Spectrum of Engineering Sciences
ISSN (e) 3007-3138 (p) 3007-312X Volume 4, Issue 3, 2026

A DECENTRALIZED NETWORK INTRUSION DETECTION SYSTEM FOR DISTRIBUTED DENIAL OF SERVICE ATTACK PREVENTION IN VEHICULAR AD HOC NETWORKS

Muhammad Shoaib Rasheed[1], Tahir Abbas[*2], Sadia Tariq[2], Nazir Ahmad[3], Ehsan Ul Haq[4], Nauman Sultan[5], Zaiba Aziz[6], Muhammad Arslan[7]

[1,3,4,5,6]*Department of Computer Science, National College of Business Administration & Economics Lahore, Multan Sub Campus, 60000, Pakistan*
[2]*Department of Communication and Cyber Security, Bahauddin Zakariya University, Multan, 60000, Pakistan*
[7]*Department of Computer Science,gc University Faislabad Layyah Campus, 60000, Pakistan*

[1]bukhari8193@gmail.com, [2]tahir.abbas@bzu.edu.pk, [3]sadia.ranatahir@gmail.com, [4]nazirgohar79@gmail.com, [5]ehsan412@gmail.com, [6]maliknomimaliknomi555@gmail.com, [7]arslandrigh668@gmail.com

DOI: http://doi.org/10.5281/zenodo.20019460

Keywords
VANETs; Decentralized Intrusion Detection; DDoS, Big Data Analytics; Random Forest; Apache Spark; HDFS; Intelligent Transportation Systems.

Article History
Received: 03 January 2026
Accepted: 13 March 2026
Published: 25 March 2026

Corresponding Author: *
Tahir Abbas

Abstract
The Vehicular-Ad-Hoc Networks (VANETs) have serious security problems because wireless links are often weak and unstable. One of the biggest threats is a Distributed Denial of Service (DDoS) attack that floods the network with fake traffic and stops real users from using important services. To protect VANETs from these attacks we need intrusion detection systems that are decentralized and can handle large amounts of data. The centralized systems usually fail because they slowdowns and depend on one main control point. This research study proposed Decentralized Network Intrusion Detection System (DNIDS) to save VANETs from DDoS attacks using big data tools and distributed learning. The system has two main parts that are a traffic collection module and a detection module. The collection module gathers network data from different nodes through micro-batch processing in real time. The detection module applies Random Forest (RF) classifier that is built on Apache Spark with the Hadoop Distributed File System (HDFS) to store and analyze attack data efficiently. This system was tested with the NSL-KDD and UNSW-NB15 datasets and attain detection accuracies upto 99.95% and 98.75% with false alarm rates upto 0.05% and 1.08% respectively. These results are better than many base line methods and prove that a decentralized design works well for keeping VANETs secure.

1. INTRODUCTION

The VANETs are a fundamental part of Intelligent Transportation Systems (ITS). They let vehicles and roadside units to communicate directly to improve safety, manage traffic and assist drivers (Guerna et al., 2022; Halawani, 2025). The applications like collision alerts, cooperative driving and emergency messages depend on fast and steady data exchange (Fu et al., 2021). But since VANETs are highly dynamic so they face many security risks (Mahi et al., 2023). One of the most serious risk is the Distributed Denial of Service (DDoS) attack that floods the network and

reduces its ability to send important safety messages (Ali et al., 2022; Karthikeyan & Usha, 2022). To maintain stable and reliable communication intrusion detection systems must scale well and respond quickly (Santhosh Kumar et al., 2023).

A. Motivation

The most traditional Intrusion Detection Systems (IDS) use a centralized design where all traffic data is collected and analyzed in one place (Goyal et al., 2024). The traditional IDS works well in fixed or wired networks but not in vehicle networks because vehicles move constantly and network conditions change quickly so the centralized systems face delays, bottlenecks and a single point of failure. To handle these challenges. A decentralized detection system is needed that can process distributed data and react quickly to attacks (Aminu et al., 2024).

1. **Scalability:** The fast movement and dense communication among vehicles create large volumes of data. A single central server cannot analyze this information in real time without bottlenecks (Fang et al., 2025).
2. **Single point of failure:** The central IDS node becomes a high-value target. A single DDoS attack could disable the entire protection layer (Kothandapani, 2023).
3. **Latency:** Many vehicular applications are time-sensitive. Even short delays in detecting malicious traffic can interrupt warnings or emergency signals (Lone et al., 2024). The DDoS threats become more severe in VANETs because attackers exploit both vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communication. The compromised vehicles or roadside units can flood legitimate nodes with fake messages overwhelm nearby vehicles or target network infrastructure (Malik, 2024). For this reason, detecting and identifying DDoS in a large flow of network data remains complicated challenge. The common mechanisms to detect and prevent DDoS include attack prevention, attack detection and attack reaction (Dalmazo et al., 2021; Tehaam et al., 2022). A NIDS has become a essential part of network security(Sohi et al., 2021). It operates by monitoring and analyzing network behavior to detect the abnormal use(Mavaluru et al., 2023). The main type of NIDS such as signature based NIDS that detects attacks by comparing traffic against known signatures of prior attacks. This method is not very effective for DDoS because attackers often change their methods which makes it difficult to define a fixed pattern (Feraudo et al., 2024).

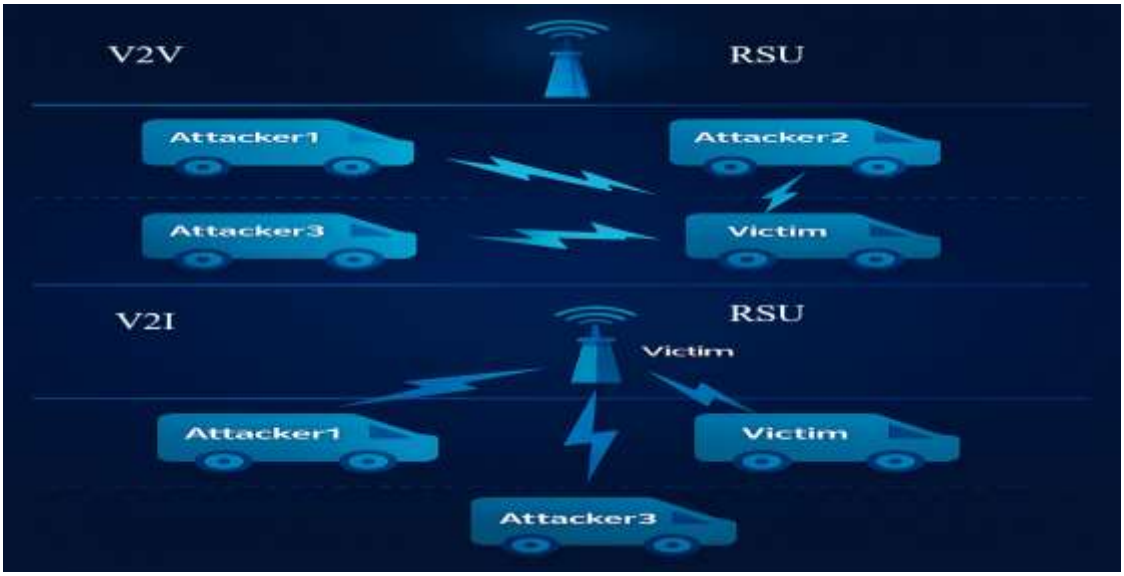

Figure 1. DDOS Attacks launched in VANET through two patterns of communication.

The Figure 1 shows how both channels are used. In V2V attacks infected vehicles send continuous malicious packets to nearby nodes. In V2I attacks the adversaries overload roadside units to block critical services. Together, these attack paths highlight why VANETs need distributed and adaptive detection systems (Farsimadan et al., 2025).

B. Problem Definition

This research study addresses the following research question

How can a scalable, decentralized and accurate intrusion detection framework to be designed to overcome DDoS attacks in VANETs?

Key challenges include

1. Managing large volumes of high-speed traffic data in real time (Abdel-Aty et al., 2023).
2. Achieving high detection accuracy with minimal false alarms (Jiao et al., 2024).
3. Adapting to the dynamic and mobile nature of VANET environments (Benkirane et al., 2023).
4. Scaling effectively across large vehicular networks (Zhang et al., 2025).

C. Contributions

The main contributions are as follows

Decentralized NIDS design: The research introduces a DNIDS that spreads data collection and analysis across several nodes and reducing reliance on a single central server (Heidari & Jabraeil Jamali, 2023).

Real-time micro-batch processing: The system includes a data collection module that uses micro-batch processing to quickly extract important features from fast-moving network traffic (Olayinka, 2021).

