IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 17 September 2023, accepted 19 November 2023, date of publication 28 November 2023,
date of current version 8 December 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3337378

RESEARCH ARTICLE

One Versus All Binary Tree Method to Classify Misbehaviors in Imbalanced VeReMi Dataset

OMESSAAD SLAMA[1], MOUNIRA TARHOUNI[1], SALAH ZIDI[1], AND BECHIR ALAYA[1,2]
[1]Hatem Bettaher Laboratory, IResCoMath, University of Gabes, Gabes 6029, Tunisia
[2]Department of Management Information Systems and Production Management, College of Business and Economics, Qassim University, Buraydah 51452, Saudi Arabia

Corresponding author: Omessaad Slama (slama.oumsaad@gmail.com)

This work was supported in part by the Deputyship for Research and Innovation, Ministry of Education; and in part by Qassim University, Saudi Arabia.

ABSTRACT Nowadays, transportation networks depend heavily on the technology known as vehicular ad hoc networks (VANETs). VANETs enhance traffic control and road safety while also enabling vehicle-to-vehicle communication using basic safety messages (BSM), which are susceptible to different kinds of attacks. This study focuses on techniques for detecting and classifying misbehavior in VANETs while dealing with unbalanced data. In order to ensure equal treatment of minority and majority categories, we provide a novel method called One vs. All Binary Tree (OVA-BT). This approach separates binary classifiers for each kind of misbehavior and provides specific assessment metrics for each kind of misbehavior. We evaluate our experiment using five-fold cross-validation with six individual models of ML and an ensemble classifier. The findings demonstrated that the use of OVA-BT enhances the classification accuracy when compared to a traditional single multi-class model and that the classifier ensemble's classification performance is greater than the best individual model on the testing set.

INDEX TERMS One versus all binary tree (OVA-BT) strategy, imbalanced data, VeReMi extension dataset, VANETs, misbehavior detection.

I. INTRODUCTION

Intelligent transportation systems currently require the use of VANETs, which offer effective interaction and cooperation between infrastructure and vehicles. Several advantages of VANETs include increased traffic control efficiency, enhanced road security, and improved conductor experience [1].

Various infrastructures form the basis of VANET such as On-Board Units (OBUs), Roadside Units (RSUs), and Certificate Authorities (CAs) [2], [3]. A vehicle is equipped with an OBU which periodically diffuses its state to those nearby. An RSU [4], a crucial component of the infrastructure, is installed on the side of the road and communicates with neighboring OBUs by sharing traffic and weather information while assisting OBUs in establishing internet access.

There is a distinguished infrastructure known as the Central Authority (CA), which offers services including the administration and registration of any network components and also the cancellation of certificates in the case of misbehavior [5]. VANETs support a variety of interaction methods, including vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I). A Basic Safety Message (BSM) [6], including the sender's actual location, speed, acceleration, and direction, as well as a temporary pseudonym to determine the sender, is broadcast every 100 milliseconds to ensure communication between vehicles.

However, because of the V2X connection, VANETs are an open-access environment that is much more vulnerable to misbehavior, causing significant safety problems.

Misbehavior in VANETs is described as any intentional or unintentional activity by a vehicle that diverges from the established norms, protocols, or guidelines of the VANET system. Misbehavior may manifest itself in a number of ways, such as careless driving, moving violations, illegal access, forgery of data, or even dangerous assaults. These illegitimate or dishonest behaviors have the potential to affect VANET's regular functioning, affect the security of vehicles and

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

passengers, and damage the network's reputation for efficiency [7].

In the context of challenging issues with categorization and when dealing with an unbalanced distribution of instances of misbehavior across various categories, it might be difficult to identify and classify misbehavior in vehicular communications.

Unbalanced databases occur when specific kinds of misbehavior are considerably underrepresented in comparison to others.

Complex issues with categorization cannot always be handled with single multi-class categorization methods; they struggle with this imbalance because they tend to perform better for the majority category while performing less well for the minority misbehaving category. As a result, it is difficult to accurately identify and categorize occurrences of misbehavior, causing possible weaknesses in security.

In contrast to single multi-class categorization, this research suggests the One-Versus-All Binary Tree (OVA-BT) strategy as an efficient solution to these problems.

The OVA technique divides the multi-class categorization problem into a group of binary sub-issues [8], [9], considering every category as an independent binary classification work. This allows binary classification algorithms to handle multi-class classification problems.

Numerous OVA-based multi-class classification techniques have been suggested using this concept [10], [11].

We can successfully tackle the issue of imbalanced data to identify and classify the misbehavior in VANETs by utilizing the OVA-BT approach.

In the context of unbalanced data issues in VANETs, the OVA binary tree technique has a number of benefits. First of all, it enables the representation and categorization of every misbehavior category separately. The classifier can concentrate on extracting the distinctive patterns and features of every kind of misbehavior by treating every category as a distinct binary categorization problem. This specialized knowledge improves the precision of categorization, especially for minority misbehavior categories that might not be well represented in the database.

The OVA-BT method, on the other hand, offers flexibility in choosing the best binary classification algorithms for each category. For accurate detection, specific algorithms must be designed for many kinds of misbehavior that may exhibit different features. The OVA-BT methodology makes use of specific binary classifiers to allow the use of methods specifically created for each type of misbehavior, improving the effectiveness of classification.

Additionally, by organizing the multi-class imbalanced dataset into a hierarchical structure, the OVA-BT technique successfully overcomes this issue. The classification process is streamlined by this hierarchical decomposition, which also lessens the complexity involved in handling unbalanced data directly. The OVA-BT approach's interpretability also makes it possible to track down misclassification choices, which helps in comprehending and improving the categorization algorithms.

In this study, we will provide a thorough examination of the security risks provided by improper behavior in VANETs, focusing on the situation of unbalanced datasets. We will look at the drawbacks of traditional classification methods and emphasize the advantages of using the OVA-BT technique. Different machine-learning approaches are used and compared for learning and detecting malicious behavior. This paper employs an artificial intelligence (AI) methodology. The key contributions of this study are, in particular:

We suggest an efficient OVA-BT method to solve the problems caused by unbalanced datasets while categorizing multi-class misbehavior in vehicular communication. By adopting a binary tree structure, this method expands on the traditional OVA technique by providing for the distinct analysis and classification of each misbehavior category.

We provide thorough experimental assessments between the OVA Binary Tree approach and traditional classification methods in terms of area under the curve (AUC), recall, F1-score, and precision. We present in-depth simulation findings describing the functionality of six distinct machine learning algorithms (Support Vector Machine (SVM), Logistic Regression (LR), Naïve Bayes (NB), K-Nearest Neighbors (KNN), Random Forest (RF) and Multi-Layer Perception (MLP)) and ensemble learning classifiers.

The remainder of the paper is structured in the following manner: A literature review is presented in Section II. Section III describes the generated dataset, the proposed OVA-BT technique, the classification methods, and the performance evaluation metrics. Section IV presents the findings and discussions, while Section V provides the conclusion.

II. LITERATURE REVIEW

In order to address the multi-class classification issue when dealing with unbalanced data, numerous experiments have been carried out. This section begins with an outline of the most recent advances in artificial intelligence for multi-class categorization, followed by an overview of imbalanced data in VANETs environment.

A. HANDLING MULTI-CLASS CLASSIFICATION

Many works have dealt with the multi-class classification issue. The decomposition-based methods such as the OVA strategy have drawn a lot of interest. The OVA division approach splits a k-category multi-class categorization issue into k distinct binary algorithms, which are trained to differentiate instances of one category against instances of all other categories.

In [10], Rifkin and Klautau show the effectiveness of the OVA technique, highlighting its competitive precision and ease of use, especially when using well-tuned regularized models such as SVM. Practical circumstances frequently involve complex databases that may not conform to the assumptions of the One vs. All technique, like unbalanced category distributions and extensive connections. Although the work offers an insightful viewpoint, it failed to address the range of difficulties observed in real-world multi-class categorization jobs, in which different approaches would be more appropriate.

To solve the shortcomings of the earlier suggested Weighted Linear Loss Twin Support Vector Machine (WLTSVM) for binary categorization, the Weighted Linear Loss Multiple Birth Support Vector Machine (WLMSVM) was introduced in [12]. The benefits of WLMSVM include improved multiclass categorization efficiency, which is attained by granulating data and using the ''all-versus-one'' technique. Comparing this method to the OVA approach employed in multiple WLTSVM, the computational complexity is greatly reduced. The solution of linear equations is also made simpler by the addition of weighted linear loss. WLMSVM's success is shown in the research through findings from experiments on a variety of databases; however, it's vital to take into account any potential limits WLMSVM may have when addressing situations that are more complicated and challenging than benchmark databases.

A revolutionary One Versus All multi-class classification technique centered on the cooperative evolution of SVM was created in [13] and is known as MC^2 ESVM. The method divides an N-categories issue into N sub-issues, which are then cooperatively improved. It has been demonstrated that the strategy works well for multiclass categorization issues while maintaining a manageable number of support vectors. The method can, however, be computationally costly, difficult to converge, and hyperparameter selection-sensitive.

The authors of [14] discuss the difficulty of carrying out diagnostics while concurrent problems exist. The suggested OVA class binarization technique has a number of benefits. It improves defect detection precision while lessening the requirement for a comprehensive set of data instances. It optimizes the data collection procedure by forecasting concurrent faults using training examples of typical and singular faults. The prospective use of the One versus All evaluation technique is shown through an investigation utilizing a support vector machine and C4.5 decision tree. Nevertheless, as laboratory-gathered data might not completely reflect the nuances of actual bearing situations, it is insufficient to determine if this method is beneficial for more complicated and varied circumstances in the real world.

Khairudin et al. [15] developed accurate detection and classification methods for classifying 4 kinds of human intestinal parasites using the OVA technique and tested their solution utilizing k-NN, SVM, and ensemble learning. The database's constrained nature, especially its particular collection of helminth species and ova pictures, should be taken into account. Real-world scenarios can contain a wider variety of parasite types and changes in the quality of images, demanding additional validation and database development to guarantee the algorithms' efficacy in different healthcare environments.

In the framework of Random Forest, Adnan and Islem [16] study the use of the OVA binarization strategy. Binarization approaches can make multiclass categorization issues simpler and enhance the performance of the Random Forest algorithm, which has the opportunity to increase the precision of predictions. The paper uses ten databases from the UCI Machine Learning Repository for a thorough experimental evaluation to show the efficacy of this method. The main shortcoming of this study's assessment is that it is concentrated on a small number of categories within each dataset instead of taking into account a wider variety of categories.

