International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Machine Learning Approach for Intelligent Transport System in IOV-Based Vehicular Network Traffic for Smart Cities

Chandrakant D. Kokane[1*], Gopal Mohadikar[2], Sonu Khapekar[3], Bharti Jadhao[4], Tushar Waykole[5], Vilas V. Deotare[6]

Submitted: 26/05/2023 **Revised**: 11/07/2023 **Accepted**: 29/07/2023

Abstract: The transportation industry will face many significant issues, including traffic congestion, pollution, and ineffective traffic management, as a result of the increasing urbanisation and demographic growth. It appears that one potential solution to these issues may be provided by intelligent transport systems (ITS) that harness the power of vehicular networks on the Internet of Vehicles (IoV). By integrating communication between vehicles and infrastructure (V2V) and vehicles and the cloud (V2C), the proposed ITS architecture seeks to create a comfortable and effective transportation ecosystem. The V2V connection helps with the transmission of data on route conditions, collision avoidance, and speed, position, and other factors. Vehicle-to-infrastructure (V2I) communication enables automobiles to connect with infrastructure components like traffic lights, road signs, and parking systems in order to optimise traffic signal timings and provide drivers with real-time information. Innovative applications like customised recommendations, dynamic navigation, and predictive maintenance are made possible via V2C communication, which makes it possible for cars to connect to the cloud. The recommended method makes use of tree-based machine learning models including Decision Tree (DT), XGBoost (XGB), and Random Forest (RF) to increase traffic detection accuracy and computational efficiency.

Keywords*: Vehicular Network, Internet of Vehicles, Machine Learning, Intelligent transport system, Network traffic*

1. Introduction

To effectively combat traffic congestion, intelligent transportation systems (ITS) offer early guidance and effective traffic planning. One essential element of ITS is the Internet of Vehicles (IoV). It enables the evaluation of the capacity of the road network, the gathering of real-time traffic data, traffic relief, and the direction of traffic participants. IoV-based systems can also calculate individual journey times and estimate traffic in order to prevent congestion. To improve network management and security, network activity classification is essential. When handling the traffic from the Internet of Things (IoT), which is made up of numerous devices connected in various ways, network monitoring and management systems run into issues. Other than that, traffic data collection systems that are improved allow for the organisation of traffic based on specific requirements. To maintain the company's security, the company's traffic needs to be correctly classified based on the equipment. The classification of network traffic is a vital task that may be applied to many different forms of data, with a focus specifically on transportation data, mobile networks, the Internet of Things, and large intelligent cities [1]. The classification process depends on automatic learning (ML), in addition to data mining techniques, data bases, and travel characteristic classifications. This includes employing automatic learning approaches for precise classification through the use of extraction, selection, and application. There are many methods available for calculating the traffic flow in an IOV system. Some techniques for predicting time series include factor analysis, index analysis, multiplicative and additive models, and model analysis. Based on historical traffic data, these techniques can be utilised to produce precise estimates.

1,3,4,5 Assistant Professor, Department of Computer Engineering, Nutan Maharashtra Institute of Engineering & Technology, Talegaon(D), Pune, India(MS), ORCID ID: 0000-0001-7957-3933
2 Sr. Assistant Professor, Department of Mechanical Engineering, Tolani Maritime Institute, Induri, Pune, India(MS), ORCID ID 0009-0004-5593-7607
3ORCID ID: 0009-0007-9677-8931
4ORCID ID: 0009-0002-3356-4710
5ORCID ID:0000-0002-6758-0252
6Principal, Nutan Maharashtra Institute of Engineering & Technology, Talegaon(D), Pune, India(MS), ORCID ID: 0000-0002-5619-0806
** Corresponding Author Email: cdkokane1992@gmail.com*

