DDoS Attacks Detection in IoV using ML-based Models with an Enhanced Feature Selection Technique

Ohoud Ali Albishi, Monir Abdullah*
College of Computing and Information Technology
University of Bisha
Bisha 67714, Saudi Arabia

***Abstract*—The Internet of Vesicles (IoV) is an open and integrated network system with high reliability and security control capabilities. The system consists of vehicles, users, infrastructure, and related networks. Despite the many advantages of IoV, it is also vulnerable to various types of attacks due to the continuous and increasing growth of cyber security attacks. One of the most significant attacks is a Distributed Denial of Service (DDoS) attack, where an intruder or a group of attackers attempts to deny legitimate users access to the service. This attack is performed by many systems, and the attacker uses high-performance processing units. The most common DDoS attacks are User Datagram Protocol (UDP) Lag and, SYN Flood. There are many solutions to deal with these attacks, but DDoS attacks require high-quality solutions. In this research, we explore how these attacks can be addressed through Machine Learning (ML) models. We proposed a method for identifying DDoS attacks using ML models, which we integrate with the CICDDoS2019 dataset that contains instances of such attacks. This approach also provides a good estimate of the model's performance based on feature extraction strategic, while still being computationally efficient algorithms to divide the dataset into training and testing sets. The best ML models tested in the UDP Lag attack, Decision Tree (DT) and Random Forest (RF) had the best results with a precision, recall, and F1 score of 99.9%. In the SYN Flood attack, the best-tested ML models, including K-Nearest Neighbor (KNN), DT, and RF, demonstrated superior results with 99.9% precision, recall, and F1-score.**

***Keywords*—Random forest; IoV; DDoS; feature selection**

I. Introduction

After the significant development in the number of vehicles, where it was found that there are one billion vehicles around the world, with an expected doubling by 2035, and the accompanying increase in congestion and traffic accidents, driving has become difficult and dangerous. The idea of the IoV has been formulated to address these challenges. IoV is at the heart of the new generation of intelligent transport systems, representing a new trend of future development. The IoVs is defined as a distributed network with an open, integrated, and credible system that provides a safe and smart environment. This system consists of vehicles, individuals, infrastructure, and networks related to smart systems. It depends on the sensors integrated into modern vehicles, which are linked to the intelligent transport network. Initially, the VENAT network was allocated with its limited ability to use the information provided by mobile devices. Currently, in the 5G era, the IoV has evolved, and its ability to deal with data during communication between vehicles and the network, vehicles with each other, or vehicles with people has significantly improved. In our opinion, safeguarding the communication between vehicles and achieving a more effective network requires the use of ML techniques to provide the necessary protection for wireless communications and efficient detection of attacks, as well as the detection of misconduct and the concept of trust. It provides electronic security services for road services, vehicles, and the information required to enhance security operations and take proactive steps against threats [1]. IoV networks are characterized by many features such as scalability, dynamic topology changes, variable network density depending on city conditions, geographical location energy, security, and privacy. The IoVs involves massive dynamic data, making security and privacy major concerns. One of the most significant challenges in reducing penetration is security and privacy. Types of security attacks include authentication attacks such as jamming, eavesdropping, and Sybil attack. As a consequence, constructing a protection system based on ML techniques, algorithms, and strong authentication is required to maintain anonymity traceability, and wireless communication protection attributes to connect securely and effectively [2]. The main contribution of this research are:

1) Developing a ML based system to prevent communication errors that could cause traffic disruptions or accidents between networks and interconnected vehicles.
2) Developping IoV protection technologies and increased security investment.
3) Ensuring the security for vehicle exchange data storage and infrastructure.

The rest of the paper is organized as follows: Section II presents related work. In Section III, describes Proposed models. Section IV presents our implementation and experiments. Section V presents an experimental evaluation of the performance our heuristic. Section VI concludes the paper and discusses some future work.

II. Related Works

A. Internet of Vehicles (IoVs)

The IoV appeared as a new attempt with the emergence of Ion technologies in the field of wireless cooperation with the

*Corresponding authors

emergence of the Internet of Things (IoT). It is a common complex network in which real communication takes place in the IoV between two or more entities in which many different technologies are used such as the navigation system, mobile, sensors, and the instruction system. IoV has gone through stages with a history of innovation and development through modifications in size, style, and decoration, while technological improvement has pushed mobile phones for cars to the latest trends. Analytical approaches have improved IoV's understanding of traffic and telemetry trends. Advances in information systems, detection and communication capabilities, and intelligent physical infrastructure create new opportunities to reduce real congestion and response challenges. Real-world data flows ingest a heterogeneous amount of data and drive data processing and secure transmission between entities based on this data. Vehicles are controlled and directed in realtime [1]. Analytical approaches have improved IoV's understanding of traffic and telemetry trends. Advances in information systems, detection and communication capabilities, and intelligent physical infrastructure create new opportunities to reduce real congestion and response challenges. Real-world data flows ingest a heterogeneous amount of data and drive data processing and secure transmission between entities based on this data. Vehicles are controlled and directed in real- time [1].

B. *IoV Architecture*

The IoV architecture is composed of four main layers: environment detection, network, computation, and application layer. The environmental detection layer is tasked with collecting data from the environment around the vehicle, such as object locations, road conditions, and driving habits, via an RFID card and sensors embedded inside vehicles. The network layer is responsible for providing all required types of connectivity, such as short-term communication (for example, Zigbee, Bluetooth, Wi-Fi) or cellular network (for example, WiMAX or 4G/LTE), between the objects of the vehicle's environment and its connection to the cloud. The computing layer is accountable for processing, storing, and resolving the collected data necessary to provide safety, comfort, risk situations, and efficiency. The application layer offers both open and closed services. Open services refer to online applications provided by Internet service providers and third-party service providers (for example, real-time traffic services and online video delivery). In contrast, closed services refer to a particular IoV application (for example, a control panel and traffic instructions) [3].

C. *Characteristics of IoV*

- High Scalability: A city can contain millions of vehicles and sensors that require an extensive network. This network must be scalable to accommodate the continuous increase of vehicles.
- Dynamic structure: Many components of an IoV interact with each other (particularly vehicles) moving at high speed, rapidly changing the network topology.
- Geocommunication: The vehicle network uses geocommunication, but in IoV nodes are not predetermined when packets are sent and their speed varies based on the geographical area of the sites [4].