Random Forest detection on Apache Spark: The detection module applies a RF algorithm built on Apache Spark. This setup depends on in-memory computing to make the classification process faster and more accurate (Dritsas & Trigka, 2025).

Experimental evaluation: The proposed framework is tested on two well-known datasets that are NSL-KDD, UNSW-NB15 and its results are compared with existing intrusion detection methods (Ahmad et al., 2021).

Parameter optimization study: The paper also studies how different Random Forest parameters such as sampling rate, tree depth and the number of trees affect detection accuracy and providing useful insights for real-world use (Salman et al., 2024).

D. Structure of the Paper

The paper is structured as follows

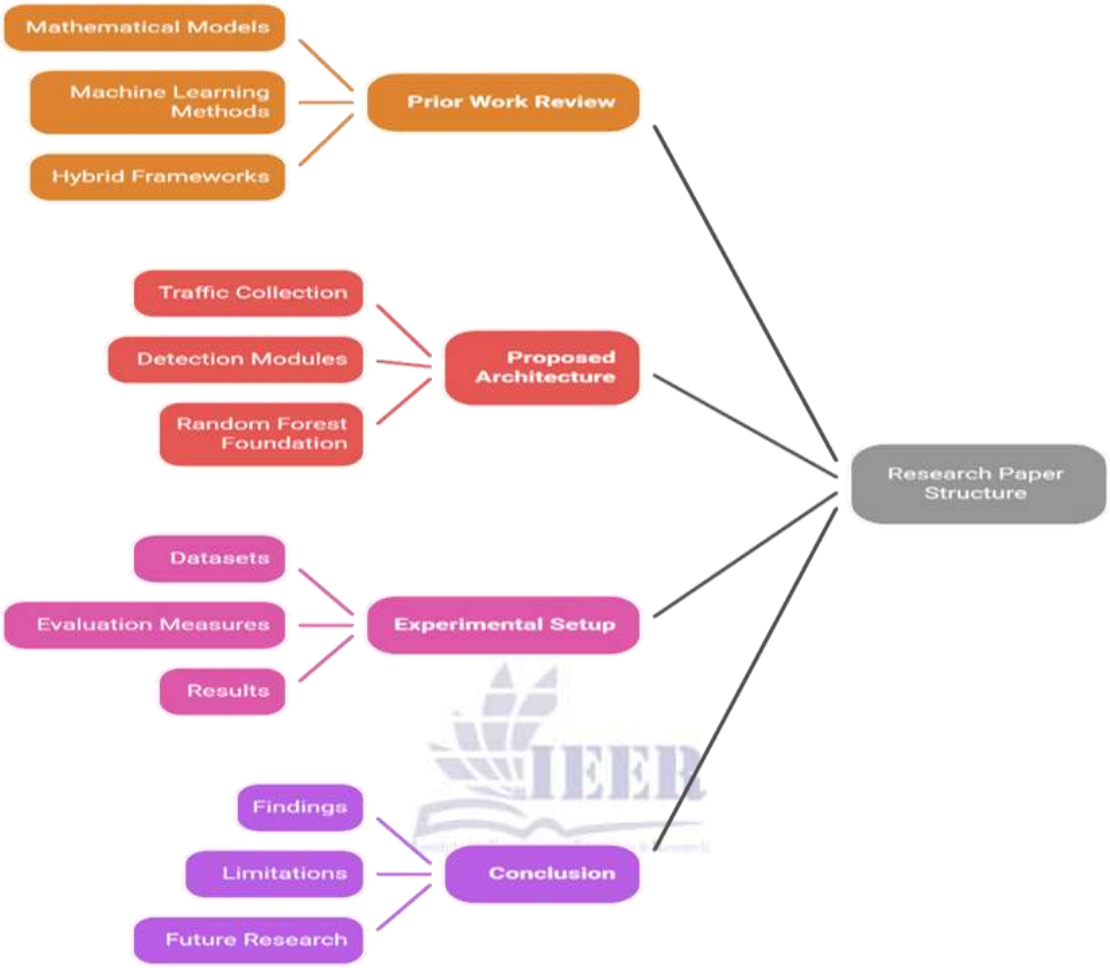

Figure 2. Illustrate main structure of research study on DDoS detection in VANETs

In Section 2 prior work on DDoS detection in VANETs is reviewed and broader intrusion detection covering mathematical models, machine learning methods, and hybrid frameworks. It details the proposed decentralized NIDS architecture, including the traffic collection and detection modules and the Random Forest (RF) foundation. Section 3 presents the experimental setup, datasets, evaluation measures and results with comparisons and parameter optimization. Section 4 presents the effectiveness of the proposed **DNID.** Section 5 concludes with findings, limitations and directions for future research. Figure 2 shows structure of this study.

2. Materials and Methods

In the past years a wide range of methods have been proposed for detecting and reducing Distributed Denial-of-Service (DDoS) attacks. These approaches are generally classified into three groups such as mathematical methods, machine learning methods and hybrid methods that integrate big data frameworks with learning models (Hussain et al., 2024; Masood et al., 2024; Tariq et al., 2025). Each group contributes important insights but each also faces limits in scalability, accuracy or adaptability in dynamic VANET environments.

A. Mathematical Methods

The mathematical methods depend on statistical measures and traffic patterns to tell normal network activity from attacks. A hop count-based filtering method that uses an IP-to-hop count (IP2HC) table was proposed by (Govindaraj et al., 2021). Their approach reached about 90% accuracy in finding malicious traffic still it can be tricked by advanced distributed attacks and may flag normal packets as threats if the IP2HC table is not kept up to date. The entropy and chi-square tests to study packet distributions and spot unusual behavior applied by (Nishiuchi et al., 2023) these tests work well in theory but their accuracy drops when attackers create traffic that looks similar to normal network activity.

B. Machine Learning Methods

The machine learning methods can recognize complex patterns in large datasets that makes them useful for intrusion detection. The machine learning methods can recognize complex patterns in large datasets that makes them useful for intrusion detection used flow mining to detect unusual network activity and showed that their approach worked well against early worms like (Ahmad et al., 2025; Sarker, 2021) developed a deep defense model that used data mining on alerts gathered from distributed intrusion prevention systems. Their method achieved high detection accuracy with few false alarms. The other researchers have also applied machine learning in this field (ul Haq et al., 2025; Zou et al., 2005) used Support Vector Machines (SVM) to predict anomalies while (Yokkampon et al., 2021) created a decision tree-based system C4.5 to detect DDoS attacks (Shanthi et al., 2022). Although these techniques improve accuracy but they depend on proper feature selection and often struggle to handle large-scale data without distributed computing support (Dustdar et al., 2022).

C. Hybrid Approaches with Big Data and Machine Learning

The hybrid methods combine big data platforms with advanced machine learning models to handle the large size and fast speed of modern network traffic (Seydali et al., 2024) used Hadoop and HBase to manage unstructured traffic data and applied a neural network for intrusion detection (Hamedani et al., 2021) improved performance by replacing Hadoop's MapReduce with Apache Spark and used Artificial Neural Networks (ANNs) for classification (Brahmane & Krishna, 2021) tested several classifiers on Apache Spark and added fuzzy logic to automatically choose the model that performed best. Their results showed that tree-based methods such as decision trees and Random Forests provide higher accuracy than other algorithms. The Random Forest (RF) algorithm introduced by (Salman et al., 2024) that combines multiple decision trees using bagging and random feature sampling. It is widely used in intrusion detection because it manages noisy data well while keeping high accuracy (Rasheed et al., 2025).

D. Comparative Summary

Table 1 summarizes representative methods for DDoS detection, highlighting their strengths and limitations. This comparison shows that mathematical methods are lightweight but lack robustness. Machine learning methods improve detection rates but face scaling limits. Hybrid methods, particularly those using Apache Spark with Random Forests, offer the best balance between scalability and detection performance.