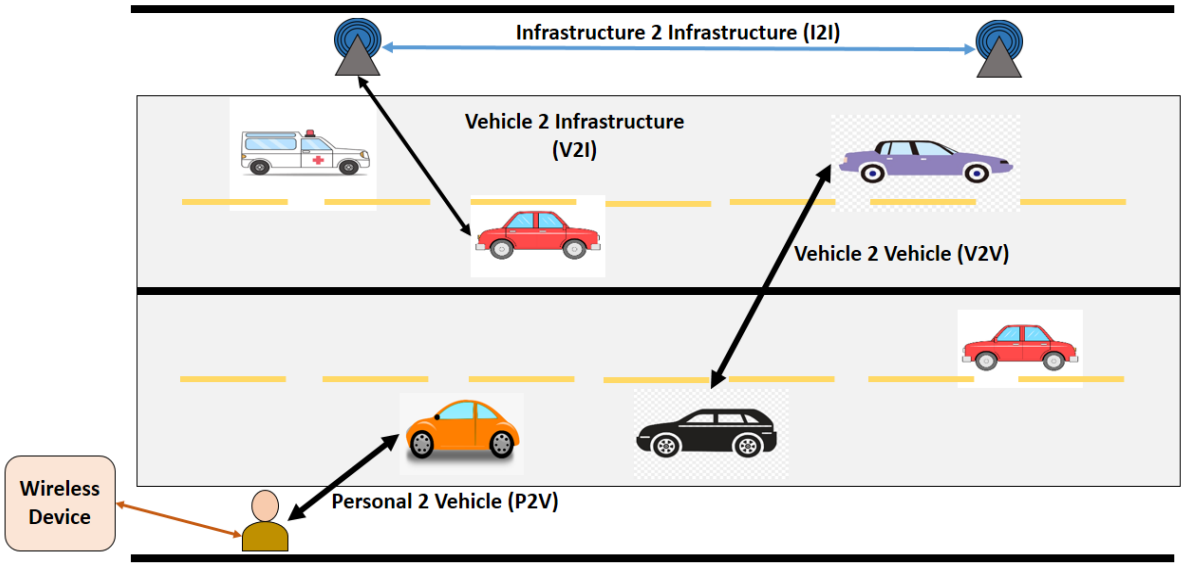

Fig 1: Internet of Vehicle (IoV) Internal Network system

A strong technique that delivers accurate classification results for network traffic order is Deep Packet Inspection (DPI). Understanding how the Internet works is essential since it plays a big part in both our personal and professional life. A rise in the complexity of network traffic patterns can be attributed to the ongoing growth of network designs, protocols, and applications. In addition to helping us satisfy our curiosity, classifying network traffic offers a wide range of crucial uses for network administrators.

It can monitor different behaviours and help with security management and network performance analysis. Utilising particular ML techniques, effective traffic analysis is achieved. To detect intrusions, assess malware activities, classify network traffic data, and examine other security traits, better artificial intelligence (AI) tools are needed. The ability of ML to successfully address network-related issues has been greatly demonstrated. Understanding and enhancing network operations and security require a thorough understanding of network traffic classification. To provide correct classification and enable different applications in network management and protection, it depends on ML approaches, data mining, and suitable dataset selection. The application of machine learning techniques to the categorization of Internet of Vehicles (IOVs) traffic within vehicular networks is the primary subject of this study. Investigating how various ML datasets and their properties affect classification performance using actual IOVs traffic data is the goal. For many network operations, such as security monitoring, defect detection, traffic engineering, and ensuring Quality of Service (QoS) standards, accurate traffic classification is crucial.

Effective network traffic classification of IOVs is essential given their explosive increase. Generating IOVs network traffic data can be difficult, though, especially when the ML models need to be trained on both malignant and benign data. In order to meet this issue, the performance of several machine learning (ML) techniques will be compared in this research, with a focus on intrusion detection systems (IDS) in the context of IOV-based vehicular networks. The goal of the study is to identify the best methodology for accurate traffic classification within IOV networks by comparing several ML approaches. The success of network operations and security depends on the categorization process' performance indicators. The study sheds light on the benefits of various ML techniques by contrasting their results.

2. Review of Literature

In order to improve performance, network security procedures, and the administration of intelligent traffic systems (ITS), network traffic categorization primarily aims to analyse and assess traffic patterns. The author explores a variety of artificial intelligence (AI) and machine learning (ML) techniques for malware analysis and traffic analysis. Effective categorization and classification of network traffic are required due to the rising demand for high-speed transmission rates. In order to do this, the author suggests a revolutionary method that relaxes the assumption of independence in Naive Bayes (NB) classifiers. This method combines technologies like intelligent transportation, databases, networks, and computing to create a system for dynamic processing centre design.

For vehicle ad hoc networks, the research proposes feature selection algorithms that remove pointless

anomaly features and cluster intelligent features [5]. A feature selection technique is used that outperforms previous filter procedures in terms of accuracy and productivity: the computerised correlation-based filter (CFS). IoT devices can quickly benefit from ML and AI techniques [3,4]. The WEKA ML programme is used to estimate the most widely used supervised ML approaches utilising a variety of traffic classification techniques. Naive Bayes, Naive Bayes Kernel Estimation, Bayesian Network, C4.5 Decision Tree, k-Nearest Neighbours (KNN), Deep Neural Networks [7], and Support Vector Machines (SVM) are a few well-liked methods for classifying vehicular network data. Due to its capability to identify distinct network structures and learn from training datasets, ML algorithms are highly valuable.

Lv.Z. et al. [2], a remote receiving terminal and a basic monitoring network make up the system. The fundamental monitoring network links taxis and streetlights as nodes and routes, respectively. Each node is given a different address as a result of the network's dynamic organisation, which acts as the node's identification inside the network. The system's design includes a simulation experiment to show how well it can fulfil criteria and send messages with the acquired data to the specified terminal while adhering to predefined settings.

The system's ability to integrate sensors through a ZigBee wireless network has the potential to encourage the construction of smart city infrastructure. This design makes it possible for data to be collected and transmitted in an efficient manner, which makes it easier to build and administer different parts of a smart city.

The Intrusion Detection Systems (IDSs) training and testing feature extraction algorithm described in this paper intends to effectively extract unique features from vehicle messages. The programme focuses on extracting two important features: variances in traffic flow and differences in position. The range of distances between cars is used to calculate the traffic flow disparities feature. The programme estimates the fluctuations in traffic flow by examining the distances between vehicles in the network. For the purpose of identifying anomalies and probable invasions, this function offers useful information. Voting filter and semi-cooperative mechanisms are both used in the extraction of the position differences feature. Together, these systems help to identify the variations in vehicle placements [11]. The voting filter process aids in the removal of erroneous or noisy position data, guaranteeing the accuracy of the extracted feature. In order to improve the precision and consistency of position difference measurements, the semi-cooperative mechanism makes use of the collaboration of surrounding vehicles.