D. *Attack Types in IoV*

IoV security is a highly developed field that requires serious attention. Any simple mistake or security failure can cause a catastrophe in terms of human and economic losses, causing damage to vehicles and road infrastructure.

1) Authentication attacks Sybil Attack: The Cyber node detects the imposition of an attack as it damages the systems in the wireless network and thus increases the likelihood of leakage of vehicle data [5], [6]. GPS deceives: This type of attack by giving deceptive information regarding vehicle speed and geographic location data of other vehicles as undeniable evidence and thus helps to avoid tracing causing unpredictable damage to property and providing false evidence [7].
2) Disguise attacks. In the network environment, each entity has its identity, in disguise attacks a similar identity is given to several nodes simultaneously causing chaos in IoV systems [2].
3) Availability attacks. Availability attacks are the main objective. These attacks is to decrease transmission power and bandwidth and thus collapse the IoV system by controlling or destroying it completely to make a significant impact on the IoV system [2].
4) Eavesdropping attacks. Resource and data are the main components of the vehicle internet system and therefore care must be taken of sensitive data and that unreliable nodes connect to it. In this type of attack, the data is stolen by intercepting and eavesdropping on it [4].
5) Jamming attacks. These are interference attacks. This type of attack aims to camouflage, replay, illusion, and tamper with data to cause chaos and confuse the movement of the regime [4].

III. DDoS Attacks Detection

Several studies and solutions have been provided by researchers in the same study area in this part, and the goal of the article, as well as the research summary, such follows: In the IoV network system setting, high performance is challenging to deliver. This suggests using the Double Deep Q-learning Network (DDQN) model. Overestimation as a Vehicle Internet is prevented. In actual complicated settings, it can deliver higher-quality network services and guarantee improved computing and processing speed. The IoVs are intelligent transport, internet is a new application of the Internet. This research offered several innovative and practical solutions in this area. The algorithm relies in its work on calculating the discharge based on the DDQN network model and then the network tasks are allocated using asynchronous processing technology [8]. The use of wireless communications between vehicle nodes and DR infrastructure makes them vulnerable to various types of attacks. In this regard, ML and its variants are gaining popularity for detecting attacks and dealing with various kinds of security issues in vehicle communications. The research also explains the basics of vehicle networks and the types of communication related to them and how to find solutions using machine learning algorithms [6]. This research focuses on applying machine learning to gather data on vehicles along a GPS route and using the Gaussian process to anticipate traffic based on three groups: training and forecasting groups, bandits, and other variables. Additionally, traffic is forecast for the present and the future, and shortly, the average speed of cars during these times is evaluated [9], [3]. The development of autonomous intelligent cars can help solve transportation problems. The IoT has developed into an advanced and intelligent system called the IoVs, but it is still vulnerable to assaults from this study. To identify dangers. K fold the study discovered that the KNN-CART algorithm delivers the greatest accuracy, with respective values of 99.79% and 99.79% [10]. The Social IoT (SIoT) is the level of enabling awareness where it permits things to interact with one another. Social IoV (SIoV) will transform the automotive industry. The scalability of relying on online technologies is the main topic of this research. It is important to concentrate on the class structure and the function of each system entity while taking into consideration the dynamic nature of the study of SIoV's structure and emphasizing the unique use cases [11].

A. *Machine Learning-based Models*

Since ML was first used as a self-learning method for checkers in 1959, it has been widely used in all areas of the network to improve work performance. The typical model of machine learning consists of three stages:

- The training stage, where the advantages are extracted from the initial data.
- The testing stage, where a new set of data is tested based on the educational experience gained in the training phase by the ML model.
- The prediction stage, where the efficiency of the ML model's work is evaluated based on quality measures.
- ML shows outstanding results in the field of detecting anomalies due to its ability to learn patterns and behavior. Thus, it is the best solution to distinguish deviant from normal behavior, classify attacks, and discover their types.

ML based DoS/DDoS security modeling is shown in Fig. 1:

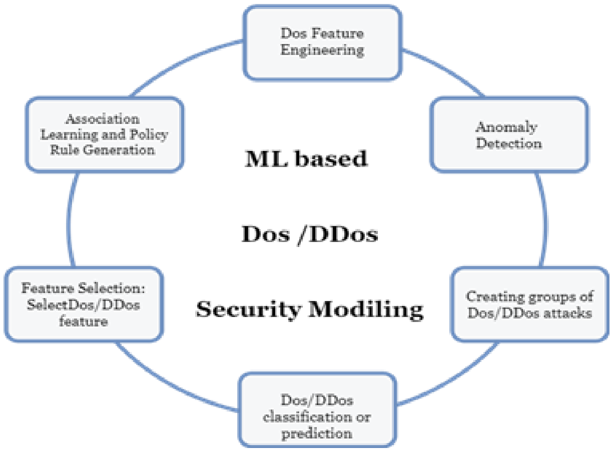

Dos Feature Engineering
Anomaly Detection
Creating groups of Dos/DDos attacks
Dos/DDos classification or prediction
Feature Selection: SelectDos/DDos feature
Association Learning and Policy Rule Generation
ML based
Dos /DDos
Security Modiling

Fig. 1. ML DoS/DDoS Security modeling process.

B. *Machine Learning Models for IoV Security*

- Supervised Learning: It is necessary to assign a value for the input and name a corresponding name for each input of the dataset through the relationship between the input models and the naming of the training group. The algorithm assigns newly acquired samples of test data and applies them to secure vehicle networks. Supervised training is classified as classification and regression, which is one category of popular classification models used in vehicle systems: KNN, DT, Naive Bayes (NB), SVM, RF, and LR models. Logistic regression and random forest models are applied in vehicle networks in applications such as driver fingerprints and types of misconduct.
- Unsupervised Learning: It consists of input values only in their training set and no labels for the dataset. Finding hidden patterns of data focuses on unclassified information. The algorithms used are more efficient and faster in data processing in aggregation applications. The most common assembly mechanisms in vehicle networks include k-means clustering, Hidden Markov Model (HMM), and NN [12].

