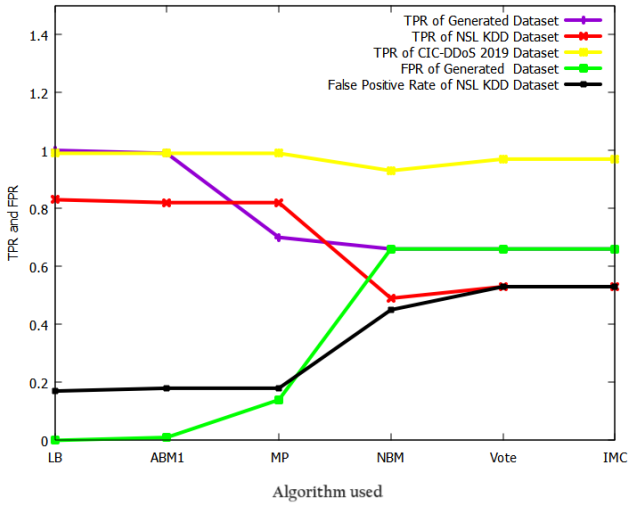


FIGURE 6. True positive rate and false positive rate.

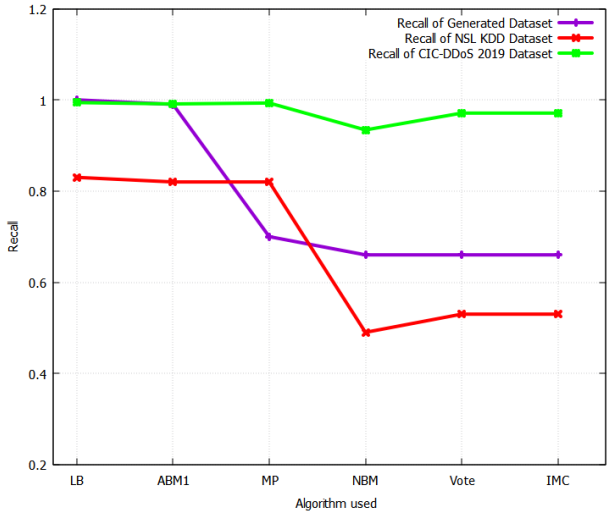


FIGURE 7. Recall values in both the datasets.

and, to some extent, precision, recall, and F1-score across different algorithms. While LogitBoost generally performs well in terms of accuracy and TP Rate, the performance of other algorithms may vary depending on the dataset. It is worth noting that the missing values for precision, recall, and F1-score limit a comprehensive analysis of these metrics for some algorithms in both datasets.

7) RESULT ANALYSIS

In the detailed result analysis, multiple datasets were scrutinized to evaluate the effectiveness of various attributes and algorithms in DDoS attack detection. For the Z-Test analysis, attributes such as Time-to-live, Time-since-previous-frame, and Protocol were found to be significant, showcasing the importance of time-related patterns and communication protocols in distinguishing normal traffic from anomalies. The inclusion of attributes like Number of file creations and Hot from the NSL-KDD dataset, as well as Flow bytes and Flag count from the CIC-DDoS 2019 dataset, further enhanced the model's accuracy. The goodness of fit tests, namely Pearson's and Hosmer-Lemeshow, indicated strong fits, affirming the reliability of the model's predictions. Individual packet probabilities were calculated, enabling the differentiation between benign and potentially malicious packets based on learned attack patterns.

In evaluating algorithm performance, LogitBoost consistently demonstrated superior accuracy, outperforming other algorithms across all datasets. Its boosting nature, emphasizing misclassified instances, resulted in high True Positive Rates (TP Rates) and low False Positive Rates (FP Rates). While LogitBoost excelled in accuracy, simpler algorithms like NaiveBayesMultinomial offered faster construction and testing times, making them suitable for real-time applications where speed is crucial. Additionally, balanced metrics such as precision, recall, and F1-score were considered. LogitBoost, AdaBoostM1, and MultilayerPerceptron exhibited balanced performance in these metrics, ensuring accurate identification of DDoS attacks while minimizing false positives.

This comprehensive analysis provides valuable insights into the attributes' significance, goodness of fit, individual packet probabilities, and algorithm performance. Understanding these findings is essential for selecting the most suitable approach based on specific application requirements, ensuring effective and reliable DDoS attack detection.