The suggested approach enables efficient and effective extraction of pertinent information from vehicle communications by utilising these two crucial aspects. The training and testing of IDSs for spotting potential attacks and guaranteeing the security of the vehicle network rely heavily on these features.

3. Publically Available Datasets

In the area of network security, intrusion detection and prevention systems (IDS and IPS) are crucial because they provide defence against increasingly sophisticated network attacks. The systems for detecting intrusions based on anomalies struggle to produce accurate and consistent performance evaluations due to the lack of reliable test and validation data. Methods based on anomalies require the availability of high-quality data bases that accurately reflect traffic and attack situations in the real network.

Table 1: Dataset Description of CIC-IDS2017 Dataset

Attack Type	Description	IOV Scenario
BENIGN	Genuine network traffic produced by actual users conducting activities like browsing or emailing.	Monitoring IOV users' regular activities, such as email, web browsing, and DNS requests.
DoS	Assaults that try to interrupt online services and frequently come before DoS/DDoS assaults.	Attacking IOV systems' network infrastructure and resulting in service degradation or unavailability.
Port-Scan	Sending client queries to various server ports on a host to look for exploitable holes.	Scanning the network ports of IOV systems to find any vulnerabilities that could allow outside access.

Brute Force Attack	Utilizing arduous trial-and-error to try and crack or compromise car network or system credentials.	Using brute force assaults to take control of IOV systems and get unauthorized access.
Web Attack	Using cross-site scripting (XSS) or SQL injection vulnerabilities in the web pages of automobiles or servers.	Stealing sensitive data or undermining the integrity of IOV systems' online interfaces as a target.
Botnet Attack	Gaining access to victim systems to create a network of controlled bots, which is frequently utilized for additional cyber-attacks.	Introducing malware into IOV systems to build a botnet and enable massively coordinated strikes.
Infiltration Attack	Targeting network infiltration, when systems are compromised by unauthorized access.	The deliberate attempt to circumvent security safeguards on IOV networks or systems.

Table 2: Dataset used in proposed method CIC-IDS2017 Dataset

Class Label	Number of Samples
BENIGN	22,731
DoS	19,035
Port-Scan	7,946
Brute-Force	2,767
Web-Attack	2,180
Botnet	1,966

This technology creates genuine, benign background traffic while profiling the abstract behaviour of human interactions. The abstract behaviour of 25 users from the CICIDS2017 dataset was developed as discussed in table 1 and table 2, concentrating on protocols including HTTP, HTTPS, FTP, SSH, and email. Overall, the CICIDS2017 dataset is a useful tool for studying network traffic. It features a variety of attack and benign traffic that closely resembles real-world events, and it was developed with the generation of realistic background traffic in mind. This dataset makes it possible for academics and professionals to thoroughly analyse and assess intrusion detection and prevention systems.

4. Methodology

The IOVs network traffic dataset is classified by classifying the traffic into various groups, such as attacks or normal. Over the past 20 years, there has been a lot of research done on the need for IOV-based vehicular network traffic classification. To effectively classify these groups, researchers have suggested a variety of ways. supervised and unsupervised learning techniques are the two primary categories of machine learning (ML) approaches used for traffic classification. Tree-based supervised ML algorithms are used in this particular study. A combination of decision tree (DT), random forest (RF), extra tree (ET), and XGBoost techniques are used in the feature-engineering stage. These algorithms provide accurate network traffic classification by training and testing intrusion detection techniques with various parameters.

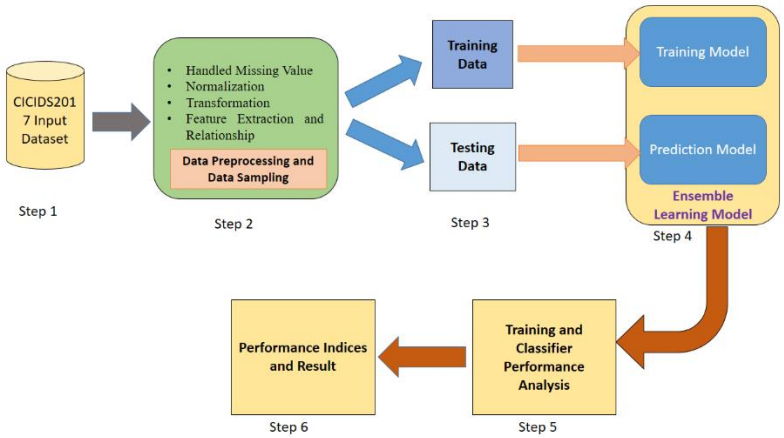

Fig 2: The network architecture proposed for connected vehicles (IoVs)

A few of the many benefits of using machine learning techniques based on trees include interpretation, scalability, and the ability to use large character spaces. The project's goal is to put these algorithms and characteristic engineering to use to achieve accurate and reliable intrusion detection in the IOV-based vehicle network. The focus of the research is on classifying IOVs for network traffic using ML-supervised tree algorithms including DT, RF, ET, and XGBoost. In the feature engineering stage, these algorithms are used to train and test intrusion detection methods with various parameter values. This method is employed by researchers to increase the accuracy and effectiveness of traffic classification in IOV-based vehicular networks.