In [17], Allwein et al. presented an approach for dealing with multi-class classification issues by breaking them down into numerous binary issues. Margin-based binary learning models are streamlined by this unification, which also offers a universal technique for mixing models. The flexible architecture can be used with several categorization techniques, like SVM, AdaBoost, regression, logistic regression, and decision-tree models. The experimental findings of AdaBoost and SVM demonstrate weakness in validation, underlining the necessity for broader methods of categorization to increase the legitimacy and application of the methodology.

For the purpose of designing neural network ensembles, McDonnell et al. [19] suggested a novel cooperative ensemble learning system (CELS). The concept behind CELS is to motivate various single networks in an ensemble to train various components or elements of a training database so that the ensemble can more effectively learn the entire training database. Both the Australian Credit Card Assessment Challenge and the Mackey-Glass time series prediction issues have been used to test CELS.

For the challenge of data categorization, Chen and Alahakoon [18] suggested a hybrid learning strategy called Learning-Neuro-Evolution of Augmenting Topologies (L-NEAT). The NEAT method is used on L-NEAT to streamline evolution by breaking down the entire problem domain into smaller tasks. Smaller tasks are then learned. The efficacy of NEAT is nevertheless affected by the selection of the evolutionary parameters and the particular database, which is inadequate.

In [19], the multi-class categorization degradation of NEAT is addressed by applying the class binarization approaches of One-vs-All and One-vs-One, where binary models are the various NEAT-evolved neural networks. OvA-NEAT and OvO-NEAT, two ensemble techniques, are created to exceed the regular NEAT in terms of precision and effectiveness.

This paper presents a novel method for detecting multi-class misbehavior in vehicular communication using the OVA-BT technology and the VeReMi extension database. Although there are other versions of the database, our research focuses on using OVA-BT for the first time on the original VeReMi extension database, a less-used technique at the time. The lack of benchmark data from other OVA-BT approaches on this database prevented direct comparisons, despite the fact that we understand the value of benchmarking against current ideas.

B. TACKLING UNBALANCED DATA

The unbalanced class number is a significant issue with supervised Machine Learning techniques. It typically happens once the database traces data relating to a real environment. In fact, in such a setting, the data is frequently unbalanced, and the models developed from the data might be more accurate for the majority category but incredibly inaccurate for the rest of the categories. Similar problems are seen in the VeReMi Extension database, which has been widely utilized to learn ML classifiers to classify misbehaving nodes in VANETs. There are several approaches to solve this issue: changing the machine learning approach, adding an inaccurate categorization fee, and data sampling [20].

In the context of VANETs, the researchers [21] attempt to oversample the minority category by employing the Synthetic Minority Oversampling Technique (SMOTE), an approach to data augmentation for the less category [22], [23]. It duplicates instances from the less categories in order to avoid the issue of unbalanced datasets. With this approach, DDoS Detection in VANETs can be realized efficiently.

The researchers in [24] used the ToN-IoT network database and proposed a detection system for vehicular communication.

Two distinct problems concerning the network data in this database are unbalances in categories and missing values. The Chi-squared and SMOTE techniques were used to tackle this problem. The ToN-IoT database is vulnerable to misbehavior than earlier datasets like NSL-KDD, KDD-CUP99, and UNSW-NB15 [25], [26]. SMOTE is used by the authors in [27] to address the issue of an imbalanced database. In SMOTE, new instances of the minority categories are combined to bring their total number to a level that is comparable to or equal to the total instances of the dominant category.

In [28], the authors utilized SMOTE and DSSTE to prevent class unbalance. The DSSTE approach is used to raise the minority occurrences and reduce the majority instances within a challenging set.

They employed the unbalanced VeReMi initial database to demonstrate the effects of an unbalanced dataset on the classification of misbehavior in Vehicular communications. This study's primary weakness is that it only considers a small subset of misbehavior types when discussing the identification of DDoS for vehicular communication. The work is helpful in tackling this particular subset of security issues, but it falls short of offering a comprehensive answer to the complex and always-changing threat environment that VANETs must contend with. To guarantee the security system's durability in a variety of real-world settings, a more successful strategy would require taking a wider range of misbehavior types and variants into account.

When using the unbalanced VeReMi Extension dataset, containing a large number of misbehavior types, in our earlier research [29], we discovered insufficient accuracy in the single multiclass classification of misbehavior in vehicular communication. In this study, we attempt to employ the OVA Binary Tree technique to avoid these issues in order to enhance the outcomes, and we track changes in the model's output. The goal of this work is to clearly demonstrate how an unbalanced dataset affects the development of intelligent algorithms for classifying cyber-attacks.

In addition, the utilization of a real-world database with a wider variety of classes is one of the key advantages of our study. Our work makes use of a more varied and complicated database with 20 classes, whereas the researcher's study may have used synthetic or smaller databases with fewer classes. One of our research findings is that we have handled a wide range of real-world classification scenarios by using various categories, which helps us better grasp the problems and solutions for multiclass classification.

III. MATERIALS AND METHODS

This section describes the generated dataset, including its attributes and statistical data, the applied methods for classification, the data preparation, the suggested approach based on an One Versus All binary tree for classification, and the performance evaluation metrics used to assess the system's effectiveness across all options and methods.

A. GENERATED DATASET

The Vehicular Reference Misbehavior Dataset Extension (VeReMi) is used in this study, which was developed using OMNET++ [30] and VEINS [31] on the Luxembourg SUMO traffic scenario (LuST) [32]. The simulations in this dataset range in density from low to high traffic times. Each simulation is made up of log and ground truth files. There is a single ground truth file explaining how a vehicle really behaves in the interconnected system, and it is used for running simulations. An attacker type is included in the ground truth file as well, separating genuine vehicles from misbehaving vehicles. On the other hand, in a simulation, the amount of log files equals the amount of the network's nodes. Each node generates a log file, including all the Basic Safety Message received from other devices.

Since there are as many log files as receivers, the first phase is to merge all of the different log files into one file. Then, log files and ground truth files must be joined by mapping the ground truth file to the log files for each simulation.

The unique message ID identifier is present in both ground truth files and log files. The attacker type in the ground truth file should be transferred to the information in the merged log file to build a merged dataset for a single scenario, as shown in Figure 1.

In order to produce a labeled database, we introduced an identifiable characteristic named "class" for our target class into the merged database.

We assigned the class number 0 to the genuine vehicles while for misbehaving vehicles, the class number is between 1 and 19.

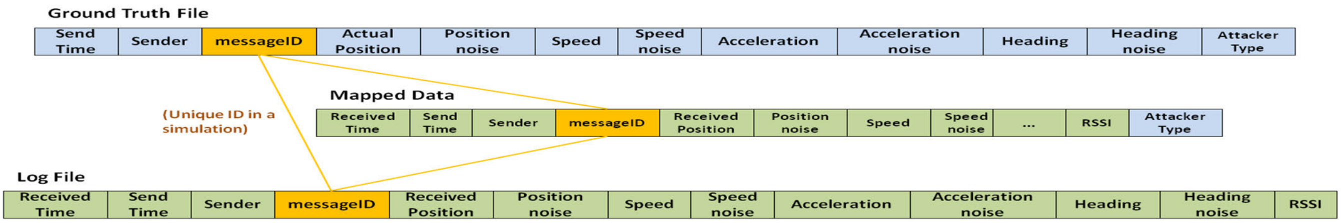


FIGURE 1. Log file and Ground truth file data extraction to generate VeReMi Extension.

TABLE 1. (a) Detailed information on the data and (b) statistics in the produced database.

(a)

Detailed Statistics	
Item	Numbers
Number of instances overall	3,194,808
Number of columns	29
Number of characteristics	13
Numerous numerical characteristics	12
Numerous categorical characteristics	1
Number of classes	20

(b)

Statistical Numbers for Each Class				
class number	Class Label	Type	Number of instances	% w.r.t. total database instances
0	Normal	Normal	1900539	59,488%
16	Constant Speed Offset	Fault	175391	5,489%
13	Eventual Stop	Attack	131305	4,109%
15	Constant Speed	Fault	129270	4,046%
14	Grid Sybil	Attack	126724	3,966%
18	Random Speed	Fault	86883	2,719%
19	Random Speed Offset	Fault	82100	2,569%
6	DoS Random	Attack	44359	1,388%
11	DoS Disruptive Sybil	Attack	44337	1,387%
17	Delayed messages	Attack	44310	1,386%
3	Random Position	Fault	43857	1,372%
1	Constant Position	Fault	43653	1,366%
2	Constant Position Offset	Fault	43567	1,363%
10	Disruptive	Attack	43264	1,354%
12	DoS Disruptive	Attack	43118	1,349%
9	Data Replay Sybil	Attack	42790	1,339%
8	Data Replay	Attack	42583	1,333%
4	Random Position Offset	Fault	42575	1,332%
7	DoS Random Sybil	Attack	42258	1,322%
5	DoS	Attack	41925	1,312%

The database includes a number of errors caused by faulty On Board Units or vehicle sensors that result in inaccurate results for position, acceleration, velocity, and heading, as well as a variety of cyber-attacks when the vehicle sends an incorrect message [7]. For easier reading, Table 1 (a, b) summarizes additional details and statistics regarding the created dataset.

However, this dataset has an unbalanced class structure; 59,488% of the dataset is made up of

records in the normal category, while classes like DoS Random Sybil and Dos account for 1,322% and 1,312% of the total dataset, respectively.

You can access the generated database of this paper at http://dx.doi.org/10.17632/k62n4z9gdz.1 which is an open-source online data repository maintained by Mendeley Data [33].

B. CLASSIFICATION APPROACHES

This section describes the various machine learning classifiers that are used in this investigation.

1) RANDOM FOREST (RF)

It is [34] and [35] a supervised ML technique that blends various decision trees to produce predictions. By choosing portions of the training data and characteristics at random, it generates a series of decision trees. For classification or regression, the results from each tree are combined by voting or averaging to produce the final prediction. Accuracy is increased, high-dimensional data is handled, and overfitting is robustly prevented via Random Forest. It also provides measurements of feature relevance. For classification and regression applications, Random forests are an effective and widely utilized method.

2) LOGISTIC REGRESSION (LR)

It is [36] a supervised ML method employed to address multivariate classification issues. It models the association between the input variables and the likelihood of a favorable outcome using a logistic function. In the training stage, the model learns the most suitable weights, and a decision boundary is created to distinguish the categories. Interpretability is a benefit of logistic regression since the coefficients show how each input variable affects the likelihood of the outcome. To avoid overfitting, it can use regularization techniques. Although Logistic Regression is straightforward, understandable, and effective, it may struggle with unbalanced classes or complex correlations.