F. COMPARATIVE ANALYSIS

We compare the results of our proposed PREVIR with the existing state-of-the-art models for validation. We consider two existing models as mentioned in Green et al. [42] and Bajracharya [55] as these two models are closely connected to PREVIR in the use probabilistic approach. We show the statistical results in Figure 8. In our experiments, we use two tests, one with our synthesized dataset and another with the publicly available dataset. In this paper, we have considered the mean values of both tests.

Comparing the classification results of PREVIR in the generated dataset and the NSL-KDD dataset, we observe significant variations in model construction time, testing time, accuracy, TP Rate, FP Rate, and, to some extent, precision, recall, and F1-score. In the generated dataset, PREVIR's LogitBoost achieves the highest accuracy of 99.99% and exhibits excellent TP and FP rates performance. AdaBoostM1 also performs well with a high TP Rate of 0.99. However, some algorithms limit precise precision, recall, and F1-score analysis due to missing values. On the other hand, in the NSL-KDD dataset, the accuracies range from 49.01% to 83%, with LogitBoost and AdaBoostM1 achieving the highest accuracies. The TP Rates and FP Rates show variations among algorithms, with some achieving better performance than others. Again, the availability of precision, recall, and F1-score values is limited, hindering a comprehensive comparative analysis. The generated dataset

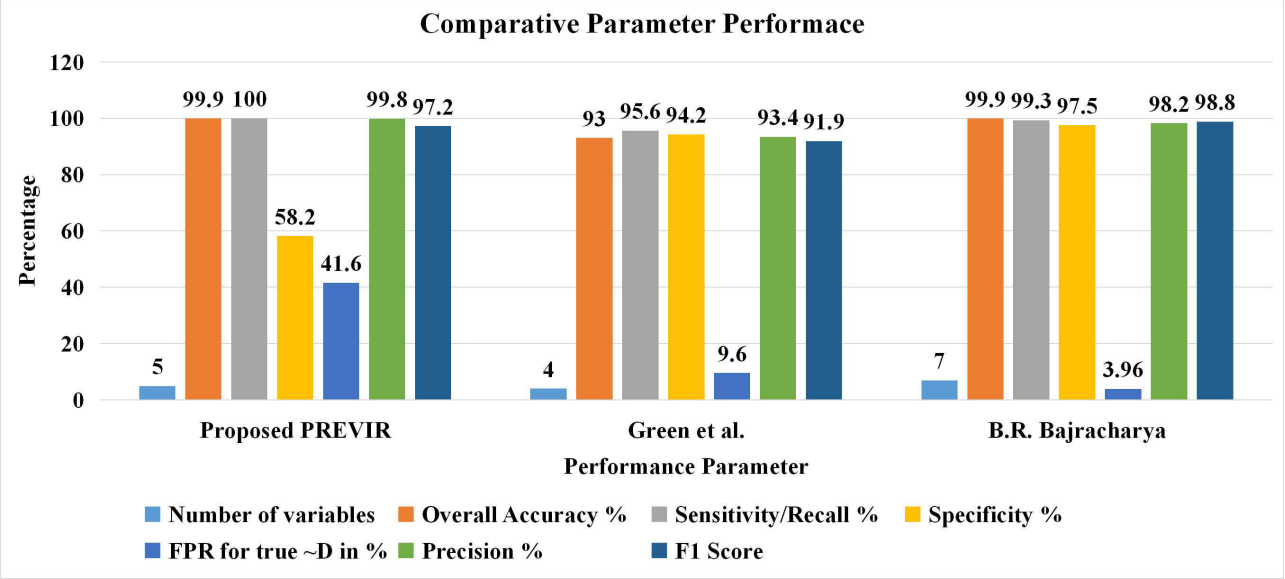


FIGURE 8. Resulting parameters of comparative studies.

demonstrates higher overall accuracy and better TP Rate and FP Rate performance than the NSL-KDD dataset. However, the analysis must include both datasets' precision, recall, and F1-score values.

- **Efficiency of PREVIR:** Model Construction Time: The text does not provide specific details about the model construction time for PREVIR in comparison to other models.
- **Testing Time:** There is no specific information about testing time efficiency for PREVIR in comparison to other models.
- **Overall Assessment:** While the accuracy and TP Rate and FP Rate performance of PREVIR, particularly with LogitBoost and AdaBoostM1, appear promising in both datasets, a comprehensive evaluation including precision, recall, and F1-score values is necessary for a more conclusive comparison with state-of-the-art models.