Table 1. Summary of DDoS Detection Approaches

Category	Method / Study	Strengths	Limitations
Mathematical	IP2HC mapping (Akano et al., 2024)	Simple; ~90% accuracy	Weak against distributed attacks; false positives
Mathematical	Entropy & chi-square (Menéndez, 2024)	Solid anomaly detection basis	Vulnerable to mimicry traffic; low adaptability

Machine Learning	Flow mining (Yezerets et al., 2025)	Effective on worm/DoS traces	Limited scalability
Machine Learning	Deep defense (Moore et al., 2023)	Strong detection, low false alarms	Computationally expensive
Machine Learning	SVM (Hosseinzadeh et al., 2021)	Early anomaly prediction	Sensitive to feature selection
Machine Learning	C4.5 decision tree (Shanthi et al., 2022)	Automated and interpretable	Struggles with complex attacks
Hybrid	Hadoop + NN (Pallamala & Rodrigues, 2022)	Handles unstructured data	High latency (MapReduce overhead)
Hybrid	Apache Spark + ANN (Dritsas & Trigka, 2025)	Faster, scalable	Costly, tuning required
Hybrid	Apache Spark + Fuzzy logic (Dritsas & Trigka, 2025)	Adaptive classifier choice	Added system overhead
Hybrid	Random Forest (RF) (Chai et al., 2021)	High accuracy, noise tolerance	Training cost increases with depth

E. Taxonomy of Detection Approaches

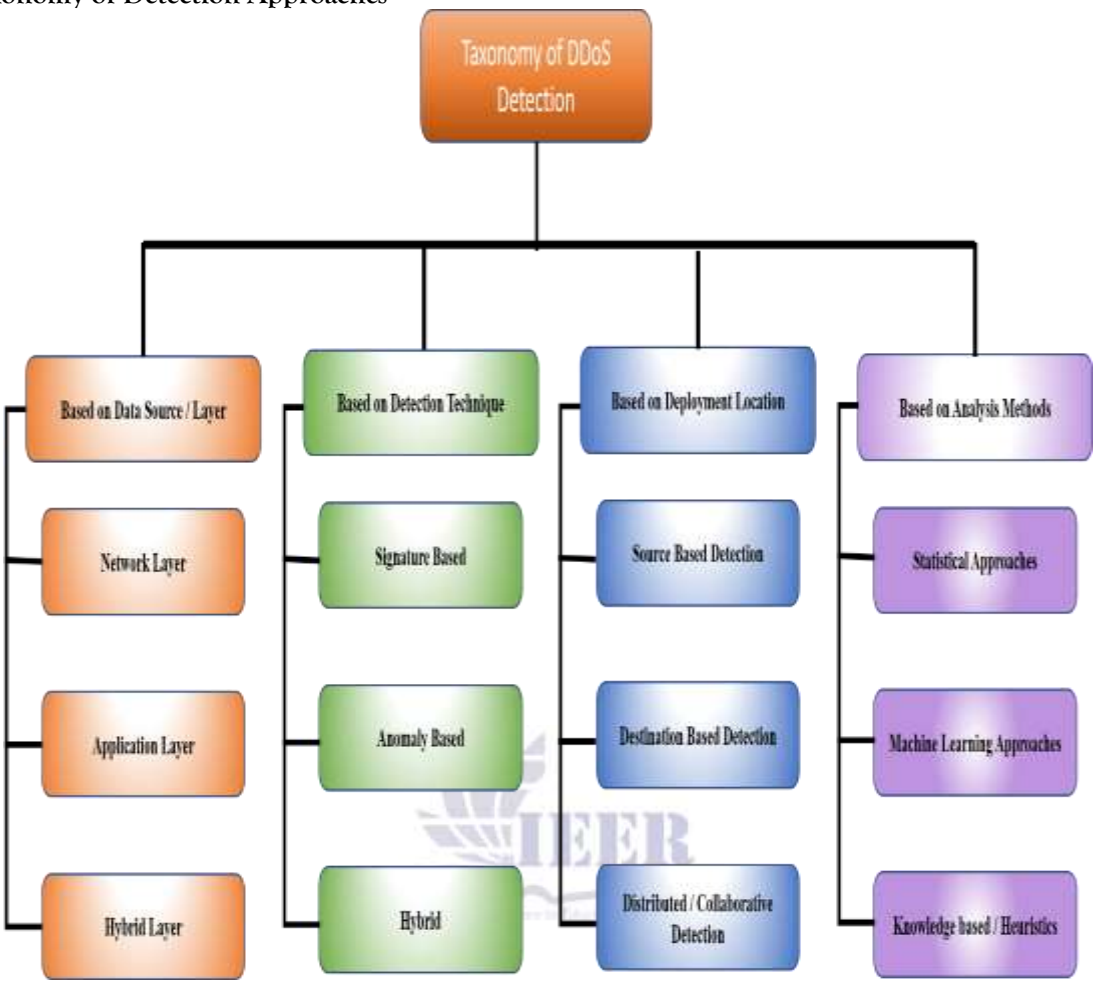

Figure 3. Taxonomy of DDoS detection methods: mathematical, machine learning, and hybrid approaches.

To illustrate this classification Figure 3 presents a taxonomy of DDoS detection approaches.

The review shows that both mathematical and machine learning methods are still important but hybrid approaches are becoming more popular. They combine the accuracy of learning models with the scalability of big data platforms. From this study it is clear that Random Forest-based hybrid systems running on distributed frameworks like Apache Spark are the most effective for detecting DDoS attacks in VANETs. Based on these findings the next section presents the proposed decentralized framework.

Proposed Decentralized NIDS for DDoS Detection

This section introduces the decentralized Network Intrusion Detection System (NIDS) designed to detect DDoS attacks in Vehicular Ad Hoc Networks (VANETs). Unlike centralized architectures, the system distributes both data collection and detection across multiple nodes to address the scale and velocity of VANET traffic. Figure 3 shows the overall architecture.

A. Collection Module

The collection module captures VANET traffic in real time and ensures scalability through three main components

Packet Collector: The Packet Collector is implemented at VANET nodes it uses LibPcap (Shaikh et al., 2024) to intercept packets. LibPcap provides a low-level API to capture Ethernet, IP, TCP and UDP traffic.

Message Queue (MQ): Since high-speed packet capture may overcome the system a distributed message queue buffers incoming traffic, separating collection from downstream parsing.

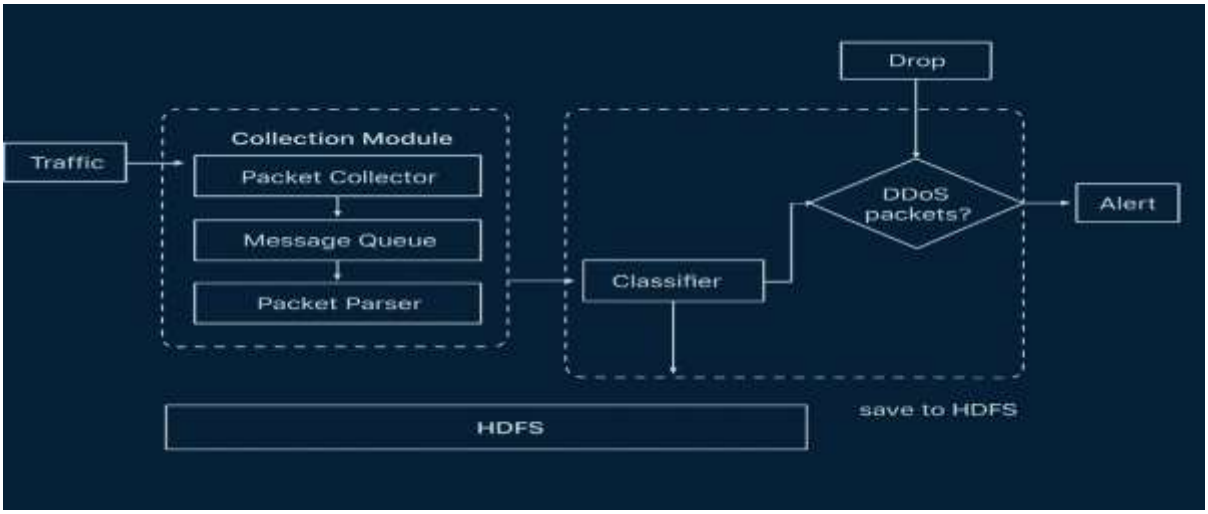

Figure 4. Architecture of the proposed decentralized NIDS for VANETs, consisting of traffic collection and detection modules on a distributed Apache Spark framework.

Packet Parser: Implements a micro-batch model. Packets are grouped into fixed-size windows (e.g., 1t seconds), parsed and transformed into statistical features (counts, averages, etc.) inspired by the KDD99 dataset (Mohammad et al., 2022). Figure 4 represents the collection modules communication.