3) SUPPORT VECTOR MACHINE (SVM)

In this classifier [37], suppose that T is a training set for binary labeled categorization: T is equal to $\{(x_1,\ y_1),\ (x_2,\ y_2), \ldots (x_n,\ y_n)\}$, in which y_i is either $+1$ *or -1*. A support vector machine looks for a hyper-plane with the maximum margin separating it from the nearest point. The definition of a hyper-plane is as follows:

$$w^T x - b = 0, \tag{1}$$

where b is the bias value and w is a norm vector. The data points can be split up if x_i meets the criteria listed below:

$$\begin{pmatrix} w^T x_i + b \geq +1, y_i = +1; \\ w^T x_i + b \leq -1, y_i = -1. \end{pmatrix}. \tag{2}$$

The ones that are nearest to the hyper-plane are referred to as support vectors. When two parallel hyper-planes are separated, the distance is represented as

$$\gamma = \frac{2}{\|w\|}. \tag{3}$$

The best hyper-plane is determined by its highest value of γ. This issue can be thought of as the related optimization method:

$$\underset{w,b}{max} \frac{2}{\|w\|} \tag{4}$$

$$s.t.y_i \left(w^T x_i + b \geq +1 \right), \quad i = 1, 2, \ldots n. \tag{5}$$

4) K-NEARST NEIGHBOURS (KNN)

For classification and regression problems, the K-Nearest Neighbors (KNN) [38] supervised machine learning classifier method is utilized. In the training set, it assigns a new observation to the majority category of its KNN. The approach entails picking a number for K, determining the separations among the new observations with the training points, determining the K nearest neighbors, determining the majority category, and making predictions for each data point in the test database.

5) NAÏVE BAYES (NB)

It is (NB) [39] a supervised ML technique for classification that is founded on the Bayes theorem and presumes feature independence. To predict new occurrences, it calculates probabilities using training data. The Naive Bayes algorithm is frequently utilized for categorization tasks since it is easy to use and effective. When the independence assumption is broken, it cannot perform as well.

6) MULTI-LAYER PERCEPTRON (MLP)

A sort of artificial neural network used in artificial intelligence is called an mlp (multi-layer perceptron). It is made up of layers of connected nodes, with an input layer for taking in data, hidden layers for doing complicated alterations, and an output layer for displaying findings. Mlps learn through data by modifying the biases and weights of the nodes during training to produce precise predictions. They are frequently employed in tasks such as regression, categorization, and pattern recognition in different fields and are well renowned for their capacity to model complex patterns in data.

7) BUILDING ENSEMBLE LEARNING

To provide an effective model, the ensemble model is built by carefully merging base models. In order to address a classification issue that cannot be easily handled by either of the single classifiers, [40] ensemble learning applies a variety of learning classifiers, which is more efficient than single models. Because it is effective and simple to use, the majority voting technique was employed [41]. Suppose class wc∗ is selected by the ensemble classifier. The mathematical formula for majority voting, commonly referred to as plurality voting, is given below (6):

With: dt,c appartient 0,1 is the t-th model's decision,
t = 1, . . . , T, and the model number is T.
c = 1, . . . , C, and the category number is C.

We used training data to train basic classifiers such as NB, SVM, MLP, LR, RF, and KNN. We used testing data following training to assess the accuracy

of our algorithms, with each classifier providing a unique prediction. The predictions made by these algorithms serve as extra input for the classifier ensemble, which functions as a merged classifier trained to generate the ultimate prediction. The suggested ensemble classifier is illustrated in Figure 2.

$$\sum\nolimits_{t=1}^{T} d_{t,c} = \max_{m} \sum\nolimits_{t=1}^{T} d_{t,c} \tag{6}$$

C. DATA PREPARATION

To prepare the raw data, which contains noise, inconsistencies, and unnecessary information, before it can be utilized for evaluation or the modeling procedure, we started with pre-processing, as explained below, before applying the previously mentioned algorithms into action. As seen in the

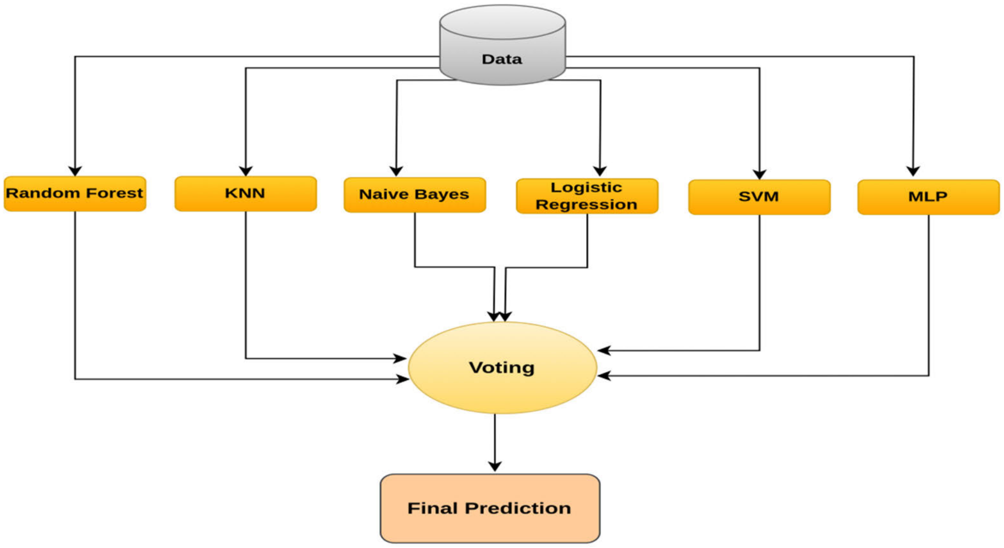


FIGURE 2. Suggested ensemble learning classifier.

details below, we then picked the relevant features for the dataset.

In addition, 70% of the database was used for training and 30% for testing.

Using a computer with 8 GB of RAM and an Intel Core i5-10300H CPU, the experimental findings were accomplished.

The Jupyter Notebook was then used to create the different ML models for misbehavior detection that were discussed in the section above.

To select the most suitable hyperparameters for every method of classification and to prevent overfitting issues, as described below, we looked at hyperparameter tuning techniques. Our proposed methodology's steps are shown in Figure 3.

1) PRE-PROCESSING

Preprocessing refers to turning unstructured raw data into a clean, organized state that may be used for modeling or analysis. To assure the accuracy and suitability of the data, we discovered, dealt with missing data, and eliminated duplicate entries from the VeReMi extension dataset as part of the essential processes. Additionally, normalizing data entails scaling numerical data to a uniform range. It makes sure that each characteristic has a comparable scale and prevents some traits from outweighing others because of their higher values. A min-max scaler was used to rescale our data to a particular range (for example, between 0 and 1).

2) FEATURE SELECTION MODELS

The selection of the best features has a big impact on efficiency when it comes to feature extraction because it greatly reduces train time and increases accuracy. On the other hand, maintaining features that are only marginally relevant can have a detrimental impact on performance.

In previous work [29], we tested three methods to select the relevant features, such as the Recursive Feature Elimination (RFE) [42], the f-test in one-way Analysis of Variance (ANOVA), and the impact of each attribute on the labeled class feature. When comparing the three methods, the RFE gives the best precision in terms of classification of misbehavior. So, we used it in this study.

Recursive Feature Elimination is a feature selection method that repeatedly removes less significant characteristics to find the ones that are most pertinent to a machine learning assignment. A model is chosen, the attributes are ordered by relevance, the least significant attributes are eliminated, and the procedure continues until the halting requirements are satisfied. RFE takes attribute interactions and dependencies into account, which enhances the efficiency of models while reducing learning time and improving interpretability. This method results in the selection of 12 characteristics from the VeReMi extension dataset: spdx, spdy, spdx_n, spdy_n, posx, posy, hedx, hedy, hedx_n, hedy_n, aclx, acly.

3) CLASSIFICATION OF MISBEHAVIOR

The OVA technique is combined with artificial intelligence algorithms in the suggested method for classifying abnormal behavior, as described in Figure 3, to overcome the difficulties provided by unbalanced data and enhance the precision of the outcomes of classification.

Misbehavior here alludes to an attack or a fault in vehicular communication. The goal is to develop a framework for classification that can correctly identify and categorize various forms of misbehavior in the unbalanced VeReMi extension dataset.

D. TUNING HYPER-PARAMETERS WITH K-FOLD CROSS-VALIDATION

The models whose hyper-parameters must be tuned for this investigation are MLP, SVM, RF, and KNN. Past research suggests that fivefold CV can be performed without losing power, essentially reducing computational time in half as

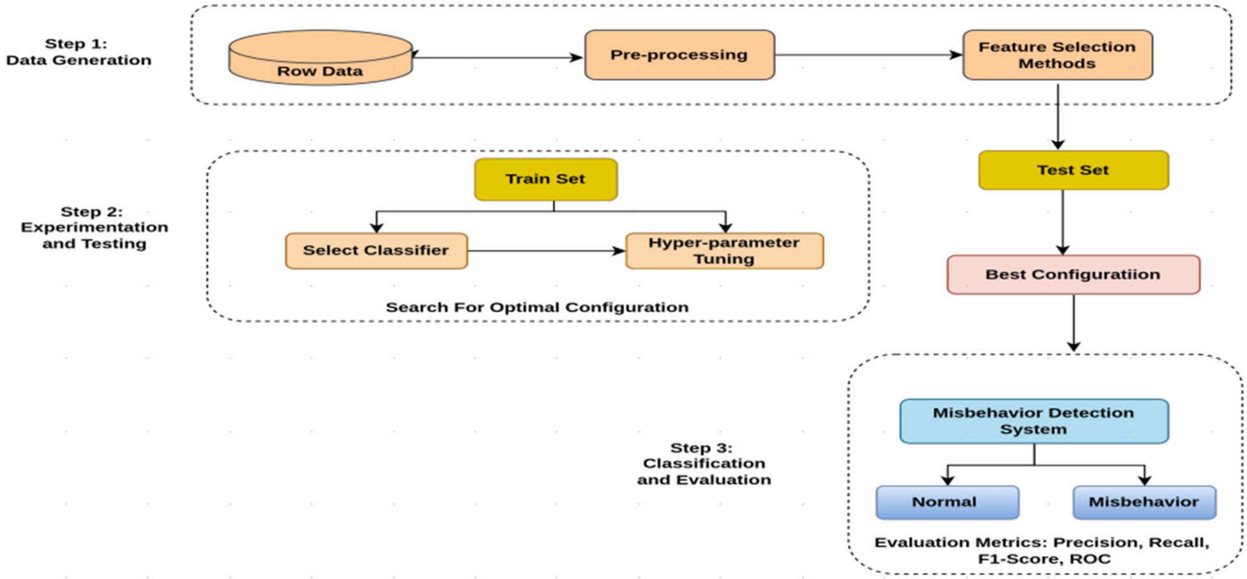

Step 1:
Data Generation
Row Data
Pre-processing
Feature Selection Methods
Test Set
Step 2:
Experimentation and Testing
Train Set
Select Classifier
Hyper-parameter Tuning
Search For Optimal Configuration
Best Configuratiion
Step 3:
Classification and Evaluation
Misbehavior Detection System
Normal
Misbehavior
Evaluation Metrics: Precision, Recall, F1-Score, ROC

FIGURE 3. Diagram of proposed system 5-fold cross validation (CV).