In addition to this comparative analysis, we compare other DDoS prevention solutions as shown in Table 11. In this table, we compare PREVIR with four existing solutions. These solutions include Bloom filters with IP-CHOCK, genetic model, reCAPTCHA controller, and HCPDS-based framework. We compare the techniques based on features and dataset, simulator type, simulation timing, area of communication and protocol, advantages and limitations, network topology, and performance accuracy. From this comparative study, we claim that our proposed PREVIR is significant for DDoS detection and obtains good accuracy.

V. THEORETICAL DISCUSSION

In this section, we discuss the implications of proposed PREVIR, its strengths, and weaknesses, and future research directions.

A. IMPLICATIONS

The findings regarding PREVIR's performance in vehicular network security have significant implications for the broader field of network security, indicating its potential for real-time threat detection and its adaptability to diverse network environments. The model's robust performance in classifying malicious packets within vehicular networks suggests its applicability to other network environments, such as industrial control systems, IoT networks, and critical infrastructure networks, with the potential to enhance security measures and contribute to the establishment of standard practices for network security. Its adaptability, efficiency, and potential for integration with existing security frameworks make it a promising candidate for bolstering network security across various domains. In addition to these implications, the following practical considerations need to be addressed. It outlines key factors such as resource efficiency, real-time threat detection, robustness, compliance with communication standards, and privacy and data security that need to be taken into account for the effective integration and operation of an IPS in the context of connected vehicle communications. These considerations are crucial for ensuring the IPS's effectiveness, scalability, reliability, and adherence to regulatory requirements within the dynamic and resource-constrained vehicular network environment.

B. STRENGTHS

The study's innovative approach stands out as a major strength, combining statistical methodologies with the LogitBoost machine learning model to pioneer a novel solution for DDoS prevention in vehicular networks. This integrated methodology not only opens new avenues for research but also presents an effective way to combat evolving cyber

TABLE 11. Comparison of Various strategies with proposed model.

Technique	Bloom filters with IP-CHOCK	Genetic Model	reCAPTCHA controller	HCPDS based framework	PREVIR
Reference	Verma et. al. (2013) [51]	Malhi et. al. (2016) [52]	Poongodi et. al. (2019) [33]	Prabakeran et. al. (2020) [53]	Proposed approach
Year	2013	2016	2019	2020	2022
Feature Set	Source IP, Destination IP, ACK/SYN	speed, direction and location of the vehicle, feedback rate, exponential backoff and associativity time	Source IP, Destination IP, Source Port, Destination Port	(packet factors, RSU zone, and vehicle dynamics)	flow packets, flow duration, total fwd packets, total backward packets
Dataset	Simulation based dataset	Simulation based dataset	Simulation based dataset	Simulation based dataset	Simulation based dataset
Advantages	Support Filtering, Prevents small and Large attacks	High accuracy, low tracking time of attackers and low network recovery time.	High efficiency, low latency & energy consumption	Suitable for security and privacy preservation	High accuracy, sensitivity and specificity
Limitation	Low acceptance rate during attack	Exact results are not provided	Memory overheads	Less accurate and slow	Not such problems
Comparison based on simulation and network topology					
Simulator	NS-2.34, NAM, tracing	MATLAB	NS-2.28	NS2	NS-3
Simulation time	80 s	100 sec	20s	100 sec	40 Sec
Communication area	200m	500m	1500 X 1500 m	1000 X 1000 m	100 X 100 m
Packet size	Not Provided	Not Provided	Not Provided	Not Provided	Default
Number of Nodes	25, 50, 75, 100, 125, and 150	50–400	50, 100, 150, and 200	50 to 400	12
Network type	DSRC with Two Ray Ground Model	Not Provided	DSRC with Two Ray Ground Model	DSRC	DSRC
Mac Type	IEEE 802.11p	IEEE 802.11p	IEEE 802.11p	IEEE 802.11	IEEE 802.11
Data transmission speed	5, 10, 15, 20, and 25 m/s	Not Provided	10kbps	Not Provided	512kbps, 20480kbps
Comparison based on Results Obtained					
Accuracy	83.30%	89.9%	94.7%	99.6%	100%