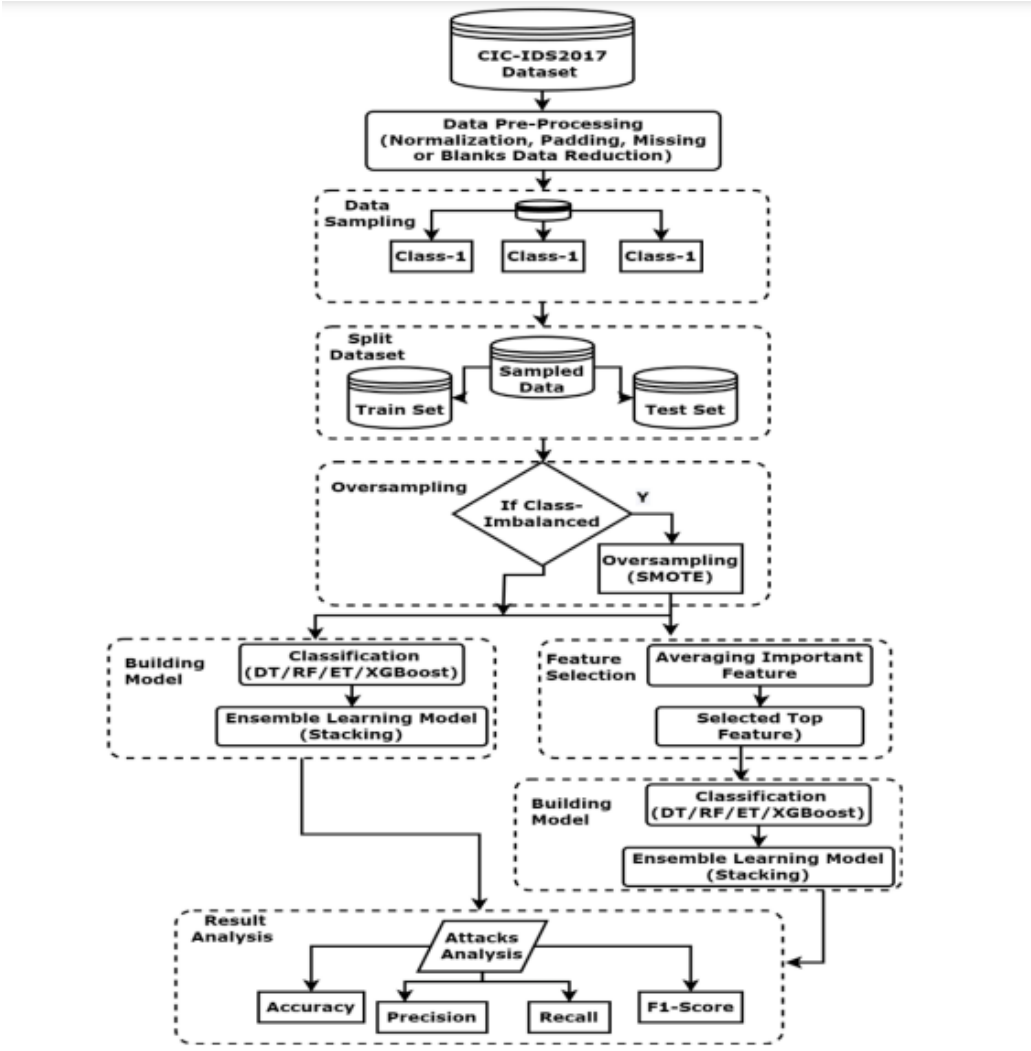

Fig 3: Process flow of proposed system in IoV Vehicular Network

A. (DT) Decision Tree

Data items are categorised by Decision Tree (DT) by assessing the values of their attributes. A decision tree is initially constructed using a set of pre-classified data. To divide the data into various classes, characteristics are picked for each node of the tree. Recursively, this partitioning procedure divides subsets of data items into smaller groups based on attribute values until each group contains only data items from the same class. With edges labelled in accordance with the parent attribute, the decision tree separates the data depending on the supplied attributes at each node. To help in classification, the decision tree's leaves are labelled with decision values. Statistical classifiers are a common classification approach used by DTs. Selected features are incorporated into the classification process, and classes that distinguish the target application based on all aspects are then determined recursively. In this instance, let X stand in for a data point's features and Y for the class. The choice is made by figuring out the ratio between X and Y, which aids in selecting the right class for the data point.

$$\text{RATIO}\,(\,X\,|\,Y)\;=\;H\,(\,X\,)\;-\;H\,(\,X\,|\,Y)\,H\,(X)$$

The conditional entropy, denoted by the symbol H (X|Y), estimates the uncertainty of variable X given variable Y. The marginal entropy, on the other hand, measures the uncertainty of variable X alone, without taking into account any other variables. This is known as H(X).