IV. Proposed Models

In IoV, vehicles can connect and communicate through Vehicle-to-Road (V2R) communication, Vehicle-to-Infrastructure (V2I) communication, as well as communication with sensors Vehicle-to-Sensor (V2S), and Vehicle-to-Vehicle (V2V) communication. All of these communications take place through the wireless network. Of course, all of these communications must have a high level of protection to preserve privacy while continuing to improve it. Current network security technologies and products, such as network firewalls, intrusion detection systems, intrusion prevention systems, web firewalls, and other security devices, are used to enhance security. The user shares much information such as location, as well as many behavioral patterns and some involuntary information such as pedestrian images and private property. This information may be subject to violation, which raises concerns, and this problem cannot be solved by reducing the sharing of information but rather by finding solutions that make it trustworthy. This part will go through the methodology that depends on detecting attacks and penetrations to take urgent measures to protect the IoVs and maintain the privacy of information by monitoring the packets that pass through the IoV network and taking proactive measures to prevent these attacks to maintain a safe communication environment and achieve security requirements [12]. Our proposed model is shown as in Fig. 2.

A. *Details of the Research Methodology*

In this part, we learn how the effectiveness of the security model, as the study was based on the efficiency of the proposed model in detecting security attacks. The CICDDoS2019 dataset with ML machine learning models to detect the ability to detect a DDoS attack [13]. We analyzed the results of DDoS attacks through the machine learning model, which goes through three stages: the training stage, where features are extracted from the raw data, then the testing stage by ML models, where

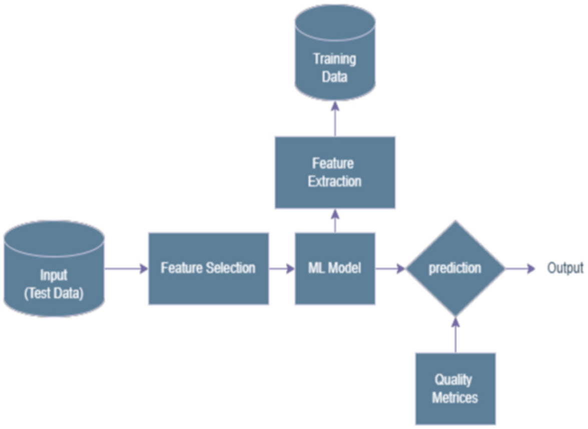


Fig. 2. The proposed method flowchart.

the dataset acquired during the training stage is tested, and the last stage is prediction, where the efficiency of the ML model is evaluated. Algorithm 1 shows the characteristics of the dataset used [14], [15]. Data preprocessing before building the ML model plays an important role in the accuracy of the machine learning models. The features were reduced from 80 to 47 using chi2, which further reduced the test time while Benign and DDoS attacks (UDPSYN) were replaced by [0.1] respectively. From a balancing act, a K-fold where k=5 was used to evaluate and compare ML models [16].

B. Intrusion Detection System (IDS)

IDS intrusion detection systems must be continuously updated to prevent attacks that develop daily. Some algorithms work well with some attacks and perform poorly with others. An ML-based IDS system can extract complex behavioral attributes that can be improved and also include dataset preprocessing [17] as in Fig. 3.

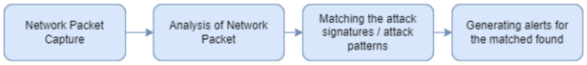

Network Packet Capture
Analysis of Network Packet
Matching the attack signatures / attack patterns
Generating alerts for the matched found

Fig. 3. Architecture of IDS.

There are two problems related to IDS:

- The high rate of false alarms, which are triggered by warnings for unlimited violations and many violations that have not yet been identified.
- New attacks are not easily detected, thus increasing the interest in using ML.

C. CICDDoS 2019 Dataset

This dataset contains the latest and most realistic DDoS attacks. It was developed at the Canadian Institute of Cybersecurity to cover normal traffic. DDoS attacks are the most common and resemble real traffic, network, and properties. It consists of a set of servers and software such as computers, switches, and traffic generators. The dataset provides a knowledge file of the attacks that were performed and models about the applications, networks, and protocols. The dataset has been studied so that it can simulate the types of attacks, consisting of 47 traffic characteristics from the original information traffic consisting of UDPSYN. The prediction and evaluation tests and performance measures are used as evidence for the results and comparisons to analyze the models [17]. To detect DDoS, a group of data was proposed, but none of them were able to detect it. The CICDDoS2019 dataset deals with these problems to achieve optimal performance. This group consists of benign and malicious DDoS attacks. The dataset specifications are listed, and the dataset files use binary classification. The dataset includes missing and duplicate data records processed by applying feature engineering or by disposing of missing records. Feature selection is done using chi-square features. It calculates chi scores to rank features. Feature selection techniques can obtain the optimal feature for target DDoS variables using machine learning algorithms [18].

D. Machine Learning Models

After obtaining the optimal feature sets, KNN, DT, NB, SVM, RF, and LR models are used as models for intrusion detection and attack classification. Using the set and features obtained, the performance of ML techniques is compared in terms of accuracy, Recall, F1, and Precision. The main objective of the research was to resolve the effect of feature selection techniques on detection accuracy, Recall, F1, and Precision. Here is a quick rundown of these methods:

1) Logistic Regression (LR): This adapted linear regression approach is commonly employed in addressing classification challenges, as it has the capability to predict the assignment of an observation to a particular class. Its practical applications include tasks like spam filtering and intrusion detection. In instances where the anticipated likelihood surpasses a predefined threshold, it is anticipated that the occurrence aligns with an attack, given its position above the threshold. Conversely, if the anticipated likelihood falls below the threshold, the occurrence is categorized as normal. This is determined by the following equation:

$$h_{(\theta(x))} = \sigma(\theta^T X) \tag{1}$$

where, $\theta(x)$ is the hypothesis, x is the input feature vector, θ is the LR parameters, and σ (r is a sigmoid function that is used for the threshold definition. The sigmoid is defined as:

where, r is the term $(\theta T x)$ in the previous equation, the output is between (0:1) [19].

$$\sigma(r) = \frac{1}{1 + e^{-r}} \tag{2}$$

2) Naive Bayes (NB): A simple but effective probabilistic algorithm with real-world applications ranging from product recommendations to controlling self-driving vehicles. Using Bayes' theorem for classification, NB is superior to other alternative techniques. NB assumes normally distributed data and defines the conditional probability of the class. Bayes' theorem provides a systematic method for calculating probability based on the advantage of independence assumptions.

$$P(L|X) = \frac{P(X|L)P(L)}{P(X)} \tag{3}$$

where, $P(L|X)$ the posterior probability of class L is, P(L) is the prior probability, $P(X|L)$ is the likelihood function, and P(X) is the probability. The training set is used to estimate these parameters [20].