TABLE 2. Optimal Hyper-Parameters.

Model	Hyper-parameters
KNN	n_neighbors = 10
SVM	C: 10 Gamma: 0.0001 kernel: rbf

RF	min_samples_split = 2
	min_samples_leaf = 1

LR	C= 0.01
	penalty = l2

MLP	hidden_layer_sizes = 20
	alpha = 0.05
	solver: sgd

compared to ten-fold CV. Thus, we tweak hyper-parameters using a 5-fold CV [43] as shown in Figure 4.

The 5-fold cross-validation procedure is below:

1. First, divide the input data into five groups.

2. For every team:

3. Select one group to serve as the foundation for testing data collection.

4. Use the rest of the groupings as your train set.

5. Fit the classifier to the train set, and then utilize the testing set to assess the model's effectiveness.

6. The average of the 5 hyper-parameters will then be used to create the final hyper-parameters.

The optimal hyper-parameter outcomes are outlined in Table 2.

E. PROPOSED APPROACH BASED OVA BINARY TREE FOR CLASSIFICATION

In this part, we introduce the suggested approach, the OVA Binary Tree (OVA-BT), which entails the building of a binary tree, the partitioning of non-leaf nodes in the tree using the OVA-BT structure, and the investigation of the binary tree for categorization.

1) ELEMENTS OF THE ONE VS. ALL BINARY TREE AND NODE PRODUCTION PROCESS

In this part, we provide a binary tree-based method

Of hierarchical multi-class categorization [44]. In hierarchical multiclass categorization, the tree structure and its construction phase have a considerable impact on classification results; hence, it is crucial to create an effective tree design [8]. It is crucial to perfectly split the trained examples into two groups (one class against the rest) for every node of the tree in order to build an ideal binary tree.

In this study, we solve multi-class categorization issues using the OVA binary tree strategy and optimize the tree structure using a top-down method.

The OVA-BT is built by beginning with a node with K categories and selecting one distinguishable category from the rest (k-1) categories on a regular basis to split a node with ($K \geq 2$) categories into two sub-nodes.

As will be shown in (7), when the database is unbalanced, the minority classes are prioritized as the "One" class at specific levels and are chosen as the "One" class at each level of the OVA binary tree. This may decrease the possibility of false negatives and increase the binary classifier's accuracy for minority classes.

Every node of a binary-tree is made up of classes and patterns that set details. Based on the data present, nodes can be classified as root, leaf, or internal. The binary tree's root node is found at the top level, containing all classes and serving as the beginning point for training. Leaf nodes only contain one category. Rest classes are found in internal nodes denoted by the symbol nleaf. A level n (where $n = 0; 1; 2; \ldots;$

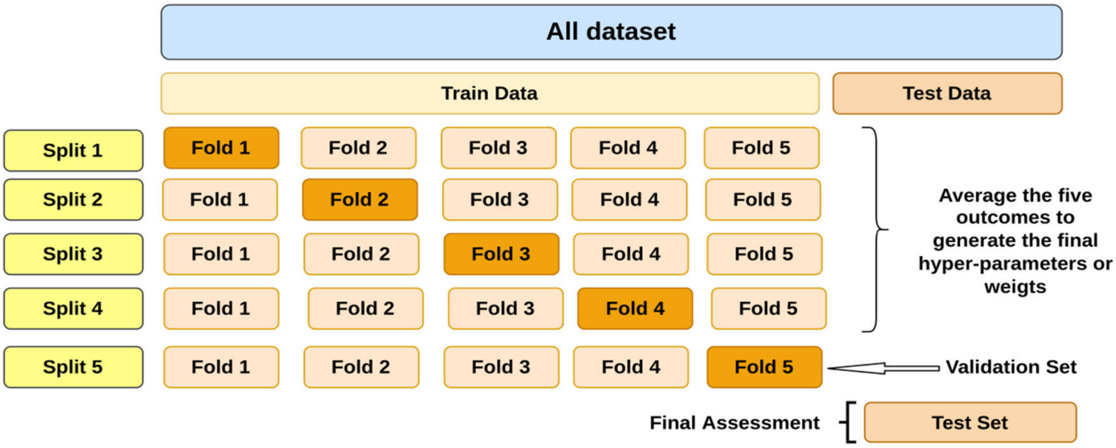


FIGURE 4. OVA-binary tree classification.

K -1) represents a layer. Only the root of the tree is present at the topmost level (n = 0). We find 2 leaf nodes at the lowest level (n=K-1). A leaf node and an internal node are both present at the intermediate levels.

2) PARTITIONING NON-LEAF NODES

The technique used for splitting a nleaf node into two child nodes in a binary tree is important for the classification results [8]. The One vs. All binary tree is used in this experiment to set the conditions for dividing nleaf nodes and building the binary tree while keeping the efficacy and efficiency of tree building into account. To establish a class for producing the leaf node (left child), as defined in (7) and shown in Figure 5,

$$Partition_n = argmin_c \left(w_1.|P_n^c| - w_2.AR\left(P_n^c\right) + w_m.AG\left(P_n^c\right)\right), \tag{7}$$

where the binary tree's nleaf node in step n contains a collection of P_n^c $(\forall n = 0, 1, \ldots, k-2)$ OVA type designs for category c,

$|P_n^c|$ shows the amount of patterns corresponding to the amount of distinct patterns observed in the database,

$AR\left(P_n^c\right)$ represents the average coverage rate of patterns that indicates how well the patterns that have been found cover the occurrences in the database. It is determined by taking into account the proportion of occurrences covered by each pattern and then averaging those coverage rates across all patterns, and $AG\left(P_n^c\right)$ represents the average degree of patterns, measuring how well or extensively the discovered patterns are used or implemented within the framework of a pattern-based organization. It displays the average number of nodes, or structural elements, for each pattern.

The weight w_j, (∀j=1, 2, . . . m) indicates the significance or impact of each feature when making predictions.

A level n nleaf node may be decomposed in a leaf node (left child node), utilizing the category with the lowest number of $Partition_n$ and an internal node (right child node) through the use of the rest categories. The shape of the concatenation of the number of patterns, the average coverage, and the average degree of the pattern set given in (7) was obtained from a scientific investigation of the experiment findings of earlier research [25]. Inferentially, a smaller number of patterns, a lower average degree of patterns, and a higher average coverage rate of patterns in the pattern set under discussion indicate that the pattern set's category differs logically from other categories.

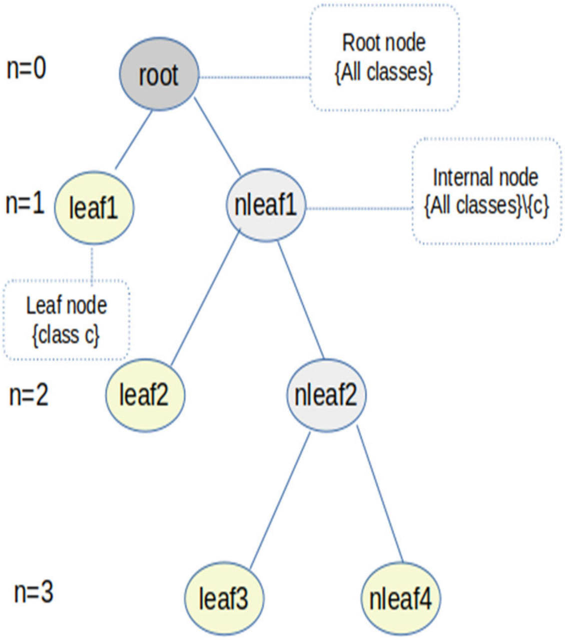


FIGURE 5. Performance of Different Models on VeReMi Extension Dataset in Classic Approach (C-LAMC) and New OVA-BT Approach.

3) MULTI-CLASS CATEGORIZATION USING ONE VS. ALL BINARY TREE EXPLORATIONS

In the root of the tree, a ML model is trained using examples from the first grouping as positive samples and examples from the second grouping as negative samples. Assigning the categories of the first set to the left sub-tree (leaf node) and the categories of the second set to the right sub-tree (nleaf node).

Once each group has been split into two sub-groups using the aforementioned approach, the procedure is repeated until there is just one category per set that represents a leaf in the binary tree.

Starting at the tree's root, each new data is identified. It is decided whether to allocate the input vector to one of the two different groups displayed through each node of the binary-tree by moving the pattern to the left or right subtree. There may be many categories among each of these groupings. This one is done repeatedly up to the instance attains the leaf node for the category it has been allocated to, which is at the bottom of the tree.

F. PERFORMANCE EVALUATION

Because of the imbalanced dataset, traditional evaluation metrics like accuracy can be misleading. A high accuracy score may be achieved by correctly classifying the majority classes while performing poorly on the minority classes. Consequently, evaluation metrics like recall, precision, F1-score, and area under the precision-recall curve (AUC-PRC) are more suitable for assessing classification performance in imbalanced multi-class datasets.

In our investigation, we chose Python because its libraries are helpful for ML. Numpy, Scikit-Learn, Matplotlib, and Pandas are the libraries that we utilized. However, Scikit is designed to work with NumPy and other scientific and mathematical libraries for Python. The Scikit-Learn library does not concentrate on uploading or summarizing the data; instead, it concentrates on modeling. The NumPy library provides operations for manipulating matrices, the Fourier transform, the algebraic domain, and arrays. To work with databases, one requires the Pandas library. It has tools to clean, examine, analyze, and transform data. Python's Matplotlib is a graphics library to visualize the results.

Traditional criteria for assessment, such as accuracy, may be misleading due to the unbalanced sample. Correctly categorizing the majority of categories while scoring poorly on the minority categories can result in a high accuracy rating. Therefore, assessment metrics including recall, precision, confusion matrix, F1-scores, Receiver Operating Characteristic (ROC), and Area Under the Curve (AUC) are better suited for evaluating the efficiency of classification in unbalanced multiclass databases.

A confusion matrix aids in the visualization of a ML classifier's classification. A row represents a predicted category while a column represents an actual category, or the other way around.