$$D\;=\;\{(x1,y1),(x2,y2),\ldots,(xN,yN)\}$$

Let xi = (xi1, xi2,..., xim) represent the feature vector of sample i in the regression tree, where xij is the feature j of sample i. The regression tree separates the input space into K regions (R1, R2,..., RK) linked to particular results (c1, c2,..., ck). We can express the regression model in the following way as a result:

$$y\;=\;f(x)\;=\;\Sigma\,ck\;*\;I(x\;\in\;Rk)$$

where y is the anticipated output variable, f(x) denotes the regression function, ck is the particular outcome connected to area R_k, and I(x ∈ Rk) denotes an indicator function that evaluates to 1 when the input x falls within region R_k and 0 otherwise.

$$f(x)\;=\;\sum_{k=1}^{k}C_k\,I(x\;\in\;Rk\,)$$

It is necessary to resolve the following optimization issues in order to determine the values of j and s

$$\min j,s\;\lfloor\min c1\sum xj\epsilon R1\;(j,s)(yi\;-\;c1)2\;+\;\min c1\sum xj\epsilon R2\;(j,s)\;(yi\;-\;c2)2\;\rceil$$

$$C1\;=\;\text{ave}((yi|xi\;\in\;Ri(j,s),),C2\;=\;\text{ave}((yi|xi\;\in\;R2(j,s))$$

The procedure entails going through all of the input variables to get the output values and choosing the best split variable j. The input space is divided into two separate areas (j, s) by each variable acting as a dividing line. The procedure is continued after segmenting each region up until a stop condition is satisfied.

B. (XGB) XGBoost Model

Boosting Trees is a potent technique for decision tree boosting. Particularly popular and powerful is an XGBoost-based Boosting Tree model. An ensemble of decision tree models is what Boosting Trees are. The first step is to create an initial base tree, indicated as y0, with initial predictions for each sample i, denoted as fo(xi) = 0. This acts as the starting point. The explanatory model is updated at each boosting step (step t). By iteratively include fresh trees in the ensemble, it is intended to enhance the predictions. Each new tree aims to capture the patterns and mistakes that the preceding trees did not sufficiently address.

$$\widehat{yi}\;=\;\sum t\,k=1\,fk(xi)\;=\;\widehat{yt}-\;1\,i\;+\;ft(xi)$$

The objective function must be minimized when solving decision trees. The criterion used to assess the effectiveness of the splits and the decision tree's overall performance is represented by the objective function.

$$\text{Objt}\;=\;\sum[i=1\text{ to }n]l(yi,\widehat{yt_i})\;+\;\sum[i=1\text{ to }t]\Omega(fi)\;+\;\text{Constant}$$

A differentiable convex loss function is often used to calculate the value of the variable l, which stands for the difference between the prediction and the target. The difference between the anticipated values and the actual target values is captured by this loss function.

A decision tree's complexity can be described as follows:

$$\Omega\;=\;\sum[i=1\text{ to }T]\Omega(wi)$$

The decision tree's overall leaf count is shown here by the letter T. Every leaf node is identified as wi, where i is a number between 1 and T. The complexity connected to each unique leaf node is described by the word (wi).

The minimal objective function:

$$\text{Obj}\;=\;-\,1/\,2\sum_{j=1}^{T}\text{G2 j Hi}\;+\;\lambda\;+\;\gamma T$$

A decision tree splits the data into two distinct subsets for each of its branches. By contrasting the target function before and after the split, it is possible to quantify the information received by the split.

$$\text{Gain} = \frac{\text{G}^2\text{L}}{\text{H}_l + \lambda} + \frac{\text{G}^2\text{R}}{\text{H}_R + \lambda} - \frac{(\text{GL} + \text{GR})^2}{\text{H}_L + + H\_R + \lambda} - \gamma$$

We may evaluate the change in information or uncertainty by comparing the values of the target function before and after the split. This adjustment offers a gauge for the knowledge acquired or uncertainty lowered as a result of the divide.

C. (RF) Random Forest:

A variable's value can be predicted or classified using the Random Forest (RF) using the outcomes of several Decision Tree (DT) algorithms. A vector of entry (x) containing the values of various characteristics assessed for each instance of formation is used to construct a number of RF regression trees. Results are analyzed and quantified. The RF regression predictor can be stated as follows after creating K trees, indicated as T(x) K_1.

$$\int_{rf}^{K} \text{f}(\text{x}) = 1/\text{K} \sum_{k=1}^{K} \text{T}(\text{x})$$

Here, RF(x) denotes the anticipated value for the input vector x based on the RF regression model. As indicated by the notation T(x), each distinct regression tree offers a forecast for the input vector x. The final forecast of the RF model is created by averaging (adding up and dividing by K) the predictions from each of the K trees. The accuracy and robustness of the predictions are increased by RF by merging the predictions from various trees, which takes advantage of the ensemble's variety and collective knowledge. The act of averaging helps to reduce the biases and inaccuracies of the individual trees, producing predictions that are more trustworthy and accurate. The bagging process generates several training data subsets to minimize the correlation between trees. In bagging, independent random vectors with the same distribution as the input sample are created by resampling random samples from the original dataset to produce various subsets, denoted as h(X, θk), k = 1,..., K. Some data points may be utilized more than once in training as a result of this resampling procedure, while others may not be used at all.