3) k-Nearest Neighbor (KNN): A method used to classify objects. Based on the learning data closest to the object, the comparison is based on previous and current data. It is a basic strategy that uses new instances from a test set to the closest instance in the training set. The number of neighbors and the distance are the two basic parameters of the KNN technique. The algorithm calculates the distance to the nearest neighbor by applying the Euclidean distance formula and is known as:

$$d(x, y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \tag{4}$$

where, $d(x, y)$ is a Euclidean distance function between the two samples, x_i is the initial observation, y_i is the second sampling of the information, and n represents the observations [21].

4) Decision Tree (DT): DT classifiers are one of the most popular ways to represent classifiers for data classification. It is one of the widely used techniques in data mining and can handle a vast amount of information. It is likened to a tree with its branches and leaves, where the inner node refers to the rules of classification, the leaves refer to the chapter label, and the branch refers to the results. The greatest degree of information acquisition is used as a measure for choosing the optimal traits and is used to construct the decision node, by creating a new sub-tree under the decision tree. The cycle continues until all the results of the subsets have the same value, at which point the process stops, and the final value is calculated as an output value. Gini inclusions were used as division criteria, as shown:

$$G(D) = \sum_{i=1}^{C}(P(i) * (1 - P(i))) \tag{5}$$

where, D is the training dataset, C is a collection of class labels, and $p(i)$ is the proportion of samples having the class label i in C. When there is just one class in C, the Gini impurity is zero [22], [23].

5) Random Forest (RF): ML technology is a supervised technique and gives excellent results. It consists of several trees planted randomly, and each leaf node is named for each tree. Each internal node contains a test that divides the data space to be classified by sending images to the bottom of the tree and collecting the leaf distributions obtained. The best way to determine the number of trees necessary is to compare forest predictions with subset predictions from the forest to produce a model that predicts the dataset more accurately and consistently. Its advantage lies in the fact that it is highly adaptable and enables it to solve classification and regression issues [23]. The general equation for a random forest model can be written as:

$$y = f(x) = \sum(i = 1\ to\ n)\ Ti\ (x)/n \tag{6}$$

where, y is the predicted outcome, x is the input feature vector, n is the number of DTs in the forest ,and $Ti(x)$ is the prediction made by the RF.

6) Support Vector Machine (SVM): Supervised learning models with machine learning analyze the data used in classification and regression analysis and can handle linear datasets. The main goal of SVM is when the problem is not linearly separable, then it will be with a nonlinear kernel such as RBF for nonlinear mapping to transform the unique form of training data into a higher dimension through the equation.

$$K(x, y) = e^{\frac{||x-y||^2}{z\sigma^2}} \tag{7}$$

where, σ is the variance and the SVM hyper-parameter, $||x - y||$ is the Euclidean distance between two points [24].

E. Executing DDoS Attacks

A subclass of DoS attacks disrupts normal traffic for a particular target. DoS attacks from multiple sources are performed simultaneously. On the IoVs, malicious vehicles can launch DDoS attacks, so it is important to detect attacks in real-time. Intended to flood threats to undermine the availability of vehicular Internet operations to perform DDoS attacks through an SSH-based master agent. The types of attacks described in the dataset are as follows: UDP-Lag attack is an attack that disrupts the communication file between the server and the client, and a SYNflood attack that controls the transmission to drain the victim's resources and affects them by not responding [25]. ML is one of the most popular methods, as it is considered a powerful model that predicts modern forms of DDoS attacks, as it analyzes them in real-time and classifies them into normal behavior or abnormal behavior. It also predicts attacks before they occur based on DDoS modeling and many algorithms such as KNN and SVM [21]. DDos attack in IoV is drawn in Fig. 4.

F. Confusion Matrix

The confusion matrix as in Fig. 5 is a measure of self-learning rating performance. It is a table of type $n * n$ where n is the number of possible labels for the data. The confusion matrix plays an important role in determining performance. In our model, we have three types of values: Benign , UDPLag, and SYN.

Most of the measures mentioned above can be calculated from the confusion matrix illustrated in Fig. 5, which is a typical tool used to record model performance. The rows in the matrix are the actual class, and the columns are the expected class. In the confusion matrix, TN, FN, FP, and TP represent true positives (the number of negative samples correctly classified, similar definition for the rest), false negatives, false positives, false negatives, and true positives, respectively. This is especially important under imbalanced learning conditions [26].

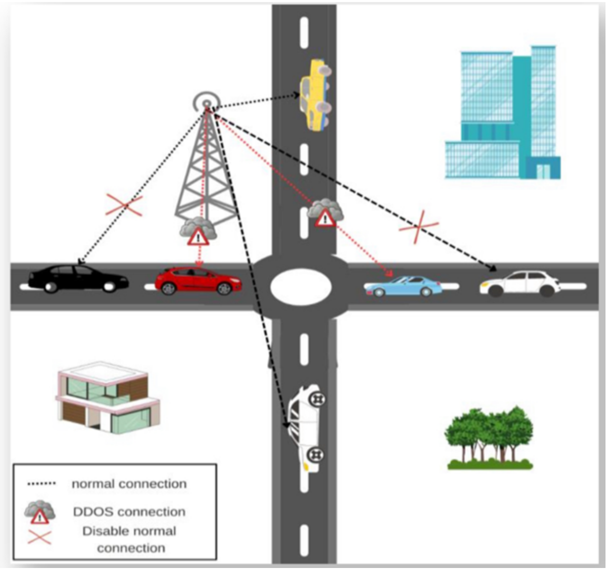
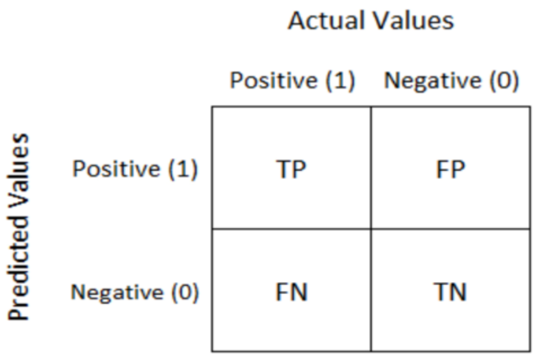


Fig. 4. DDoS Attack in IoV.



		Actual Values	
		Positive (1)	Negative (0)
Predicted Values	Positive (1)	TP	FP
	Negative (0)	FN	TN

Fig. 5. Confusion matrix [24].