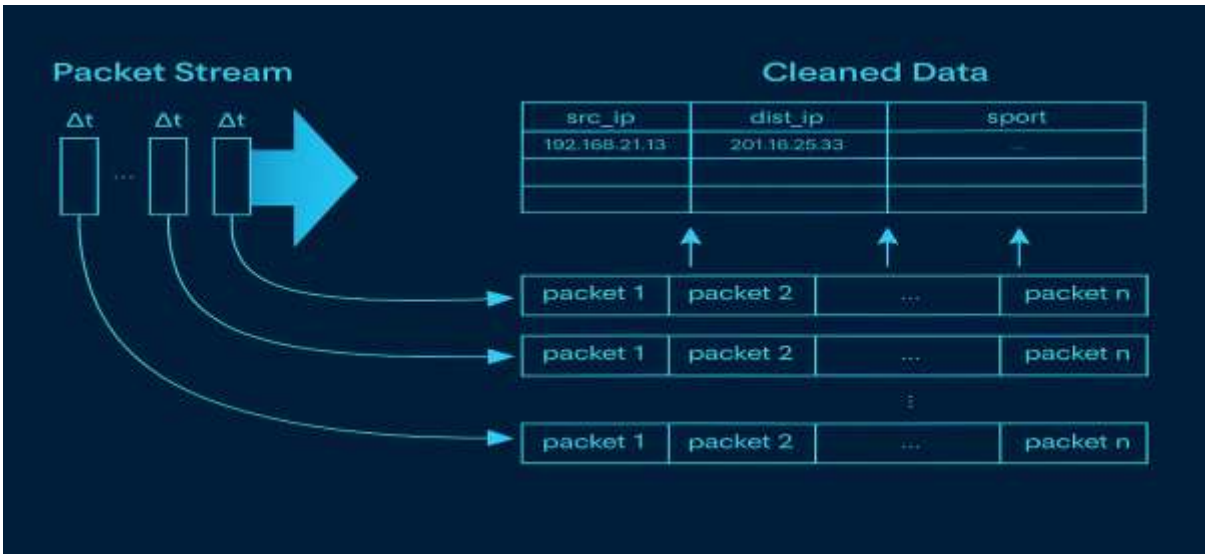

Figure 5. Packet stream processed in periodic micro batches.

The Figure 5 illustrates packet stream processing through periodic micro batches. The traffic is cached in a queue divided into small batches and then analyzed in near real time to detect DDoS activity. The network nodes handle large amounts of traffic. If the traffic is not processed on time the data at a node keeps building up and this affects the stability and availability of the system. To manage this we use the distributed message queue (MQ) (Maharjan et al., 2023) to temporarily store the collected traffic. The message queuing separates capturing packets from parsing them and this balances the differences between collecting and processing traffic data. The packet parser works with a micro – batch data processing model. The traffic packets are divided into small datasets using fixed time windows. All packets in one dataset are parsed together to extract their attributes. The packets captured over a Δt interval form one dataset. The extracted attributes are appended to the cleaned data Figure 3. The cleaned data is then used by the detection module. The Packet Parser algorithm starts with the cached packets as input. After initialization a tumbling window is created every Δt each window triggers a calculation. The packets in the window are retrieved from the MQ. Each packet is parsed according to the TCP/IP model. Attributes are extracted according to protocols such as Ethernet IP TCP and UDP. At the same time statistical attributes of traffic in the same window are calculated inspired by the KDD99 dataset (Mohammad et al., 2022). The basic attributes and the statistical attributes are then sent to the detection module for abnormality detection.

Algorithm 1. Feature Extraction in the Collection Module

Input: Traffic packets in binary format

1: loop

2: Load packets from MQ in window of size Δt → P

3: For i = 1 to n do (n = number of packets)

4: Parse packet(i) by protocol (Ethernet, IP, TCP, UDP)

5: Extract features and compute statistics

6: end for

7: end loop

Output: Structured packet dataset with features

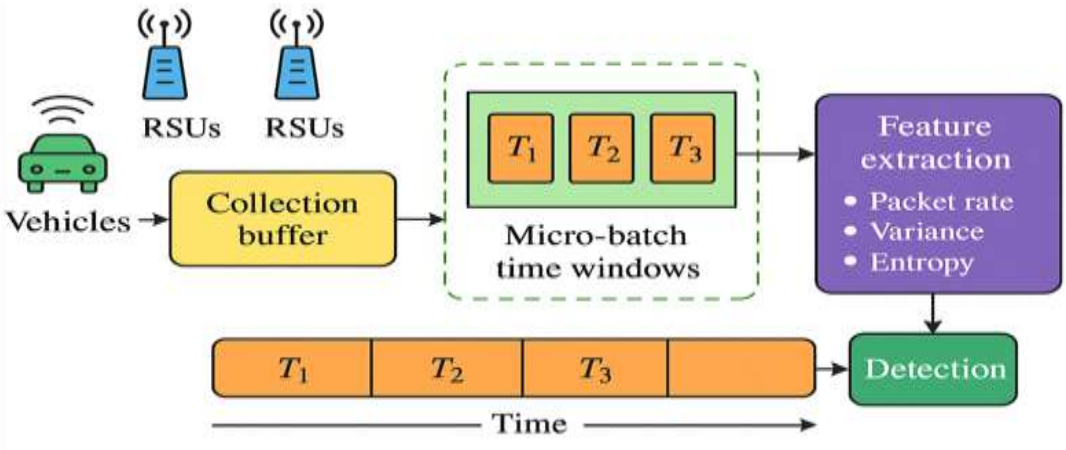

Figure 6. Feature extraction process in the collection module using micro-batch time windows.

Figure 6 show the process of feature extraction. This process captures raw traffic, buffers it through MQ and parses it into structured datasets before forwarding to the detection pipeline.

B. Detection Module

Once structured features are generated the detection module processes and classifies them into two stages

Data Preprocessing:

1. The attributes are converted to floating-point values.
2. The data is stored as Apache Spark Data Frames.
3. The Vector Assembler (Demırcı & Acarturk, 2022) merges multiple features into single vectors.
4. Features are normalized to the [0,1] range using MinMaxScaler (Asha et al., 2025)

$$\text{Rescaled(ai)} = \frac{\text{ai} - \text{Amin}}{\text{Amax} - \text{Amin}} \text{X(max} - \text{min)} + \text{min} \quad (1)$$

where max = 1, min 0.

Classification with Random Forest (RF)

1. RF operates as a binary classifier such as 0= normal traffic and 1= DDoS traffic.
2. Detected attack traffic is stored in HDFS for traceability and retraining.
3. Administrators receive real-time alerts when threats are detected.

C. Apache Spark-ML Random Forest Algorithm

The selection of Random Forest (RF) is because it is robust to noise, resists overfitting and supports parallelization. In Apache Spark, each decision tree is trained on a subset of features and samples.
For an input vector V=(V1,V2, ...Vp)^T and label Z the classifier minimizes the expected loss
L (Z, f(V)) is the classification loss.

$$\text{Ev, z [L(Z, f(V))]} \quad (2)$$

Training Workflow

1. Extract label and feature columns into Apache Spark RDDs.

2. Compute candidate splits, bins, and feature samples.
3. Apply bootstrap sampling to build training subsets.
4. Train multiple trees in parallel using Apache Spark workers.
5. For each split, compute Gini impurity

$$I(X) = 1 - \sum_{i=1}^{m} pi^2 \quad (3)$$

where p is the probability of class i for a feature of A

$$I(X, A) = \sum_{i=1}^{m} \frac{|Xi|}{|X|} I(Xi) \quad (4)$$

6. Select the feature split minimizing impurity.
7. Repeat until maximum depth or minimum node size is reached.

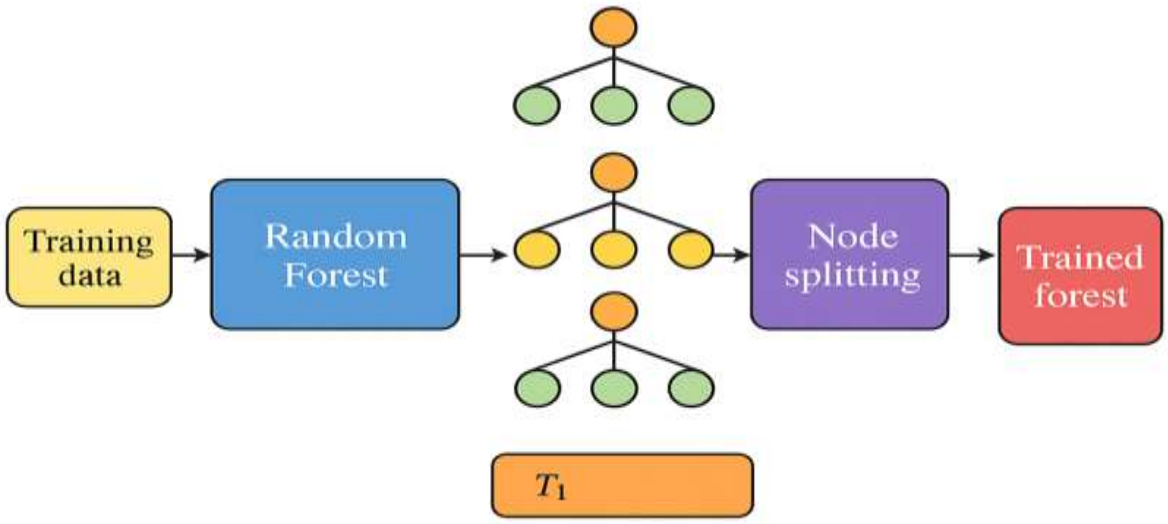

Figure 7. Workflow of Random Forest training in Apache Spark, showing parallel tree construction and node splitting.