threats. Additionally, the research introduces a robust analysis framework, leveraging statistical methods like the Z-Test, Goodness of Fit, and Hosmer-Lemenshow Goodness of Fit, ensuring a deeper understanding of variable relationships. This framework enhances the research's validity and reliability, marking a significant contribution to the domain. Furthermore, PREVIR effectively tackles overfitting and underfitting issues through meticulous feature selection and the integration of the LogitBoost ML classifier, providing an efficient and reliable defense against DDoS attacks. The system's proactive defense against zero-day attacks adds another layer of strength, showcasing its ability to adapt and defend against emerging threats in real-time. Moreover, the exceptional accuracy and precision demonstrated by LogitBoost, with perfect scores in the precision, recall, and F1-score, validate its capability to identify positive instances accurately while maintaining a balanced approach between precision and recall, solidifying its effectiveness as a defense mechanism.

C. WEAKNESSES

While the study presents several strengths, there are notable limitations to consider. One weakness lies in the potential scalability challenges of the proposed integrated model. Implementing this sophisticated solution in large-scale vehicular networks might cause issues related to computational resources and processing speeds. Additionally, the research could benefit from further exploration of its applicability in diverse network environments. Testing the proposed methodology on various datasets and network configurations would provide a clearer understanding of its adaptability and effectiveness across different scenarios. Furthermore, addressing the potential limitations of the LogitBoost model, such as its sensitivity to noisy data and outliers, would enhance the overall robustness of the system. Lastly, a critical consideration is the system's real-time implementation feasibility. Evaluating its performance in real-world, time-sensitive situations and addressing any latency issues would be essential for practical deployment and usability in live vehicular networks.

In addition to these limitations, some practical implementation challenges need redressing before implementation.

- Complexity: Managing a three-tier architecture can be complex, especially concerning the coordination and communication between different tiers. Integration and synchronization of components across tiers might be challenging, leading to complexities in system design and maintenance.
- Resource intensiveness: Implementing and managing three tiers simultaneously can be resource-intensive in

terms of computational power, memory, and network bandwidth. This could be a concern, especially in resource-constrained environments, where allocating resources to multiple tiers might affect the overall system performance.

- Increased latency: The processing involved in multiple tiers can introduce latency in the system. As data passes through each tier for analysis and security checks, the response time might increase. In time-sensitive applications like VANETs, increased latency can be a significant drawback.

D. FUTURE RESEARCH

In future research, it is essential to focus on improving the performance metrics of the PREVIR model by addressing missing values in precision, recall, and F1-score evaluation. PREVIR should also be evaluated on diverse datasets to assess its performance in different network environments. Further efforts should be directed towards implementing PREVIR in real-time DDoS detection systems, considering scalability and latency issues. Lastly, mitigating the false positive rate of the model is vital to minimize disruptions to legitimate network traffic and improve its usability in practical scenarios. In addition to the above, the following areas can be extended from our proposed PREVIR.

1) ADAPTATION TO IOT SECURITY

Investigating the adaptation of the PREVIR model to address security challenges within IoT networks, particularly in the context of anomaly detection and mitigation of diverse cyber threats targeting IoT devices and infrastructure.

2) INTEGRATION WITH IDS

Exploring the potential integration of the PREVIR approach with traditional intrusion detection systems in network security, aiming to enhance the overall efficacy of threat detection and response mechanisms across various network environments.

3) EXTENSION TO CRITICAL INFRASTRUCTURE PROTECTION

Researching the applicability of the PREVIR model to safeguard critical infrastructure networks, such as energy, water, and transportation systems, by focusing on the detection and prevention of cyber-physical attacks and network anomalies.

By delving into these future research directions, there is a promising opportunity to further advance the capabilities of the PREVIR model and its applicability to diverse network security domains, and contribute to the development of sophisticated and proactive measures for mitigating cyber threats across interconnected systems.

VI. CONCLUSION AND FUTURE WORK

In this paper, we address the escalating concerns surrounding data breaches in expanding vehicular networks, particularly DDoS intrusions. Traditional security measures have primarily focused on intrusion detection, leaving a critical gap in proactive prevention strategies. To bridge this gap, we introduce Predictive Risk Evaluation for Vehicular Infrastructure Resilience (PREVIR), a pioneering amalgamation of statistical analysis (logit method) and machine learning (LogitBoost method).