G. Data Oversampling

Sampling is the most used method to solve the problem of class imbalance. The process of data sampling involves creating a data set by adjusting the number of samples of the majority class in the unbalanced data set and it occupies the largest part while the minority class occupies the smallest part. The sampling method is classified as a reduction or oversampling method, depending on which of the two categories is the number of samples [27].

• Random oversampling Random oversampling is done by increasing the samples of the minority group randomly, which means increasing the cases corresponding to the minority group by repeating them at a certain rate. It is considered an additional advantage as it does not cause the loss of any information. (a) Oversampling increases the number of instances of the training set, and random oversampling increases the training time of the model [28], [29]. Algorithm 1 shows the random sampling for the initialization of the backing sample.

H. MinMax Scaler

MinMax Scaler is one of the most popular scaling algorithms. The main idea of the linear data conversion algorithm where the algorithm assigns the value of V from the variable to the value of V using the formula:

$$X_{scaled} = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{8}$$

The goal is to measure the variable MinMax in the interval [0,1] using linear assignment, meaning that the minimum and maximum value of a feature/variable is going to be 0 and 1, respectively [30].

I. Feature Extraction

The feature plays a big and important role in the performance of the model. Excluding or including features leads to the deterioration or improvement of the model. Accordingly, the features are the only ones relevant to the improvement of the model. The main objective of the classification is to know the benign and malicious traffic. The model is trained using the selected features before the training ends. The K1Fold is validated to divide the model into training and testing and also serves to help evaluate the model. The model is divided into five groups of equal size, four groups are trained, and one group is tested. The process is repeated ten times. The performance measures used in the model are feature selection. Reducing the number of features contributes to reducing the processing time that machine learning algorithms take. We can calculate the Chi-square between each element and the target, then select the ideal number of features with the best Chi square scores [31], [32].

$$FE = REM + RMD + DER \tag{9}$$

Were,

- FE Feature Extraction as shown in Algorithm 3,
- REM: Review Existing model,
- RMD: Remove missing data, and
- DER: Domain expert review

The argmax function returns the index of the element in the list that has the maximum value. You can use any appropriate performance metric to evaluate the models, such as accuracy, precision, recall, F1-score.

V. Experiments

In this section, we will learn how to measure the effectiveness of the security model, as the study was based on the efficiency of the proposed model in detecting security attacks. We analyzed the results of the attacks through a typical ML machine learning model where the features are extracted from the raw data and tested by the ML model. The CICDDoS2019 dataset is then tested and predicted, and the working efficiency of the ML model is evaluated.

Algorithm 1 Feature Extraction to optimize features
Input: A large Number of Features
Output: Optimized Features

```
1) Start
2) Extract Datasets
3) Delete missing data, Feature selection using domain expert
4) Data pre-processing
5) Use 10-fold cross-validation.
6) While all data sets are trained and test
  a. Split data into k-5 and 10-fold cross-validation.
  b. Model fitting
  c. Model Evaluation
7) End while
8) End
```

A. Models Implementation

ML models and configurations are evaluated based on evaluation scales: TP represents the true positives; TN represents the true negatives through the criteria.

1) Data preprocessing: Processing the dataset is the main stage before entering the data into the ML to achieve high performance. There are many challenges in the dataset such as missing values, categorical features, and class imbalance. Also, useless features may affect the performance of the selected ML.

2) Feature selection: Feature selection is necessary to detect intrusions, get the best score for the prospective feature, and choose the best. Where the different features should be checked gives a positive and negative category for each of them and thus get rid of the useless ones to improve the performance. The feature is selected using Chi2 technology, as it achieves better performance for many classification problems. A selection strategy is used to exclude the features using the null hypothesis. A higher Chi2 value means that the feature is more significant [33].

$$x^2 = \sum_{i=1}^{m}\sum_{j=1}^{n}(\frac{O_i - E_i}{E_i})^2 \tag{10}$$

Where: m represents the number of features, and n represents the number of classes and O_i is any observed frequency and E_i expected frequency [34].

3) Data normalization: The numerical values in the dataset pose a challenge to the classifier during training. Maximum values must be set for each property within the range of (0, 1). Values outside the range can lead to incorrect results, as the technique may skew to the higher advantage. Data normalization plays a vital role in outperforming features with higher values over features with lower values. The data is oversampled to balance the class distribution, as presented in Eq [35], [36]

$$Z = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{11}$$

where x is the feature value, Z is the value after normalization, and x_max and x_min are the maximum and minimum values of the feature.

4) Data cleaning: The CICDDoS2019 dataset contains missing values and infinite values. The values are processed in two ways: In the second dataset, the infinite values are replaced by extreme values, and the missing values are replaced by averages. Only attack information packets were used to evaluate the proposed approach. Data packets representing normal network traffic are discarded in both groups, which improves accuracy and reduces computing time.

B. Proposed Models

In the dataset, the selected methods were used for training and tested by different parameters in feature engineering for intrusion detection. We selected different workbooks using: Accuracy, Recall, Precision, and F1 point. The methods used have shown strong performance in creating IDS. We explore the following strategies: K- Nearest Neighbor (KNN), DT, NB, SVM, RF, and LR.

C. Experiments

The CICDDoS2019 dataset and ML machine learning models were used to detect DDoS attacks. The implementation was done using Python 3.10 with many libraries such as Pandas, NumPy, Seaborn, and Matplotlib.pyplot.

VI. Results and Discussions

In this section, we review all the features for analyzing system performance, detecting events that are not compatible with normal behavior, confirming auditing and examining this data, and quality measures for the fully utilized ML model to be able to take a proactive step to avoid potential damage to vehicular Internet networks. Outstanding results appear in the field of discovering anomalies in time series data due to its ability to learn patterns and complex behavior. Therefore, it is the appropriate solution to distinguish deviant behavior from normal behavior.