Figure 7 represents the work flow of RF in Apache Spark. This workflow illustrates Apache Spark's parallelism, which accelerates decision tree training while ensuring accuracy through Gini-based optimization.

D. Advantages of the Proposed Design

Scalability: The packet capturing and classification are shared across multiple nodes by allowing the system to handle large amounts of data efficiently.

Accuracy: The Random Forest algorithm increases detection precision and helps reduce false alarms.

Resilience: The collected attack data is stored and used for retraining and allowing the system to adapt to new and evolving threats.

Real-Time Capability: The Apache Spark's in memory processing enables fast detection with minimal delay.

With the established decentralized NIDS framework, the next section evaluates its performance through experiments on standard benchmark datasets.

A. Datasets

This section evaluates how well the proposed decentralized NIDS performs using two standard datasets that are NSL-KDD and UNSW-NB15. It first describes the datasets and then explains the

experimental setup by evaluation metrics and results. The section also compares the proposed system with other recent methods and includes an analysis of how different parameters affect performance.

1) NSL-KDD Dataset

The NSL-KDD dataset (Masoodi et al., 2021) is an improved version of the KDDCUP 99 dataset and is widely used for testing intrusion detection systems. Unlike the original dataset NSL-KDD removes repeated and duplicate records to reduce learning bias. It includes 41 features that describe both packet-level and statistical network behavior. The attacks are divided into four main types such as Denial of Service (DoS), User to Root (U2R), Remote to Local (R2L) and Probing. Table 1 shows the distribution of the NSL-KDD dataset. In this study only Normal and DoS traffic are used to create a binary classification problem, focusing on detecting large-scale denial-of-service attacks relevant to decentralized NIDS in VANETs.

Table 1. Attack categories in NSL-KDD dataset

Category	Attack Types	Example	Used in this study
DoS	smurf, teardrop, neptune, back, land	Network flooding	Yes
U2R	buffer_overflow, loadmodule, rootkit	Privilege escalation	No
R2L	ftp_write, imap, guess_passwd	Unauthorized access	No
Probing	portsweep, ipsweep, satan, nmap	Reconnaissance	No
Normal		Legitimate traffic	Yes

2) UNSW-NB15 Dataset

The UNSW-NB15 dataset (Aleesa et al., 2021) was created to capture modern attack scenarios with realistic traffic patterns. It contains 49 features extracted with IXIA PerfectStorm in a controlled cyber range. Compared with NSL-KDD, UNSW-NB15 is more imbalanced and includes nine attack categories.

Table 2. The Distribution of Traffic Records in UNSW-NB15 Dataset

Traffic Class	Training Samples	Testing Samples
Normal	56000	37000
Generic	40000	18871
Exploits	33393	11132
Fuzzers	18184	6062
DoS	12264	4089
Reconnaissance	10491	3496
Analysis	2000	677
Backdoor	1746	583
Shellcode	1133	378
Worms	130	44
Total	**175341**	**82332**

Table 2 presents the UNSW-NB15 dataset captures a wider variety of modern attack categories compared to older benchmarks. For the purposes of this study, only the Normal and DoS classes were used to align with the focus on DDoS detection in decentralized NIDS. This binary selection allows the evaluation to remain consistent with the system's objective of preventing large-scale denial-of-service attacks in vehicular networks.

B. Experimental Settings

1) Platform

The experiments were conducted on a server with Intel Xeon E5-2603 v3 @ 1.60GHz, 10 GB RAM per VM and a five-node Hadoop cluster. Each node ran HDFS, YARN and Apache Spark. Detection algorithms were implemented in Scala 2.12.6 and executed in yarn-cluster mode.

2) Data Preprocessing

1. **Filtering:** Only Normal and DoS samples retained
2. **Encoding:** Conversion of categorical attributes to numeric values
3. **Normalization:** Features rescaled to [0,1] using MinMaxScaler (Asha et al., 2025)

3) Evaluation Metrics

The four standard classification metrics were used to evaluate the proposed NIDS that based on the following definitions

- **True Positives (TP):** Number of correctly classified DDoS samples.
- **True Negatives (TN):** Number of correctly classified normal samples.
- **False Positives (FP):** Number of normal samples incorrectly classified as DDoS.
- **False Negatives (FN):** Number of DDoS samples incorrectly classified as normal.

From these elements standard metrics including Accuracy, Precision, Recall and F1 – score is calculated to assess performance comprehensively.

Accuracy: The Accuracy measures the percentage of samples that are correctly identified. It is defined in Equation (5)

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \quad (5)$$

Recall: The Recall measures the proportion of DDoS traffic correctly identified. It is defined in Equation (6)

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (6)$$

False Alarm Rate (FAR): The False Alarm Rate (FAR) measures the proportion of normal traffic incorrectly classified as DDoS traffic. It is defined in Equation (7)

$$\text{FAR} = \frac{\text{FP}}{\text{FP} + \text{TN}} \quad (7)$$

F1-Measure: The F1-Measure is the weighted harmonic mean of Precision and Recall. It is defined in Equation (8)

$$\text{F1} = \frac{2 \text{ X Precision X Recall}}{\text{Precision X Recall}} \quad (8)$$

Precision: The Precision measures the fraction of correctly classified DDoS traffic out of all traffic labeled as DDoS. It is defined in Equation (9)

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (9)$$

3. Results

1) Comparison with Other Classifiers

The Random Forest (RF) was compared with Naïve Bayes (NB), Logistic Regression (LR), Support Vector Machine
(SVM), Gradient Boosting Decision Tree (GBDT), and XGBoost.

Table 3. Classifier Performance on NSL-KDD Dataset (KDDTestC and KDDTest-21)

Classifier	Accuracy (%)	Recall (%)	FAR (%)	F1-Measure (%)	Accuracy (%)	Recall (%)	FAR (%)	F1-Measure (%)
	KDDTestC				KDDTest-21			
RF	**99.95**	**99.95**	**0.05**	**99.94**	**99.89**	**99.93**	**0.19**	**99.92**
NB	94.64	99.28	8.28	93.47	86.34	98.65	28.51	88.76
LR	**99.96**	**99.95**	**0.03**	**99.95**	99.86	99.91	0.23	**99.99**
SVM (RBF)	86.64	92.81	16.70	82.99	64.88	85.50	51.80	68.52
SVM (Linear)	80.44	92.63	12.56	87.12	72.28	86.40	44.14	77.02
SVM (Sigmoid)	86.66	91.22	15.97	83.31	64.92	82.89	51.88	69.55
GBDT	93.11	98.83	10.31	91.48	81.78	97.73	34.79	84.54
XGBoost	91.65	96.51	11.30	89.71	77.92	93.27	38.55	81.38

Table 3 shows that Random Forest gave the best overall results with the highest accuracy and lowest false alarms on both datasets. The Logistic Regression was close but other models like Naive Bayes, GBDT, XGBoost and SVM performed worse so it confirming the Random Forest as the best option for decentralized intrusion detection in VANETs.

Table 4. Confusion Matrices of Classifiers on NSL-KDD Dataset

Classifier	Actual Class	Predicted: DDoS	Predicted: Normal
RF (KDDTestC)	DDoS	**7454**	**5**
	Normal	**4**	**9706**
RF (KDDTest-21)	DDoS	**6586**	872
	Normal	48	9663
LR (KDDTestC)	DDoS	**7454**	**3**
	Normal	**4**	**9708**
LR (KDDTest-21)	DDoS	5598	1860
	Normal	434	9277
XGBoost (KDDTest-21)	DDoS	6133	1325
	Normal	488	9223
GBDT (KDDTest-21)	DDoS	5717	1741
	Normal	550	9161
NB (KDDTest-21)	DDoS	6350	1108
	Normal	**75**	9636
SVM (RBF, KDDTest-21)	DDoS	6250	1208
	Normal	226	9485

Table 4 shows the confusion matrix results for all classifiers on KDDTestC and KDDTest-21. The Random Forest (RF) achieved the strongest balance, with near-perfect results on KDDTestC and correct detection of 6586 DDoS and 9663 normal samples on KDDTest-21. The Logistic Regression (LR) also performed well on KDDTestC but dropped on KDDTest-21 by misclassifying 1860 DDoS and 434 normal records. The Gradient Boosting (GBDT), XGBoost and Naive Bayes offered competitive results yet produced higher false predictions while SVM variants showed the weakest detection under complex traffic conditions.