In our study, LogitBoost consistently demonstrates the highest accuracy values across different datasets, achieving a perfect score of 99.99% in the Generated Dataset and an accuracy of 83% in the NSL-KDD Dataset. In the CIC-DDoS 2019 Dataset, LogitBoost, AdaBoostM1, and MultilayerPerceptron performed exceptionally well, with accuracy scores above 99%. These results provide empirical evidence of LogitBoost's superior performance in accurately predicting outcomes compared to other evaluated algorithms. Similarly, the evaluation of machine learning algorithms on multiple datasets reveals that LogitBoost consistently performs excellently in correctly identifying positive instances and avoiding false positives, as evidenced by its perfect TP rate of 1 and FP rate of 0 in the generated dataset. AdaBoostM1 also exhibits strong performance with a high TP rate of 0.99 and a low FP rate of 0.01. However, other algorithms such as MultilayerPerceptron, NaiveBayesMultinomial, Vote, and InputMappedClassifier show lower TP and FP rates.

In our analysis, LogitBoost achieved perfect precision, recall, and an F1-score of 1 in the generated dataset, indicating its exceptional performance. In the NSL-KDD public dataset, LogitBoost, AdaBoostM1, MultilayerPerceptron, and NaiveBayesMultinomial exhibited similar precision, recall, and F1 scores. Among the algorithms for the specified metric, the Multi-Layer Perceptron (MLP) algorithm achieved the highest precision and F1 score in the CIC-DDoS 2019 Dataset. We also compare our proposed PREVIR with non-amalgamated and non-ML contemporary solutions for DDoS prevention. Our proposed PREVIR outperforms all the frameworks and shows 100% accuracy of probability analysis and classification.

REFERENCES

[1] G. Emin, A. Yaar, G. Airta, and C. Bayilmi, "IoT based and socket programming-based thermal sensor application on mobile platform," *Univ. J. Sci. Technol.*, vol. 11, no. 1, pp. 457–465, 2023.

[2] E. Güney, C. Bayilmis, and B. Çakan, "An implementation of real-time traffic signs and road objects detection based on mobile GPU platforms," *IEEE Access*, vol. 10, pp. 86191–86203, 2022.

[3] V. Bibhu, K. Roshan, K. B. Singh, and D. K. Singh, "Performance analysis of black hole attack in vanet," *Int. J. Comput. Netw. Inf. Secur.*, vol. 4, no. 11, pp. 47–54, Oct. 2012.

[4] H. Hasrouny, A. E. Samhat, C. Bassil, and A. Laouiti, "VANet security challenges and solutions: A survey," *Veh. Commun.*, vol. 7, pp. 7–20, Jan. 2017.

[5] A. A. Marianne, N. G. El-din, S. Mousa, and A. A. El-Kosairy, "Jamming attacks on VANETs," *Acad. Sci. Eng.*, 2014.

[6] F. Sakiz and S. Sen, "A survey of attacks and detection mechanisms on intelligent transportation systems: VANETs and IoV," *Ad Hoc Netw.*, vol. 61, pp. 33–50, Jun. 2017.

[7] D. Kshirsagar and A. Patil, "Blackhole attack detection and prevention by real time monitoring," in *Proc. 4th Int. Conf. Comput., Commun. Netw. Technol. (ICCCNT)*, Jul. 2013, pp. 1–5.

[8] J. Sen, M. Girish Chandra, S. G. Harihara, H. Reddy, and P. Balamuralidhar, "A mechanism for detection of gray hole attack in mobile ad hoc networks," in *Proc. 6th Int. Conf. Inf., Commun. Signal Process.*, Dec. 2007, pp. 1–5.

[9] M. N. Mejri and J. Ben-Othman, "Entropy as a new metric for denial of service attack detection in vehicular ad-hoc networks," in *Proc. 17th ACM Int. Conf. Model., Anal. Simul. Wireless Mobile Syst.*, Sep. 2014, pp. 73–79.

[10] D. Mahajan and M. Sachdeva, "DDoS attack prevention and mitigation techniques—A review," *Int. J. Comput. Appl.*, vol. 67, no. 19, pp. 21–24, Apr. 2013.

[11] F. Paul and D. Senie, "RFC2827: Network ingress filtering: Defeating denial of service attacks which employ IP source address spoofing," 2000.