A. Results Measurement Formulas

- Accuracy: It is responsible for evaluating classification models by depicting the proportion of correct predictions in the dataset, and is based on:
$$Accuricy = \frac{TP + TN}{TP + TN + FP + FN} \tag{12}$$
- Recall: measures the ratio of correctly identified labels to the total number of instances and is based on the following:
$$Recall = \frac{TP}{TP + FN} \tag{13}$$
- Precision: measures the ratio of correctly selected labels to the total number of positive ratings:
$$Precision = \frac{TP}{TP + FP} \tag{14}$$
- F1: points measure the harmonic mean of precision and recall [37].

$$F1 = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{15}$$

B. Results Analysis

Exploit-based attacks are attacks in which the attacker's identity is kept hidden by a third party. Packets are sent by the attacker to mirroring servers with the source IP address changed to the target victim's IP to confuse it. These attacks are carried out through transport layer protocols such as TCP and UDP. These include exploits based on SYN floods and flooding attacks such as UDP floods. The dataset includes CICDDoS2019 token 25 which consists of UDP, and SYN traffic. It is used to analyze system performance and discover events that are not consistent with the normal behavior of the network. Through mathematical models of ML algorithms: LR, KNN, DT, NB, RF, and SVM. we trained the models and performed validation to calculate the evaluation metrics.

C. Description of Network Attacks

- UDP Lag: UDP Lag attack is an attack that disrupts file communication between a client and a server. The attack can be carried out in two ways: through hardware switching, known as delay switching, or through software running on the network and consuming the bandwidth of others. It involves a special UDP stream that consumes more bandwidth while decreasing the number of packets.
- SYN Flood: In addition, SYN Flood is a type of TCP flood that targets the initial handshake of the TCP connection. The SYN flood sends a large volume of packets to the target server.

D. Dataset Scenarios

The files contain all the packets, and the CSV files provide a simpler way to load the data. These files consist of features extracted from the original pcap and are fixed- size files. The files are converted from pcap to CSV by capturing all sides of the network traffic data. Along with the innocuous packets, the traffic is then broken down into smaller data through parallel conversion using TCP Dump. The features are then extracted using chi2 and stored in separate CSV files. The extracted features are used to aggregate the captured values to reduce discrepancies in data size.

E. Results Discussion

In this section, we present the evaluation of the performance of classification algorithms, namely LR, KNN, DT, NNB, RF, and SVM models.

We trained the models and performed validation to calculate the evaluation metrics. The evaluation scheme is a performance evaluation, as it determines the efficiency and robustness of the proposed scheme. A dataset with identical characteristics is needed for real traffic and DDoS traffic flows, so we evaluated the performance of classification algorithms using the CICDDoS2019 dataset. The performance of the six-model considering UDP-Lag attack is shown in Fig. 6.

We trained the models and performed validation to calculate the evaluation metrics. The evaluation scheme is a performance evaluation, as it determines the efficiency and robustness of the proposed scheme. A dataset with identical characteristics is needed for real traffic and DDoS traffic flows, so we evaluated the performance of classification algorithms using the CICDDoS2019 dataset.

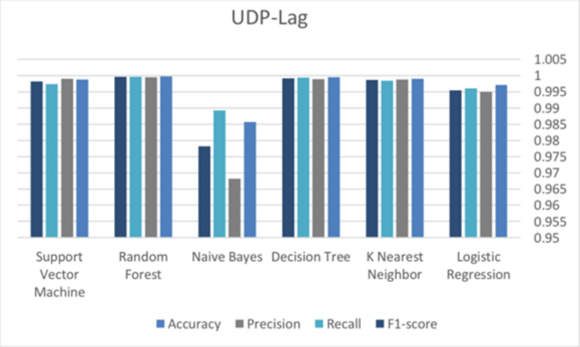


Fig. 6. Performance for proposed ML models for UDP-Lag attack.

We adopted six ML models for binary classification (benign or malicious). The results showed high accuracy in Random Forest, k Nearest Neighbor algorithm, and Decision Tree. These results demonstrate how ML models can be used to classify attacks against IoV. These models may face challenges in classifying other attacks as benign or malicious, and despite the similarity in patterns, the classification is successful. The accurate results are shown in Table I.

TABLE I. PROPOSED ML MODELS RESULTS FOR UDP-LAG ATTACK

Model + chi2 FE	Accuracy	Precision	Recall	F1score
LR	0.9950	0.992	0.9856	0.9875
LR+chi2	0.9954	0.996	0.9949	0.9971
KNN	0.9976	0.9974	0.9967	0.989
KNN+chi2	0.9986	0.9984	0.9987	0.999
DT	0.9971	0.9954	0.9949	0.9925
DT+chi2	0.9991	0.9994	0.9989	0.9995
NB	0.9722	0.9792	0.9661	0.9817
NB+chi2	0.9782	0.9892	0.9681	0.9857
KNN	0.9980	0.9924	0.9957	0.991
KNN+chi2	0.9986	0.9984	0.9987	0.999
RF	0.9976	0.9962	0.9943	0.9972
RF+chi2	0.9996	0.9996	0.9995	0.9997
SVM	0.9962	0.9943	0.991	0.9916
SVM+chi2	0.9982	0.9973	0.999	0.9988

The best ML models tested in the UDP Lag attack outperformed. The DT model, and RF model had the best results with a precision, recall, and F1 score of 99.9%. For the SYN flood, the performance of the six models is presented in Fig. 7.

In the SYN flood attack, the best tested ML models appeared superior, with KNN, DT, and RF models having the best results with 99.9% precision, recall, and F1-score. The details results are shown in Table II.

The confusion matrix plays an important role in determining performance. The Confusion matrix for UDP-Lag and SYN Flood are shown in Fig. 8 and 9 .

VII. CONCLUSIONS AND FUTURE WORK

We presented a new and large-scale IoV data set for the training and evaluation of threat detection systems. The results reveal high response rates for the models with the selected features. A system based on ML models has been developed

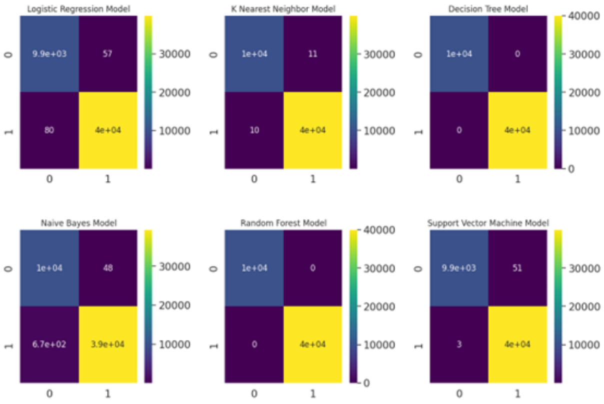


Fig. 8. Confusion matrix for UDP lag attack.