D. Ensemble Method

To build a more potent and reliable model, ensemble approaches combine various distinct models. The approaches DT, RF, and XGBoost all fall under the umbrella of ensemble methods. Here is a quick description of each method:

Algorithm 1: Ensemble Method

Input: *Network Traffic Datasets*

Output: *Classified Data with having No.of features*

Begin

Initialization

Load Network traffic

Data Divide the network traffic datasets

Training Dataset $=$ 80% *of network traffic data*

Test dataset $=$ 20% *of network traffic data*

Training dataset

Train the dataset by using the Bayes theorem

Extract the features from the train data

Measure the probabilities by the Likelihood method

The group predicted by a higher probability

$[ID\ Value] = max(probabilities\ of\ each\ class)$

End

These approaches are examples of ensemble methods, which make use of the variety and combination of multiple models to increase prediction accuracy overall, decrease overfitting, and deliver more trustworthy findings.

5. Results and Discussion

Performance Indices:

The accuracy (ACC) is calculated as the percentage of correctly classified instances, whether they are normal or attacks, and is determined by the following formula:

$$ACC = \frac{(TP + TN)}{(TP + TN + FP + FN)}$$

The formula for calculating precision (P), which is the proportion of pertinent instances among the identified instances:

$$P = \frac{TP}{(TP + FP)}$$

Recall (R) is calculated as the ratio of the number of relevant instances over the total number of relevant instances discovered:

$$\text{R} = \frac{\text{TP}}{(\text{TP} + \text{FN})}$$

The F1-Score is a metric that combines recall and precision into one number. It can be calculated using the formula below as the weighted average of recall and precision:

$$\text{F1Score} = \frac{(2 * \text{P} * \text{R})}{(\text{P} + \text{R})}$$

In particular, when α = 1, the formula for the F1-Score simplifies. Overall, these formulas allow us to calculate

accuracy, precision, recall, and the F1-Score, which are commonly used metrics for evaluating classification performance.

Table 3: Performance Evaluation of Proposed method using CIC-IDS2017 2017 Dataset

Algorithm	Accuracy in (%)	Recall in (%)	Precision in (%)	F1 Score in (%)
DT	98.05	97.04	98.03	98.04
RF	97.81	97.79	98.81	98.78
XGBoost	96.07	96.06	96.16	95.97
Ensemble	98.05	98.05	98.04	98.04

Several performance criteria, including accuracy, recall, precision, and F1 score, were evaluated in the comparison of various machine learning techniques, including DT (Decision Tree), RF (Random Forest), XGBoost, and an ensemble method as shown in table 3.

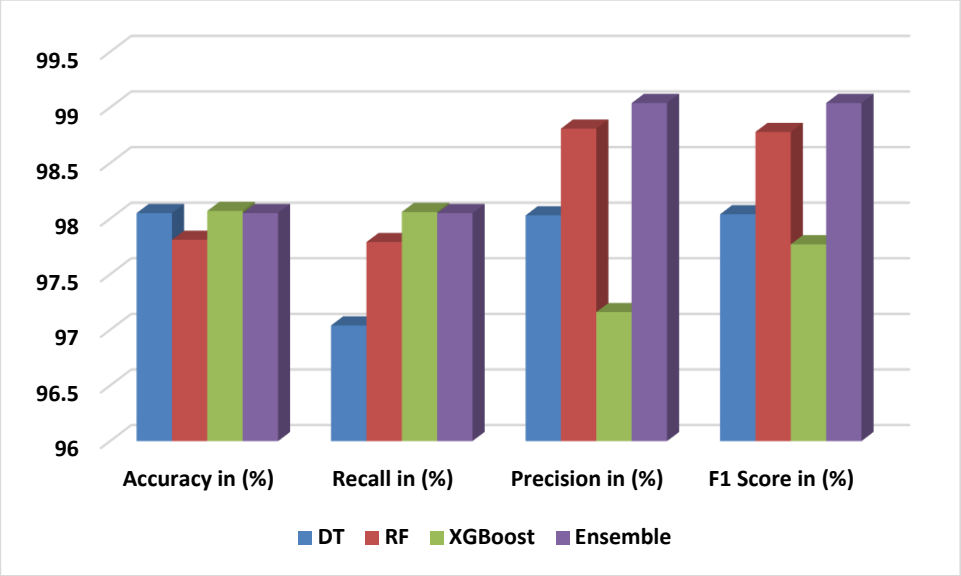

Fig 4: Graphical representation of Performance Evaluation metrics

98.05% accuracy means that 98.05% of the occurrences were properly categorised by the Decision Tree method. The precision rate, which shows the proportion of pertinent instances among the detected instances, was 98.03%, while the recall rate, which measures the percentage of pertinent examples properly identified, was 97.04%. The precision and recall combined measure, or F1 score, attained 98.04%, showing a balanced performance in both areas.

A high level of accuracy was also shown by the Random Forest method, which had a recall rate of 97.79% and a precision of 97.81%. The precision rate was 98.81%, indicating that a significant portion of the discovered instances were meaningful. The F1 score of 98.78% demonstrated a perfect harmony between recall and precision.