2) Parameter Sensitivity in Random Forest

- **Sampling Rate**: The accuracy improved and FAR decreased until stabilizing after 0.55.
- **Number of Trees**: The performance plateaued around 100 trees balancing accuracy and computational cost.
- **Tree Depth**: The accuracy increased with depth, stabilizing beyond 16 levels (Figure 6).

3) Comparison with State-of-the-Art

The proposed method was compared with three advanced DDoS detection systems Entropy-based(Mohammad et al., 2022), Hybrid-ADE (Sahoo et al., 2022) and Marliboost (Premkumar et al., 2022).

Table 5. Confusion matrix results of classifiers on NSL-KDD dataset

Classifier	DDoS → DDoS	DDoS → Normal	Normal → DDoS	Normal → Normal
RF	**7454**	**5**	**4**	**9706**
NB	6586	872	48	9663
LR	**7454**	**3**	**4**	**9708**
SVM (RBF)	5598	1860	434	9277
SVM (Linear)	6133	1325	488	9223
SVM (Sigmoid)	5717	1741	550	9161
GBDT	6350	1108	75	9636
XGBoost	6250	1208	226	9485

Table 5 shows that both Random Forest (RF) and Logistic Regression (LR) achieved the strongest results, correctly classifying nearly all DDoS and Normal instances, with only a handful of misclassifications. RF recorded 7454 true DDoS and 9706 true Normal detections, while LR slightly surpassed it with 7454 true DDoS and 9708 true Normal**.** In contrast the Naive Bayes, GBDT, XGBoost and all SVM variants showed significantly higher misclassifications especially in labeling Normal traffic as DDoS that undermines reliability. These findings reinforce RF and LR as the most effective classifiers for decentralized NIDS in VANETs with RF offering a better balance between recall and false alarm control.

Table 6. Confusion matrix results of classifiers on UNSW-NB15 dataset

Classifier	True DDoS → Predicted DDoS	True DDoS → Predicted Normal	True Normal → Predicted DDoS	True Normal → Predicted Normal
RF	**4339**	**3**	**3**	**2148**
NB	3503	839	48	2104
LR	**4338**	**5**	**4**	**2147**
SVM (RBF)	2482	1860	421	1731
SVM (Linear)	3017	1325	475	1677
SVM (Sigmoid)	2601	1741	537	1615
GBDT	3234	1108	75	2077
XGBoost	3134	1208	226	1926

Table 6 shows that Random Forest and Logistic Regression reached almost perfect detection results on the UNSW-NB15 dataset with very few errors in identifying DDoS and Normal traffic. The other classifiers especially SVM models made more false predictions. These results confirm that Random Forest is the most dependable choice for a decentralized NIDS in VANETs.

Table 7: Confusion Matrix Results on the UNSW-NB15 Dataset

Classifier	Actual DDoS → Predicted DDoS	Actual DDoS → Predicted Normal	Actual Normal → Predicted DDoS	Actual Normal → Predicted Normal
RF	**8270**	**3994**	496	55504
NB	11658	606	246	55754
LR	0	12264	0	**56000**
SVM (RBF)	8258	4006	**38**	55962
SVM (Linear)	9422	2842	187	55813
SVM (Sigmoid)	7706	4558	17	55983
GBDT	11425	839	138	55862
XGBoost	11471	**793**	168	55832

Table 7 presents the confusion matrix results for the UNSW-NB15 dataset. The Logistic Regression correctly classified all normal traffic but unable to detect DDoS attacks. The Random Forest and SVM (RBF) provided balanced performance across both classes. The XGBoost and GBDT achieved strong DDoS detection but produced more false positives than Random Forest.

4) Discussion on the Efficiency of Random Forest Parameters

This section studies how different parameters affect the performance of the Random Forest model focusing on sampling rate, tree depth and the number of trees. The UNSW-NB15 dataset is used for this evaluation. To test the sampling rate the tree depth is fixed at 20 and the number of trees is set to 90, 100 and 110. The accuracy and false alarm rate are measured as the sampling rate changes from 0.2 to 0.95. The results shows that the performance becomes stable when the tree depth is 20 and the number of trees is between 90 and 110. As the sampling rate increases the accuracy improves and false alarms decrease. When the sampling rate reaches 0.55 both values level off with the accuracy at 98.67% and a false alarm rate of 1.17%. The low sampling rates cause the model to underfit during training. Moreover to examine the impact of tree depth the sampling rate is fixed at 0.9. The results show that increasing tree depth improves accuracy until it reaches a depth of 16 where both accuracy and false alarm rate stay consistent.

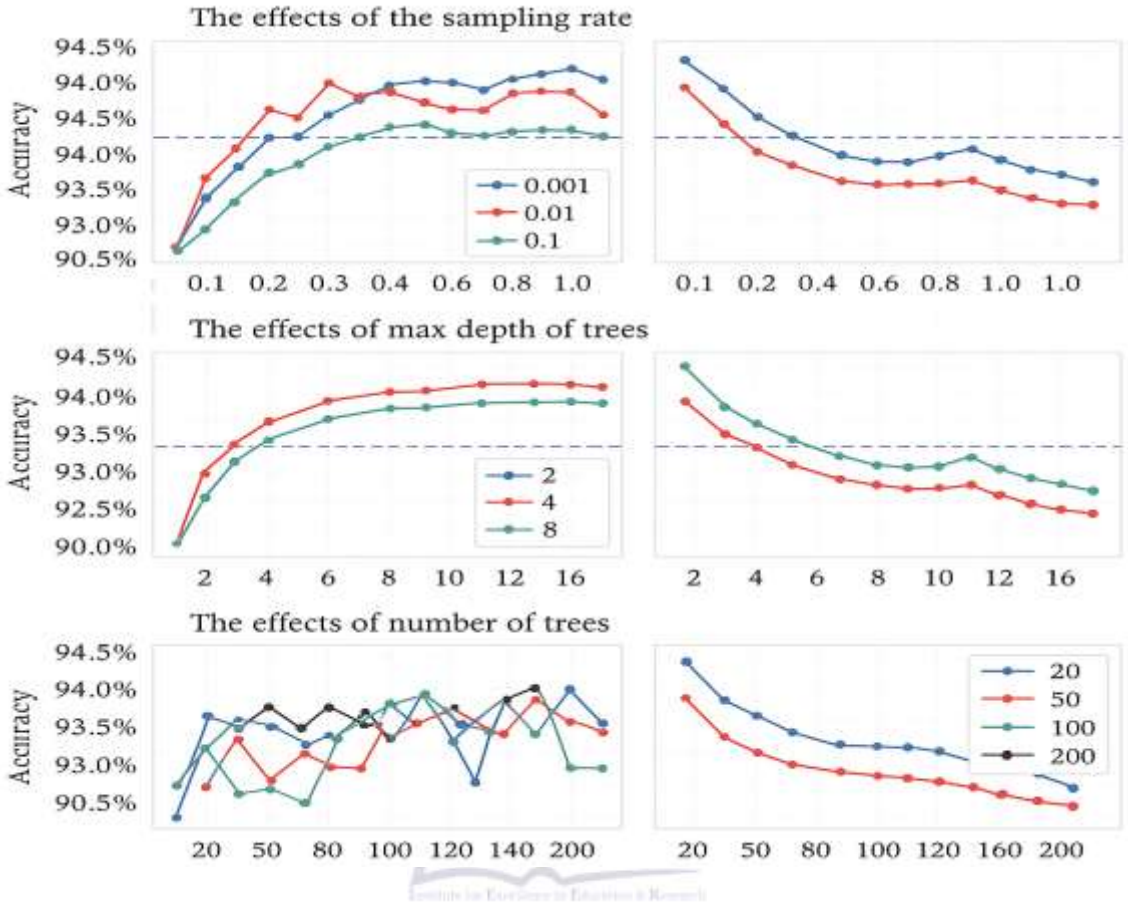

Figure 8. The effects of the sampling rate, max depth of trees and number of trees.

Finally, to assess the effect of the number of trees, the maximum depth is set to 20 while varying the sampling rate from 0.6 to 0.9. As shown in Figure 6 accuracy improves as the number of trees increases but eventually levels off. These findings indicate that more trees enhance model fitting and accuracy but also raise training costs.

4. Discussion

Comparison with State-of-the-Art Methods: To evaluate the effectiveness of the proposed Decentralized Network Intrusion Detection System (DNIDS) for DDoS attack prevention in Vehicular Ad Hoc Networks, it was compared with three recent detection methods using the NSL-KDD dataset, a commonly used benchmark for intrusion detection research. The selected approaches include Hybrid-ADE (Sahoo et al., 2022)and Marliboost (Premkumar et al., 2022). The results are presented in Table 8.