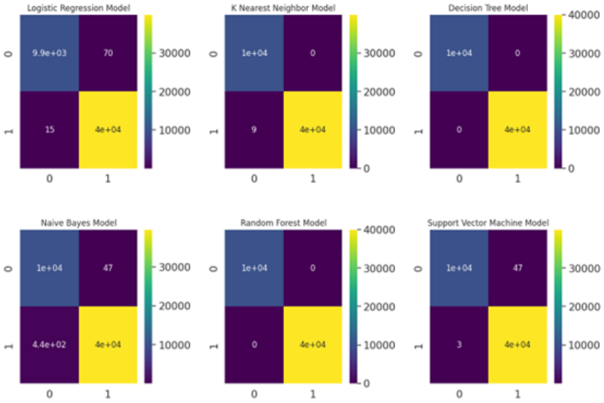


Fig. 9. Confusion matrix for SYN flood attack.

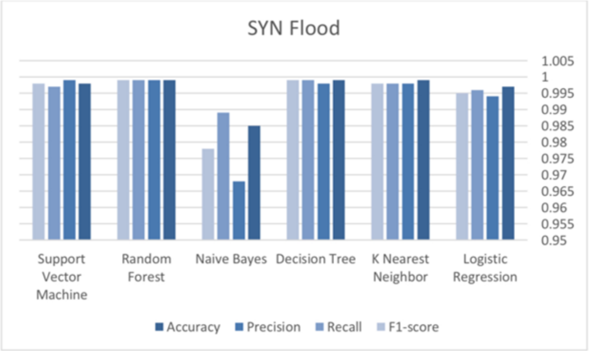


Fig. 7. Performance for proposed ML models for SYN flood attack.

TABLE II. PROPOSED ML MODELS RESULTS FOR SYN FLOOD ATTACK

Model + chi2 FE	Accuracy	Precision	Recall	F1score
LR	0.9970	0.9972	0.9943	0.9932
LR+chi2	0.9982	0.9982	0.9963	0.9972
KNN	0.9976	0.9982	0.9967	0.9955
KNN+chi2	0.9996	0.9992	0.9997	0.9995
DT	0.9990	0.9975	0.9898	0.9796
DT+chi2	0.9998	0.9995	0.9998	0.9996
NB	0.9900	0.9778	0.9901	0.9819
NB+chi2	0.9902	0.9781	0.9921	0.9849
RF	0.9990	0.9985	0.9898	0.9976
RF+chi2	0.9997	0.9995	0.9998	0.9996
SVM	0.9969	0.9970	0.9963	0.9970
SVM+chi2	0.9989	0.9990	0.9975	0.9982

to prevent communication errors that could cause traffic disruptions or accidents between networks and interconnected vehicles. Development of IoV protection technologies and increased security investment. Ensuring security for vehicle exchange data storage and infrastructure. For the UDP Lag, DT, and RF models had the best results with a precision, recall, and F1 score of 99.9%. In the SYN flood attack, the best tested ML models appeared superior, with KNN, DT, and RF having the best results with 99.9% precision, recall, and F1score. This work opens the door to the development of many future endeavors. For example, optimizing ML models, analyzing features and their impact on different ML models, interpreting ratings, and assessing portability based on comparisons to other datasets.

ACKNOWLEDGMENT

The authors are thankful to the Deanship of Graduate Studies and Scientific Research at University of Bisha for supporting this work through the Fast-Track Research Support Program.

REFERENCES

[1] A. Arooj, M. S. Farooq, A. Akram, R. Iqbal, A. Sharma, and G. Dhiman, “Big data processing and analysis in internet of vehicles: architecture, taxonomy, and open research challenges,” *Archives of Computational Methods in Engineering*, vol. 29, no. 2, pp. 793–829, 2022.

[2] L. Yadav, S. Kumar, A. KumarSagar, and S. Sahana, “Architechture, applications and security for iov: A survey,” in *2018 International Conference on Advances in Computing, Communication Control and Networking (ICACCCN)*. IEEE, 2018, pp. 383–390.

[3] S. J. Kamble and M. R. Kounte, “Machine learning approach on traffic congestion monitoring system in internet of vehicles,” *Procedia Computer Science*, vol. 171, pp. 2235–2241, 2020.

[4] A. Samad, S. Alam, S. Mohammed, and M. Bhukhari, “Internet of vehicles (iov) requirements, attacks and countermeasures,” in *Proceedings of 12th INDIACom; INDIACom-2018; 5th international conference on “computing for sustainable global development” IEEE conference, New Delhi*, 2018, pp. 1–4.

[5] N. Hafsa, S. Rushd, M. Al-Yaari, and M. Rahman, “A generalized method for modeling the adsorption of heavy metals with machine learning algorithms,” *Water*, vol. 12, no. 12, p. 3490, 2020.

[6] A. Talpur and M. Gurusamy, “Machine learning for security in vehicular networks: A comprehensive survey,” *IEEE Communications Surveys & Tutorials*, vol. 24, no. 1, pp. 346–379, 2021.

[7] O. Kaiwartya, A. H. Abdullah, Y. Cao, A. Altameem, M. Prasad, C.-T. Lin, and X. Liu, “Internet of vehicles: Motivation, layered architecture, network model, challenges, and future aspects,” *IEEE access*, vol. 4, pp. 5356–5373, 2016.

[8] H. Xi and H. Sun, “Resource allocation strategy of internet of vehicles using reinforcement learning.” *Journal of Information Processing Systems*, vol. 18, no. 3, 2022.

[9] J. A. Fadhil and Q. I. Sarhan, “Internet of vehicles (iov): a survey of challenges and solutions,” in *2020 21st International Arab Conference on Information Technology (ACIT)*. IEEE, 2020, pp. 1–10.

[10] K. Aswal, D. C. Dobhal, and H. Pathak, “Comparative analysis of machine learning algorithms for identification of bot attack on the internet of vehicles (iov),” in *2020 International Conference on Inventive Computation Technologies (ICICT)*. IEEE, 2020, pp. 312–317.

[11] T. A. Butt, R. Iqbal, S. C. Shah, and T. Umar, “Social internet of vehicles: Architecture and enabling technologies,” *Computers & Electrical Engineering*, vol. 69, pp. 68–84, 2018.

[12] E. C. P. Neto, S. Dadkhah, R. Ferreira, A. Zohourian, R. Lu, and A. A. Ghorbani, “Ciciot2023: A real-time dataset and benchmark for large-scale attacks in iot environment,” 2023.