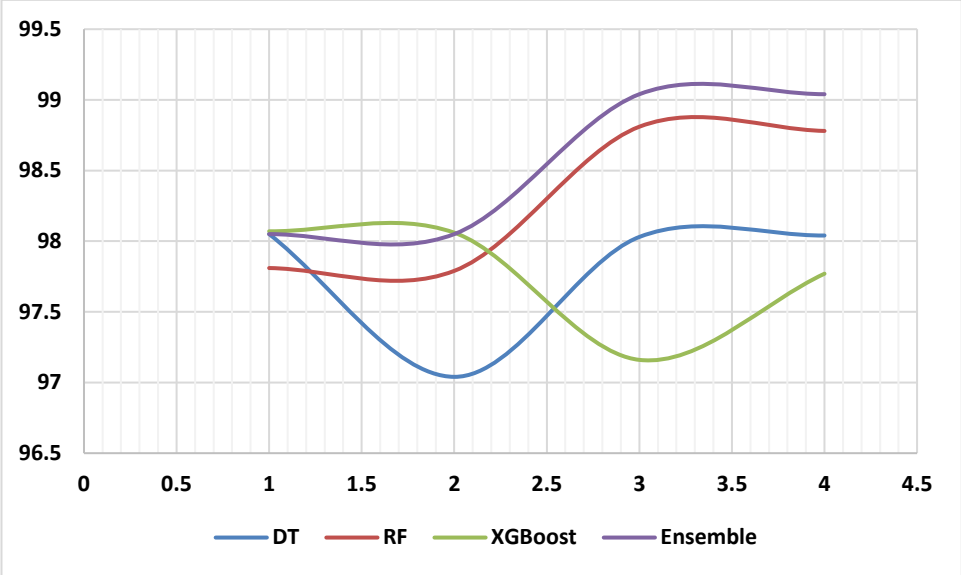

Fig 5: Comparison of Different method with Ensemble method

In comparison to the other algorithms, the XGBoost method displayed a significantly lower accuracy of 96.07%. Both the recall and precision rates were 96.06%. The precision and recall weighted average, also known as the F1 score, was 95.97%. In terms of accuracy, recall, precision, and F1 score of 98.5%, the combined method performed on par with the decision tree method. When several algorithms are employed, the prediction accuracy improves. The evaluation findings demonstrated that the combined Decision Tree, Random Forest, and algorithms consistently achieved high F1 precision, recall, and precision scores. The XGBoost algorithm fared remarkably well by these measures despite being very efficient. This finding demonstrates how these autonomous learning systems are capable of identifying important events and making accurate forecasts.

6. Conclusion

The outcomes demonstrated that DT and RF both had incredibly high precision, with DT having a precision of 98,05% and RF having a precision of 97,81%. With DT and RF of 97,4% and 97,79%, respectively, these algorithms showed startlingly high recall rates. The two algorithms had remarkable precision rates, with accuracy rates of 98,03% and 98,81%, respectively. With scores of 98,04% and 98,78%, respectively, DT and RF received very high F1 ratings. A variety of machine learning algorithms, including Decision Tree (DT), Random Forest (RF), XGBoost, and other methods, have demonstrated to be quite successful in handling classification jobs. These algorithms were assessed using a number of performance criteria, including F1 score, recall, accuracy, and precision. The ensemble technique, which performed as well as DT, proved that integrating numerous algorithms can increase prediction accuracy. Overall, these results show how precisely the ensemble, DT, RF, and XGBoost approaches classify situations. Each approach may be a good option, depending on the specific criteria and conditions of the classification challenge. Academics and professionals can utilise these insights to select the appropriate algorithm based on the characteristics of the dataset and the required performance criteria.

References:

[1] M. Shafiq, Z. Tian, A.K. Bashir, A. Jolfaei, X. Yu,“ Data mining and machine learning methods for sustainable smart cities traffic classification: a survey”, Sustain Cities Soc, 60 (2020), Article 102177

[2] Z. Lv, B. Hu, H. Lv, “Infrastructure monitoring and operation for smart cities based on IoT system”, IEEE Trans Ind Inf, 16 (3) (2020), pp. 1957-1962, 10.1109/TII.2019.2913535

[3] N. Hussain, P. Rani, H. Chouhan, U.. Gaur, “Cyber security and privacy of connected and automated vehicles (CAVs)-based federated learning: challenges, opportunities, and open issues”, Federated learning for IoT applications, Springer (2022), pp. 169-183

[4] P. Rani, N. Hussain, R.A.H. Khan, Y. Sharma, P.K. Shukla, “Vehicular intelligence system: time-based vehicle next location predicion in software-defined internet of vehicles (SDN-IOV) for the smart cities”, Intelligence of things: AI-IoT based critical-applications and innovations, Springer International Publishing, Cham (2021), pp. 35-54.

[5] O.A. Wahab, A. Mourad, H. Otrok, J. Bentahar, “CEAP: SVM-based intelligent detection model for clustered vehicular adhoc networks”, Expert Syst

Appl, 50 (2016), pp. 40-54, 10.1016/j.eswa.2015.12.006

[6] J. Yang, Z. Fei, “Broadcasting with prediction and selective forwarding in vehicular networks”, Int J Distrib Sens Netw, 9 (12) (2013), Article 309041.