[12] K. Park and H. Lee, "On the effectiveness of route-based packet filtering for distributed DoS attack prevention in power-law internets," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 31, no. 4, pp. 15–26, Oct. 2001.

[13] T. Peng, C. Leckie, and K. Ramamohanarao, "Protection from distributed denial of service attacks using history-based IP filtering," in *Proc. IEEE Int. Conf. Commun.*, May 2003, pp. 482–486.

[14] C. Jin, H. Wang, and K. G. Shin, "Hop-count filtering: An effective defense against spoofed DDoS traffic," in *Proc. 10th ACM Conf. Comput. Commun. Secur.*, Oct. 2003, pp. 30–41.

[15] A. D. Keromytis, V. Misra, and D. Rubenstein, "SOS: An architecture for mitigating DDoS attacks," *IEEE J. Sel. Areas Commun.*, vol. 22, no. 1, pp. 176–188, Jan. 2004.

[16] N. Weiler, "Honeypots for distributed denial-of-service attacks," in *Proc. 11th IEEE Int. Workshops Enabling Technol., Infrastruct. Collaborative Enterprises*, Jun. 2002, pp. 109–114.

[17] A. Verma, R. Saha, G. Kumar, and T.-H. Kim, "The security perspectives of vehicular networks: A taxonomical analysis of attacks and solutions," *Appl. Sci.*, vol. 11, no. 10, p. 4682, May 2021.

[18] D. B. Chapman, "Network in security through IP packet filtering," in *Proc. USENIX Summer*, vol. 21, 1992, pp. 1–14.

[19] V. Abhilash, "Production Honeypots: An organization's view," in *Global Information Assurance Certification Paper*. Rockville, MD, USA: SANS Institute, 2004.

[20] D. Kevin, G. Francia, L. Ertaul, L. H. Encinas, and E. El-sheikh, *Computer and Network Security Essentials*. Cham, Switzerland: Springer, 2018.

[21] H.-G. Kim, D.-J. Kim, S.-J. Cho, M. Park, and M. Park, "An efficient visitation algorithm to improve the detection speed of high-interaction client honeypots," in *Proc. ACM Symp. Res. Appl. Comput.*, Nov. 2011, pp. 266–271.

[22] C. K. Ng, L. Pan, and Y. Xiang, *Honeypot Frameworks and Their Applications: A New Framework*. Cham, Switzerland: Springer, 2018.

[23] L. Spitzner, "Honeytokens: The Other Honeypot," in *Proc. Annu. Comput. Secur. Appl. Conf.*, 2003.

[24] L. Spitzner, "Honeypots: Catching the insider threat," in *Proc. 19th Annu. Comput. Secur. Appl. Conf.*, 2003, pp. 170–179.

[25] X. Jiang, D. Xu, and Y.-M. Wang, "Collapsar: A VM-based honeyfarm and reverse honeyfarm architecture for network attack capture and detention," *J. Parallel Distrib. Comput.*, vol. 66, no. 9, pp. 1165–1180, Sep. 2006.

[26] A. Le, E. Al-Shaer, and R. Boutaba, "On optimizing load balancing of intrusion detection and prevention systems," in *Proc. IEEE Conf. Comput. Commun. Workshops*, Apr. 2008, pp. 1–6.

[27] S. Fortunati, F. Gini, M. S. Greco, A. Farina, A. Graziano, and S. Giompapa, "An improvement of the state-of-the-art covariance-based methods for statistical anomaly detection algorithms," *Signal, Image Video Process.*, vol. 10, no. 4, pp. 687–694, Apr. 2016.

[28] T.-H. Lee, L.-H. Chang, and C.-W. Syu, "Deep learning enabled intrusion detection and prevention system over SDN networks," in *Proc. IEEE Int. Conf. Commun. Workshops*, Jun. 2020, pp. 1–6.

[29] A. Ali and M. M. Yousaf, "Novel three-tier intrusion detection and prevention system in software defined network," *IEEE Access*, vol. 8, pp. 109662–109676, 2020.

[30] M. H. Rohit, S. Md. Fahim, and A. H. A. Khan, "Mitigating and detecting DDoS attack on IoT environment," in *Proc. IEEE Int. Conf. Robot., Autom., Artif.-Intell*, Nov. 2019, pp. 5–8.