As demonstrated in (8), low precision demonstrates that the algorithms output a lot of false positives, whereas high precision shows that the system is capable of differentiating between genuine and malicious nodes.

$$Precision = \frac{TP}{(TP + FP)}. \tag{8}$$

As suggested in (9), recall influences the algorithm's capability to detect malicious nodes; a low recall signifies that inappropriate behavior is more difficult to detect.

$$Recall = \frac{TP}{(TP + FN)}. \tag{9}$$

The harmonic combination of recall and accuracy is known as the F1-score. It sets a trade-off among recall and accuracy, which means a higher F1-score suggests a higher recall and precision value, meaning that the algorithm is going to be more efficient, as stated in (10).

$$F1 - Score = \frac{2 * (Precision * recall)}{(Precision + Recall)}. \tag{10}$$

In True Positive (TP), the system predicted misbehaving accurately; however, in False Negative (FN), the malicious vehicle was misclassified as normal by the algorithm. The model in False Positive (FP) wrongly classified the data instances as illegitimate, but in True Negative (TN), the machine learning models predict the negative instances properly.

We used the ROC and the AUC to visualize and evaluate classifiers; ROC plots are a very helpful tool. By displaying the true positive rate (vertical axis) versus the false positive rate (horizontal axis), it assesses the model's capacity to differentiate between positive and negative categories. The AUC score ranges from 0 to 1, with 1 being the best performance and 0 representing the worst. The ROC AUC measurement makes it simple to compare various models and is excellent for assessing unbalanced databases. Equation 11 below provides the AUC calculation method.

$$AUC = \frac{\sum Rank_{i \in positiveclass} - \frac{M(1+M)}{2}}{M * N}, \tag{11}$$

where Rank i indicates the instance's rank value, M is the amount of positive occurrences, and N indicates the amount of negative instances.

IV. RESULTS AND DISCUSSION

This section addresses multi-class unbalanced classification issues by developing a complete empirical investigation. Using the novel One-Versus-All Binary Tree (OVA-BT) approach for multi-class unbalanced classification in comparison to classic methods, we would like to assess the efficacy of the aforementioned ML classifiers and ensemble learning.

In a prior study [29], we used the VeReMi extension database to build a single multi-class classification called "Classic Learning Approach for Multi-Class Classification" (C-LAMC), but the results were poor since we learned from an unbalanced dataset. The OVA-BT technique, which separates the multi-class categorization issue into a group of binary sub-issues, is developed in this study in an effort to improve these results.

In fact, the performance of our suggested OVA binary tree technique and the Classic were compared in our study using the unbalanced VeReMi Extension database. Our main goal was to evaluate how well our new approach handled multi-class misbehavior identification in this specific database.

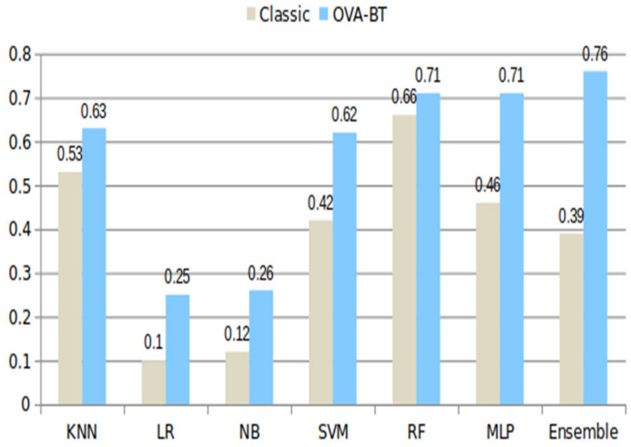


FIGURE 6. Confusion Matrix and Classification Repot for best.

TABLE 3. Performance of dfferent models on veremi extension dataset in classic approach (c-lamc) and new ova-bt approach.

	KNN	LR	NB	SVM	RF	MLP	Ensemble
Classic	0.53	0.1	0.12	0.42	0.66	0.46	0.39
OVA-BT	0.63	0.25	0.26	0.62	0.71	0.71	**0.76**

In this paper, we desire to demonstrate how OVA-BT can effectively support learning from multi-class unbalanced datasets.

The models used were created with Scikit-learn, and the hyper-parameters given in the previous section were maintained. We mention that the preceding section discussed the evaluation metrics. The testing set was utilized to evaluate the suggested method.

As demonstrated in Table 3 and Figure 6, the precision scores demonstrating the rate of identified true attack, we compare the results achieved in the C-LAMC to our new proposed OVA-BT scheme.

The higher precision of OVA-BT is provided by the ensemble classifier (Ensemble_OVA-BT), with 76%. This precision was 39% in C-LAMC. The employ of the OVA-BT strategy is particularly helpful in detection of misbehavior in VANETs. In this instance, the rate of precision increase is 37%.

Ensemble classifier was followed by MLP_OVA-BT with 0.71%, which is an increase of 25% over the classic MLP.

The precision of the KNN_OVA-BT, SVM_OVA-BT, and RF_OVA-BT classifiers was increased by 10%, 20%, and 4%, respectively, whereas the precision of the Naïve Bayes and Logistic Regression models remains with insufficient, both with the use of OVA-

BT and the classic approach C-LAMC.

These outcomes demonstrate the benefits of the OVA-BT approach in solving the difficulties of multi-class misbehavior detection in vehicular communication. We acknowledge that, when looked at separately, the best precision obtained with OVA-BT may appear to be very small. It's important to stress that, when used with the same VeReMi dataset, this precision greatly exceeds classic (C-LAM) methods.

The accuracy of 0.39 attained by the traditional approach serves as a baseline that was previously employed in VANET misbehavior identification. With a precision of 0.76 in the ensemble learning classifier, the new OVA-BT technique significantly outperforms the baseline. This increase is significant since it shows how the OVA-BT method may improve the precision of misbehavior identification.

Our findings show a considerable improvement in the field of Vehicular Ad hoc Network misbehavior identification when compared with classic methods. The precision and robustness of misbehavior detection systems in actual production contexts may be improved, but our work offers a potential starting point for further investigation and enhancement approaches.

The confusion matrix for the better and worse models, MLP and Logistic Regression using the proposed OVA-BT scheme, is shown in Figure 7.

The outcomes shown in Table 4 show the evaluation performance on training and testing sets using the proposed OVA-BT approach of Naïve Bayes, Logistic Regression, Random Forest, KNN, MLP, SVM, and ensemble learning classifiers, as denoted KNN_OVA-BT, LR_OVA-BT, NB_OVA-BT, SVM_OVA-BT, RF_OVA-BT, MLP_OVA-BT, and Ensemble_OVA-BT, respectively.

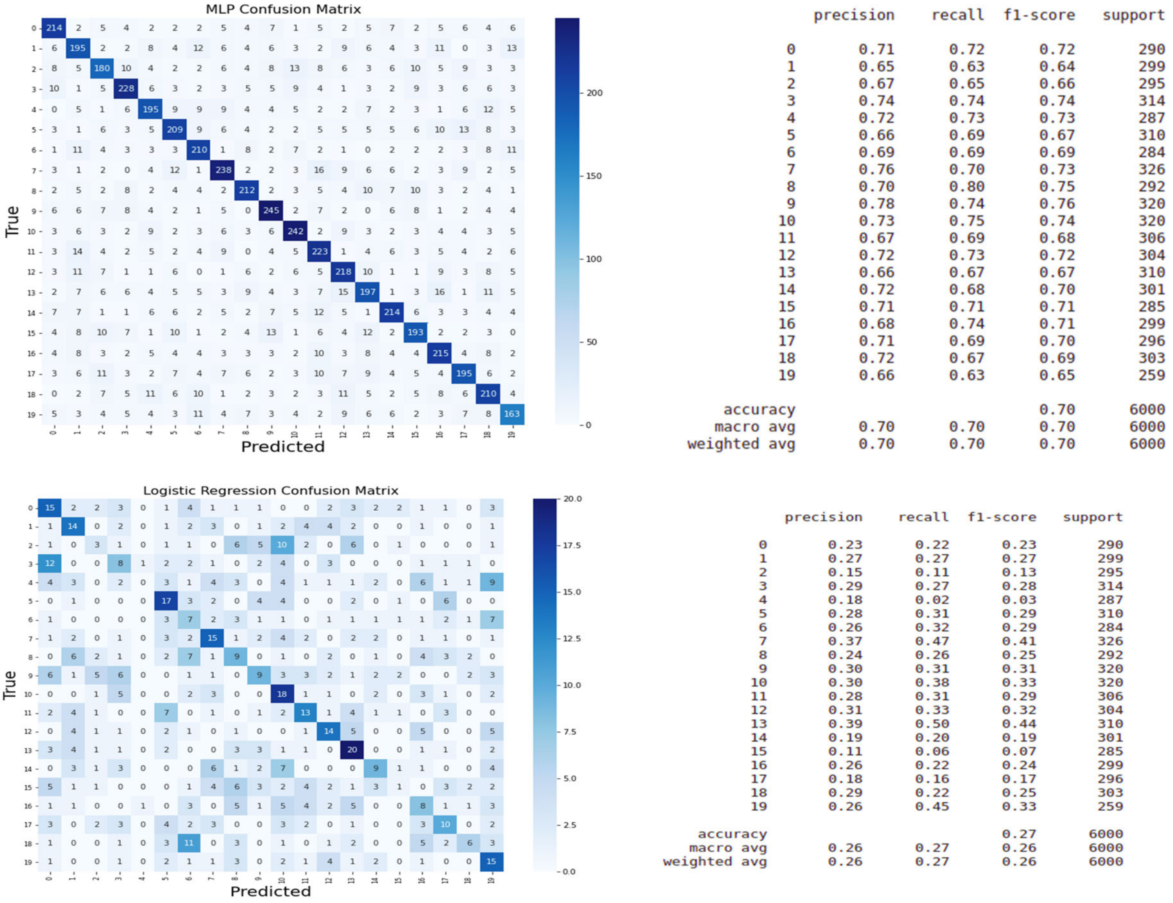
The MLP_OVA-BT gave the best result with 71.37%, 71.31%, and 71.28% in precision, recall, and F1-score, respectively. Random Forest (RF_OVA-BT) comes in second place with 71.2% in precision, 70.1% in recall, and 71.15% in F1-score. The SVM_OVA-BT comes in third place, whereas the NB_OVA-BT and LR_OVA-BT algorithms generate the worst score with 24.71%, 26.58%, and 24.7% in precision, recall, and F1-score, respectively, in LR_OVA-BT.