[7] H.M. Song, J. Woo, H.K. Kim, “In-vehicle network intrusion detection using deep convolutional neural network”, Veh. Commun., 21 (2020), Article 100198

[8] M.V. Mahoney, “A machine learning approach to detecting attacks by identifying anomalies in network traffic”, Florida Institute of Technology (2003)

[9] H. Ye, L. Liang, G. Ye Li, J. Kim, L. Lu, M. Wu, “Machine learning for vehicular networks: recent advances and application examples”, IEEE Veh Technol Mag, 13 (2) (2018), pp. 94-101.

[10] J. Erman, A. Mahanti, M. Arlitt, “QRP05-4: internet traffic identification using machine learning”, Proceedings of the *IEEE globecom 2006*, San Francisco, CA, USA (2006), pp. 1-6, 10.1109/GLOCOM.2006.443

[11] J. Liang, J. Chen, Y. Zhu, R. Yu, “A novel Intrusion Detection System for Vehicular Ad Hoc Networks (VANETs) based on diffrences of traffic flow and position”, Appl Soft Comput, 75 (2019), pp. 712-727, 10.1016/j.asoc.2018.12.001

[12] P. Khobragade, P. Ghutke, V. P. Kalbande and N. Purohit, "Advancement in Internet of Things (IoT) Based Solar Collector for Thermal Storage System Devices: A Review", 2022 2nd International Conference on Power Electronics & IoT Applications in Renewable Energy and its Control (PARC), pp. 1-5, 2022.

[13] Mohammed B, et al. Edge computing intelligence using robust feature selection for network traffic classification in intenet-of-things. IEEE Access 2020;8: 224059–70.

[14] Gad AR, Nashat AA, Barkat TM. Intrusion detection system using machine learning for vehicular Ad hoc networks based on ToN-IoT dataset. IEEE Access 2021; 9:142206–17.

[15] Yang L, Moubayed A, Hamieh I, Shami A. Tree-based intelligent intrusion detection system in internet of vehicles. In: Proceedings of the ieee global communications conference (GLOBECOM); Dec. 2019. p. 1–6.

[16] Gao Y, Wu H, Song B, Jin Y, Luo X, Zeng X. A distributed network intrusion detection system for distributed denial of service attacks in vehicular Ad Hoc network. IEEE Access 2019;7:154560–71.

[17] Peng R, Li W, Yang T, Huafeng K. An internet of vehicles intrusion detection system based on a convolutional neural network. In: Proceedings of the IEEE intl conf on parallel & distributed processing with applications, big data & cloud computing, sustainable computing & communications, social computing & networking (ISPA/BDCloud/SocialCom/SustainCom); 2019. p. 1595–9.

[18] W. Wu et al., "A survey of intrusion detection for in-vehicle networks", IEEE Trans. Intell. Transp. Syst., vol. 21, no. 3, pp. 919-933, Mar. 2020.

[19] J. Liu, S. Zhang, W. Sun and Y. Shi, "In-vehicle network attacks and countermeasures: Challenges and future directions", IEEE Netw., vol. 31, no. 5, pp. 50-58, Sep. 2017

[20] D. Kosmanos et al., "A novel intrusion detection system against spoofing attacks in connected electric vehicles", Array, vol. 5, Mar. 2020.

[21] Y. Sun et al., "Attacks and countermeasures in the Internet of vehicles", Ann. Telecommun., vol. 72, no. 5, pp. 283-295, 2017.

[22] E. Seo, H. M. Song and H. K. Kim, "GIDS: GAN based intrusion detection system for in-vehicle network", Proc. 16th Annu. Conf. Privacy Secur. Trust (PST), pp. 1-6, Aug. 2018.

[23] M.-J. Kang and J.-W. Kang, "Intrusion detection system using deep neural network for in-vehicle network security", PLoS ONE, vol. 11, no. 6, Jun. 2016

[24] H. Lee, S. H. Jeong and H. K. Kim, "OTIDS: A novel intrusion detection system for in-vehicle network by using remote frame", Proc. 15th Annu. Conf. Privacy Secure Trust (PST), pp. 57-5709, Aug. 2017.

[25] T. Marsden, N. Moustafa, E. Sitnikova and G. Creech, "Probability risk identification based intrusion detection system for SCADA systems", Proc. Int. Conf. Mobile Netw. Manage., pp. 353-363, 2017.

[26] Yoon, H. . (2023). A Quantitative Evaluation for Usability under Software Quality Models. International Journal on Recent and Innovation Trends in Computing and Communication, 11(3), 24–29. https://doi.org/10.17762/ijritcc.v11i3.6194

[27] Deshpande, V. (2021). Layered Intrusion Detection System Model for The Attack Detection with The

Multi-Class Ensemble Classifier . Machine Learning Applications in Engineering Education and Management, 1(2), 01–06. Retrieved from http://yashikajournals.com/index.php/mlaeem/article/view/10

[28] Kumbhkar, M., Shukla, P., Singh, Y., Sangia, R. A., & Dhabliya, D. (2023). Dimensional reduction method based on big data techniques for large scale data. Paper presented at the 2023 IEEE International Conference on Integrated Circuits and Communication Systems, ICICACS 2023, doi:10.1109/ICICACS57338.2023.10100261 Retrieved from www.scopus.com