[31] A. R. Jamader, P. Das, and B. R. Acharya, "BcIoT: Blockchain based DDos prevention architecture for IoT," in *Proc. Int. Conf. Intell. Comput. Control Syst. (ICCS)*, May 2019, pp. 377–382.

[32] Z. Liu, Y. Cao, M. Zhu, and W. Ge, "Umbrella: Enabling ISPs to offer readily deployable and privacy-preserving DDoS prevention services," *IEEE Trans. Inf. Forensics Security*, vol. 14, no. 4, pp. 1098–1108, Apr. 2019.

[33] M. Poongodi, V. Vijayakumar, F. Al-Turjman, M. Hamdi, and M. Ma, "Intrusion prevention system for DDoS attack on VANET with reCAPTCHA controller using information based metrics," *IEEE Access*, vol. 7, pp. 158481–158491, 2019.

[34] M. Poongodi, M. Hamdi, A. Sharma, M. Ma, and P. K. Singh, "DDoS detection mechanism using trust-based evaluation system in VANET," *IEEE Access*, vol. 7, pp. 183532–183544, 2019.

[35] Md. M. Rahman, S. Roy, and M. A. Yousuf, "DDoS mitigation and intrusion prevention in content delivery networks using distributed virtual honeypots," in *Proc. 1st Int. Conf. Adv. Sci., Eng. Robot. Technol. (ICASERT)*, May 2019, pp. 1–6.

[36] N. Ahuja and G. Singal, "DDOS attack detection & prevention in SDN using OpenFlow statistics," in *Proc. IEEE 9th Int. Conf. Adv. Comput. (IACC)*, Dec. 2019, pp. 147–152.

[37] N.-N. Dao, D.-N. Vu, Y. Lee, M. Park, and S. Cho, "MAEC-X: DDoS prevention leveraging multi-access edge computing," in *Proc. Int. Conf. Inf. Netw. (ICOIN)*, Jan. 2018, pp. 245–248.

[38] M. Suchitra, S. M. Renuka, and L. K. Sreerekha, "DDoS prevention using D-PID," in *Proc. 2nd Int. Conf. Intell. Comput. Control Syst. (ICICCS)*, Jun. 2018, pp. 453–457.

[39] W. Zhijun, W. Jingjie, and Y. Meng, "Prevention of DoS attacks in information-centric networking," in *Proc. IEEE Conf. Appl., Inf. Netw. Secur. (AINS)*, Nov. 2018, pp. 105–110.

[40] N. R. Norman, "Econometrics by example, by DamodarGujarati," *Econ. Rec.*, vol. 90, no. 290, pp. 404–406, Sep. 2014.

[41] R. Obeid, "Early warning of bank failure in the Arab region: A logit regression approach," *Asian J. Econ. Empirical Res.*, vol. 9, no. 2, pp. 91–99, Aug. 2022.

[42] I. Green, T. Raz, and M. Zviran, "Analysis of active intrusion prevention data for predicting hostile activity in computer networks," *Commun. ACM*, vol. 50, no. 4, pp. 63–68, Apr. 2007.

[43] A. Mukhopadhyay and G. K. Shukla, "Mitigating security breaches through insurance: Logit and Probit models for quantifying e-risk," in *Proc. AMCIS*, 2009, p. 767.

[44] J. Geng and P. Luo, "A novel vulnerability prediction model to predict vulnerability loss based on probit regression," *Wuhan Univ. J. Natural Sci.*, vol. 21, no. 3, pp. 214–220, Jun. 2016.

[45] A. Mukhopadhyay, S. Chatterjee, K. K. Bagchi, P. J. Kirs, and G. K. Shukla, "Cyber risk assessment and mitigation (CRAM) framework using logit and probit models for cyber insurance," *Inf. Syst. Frontiers*, vol. 21, no. 5, pp. 997–1018, Oct. 2019.

[46] P. P. Om, K. Sasirekha, and D. Vistro, "A DDoS prevention system designed using machine learning for cloud computing environment," *Int. J. Manag.*, vol. 11, no. 10, pp. 1–3, 2020.

[47] K. Sharma and A. Mukhopadhyay, "Cyber risk assessment and mitigation using logit and probit models for DDoS attacks," in *Proc. AMCIS*, 2020, pp. 1–10.

[48] I. Cvitic, D. Perakovic, B. B. Gupta, and K. R. Choo, "Boosting-based DDoS detection in Internet of Things systems," *IEEE Internet Things J.*, vol. 9, no. 3, pp. 2109–2123, Feb. 2022.