Table 8: Comparison of the Proposed Apache Spark-ML Random Forest–Based Algorithm with Existing DDoS Detection Methods

Approach	Accuracy (%)	FAR (%)
Proposed NIDS	**99.95**	**0.05**
Mohamed (Mohamed et al., 2023)	98.23	0.33
Hybrid-ADE (Idhammad et al., 2018)	97.40	0.45
Marliboost (Nirmala et al., 2023)	96.38	**0.01**

Table 8 shows that the proposed NIDS reached the highest accuracy of 99.95% with a very low false alarm rate of 0.05% and performing better than both Mohamed's method and Hybrid-ADE. Although Marliboost achieved the lowest false alarm rate of 0.01% its overall accuracy was lower that confirming the strength of the proposed system. The Random Forestp-based decentralized NIDS achieved 99.95% accuracy and a 0.05% false alarm rate that surpassing all other models. Mohamed Idhammad's approach improved detection by filtering unnecessary traffic using entropy estimation while the Hybrid-ADE model increased accuracy through density estimation in the autoencoders hidden layer but produced more false alarms. The Marliboost used ensemble learning to reduce false alarms to 0.01% but its accuracy remains below that of the proposed framework. Overall the decentralized NIDS offers the best balance between accuracy and reliability showing its effectiveness in preventing DDoS attacks in Vehicular Ad Hoc Networks.

V. Conclusion and Future Work

The VANETs are a main part of intelligent transportation systems. They enable vehicles so that communicate with each other V2V and with roadside units V2I to improve road safety, manage traffic and support better driving decisions. Because these networks use open wireless channels and they are exposed to Distributed DDoS attacks that can block communication and threaten human safety. To address this problem intrusion detection systems needs to be accurate, scalable and resilient under heavy network traffic. This study proposes a DNIDS for detecting DDoS attacks in VANETs. The system uses distributed traffic collection with micro-batch processing to lower delay and improve fault tolerance. The Apache Spark is used for feature extraction and model training that allowing the system to handle large-scale networks efficiently. The detection is carried out using a RF classifier that are chosen for its strong performance with noisy and high-dimensional data.

REFERENCES

Abdel-Aty, M., Zheng, O., Wu, Y., Abdelraouf, A., Rim, H., & Li, P. (2023). Real-time big data analytics and proactive traffic safety management visualization system. *Journal of transportation engineering, Part A: Systems*, *149*(8), 04023064.

Ahmad, M., Riaz, Q., Zeeshan, M., Tahir, H., Haider, S. A., & Khan, M. S. (2021). Intrusion detection in internet of things using supervised machine learning based on application and transport layer features using UNSW-NB15 data-set. *EURASIP Journal on Wireless Communications and Networking*, *2021*(1), 10.

Ahmad, N., Rajwana, M. A., Kashif, M. I., Malik, H., Bukhri, W. A., Tariq, S., Ejaz, S., ul Haq, E., & Rasheed, M. S. (2025). GAT–CNN: Graph Attention Network and Convolutional Neural Network (GAT–CNN) Model for Intrusion Detection in Internet of Things Using BoT-IoT Dataset. *STATISTICS, COMPUTING AND INTERDISCIPLINARY RESEARCH*, *7*(2), 285-299.

Akano, O., Olayinka, T., Adeniji, O., & Ogunjinmi, B. (2024). Mitigating Insider Threat's IP Spoofing through Enhanced Dynamic Cluster Algorithm (EDPU Based HCF). *Advances in Research*, *25*(3), 85-90.

Aleesa, A., Younis, M., Mohammed, A. A., & Sahar, N. (2021). Deep-intrusion detection system with enhanced UNSW-NB15 dataset based on deep learning techniques. *Journal of Engineering Science and Technology*, *16*(1), 711-727.

Ali, M. H., Jaber, M. M., Abd, S. K., Rehman, A., Awan, M. J., Damaševičius, R., & Bahaj, S. A. (2022). Threat analysis and distributed denial of service (DDoS) attack recognition in the internet of things (IoT). *Electronics*, *11*(3), 494.

Aminu, M., Akinsanya, A., Dako, D. A., & Oyedokun, O. (2024). Enhancing cyber threat detection through real-time threat intelligence and adaptive defense mechanisms. *International Journal of Computer Applications Technology and Research*, *13*(8), 11-27.

Asha, V., Vasumathi, M., Prasad, A., AP, Y., & Sivani, M. (2025). Evaluation of ML Models using SMOTE and Feature Scaling for Intrusion Detection System (IDS). 2025 International Conference on Visual Analytics and Data Visualization (ICVADV),

Benkirane, S., Guezzaz, A., Azrour, M., Gardezi, A. A., Ahmad, S., Sayed, A. E., & Shafiq, M. (2023). Adapted speed system in a road bend situation in VANET environment. *CMC-COMPUTERS MATERIALS & CONTINUA*, *74*(2), 3781-3794.

Brahmane, A. V., & Krishna, B. C. (2021). Big data classification using deep learning and apache spark architecture. *Neural Computing and Applications*, *33*(22), 15253-15266.

Chai, J., Zhao, Y., Chen, X., Du, J., Li, J., Yang, T., Wang, L., & Zhang, Z. (2021). Cost-effective OSNR monitoring with large chromatic dispersion tolerance using random forest for intermediate nodes. *Optics Communications*, *479*, 126469.

Dalmazo, B. L., Marques, J. A., Costa, L. R., Bonfim, M. S., Carvalho, R. N., da Silva, A. S., Fernandes, S., Bordim, J. L., Alchieri, E., & Schaeffer-Filho, A. (2021). A systematic review on distributed denial of service attack defense mechanisms in programmable networks. *International Journal of Network Management*, *31*(6), e2163.

Demırcı, D., & Acarturk, C. (2022). Static malware detection using stacked BiLSTM and GPT-2. *IEEE Access*, *10*, 58488-58502.

Dritsas, E., & Trigka, M. (2025). Applying Machine Learning on Big Data with Apache Spark. *IEEE Access*.

Dustdar, S., Pujol, V. C., & Donta, P. K. (2022). On distributed computing continuum systems. *IEEE Transactions on Knowledge and Data Engineering*, *35*(4), 4092-4105.

Fang, Z., Hu, S., Wang, J., Deng, Y., Chen, X., & Fang, Y. (2025). Prioritized information bottleneck theoretic framework with distributed online learning for edge video analytics. *IEEE Transactions on Networking*.

Farsimadan, E., Moradi, L., & Palmieri, F. (2025). A review on security challenges in V2X communications technology for VANETs. *IEEE Access*.

Feraudo, A., Romandini, N., Mazzocca, C., Montanari, R., & Bellavista, P. (2024). DIVA: A DID-based reputation system for secure transmission in VANETs using IOTA. *Computer Networks*, *244*, 110332.

Fu, Y., Li, C., Yu, F. R., Luan, T. H., & Zhang, Y. (2021). A survey of driving safety with sensing, vehicular communications, and artificial intelligence-based collision avoidance. *IEEE transactions on intelligent transportation systems*, *23*(7), 6142-6163.

Govindaraj, L., Sundan, B., & Thangasamy, A. (2021). An intrusion detection and prevention system for ddos attacks using a 2-player bayesian game theoretic approach. 2021 4th international conference on computing and communications technologies (ICCCT),

Goyal, R., Elawadhi, O., Sharma, A., Bhutani, M., & Jain, A. (2024). Cloud-connected central unit for traffic control: interfacing sensing units and centralized control for efficient traffic management. *International Journal of Information Technology*, *16*(2), 841-851.

Guerna, A., Bitam, S., & Calafate, C. T. (2022). Roadside unit deployment in internet of vehicles systems: A survey. *Sensors*, *22*(9), 3190.

Halawani, A. T. (2025). Enhancing Traffic Safety by Strategically Placing Roadside Units for Connected Vehicles: Intercity Roads. 2025 2nd International Conference on Advanced Innovations in Smart Cities (ICAISC),

Hamedani, A. F., Aziz, M., Wieder, P., & Yahyapour, R. (2021). BFDD-S: Big Data Framework to Detect and Mitigate DDoS Attack in SDN Network. *open science index 15 2021*, *18*, 30.

Heidari, A., & Jabraeil Jamali, M. A. (2023). Internet of Things intrusion detection systems: a comprehensive review and future directions. *Cluster Computing*, *26*(6), 3753-3780.

Hosseinzadeh, M., Rahmani, A. M., Vo, B., Bidaki, M., Masdari, M., & Zangakani, M. (2021). Improving security using SVM-based anomaly detection: issues and challenges. *Soft Computing-A Fusion of Foundations, Methodologies & Applications*, *25*(4).

Hussain, M., O'Nils, M., Lundgren, J., & Mousavirad, S. J. (2024). A comprehensive review on deep learning-based data fusion. *IEEE Access*.