Table 4 shows that for these 3 metrics on the testing set, ensemble learning has the highest predictive accuracy (recall = 0.783, precision = 0.7568, and F1-score = 0.768). It is important to note that recall, precision, and the F1-score are more helpful if the category distribution is unequal.

The confusion matrix on the testing set of the ensemble learning classifier; shown in Figure 8, shows that misclassification is the primary factor influencing these three measures. This is most likely due to overlapping classification rules and the low number of classes in some categories (see Table 1, b) above for the distribution of classes).

The individual models' and ensemble learning's ROC curves and AUC values are exposed in Figure 9. It is clear that, when compared to other individual models, MLP_OVA-BT performed better, followed by SVM_OVA-BT, then KNN_OVA-BT, while Naive Bayes (NB_OVA-BT) and Logistic Regression (LR_OVA-BT) produced the worst results. In comparison to MLP_OVA-BT, LR_OVA-BT, NB_OVA-BT, KNN_OVA-BT, SVM_OVA-BT, and RF_OVA-BT, the ensemble learning ensemble_OVA-BT achieves the greatest AUC value of 96%, which has increased by 5%, 16%, 15%, 6%, 1%, and 3%, respectively.

Although the MLP_OVA-BT classifier's ROC curve has a lower AUC of 91% as compared to ensemble learning



MLP Confusion Matrix

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
0	214	2	5	4	2	2	2	5	4	7	1	5	2	5	7	2	5	6	4	6
1	6	195	2	2	8	4	12	6	4	6	3	2	9	6	4	3	11	0	3	13
2	8	5	180	10	4	2	2	6	4	8	13	8	6	3	6	10	5	9	3	3
3	10	1	5	228	6	3	2	3	5	5	9	4	1	3	2	9	3	6	6	3
4	0	5	1	6	195	9	9	9	4	4	5	2	2	7	2	3	1	6	12	5
5	3	1	6	3	5	209	9	6	4	2	2	5	5	5	5	6	10	13	8	3
6	1	11	4	3	3	3	210	1	8	2	7	2	1	0	2	2	2	3	8	11
7	3	1	2	0	4	12	1	238	2	2	3	16	9	6	6	2	3	9	2	5
8	2	5	2	8	2	4	4	2	212	2	3	5	4	10	7	10	3	2	4	1
9	6	6	7	8	4	2	1	5	0	245	2	7	2	0	6	8	1	2	4	4
10	3	6	3	2	9	2	3	6	3	6	242	2	9	3	2	3	4	4	3	5
11	3	14	4	2	5	2	4	9	0	4	5	223	1	4	6	3	5	4	2	6
12	3	11	7	1	1	6	0	1	6	2	6	5	218	10	1	1	9	3	8	5
13	2	7	6	6	4	5	5	3	9	4	3	7	15	197	1	3	16	1	11	5
14	7	7	1	1	6	6	2	5	2	7	5	12	5	1	214	6	3	3	4	4
15	4	8	10	7	1	10	1	2	4	13	1	6	4	12	2	193	2	2	3	0
16	4	8	3	2	5	4	4	3	3	3	2	10	3	8	4	4	215	4	8	2
17	3	6	11	3	2	7	4	7	6	2	3	10	7	9	4	5	4	195	6	2
18	0	2	7	5	11	6	10	1	2	3	2	3	11	5	2	5	8	6	210	4
19	5	3	4	5	4	3	11	4	7	3	4	2	9	6	6	2	3	7	8	163

```
              precision    recall  f1-score   support

           0       0.71      0.72      0.72       290
           1       0.65      0.63      0.64       299
           2       0.67      0.65      0.66       295
           3       0.74      0.74      0.74       314
           4       0.72      0.73      0.73       287
           5       0.66      0.69      0.67       310
           6       0.69      0.69      0.69       284
           7       0.76      0.70      0.73       326
           8       0.70      0.80      0.75       292
           9       0.78      0.74      0.76       320
          10       0.73      0.75      0.74       320
          11       0.67      0.69      0.68       306
          12       0.72      0.73      0.72       304
          13       0.66      0.67      0.67       310
          14       0.72      0.68      0.70       301
          15       0.71      0.71      0.71       285
          16       0.68      0.74      0.71       299
          17       0.71      0.69      0.70       296
          18       0.72      0.67      0.69       303
          19       0.66      0.63      0.65       259

    accuracy                           0.70      6000
   macro avg       0.70      0.70      0.70      6000
weighted avg       0.70      0.70      0.70      6000
```

Logistic Regression Confusion Matrix

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
0	15	2	2	3	0	1	4	1	1	1	0	0	2	3	2	2	1	1	0	3
1	1	14	0	2	0	1	2	3	0	1	2	4	4	2	0	0	1	0	0	1
2	1	0	3	1	0	1	1	0	6	5	10	2	0	6	0	1	0	0	0	1
3	12	0	0	8	1	2	2	1	0	2	4	0	3	0	0	0	1	1	1	0
4	4	3	0	2	0	3	1	4	3	0	4	1	1	1	2	0	6	1	1	9
5	0	1	0	0	0	17	3	2	0	4	4	0	0	2	0	1	0	6	0	0
6	1	0	0	0	0	3	7	2	3	1	1	0	1	1	1	1	1	2	1	7
7	1	2	0	1	0	3	2	15	1	2	4	2	0	2	2	0	1	1	0	1
8	0	6	2	1	0	2	7	1	9	0	1	0	2	0	1	0	4	3	2	0
9	6	1	5	6	0	0	1	1	0	9	3	3	2	1	2	2	0	0	2	3
10	0	0	1	5	0	0	2	3	0	0	18	1	1	0	2	0	3	1	0	2
11	2	4	1	0	0	7	0	1	0	1	2	13	1	4	1	1	0	3	0	0
12	0	4	1	1	0	2	1	0	1	0	0	1	14	5	0	0	5	0	0	5
13	3	4	1	1	0	2	0	0	3	3	1	1	0	20	0	0	1	1	0	2
14	0	3	1	3	0	0	0	6	1	2	7	0	0	0	9	1	1	0	0	4
15	5	1	1	0	0	0	1	4	6	3	2	4	2	1	3	1	0	3	2	2
16	1	1	0	0	1	0	3	0	5	1	5	4	2	5	0	0	8	1	1	3
17	3	0	2	3	0	4	2	3	0	0	3	2	0	1	0	0	3	10	0	2
18	1	0	0	1	0	3	11	0	3	0	1	0	1	2	0	0	5	2	6	3
19	1	0	0	1	0	2	1	1	3	0	2	1	4	1	2	0	1	0	0	15

```
              precision    recall  f1-score   support

           0       0.23      0.22      0.23       290
           1       0.27      0.27      0.27       299
           2       0.15      0.11      0.13       295
           3       0.29      0.27      0.28       314
           4       0.18      0.02      0.03       287
           5       0.28      0.31      0.29       310
           6       0.26      0.32      0.29       284
           7       0.37      0.47      0.41       326
           8       0.24      0.26      0.25       292
           9       0.30      0.31      0.31       320
          10       0.30      0.38      0.33       320
          11       0.28      0.31      0.29       306
          12       0.31      0.33      0.32       304
          13       0.39      0.50      0.44       310
          14       0.19      0.20      0.19       301
          15       0.11      0.06      0.07       285
          16       0.26      0.22      0.24       299
          17       0.18      0.16      0.17       296
          18       0.29      0.22      0.25       303
          19       0.26      0.45      0.33       259

    accuracy                           0.27      6000
   macro avg       0.26      0.27      0.26      6000
weighted avg       0.26      0.27      0.26      6000
```

FIGURE 7. (MLP_OVA-BT) and worst (LR_OVA-BT) classifiers.

TABLE 4. Evaluation of various classifiers' performance on training and testing sets using OVA-BT.

		Model						
Set	Metric	KNN_OVA-BT	LR_OVA-BT	NB_OVA-BT	SVM_OVA-BT	RF_OVA-BT	MLP_OVA-BT	Ensemble_OVA-BT
Training	*Precision*	0.66408	0.2643	0.2649	0.7865	0.934	0.98766	0.979
	Recall	0.65628	0.2778	0.27204	0.785	0.981	0.98766	0.9789
	F1-score	0.6558	0.2564	0.2584	0.7848	0.9815	0.9876	0.9789
Testing	*Precision*	0.6264	0.2471	0.2552	0.61727	0.712	0.71375	0.7568
	Recall	0.6235	0.2658	0.2682	0.6183	0.701	0.71316	0.783
	F1-score	0.6225	0.247	0.2556	0.6166	0.7115	*0.71286*	0.768

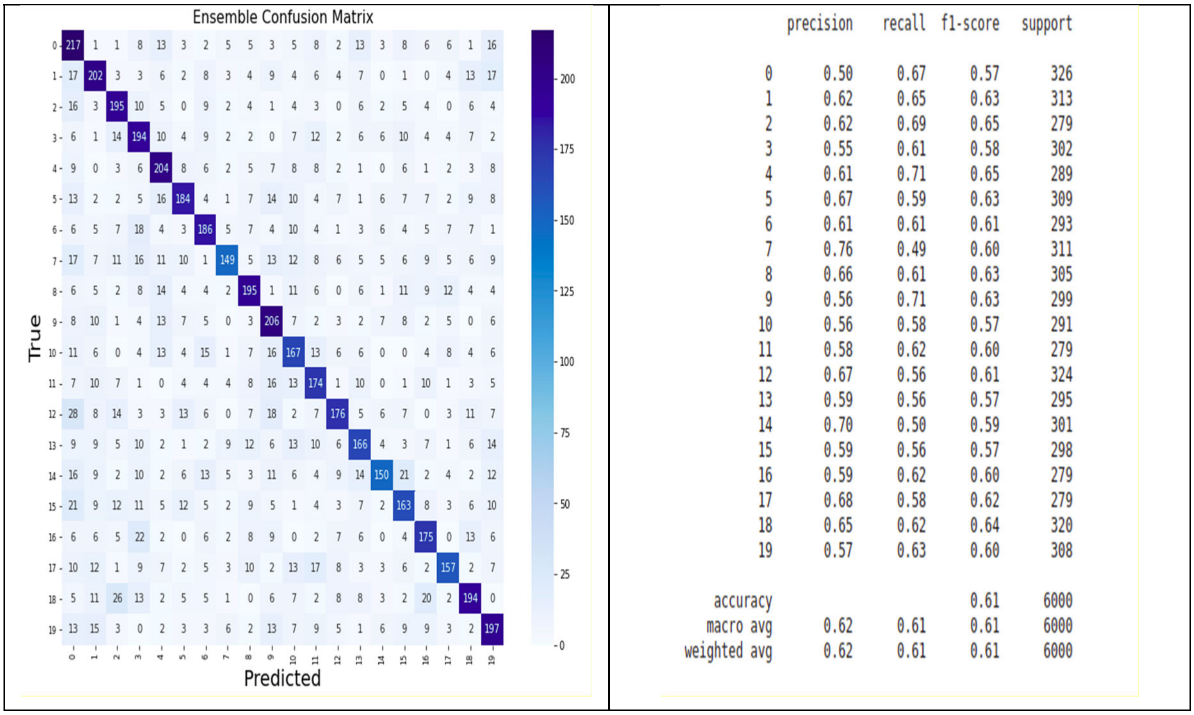
ensemble_OVA_BT with 96%, it performs best locally (on the left top corn). The Ensemble learning would then be preferred in this scenario with regard to the AUC measure (the higher AUC, better the model to differentiate between normal vehicles and misbehaving), but the MLP_OVA-BT model would obviously outperform in certain specific scenarios of misclassification costs.