[13] T. E. Ali, Y.-W. Chong, and S. Manickam, “Machine learning techniques to detect a ddos attack in sdn: A systematic review,” *Applied Sciences*, vol. 13, no. 5, p. 3183, 2023.

[14] H. J. Hadi, U. Hayat, N. Musthaq, F. B. Hussain, and Y. Cao, “Developing realistic distributed denial of service (ddos) dataset for machine learning-based intrusion detection system,” in *2022 9th International Conference on Internet of Things: Systems, Management and Security (IOTSMS)*. IEEE, 2022, pp. 1–6.

[15] Z. Li, Y. Kong, C. Wang, and C. Jiang, “Ddos mitigation based on space-time flow regularities in iov: A feature adaption reinforcement learning approach,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 3, pp. 2262–2278, 2021.

[16] N. Bindra and M. Sood, “Detecting ddos attacks using machine learning techniques and contemporary intrusion detection dataset,” *Automatic Control and Computer Sciences*, vol. 53, pp. 419–428, 2019.

[17] A. Thakkar and R. Lohiya, “A review of the advancement in intrusion detection datasets,” *Procedia Computer Science*, vol. 167, pp. 636–645, 2020.

[18] I. Sharafaldin, A. H. Lashkari, S. Hakak, and A. A. Ghorbani, “Developing realistic distributed denial of service (ddos) attack dataset and taxonomy,” in *2019 International Carnahan Conference on Security Technology (ICCST)*. IEEE, 2019, pp. 1–8.

[19] T. Zhang, C. Xu, P. Zou, H. Tian, X. Kuang, S. Yang, L. Zhong, and D. Niyato, “How to mitigate ddos intelligently in sd-iov: a moving target defense approach,” *IEEE Transactions on Industrial Informatics*, vol. 19, no. 1, pp. 1097–1106, 2022.

[20] I. Wickramasinghe and H. Kalutarage, “Naive bayes: applications, variations and vulnerabilities: a review of literature with code snippets for implementation,” *Soft Computing*, vol. 25, no. 3, pp. 2277–2293, 2021.

[21] A. R. Lubis, M. Lubis *et al.*, “Optimization of distance formula in k-nearest neighbor method,” *Bulletin of Electrical Engineering and Informatics*, vol. 9, no. 1, pp. 326–338, 2020.

[22] B. Charbuty and A. Abdulazeez, “Classification based on decision tree algorithm for machine learning,” *Journal of Applied Science and Technology Trends*, vol. 2, no. 01, pp. 20–28, 2021.

[23] H. Narayanan, M. Sokolov, A. Butté, and M. Morbidelli, “Decision tree-pls (dt-pls) algorithm for the development of process: Specific local prediction models,” *Biotechnology progress*, vol. 35, no. 4, p. e2818, 2019.

[24] E. Y. Boateng, J. Otoo, and D. A. Abaye, “Basic tenets of classification algorithms k-nearest-neighbor, support vector machine, random forest and neural network: a review,” *Journal of Data Analysis and Information Processing*, vol. 8, no. 4, pp. 341–357, 2020.

[25] G. Nattino, M. L. Pennell, and S. Lemeshow, “Assessing the goodness of fit of logistic regression models in large samples: A modification of the hosmer-lemeshow test,” *Biometrics*, vol. 76, no. 2, pp. 549–560, 2020.

[26] M. Ghurab, G. Gaphari, F. Alshami, R. Alshamy, and S. Othman, “A detailed analysis of benchmark datasets for network intrusion detection system,” *Asian Journal of Research in Computer Science*, vol. 7, no. 4, pp. 14–33, 2021.

[27] K. Bouzoubaa, Y. Taher, and B. Nsiri, “Predicting dos-ddos attacks: Review and evaluation study of feature selection methods based on wrapper process,” *Int. J. Adv. Comput. Sci. Appl*, vol. 12, no. 5, pp. 131–145, 2021.

[28] P. J. Huang, *Classification of imbalanced data using synthetic oversampling techniques*. University of California, Los Angeles, 2015.

[29] I. Bolodurina, A. Shukhman, D. Parfenov, A. Zhigalov, and L. Zabrodina, “Investigation of the problem of classifying unbalanced datasets in identifying distributed denial of service attacks,” in *Journal of Physics: Conference Series*, vol. 1679, no. 4. IOP Publishing, 2020, p. 042020.

[30] S. Park and H. Park, “Combined oversampling and undersampling method based on slow-start algorithm for imbalanced network traffic,” *Computing*, vol. 103, no. 3, pp. 401–424, 2021.

[31] S. Solanki, V. Dehalwar, and J. Choudhary, “Cooperative spectrum sensing for pu detection in cognitive radio using svm,” in *Data Engineering and Communication Technology: Proceedings of ICDECT 2020*. Springer, 2021, pp. 61–69.

[32] L. Munkhdalai, T. Munkhdalai, K. H. Park, H. G. Lee, M. Li, and K. H. Ryu, “Mixture of activation functions with extended min-max normalization for forex market prediction,” *IEEE Access*, vol. 7, pp. 183 680–183 691, 2019.

[33] U. Shrestha, A. Alsadoon, P. Prasad, S. Al Aloussi, and O. H. Alsadoon, “Supervised machine learning for early predicting the sepsis patient: modified mean imputation and modified chi-square feature selection,” *Multimedia Tools and Applications*, vol. 80, pp. 20 477–20 500, 2021.

[34] C. Ioannou and V. Vassiliou, “Accurate detection of sinkhole attacks in iot networks using local agents,” in *2020 Mediterranean Communication and Computer Networking Conference (MedComNet)*. IEEE, 2020, pp. 1–8.

[35] D.-C. Li, S.-Y. Wang, K.-C. Huang, and T.-I. Tsai, “Learning class-imbalanced data with region-impurity synthetic minority oversampling technique,” *Information Sciences*, vol. 607, pp. 1391–1407, 2022.

[36] A. R. Gad, A. A. Nashat, and T. M. Barkat, “Intrusion detection system using machine learning for vehicular ad hoc networks based on ton-iot dataset,” *IEEE Access*, vol. 9, pp. 142 206–142 217, 2021.

[37] A. Thakkar and R. Lohiya, “Attack classification using feature selection techniques: a comparative study,” *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, pp. 1249–1266, 2021.