Idhammad, M., Afdel, K., & Belouch, M. (2018). Semi-supervised machine learning approach for DDoS detection. *Applied Intelligence*, *48*(10), 3193-3208.

Jiao, Y., Calvert, S. C., & Van Lint, H. (2024). Minimising Missed and False Alarms: A Vehicle Spacing based Approach to Conflict Detection. 2024 IEEE Intelligent Vehicles Symposium (IV),

Karthikeyan, H., & Usha, G. (2022). Real-time DDoS flooding attack detection in intelligent transportation systems. *Computers and Electrical Engineering*, *101*, 107995.

Kothandapani, H. P. (2023). Emerging trends and technological advancements in data lakes for the financial sector: An in-depth analysis of data processing, analytics, and infrastructure innovations. *Quarterly Journal of Emerging Technologies and Innovations*, *8*(2), 62-75.

Lone, F., Verma, H. K., & Sharma, K. P. (2024). A systematic study on the challenges, characteristics and security issues in vehicular networks. *International Journal of Pervasive Computing and Communications*, *20*(1), 56-98.

Maharjan, R., Chy, M. S. H., Arju, M. A., & Cerny, T. (2023). Benchmarking message queues. Telecom,

Mahi, M. J. N., Chaki, S., Humayun, E., Imran, H., Barros, A., & Whaiduzzaman, M. (2023). A review on VANET security: Future challenges and open issues. *Indonesian Journal of Electrical Engineering and Informatics (IJEEI)*, *11*(1), 180-193.

Malik, S. A. (2024). *Analysis and Simulation of Cyber Attacks Against Connected and Autonomous Vehicles and Their Platoon System* The University of Toledo].

Masood, J. A. I. S., Chakravarthy, N. K., Asirvatham, D., Marjani, M., Shafiq, D. A., & Nidamanuri, S. (2024). A hybrid deep learning model to predict high-risk students in virtual learning environments. *IEEE Access*, *12*, 103687-103703.

Masoodi, F., Bamhdi, A. M., & Teli, T. A. (2021). Machine learning for classification analysis of intrusion detection on NSL-KDD dataset. *Turkish Journal of Computer and Mathematics Education*, *12*(10), 2286-2293.

Mavaluru, D., Mubarakali, A., Narapureddy, B. R., Ramakrishnan, J., John, R., Ravishankar, N., & Karthika, P. (2023). Deep convolutional neural network based real-time abnormal behavior detection in social networks. *Computers and Electrical Engineering*, *111*, 108987.

Menéndez, A. G. (2024). A Hybrid Framework for Statistical Feature Selection and Image-Based Noise-Defect Detection. *arXiv preprint arXiv:2412.08800*.

Mohamed, H. G., Alrowais, F., Al-Hagery, M. A., Al Duhayyim, M., Hilal, A. M., & Motwakel, A. (2023). Optimal Wavelet Neural Network-Based Intrusion Detection in Internet of Things Environment. *Computers, Materials & Continua*, *75*(2).

Mohammad, A. H., Alwada'n, T., Almomani, O., Smadi, S., & ElOmari, N. (2022). Bio-inspired hybrid feature selection model for intrusion detection. *Computers, Materials and Continua*, *73*(1), 133-150.

Moore, S. J., Cruciani, F., Nugent, C. D., Zhang, S., Cleland, I., & Sani, S. (2023). Deep learning for network intrusion: A hierarchical approach to reduce false alarms. *Intelligent Systems with Applications*, *18*, 200215.

Nirmala, E., Suresh, M., & Maragatharajan, M. (2023). Deployment of Robust Security Scheme in SDN Based 5G Network to Detect DDoS Attack. 2023 4th International Conference on Smart Electronics and Communication (ICOSEC),

Nishiuchi, T., Fujita, S., Watanabe, Y., Iwamoto, M., & Sawada, K. (2023). Packet Analysis and Information Theory on Attack Detection for Modbus TCP. IECON 2023-49th Annual Conference of the IEEE Industrial Electronics Society,

Olayinka, O. H. (2021). Big data integration and real-time analytics for enhancing operational efficiency and market responsiveness. *Int J Sci Res Arch*, *4*(1), 280-296.

Pallamala, R. K., & Rodrigues, P. (2022). An investigative testing of structured and unstructured data formats in big data application using apache spark. *Wireless Personal Communications*, *122*(1), 603-620.

Premkumar, M., Sundararajan, T. V. P., & Mohanbabu, G. (2022). Dynamic defense mechanism for DoS attacks in wireless environments using hybrid intrusion detection system and statistical approaches. *Tehnički vjesnik*, *29*(3), 965-970.

Rasheed, M. S., Rajwana, M. A., Kashif, M. I., Malik, H., Bukhari, W. A., Tariq, S., Ahmad, N., & ul Haq, E. (2025). BiLSTM-CRF: Improving Cybersecurity Text Analysis through Bidirectional LSTM-CRF (BiLSTM-CRF) Based Named Entity Recognition. *STATISTICS, COMPUTING AND INTERDISCIPLINARY RESEARCH*, *7*(2), 359-376.

Sahoo, D. K., Sahu, R. K., & Panda, S. (2022). Chaotic Harris hawks optimization based type-2 fractional order fuzzy PID controller for frequency regulation of power systems. *International Journal of Ambient Energy*, *43*(1), 3832-3844.

Salman, H. A., Kalakech, A., & Steiti, A. (2024). Random forest algorithm overview. *Babylonian Journal of Machine Learning*, *2024*, 69-79.

Santhosh Kumar, S., Selvi, M., & Kannan, A. (2023). A comprehensive survey on machine learning-based intrusion detection systems for secure communication in internet of things. *Computational Intelligence and Neuroscience*, *2023*(1), 8981988.

Sarker, I. H. (2021). Machine learning: Algorithms, real-world applications and research directions. *SN Computer Science*, *2*(3), 160.

Seydali, M., Khunjush, F., & Dogani, J. (2024). Streaming traffic classification: a hybrid deep learning and big data approach. *Cluster Computing*, *27*(4), 5165-5193.

Shaikh, J., Yandrapalli, V., Gole, N. T., Patil, A., Bodne, N. P., Matte, S., & Umar, S. (2024). Tabular Deep Learning-Based Light-Weight Intrusion Detection System for VANET System. International Conference on ICT for Sustainable Development,

Shanthi, J., Rani, D. G. N., & Rajaram, S. (2022). A C4. 5 decision tree classifier based floorplanning algorithm for System-on-Chip design. *Microelectronics journal*, *121*, 105361.

Sohi, S. M., Seifert, J.-P., & Ganji, F. (2021). RNNIDS: Enhancing network intrusion detection systems through deep learning. *Computers & Security*, *102*, 102151.

Tariq, S., Rajwana, M. A., Kashif, M. I., Malik, H., Bukhari, W. A., Khan, A., Ahmad, N., & ul Haq, E. (2025). CNN-RL: A Deep Learning–Based Adaptive Crowdsourcing Model for Cybersecurity Education. *STATISTICS, COMPUTING AND INTERDISCIPLINARY RESEARCH*, *7*(2), 301-317.

Tehaam, M., Ahmad, S., Shahid, H., Saboor, M. S., Aziz, A., & Munir, K. (2022). A review of ddos attack detection and prevention mechanisms in clouds. 2022 24th International Multitopic Conference (INMIC),

ul Haq, E., Rajwana, M. A., Iqbal, M. A., & Kashif, M. (2025). Emerging Algorithms Reshaping Computer Vision: The Next Frontier in Visual Recognition. *STATISTICS, COMPUTING AND INTERDISCIPLINARY RESEARCH*, *7*(2), 387-410.

Yezerets, E., Mudrik, N., & Charles, A. S. (2025). Decomposed Linear Dynamical Systems (dLDS) models reveal instantaneous, context-dependent dynamic connectivity in C. elegans. *Communications Biology*, *8*(1), 1218.

Yokkampon, U., Chumkamon, S., Mowshowitz, A., Fujisawa, R., & Hayashi, E. (2021). Anomaly detection using support vector machines for time series data. *Journal of Robotics, Networking and Artificial Life*, *8*(1), 41-46.

Zhang, R., Zhao, C., Du, H., Niyato, D., Wang, J., Sawadsitang, S., Shen, X., & Kim, D. I. (2025). Embodied AI-enhanced vehicular networks: An integrated large language models and reinforcement learning method. *arXiv preprint arXiv:2501.01141*.

Zou, C. C., Gong, W., Towsley, D., & Gao, L. (2005). The monitoring and early detection of internet worms. *IEEE/ACM Transactions on Networking*, *13*(5), 961-974.