[49] R. P. Nayak, S. Sethi, S. K. Bhoi, K. S. Sahoo, and A. Nayyar, "ML-MDS: Machine learning based misbehavior detection system for cognitive software-defined multimedia VANETs (CSDMV) in smart cities," *Multimedia Tools Appl.*, vol. 82, no. 3, pp. 3931–3951, Jan. 2023.

[50] S. Sadhwani, B. Manibalan, R. Muthalagu, and P. Pawar, "A lightweight model for DDoS attack detection using machine learning techniques," *Appl. Sci.*, vol. 13, no. 17, p. 9937, Sep. 2023, doi: 10.3390/app13179937.

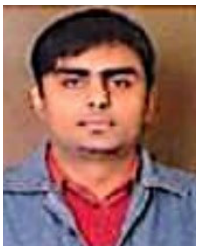
[51] K. Verma, H. Hasbullah, and A. Kumar, "Prevention of DoS attacks in VANET," *Wireless Pers. Commun.*, vol. 73, no. 1, pp. 95–126, Nov. 2013.

[52] A. K. Malhi and S. Batra, "'Genetic' based framework for prevention of masquerade and DDoS attacks in vehicular ad-hoc networks," *Secur. Commun. Netw.*, vol. 9, no. 15, pp. 2612–2626, 2016.

[53] S. Prabakeran and T. Sethukarasi, "Optimal solution for malicious node detection and prevention using hybrid chaotic particle dragonfly swarm algorithm in VANETs," *Wireless Netw.*, vol. 26, no. 8, pp. 5897–5917, Nov. 2020.

[54] M. Al-Mehdhara and N. Ruan, "MSOM: Efficient mechanism for defense against DDoS attacks in VANET," *Wireless Commun. Mobile Comput.*, vol. 2021, pp. 1–17, Apr. 2021.

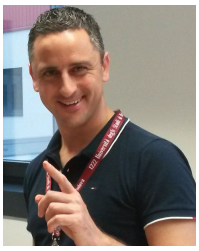
[55] B. Bajracharya, "Detecting DDoS attacks using logistic regression," *A seminar report*, 2020.



AMANDEEP VERMA received the M.Sc. and M.Phil. degrees in computer science. He is currently pursuing the Ph.D. degree with Lovely Professional University, Punjab, India. He is a Research Scholar with Lovely Professional University. He is also with the Education Department with experience of more than 15 years. He has published some articles in the area of vehicular network security.

RAHUL SAHA (Member, IEEE) is currently an Associate Professor with UNIPD, Italy, and Lovely Professional University, Punjab, India. He is currently a young and enthusiastic researcher in the fields of cryptography, network security, the IoT, blockchain, and information security. He is also working on many projects and has completed more than 60 research articles with a total citation of more than 320 with an H-index of 12 and i10-index of 17.

GULSHAN KUMAR is currently an Associate Professor with Lovely Professional University, Punjab, India. He is an energetic and young researcher in the fields of cyber-physical systems, blockchain technology and the IoT, edge and cloud computing, localization and ad hoc positioning, and wireless systems. He has completed more than 85 research articles with a total citation of more than 571 with an H-index of 13 and i10-index of 20.



MAURO CONTI (Fellow, IEEE) is currently a Full Professor with the University of Padua, Italy. He is also affiliated with TU Delft, The Netherlands, and UW, USA; the CEO and the Co-Founder of CHISITO; the Co-Founder of DYALOGHI EU; the Marie Curie Fellow Alumni; and the DAAD Fellow Alumni. He has approximately 600 research articles published with his name and with an overall citation of 17131, an H-index of 64, and an i-10 index of 253. He has lots of stars on his shoulders, like the Young Academy of Europe Fellow, the Editor-in-Chief of IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, an UniPD Academic Advisor for Entrepreneurship Development, the Study Program Coordinator of the M.Sc. degree in cybersecurity, the Head of the SPRITZ Security and Privacy Research Group, and the Director of the Cybersecurity National Laboratory (CINI), UniPD Node.

TAI-HOON KIM (Member, IEEE) photograph and biography not available at the time of publication.

Open Access funding provided by 'Università degli Studi di Padova' within the CRUI CARE Agreement