We evaluated the performance of the classification algorithms Random Forest, KNN, SVM, Naive Bayes,

logistic regression, and Multi-Layer Perceptron (MLP) in our research on misbehavior detection in Vehicular communication using unbalanced databases and the One vs. All Binary Tree (OVA-BT) method for metrics of assessment.

We discovered that because of their bias towards the majority category, Naive Bayes and logistic regression classifiers performed poorly in identifying misbehaving. They had significant precision for cases of normal behavior but poor sensitivity and recall metrics for instances of misbehavior.

On the other hand, MLP performed better in identifying inappropriate behavior. It was able to overcome the difficulties presented by class imbalance due to its capacity to recognize complicated patterns and learn non-linear decision limits. By concentrating on the unique nuances of each class, the OVA-BT method, which assesses each misbehavior class separately, greatly enhanced MLP's performance.



	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
0	217	1	1	8	13	3	2	5	5	3	5	8	2	13	3	8	6	6	1	16
1	17	202	3	3	6	2	8	3	4	9	4	6	4	7	0	1	0	4	13	17
2	16	3	195	10	5	0	9	2	4	1	4	3	0	6	2	5	4	0	6	4
3	6	1	14	194	10	4	9	2	2	0	7	12	2	6	6	10	4	4	7	2
4	9	0	3	6	204	8	6	2	5	7	8	8	2	1	0	6	1	2	3	8
5	13	2	2	5	16	184	4	1	7	14	10	4	7	1	6	7	7	2	9	8
6	6	5	7	18	4	3	186	5	7	4	10	4	1	3	6	4	5	7	7	1
7	17	7	11	16	11	10	1	149	5	13	12	8	6	5	5	6	9	5	6	9
8	6	5	2	8	14	4	4	2	195	1	11	6	0	6	1	11	9	12	4	4
9	8	10	1	4	13	7	5	0	3	206	7	2	3	2	7	8	2	5	0	6
10	11	6	0	4	13	4	15	1	7	16	167	13	6	6	0	0	4	8	4	6
11	7	10	7	1	0	4	4	4	8	16	13	174	1	10	0	1	10	1	3	5
12	28	8	14	3	3	13	6	0	7	18	2	7	176	5	6	7	0	3	11	7
13	9	9	5	10	2	1	2	9	12	6	13	10	6	166	4	3	7	1	6	14
14	16	9	2	10	2	6	13	5	3	11	6	4	9	14	150	21	2	4	2	12
15	21	9	12	11	5	12	5	2	9	5	1	4	3	7	2	163	8	3	6	10
16	6	6	5	22	2	0	6	2	8	9	0	2	7	6	0	4	175	0	13	6
17	10	12	1	9	7	2	5	3	10	2	13	17	8	3	3	6	2	157	2	7
18	5	11	26	13	2	5	5	1	0	6	7	2	8	8	3	2	20	2	194	0
19	13	15	3	0	2	3	3	6	2	13	7	9	5	1	6	9	9	3	2	197

```
              precision    recall  f1-score   support

           0       0.50      0.67      0.57       326
           1       0.62      0.65      0.63       313
           2       0.62      0.69      0.65       279
           3       0.55      0.61      0.58       302
           4       0.61      0.71      0.65       289
           5       0.67      0.59      0.63       309
           6       0.61      0.61      0.61       293
           7       0.76      0.49      0.60       311
           8       0.66      0.61      0.63       305
           9       0.56      0.71      0.63       299
          10       0.56      0.58      0.57       291
          11       0.58      0.62      0.60       279
          12       0.67      0.56      0.61       324
          13       0.59      0.56      0.57       295
          14       0.70      0.50      0.59       301
          15       0.59      0.56      0.57       298
          16       0.59      0.62      0.60       279
          17       0.68      0.58      0.62       279
          18       0.65      0.62      0.64       320
          19       0.57      0.63      0.60       308

    accuracy                           0.61      6000
   macro avg       0.62      0.61      0.61      6000
weighted avg       0.62      0.61      0.61      6000
```

FIGURE 8. Confusion matrix and Classification Report of the classifier ensemble on the testing set.

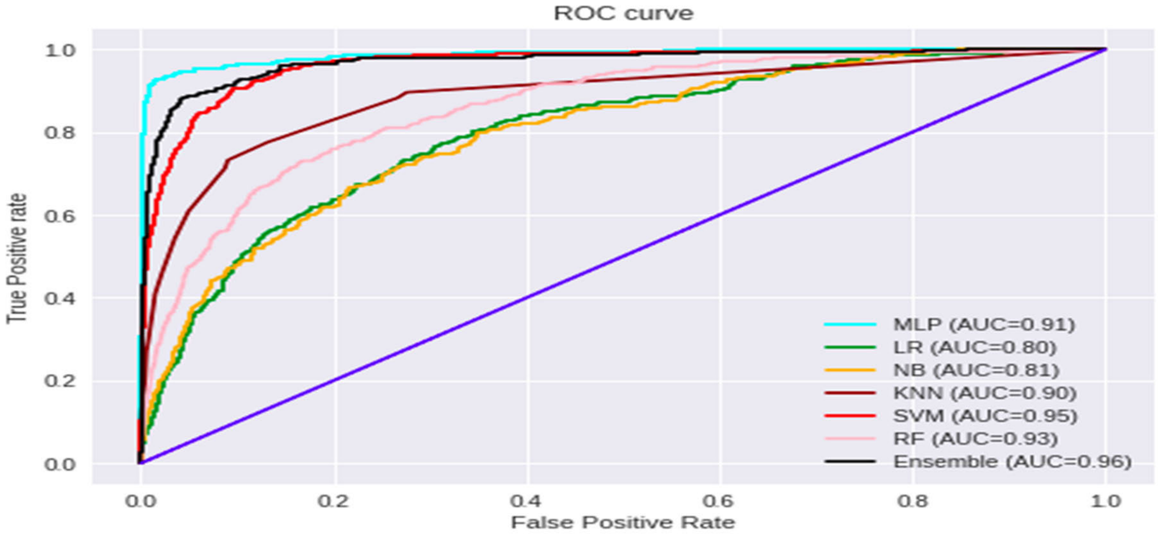


FIGURE 9. ROC curves produced by various individual model and ensemble learning on the testing set.

Our results highlight the significance of picking appropriate algorithms for misbehavior identification in unbalanced databases. Due to their limitations in handling class imbalances and detecting complicated inappropriate behavior patterns, naive Bayes and logistic regression aren't always the best options. A more efficient method for identifying occurrences of misbehavior is MLP using the OVA-BT approach.

V. CONCLUSION AND FUTURE RESEARCH

In this paper, we addressed the issue of misbehavior detection in VANETs employing the One vs. All binary tree (OVA-BT) approach for unbalanced data.

We employed a fivefold CV to tune the hyper-parameter of the individual classifiers in order to improve the classification accuracy of the VeReMi extension dataset. Evaluation metrics like the confusion matrix, precision, recall, F1-Score, ROC curve, and AUC values are used to assess the performance of the ensemble learning and six individual machine learning models: Random Forest, MLP, k-NN, Logistic Regression, SVM, and Naïve Bayes.

Additionally, we examined the impact of dividing the multi-class issue into binary models and the voting process inside various ML algorithms. Data is easier to understand when it is divided into multiple binary issues. When compared to a classic single, multi-class classification, the testing results demonstrate that our newly developed method using the OVA binary tree outperforms.

As future work, we propose to offer a trade-off between trying to balance the database and the objective that all the categories must be presented in an equitable split in order to handle the difficulties involved with an unbalanced database when utilizing the OVA-BT strategy. Moreover, we shall enhance the One Versus One classifiers to identify which categories are challenging to distinguish as well as which ones differentiate efficiently from each other. We will also try to test using SMOTE and DSSTE techniques to balance the data.

REFERENCES

[1] J. Santa, F. Pereñíguez, A. Moragón, and A. F. Skarmeta, "Experimental evaluation of CAM and DENM messaging services in vehicular communications," *Transp. Res. C, Emerg. Technol.*, vol. 46, pp. 98–120, Sep. 2014.

[2] S. Al-Sultan, M. M. Al-Doori, A. H. Al-Bayatti, and H. Zedan, "A comprehensive survey on vehicular ad hoc network," *J. Netw. Comput. Appl.*, vol. 37, pp. 380–392, Jan. 2014.

[3] S. Sepasgozar and S. Pierre, "An intelligent network traffic prediction model considering road traffic parameters using artificial intelligence methods in VANET," *IEEE Access*, vol. 10, pp. 8227–8242, 2022, doi: 10.1109/ACCESS.2022.3144112.

[4] B. Yelure and S. Sonavane, "SARP: Secure routing protocol using anonymous authentication in vehicular ad-hoc networks," *J. Ambient Intell. Humanized Comput.*, vol. 14, no. 4, pp. 3567–3581, Apr. 2023, doi: 10.1007/s12652-021-03486-1.

[5] S. Zeadally, R. Hunt, Y.-S. Chen, A. Irwin, and A. Hassan, "Vehicular ad hoc networks (VANETS): Status, results, and challenges," *Telecommun. Syst.*, vol. 50, no. 4, pp. 217–241, Aug. 2012, doi: 10.1007/s11235-010-9400-5.

[6] M. Zongo, J. Nlong II, R. Ndoundam, and A. Foerster, "DSRC applicability in Cameroon road traffic: A simulation study," *EAI Endorsed Trans. Mobile Commun. Appl.*, vol. 6, no. 19, p. 19, 2021, doi: 10.4108/eai.11-6-2021.170233.

[7] O. Slama, B. Alaya, S. Zidi, and M. Tarhouni, "Comparative study of misbehavior detection system for classifying misbehaviors on VANET," in *Proc. 8th Int. Conf. Control, Decis. Inf. Technol. (CoDIT)*, vol. 1, May 2022, pp. 243–248.

[8] A. C. Lorena, A. C. P. L. F. de Carvalho, and J. M. P. Gama, "A review on the combination of binary classifiers in multiclass problems," *Artif. Intell. Rev.*, vol. 30, nos. 1–4, pp. 19–37, Dec. 2008.

[9] M. Galar, A. Fernández, E. Barrenechea, H. Bustince, and F. Herrera, "An overview of ensemble methods for binary classifiers in multi-class problems: Experimental study on one-vs-one and one-vs-all schemes," *Pattern Recognit.*, vol. 44, no. 8, pp. 1761–1776, Aug. 2011.

[10] R. Rifkin and A. Klautau, "In defense of one-vs-all classification," *J. Mach. Learn. Res.*, vol. 5, pp. 101–141, Jan. 2004.

[11] N. Silva, J. Soares, V. Shah, M. Y. Santos, and H. Rodrigues, "Anomaly detection in roads with a data mining approach," *Proc. Comput. Sci.*, vol. 121, pp. 415–422, 2017.

[12] S. Ding, X. Zhang, Y. An, and Y. Xue, "Weighted linear loss multiple birth support vector machine based on information granulation for multi-class classification," *Pattern Recognit.*, vol. 67, pp. 32–46, Jul. 2017.

[13] A. Rosales-Perez, S. Garcia, H. Terashima-Marin, C. A. C. Coello, and F. Herrera, "MC2ESVM: Multiclass classification based on cooperative evolution of support vector machines," *IEEE Comput. Intell. Mag.*, vol. 13, no. 2, pp. 18–29, May 2018.

[14] S. S. Ng, P. W. Tse, and K. L. Tsui, "A one-versus-all class binarization strategy for bearing diagnostics of concurrent defects," *Sensors*, vol. 14, no. 1, pp. 1295–1321, 2014.

[15] N. A. A. Khairudin, C. C. Lim, A. S. A. Nasir, and Z. Mohamed, "Efficient classification techniques in classifying human intestinal parasite ova," *J. Telecommun., Electron. Comput. Eng. (JTEC)*, vol. 14, no. 3, pp. 17–23, 2022.

[16] M. N. Adnan and M. Z. Islem, "One-vs-all binarization technique in the context of random forest," in *Proc. ESANN*, 2015, pp. 385–390.

[17] E. L. Allwein, R. E. Schapire, and Y. Singer, "Reducing multiclass to binary: A unifying approach for margin classifiers," *J. Mach. Learn. Res.*, vol. 1, pp. 113–141, Dec. 2000.

[18] L. Chen and D. Alahakoon, "NeuroEvolution of augmenting topologies with learning for data classification," in *Proc. Int. Conf. Inf. Autom.*, Dec. 2006, pp. 367–371.

[19] T. McDonnell, S. Andoni, E. Bonab, S. Cheng, J. H. Goode, and J. Schrum, "Divide and conquer: Neuroevolution for multiclass classification," in *Proc. Genetic Evol. Comput. Conf.*, Jul. 2018, pp. 474–481.

[20] P. Bhardwaj, J. Kelleher, D. Costabello, and O. Sullivan, "Poisoning knowledge graph embeddings via relation inference patterns (2021a)," in *Proc. Annu. Meeting Assoc. Comput. Linguistics Int. Joint Conf. Natural Lang. Process.*, vol. 1, 2021, pp. 1875–1888.

[21] N. Alkadiri and M. Ilyas, "Machine learning-based architecture for DDoS detection in VANETs system," in *Proc. Int. Conf. Artif. Intell. Things (ICAIoT)*, Dec. 2022, pp. 1–7.

[22] J. Wang, H. Xu, and J. Zhang, "Classification of imbalanced data by using the SMOTE algorithm and locally linear embedding," Presented at the 8th Int. Conf. Signal Process., 2006, pp. 16–20, doi: 10.1109/ICOSP.2006.345752.

[23] S. Bagui and K. Li, "Resampling imbalanced data for network intrusion detection datasets," *J. Big Data*, pp. 1–41, Dec. 2021, doi: 10.1186/s40537-020-00390-x.

[24] 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. 142206–142217, 2021, doi: 10.1109/ACCESS.2021.3120626.

[25] N. Moustafa and J. Slay, "UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," in *Proc. Mil. Commun. Inf. Syst. Conf. (MilCIS)*, Jul. 2015, pp. 1–6, doi: 10.1109/MilCIS.2015.7348942.

[26] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," *Proc. IEEE Symp. Comput. Intell. Secur. Defense Appl.*, Dec. 2009, pp. 1–6, doi: 10.1109/CISDA.2009.5356528.

[27] F. Mohsen, A. Shtayyeh, M. Struijk, R. Naser, and L. Mohammad, "An approach to guide users towards less revealing internet browsers," in *Emerging Trends in Cybersecurity Application*. Cham, Switzerland: Springer, Jul. 2022, pp. 69–94.

[28] S. C. Sangapu, K. S. N. Prasard, and R. J. Kannan, "Impact of class imbalance in VeReMi dataset for misbehavior detection in autonomous vehicles," *Soft Comput.*, vol. 27, pp. 1–11, 2023.

[29] O. Slama, B. Alaya, and S. Zidi, "Towards misbehavior intelligent detection using guided machine learning in vehicular ad-hoc networks (VANET)," *Inteligencia Artif.*, vol. 25, no. 70, pp. 138–154, Dec. 2022, doi: 10.4114/intartif.vol25iss70pp138-154.

[30] K. Wehrle, M. Günes, and J. Gross, "OMNeT++: Modeling and tools for network simulation," in *Modeling and Tools for Network Simulation*. London, U.K.: Springer, 2010, pp. 35–59.

[31] C. Sommer, D. Eckhoff, A. Brummer, D. S. Buse, and F. Hagenauer, *Veins: The Open Source Vehicular Network Simulation Framework*. Cham, Switzerland: Springer, 2019, pp. 215–252.

[32] L. Codeca, R. Frank, S. Faye, and T. Engel, "Luxembourg SUMO traffic (LuST) scenario: Traffic demand evaluation," *IEEE Intell. Transp. Syst. Mag.*, vol. 9, no. 2, pp. 52–63, May 2017.

[33] O. Slama, M. Tarhouni, S. Zidi, and B. Alaya. (2023). *VeReMi_Extension: Dataset for Misbehaviors in VANETs*. [Online]. Available: http://dx.doi.org/10.17632/k62n4z9gdz.1

[34] N. A. Farhana, F. M. Afendi, A. Fitrianto, and S. H. Wijaya, "Classification modeling of support vector machine (SVM) and random forest in predicting pharmacodynamics interactions," *J. Phys., Conf. Ser.*, vol. 1863, no. 1, Mar. 2021, Art. no. 012067.

[35] W. Dhifallah, M. Tarhouni, T. Moulahi, and S. Zidi, "A novel realistic dataset for intrusion detection in IoT based on machine learning," in *Proc. Int. Symp. Netw., Comput. Commun. (ISNCC)*, Oct. 2021, pp. 1–6.

[36] S. Salzberg, *C4. 5: Programs for Machine Learning*, J. R. Quinlan, Ed. Burlington, MA, USA: Morgan Kaufmann Publishers, pp. 235-240, 1994.

[37] V. N. Vapnik, "An overview of statistical learning theory," *IEEE Trans. Neural Netw.*, vol. 10, no. 5, pp. 988–999, Sep. 1999.

[38] G. Guo, H. Wang, D. Bell, Y. Bishop-Hurley, and K. Greer, "KNN model-based approach in classification," in *On The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE*, Catania, Sicily, Italy, Berlin, Germany: Springer, Nov. 2003, pp. 986–996.

[39] M.-L. Zhang, J. M. Peña, and V. Robles, ‘‘Feature selection for multi-label Naive Bayes classification,’’ *Inf. Sci.*, vol. 179, no. 19, pp. 3218–3229, Sep. 2009, doi: 10.1016/j.ins.2009.06.010.

[40] T. G. Dietterich, ‘‘Ensemble learning,’’ *Handbook Brain Theory Neural Netw.*, vol. 2, no. 1, pp. 110–125, Mar. 2002.

[41] L. Kuncheva, *Combining Pattern Classifiers: Methods and Algorithms*. Hoboken, NJ, USA: Wiley, 2004.

[42] J. Habbema and J. Hermans, ‘‘Selection of variables in discriminant analysis by F-statistic and error rate,’’ *Technometrics*, vol. 19, no. 4, pp. 487–493, 1977, doi: https://doi.org/10.2307/1267890.

[43] A. A. Motsinger and M. D. Ritchie, ‘‘The effect of reduction in cross-validation intervals on the performance of multifactor dimensionality reduction,’’ *Genetic Epidemiol.*, vol. 30, no. 6, pp. 546–555, Sep. 2006.

[44] B. Fei and J. Liu, ‘‘Binary tree of SVM: A new fast multiclass training and classification algorithm,’’ *IEEE Trans. Neural Netw.*, vol. 17, no. 3, pp. 696–704, May 2006.

OMESSAAD SLAMA received the M.S. degree from the Faculty of Economics and Management of Sfax, University of Sfax, Tunisia, in 2010. She is currently pursuing the Ph.D. degree in electric engineering with the National Engineering School of Gabes, University of Gabes, Tunisia. Her current research interests include intelligent transportation systems, artificial intelligence, machine learning, deep learning, blockchain, and federated learning. She was a Reviewer at the International Conference of ODIT in 2022.

MOUNIRA TARHOUNI received the Engineering degree from the National Engineering School of Tunis, Tunisia, in 2007, and the Ph.D. degree in electrical engineering from the Faculty of Sciences of Tunis, Tunisia. She joined the Higher Institute of Informatics and Multimedia, Gabes, Tunisia, in 2014. Her current research interests include artificial intelligence, classification, deep learning, and blockchain.

SALAH ZIDI received the master’s degree in new technologies of dedicated IT systems, in 2004, and the Ph.D. and H.D.R. degrees in computer science from the University of Lille 1, France, in 2007 and 2017, respectively.

He joined the Higher Institute of Industrial Systems of Gabes, Tunisia. His research interests include the fields of artificial intelligence, deep learning, machine learning, categorization, feature extraction, optical character recognition, optimization, blockchain, and federated learning.

BECHIR ALAYA received the master’s and Ph.D. degrees in computer science from the University of Le Havre, France, in 2007 and 2012, respectively. He joined the College of Business and Economics, Qassim University, in 2014. His research interests include decentralized multimedia frameworks, real-time datasets, real-time nested transactions, streaming video, intelligent transportation, and quality of service management in multimedia networks.