Hindawi
Security and Communication Networks
Volume 2022, Article ID 9440886, 18 pages
https://doi.org/10.1155/2022/9440886

Research Article

Message Authentication and Network Anomalies Detection in Vehicular Ad Hoc Networks

Leonid Legashev,[1] **Irina Bolodurina**,[1] **Lubov Zabrodina**,[1] **Yuri Ushakov**,[1] **Alexander Shukhman**,[1] **Denis Parfenov**,[1] **Yong Zhou**,[2] **and Yan Xu**[2]

[1]*Faculty of Mathematics and Information Technologies, Orenburg State University, Orenburg 460018, Russia*
[2]*School of Computer Science and Technology, Anhui University, Hefei 230601, China*

Correspondence should be addressed to Leonid Legashev; silentgir@gmail.com

Received 3 June 2021; Revised 13 January 2022; Accepted 15 January 2022; Published 24 February 2022

Academic Editor: Mamoun Alazab

Copyright © 2022 Leonid Legashev et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Intelligent transport systems are the future in matters of safe roads and comfortable driving. Integration of vehicles into a unified intelligent network leads to all kinds of security issues and cyber threats common to conventional networks. Rapid development of mobile ad hoc networks and machine learning methods allows us to ensure security of intelligent transport systems. In this paper, we design an authentication scheme that can be used to ensure message integrity and preserve conditional privacy for the vehicle user. The proposed authentication scheme is designed with lightweight cryptography methods, so that it only brings little computational and communication overhead. We also conduct experiments on vehicular ad hoc network segment traffic generation in OMNeT++ tool and apply up-to-date machine learning methods to detect malicious behavior in a given simulated environment. The results of the study show high accuracy in distributed denial-of-service attack detection.

1. Introduction

The rapid development in the field of mobile devices, sensors, and 5G networks [1] allows incorporating computational nodes into wireless ad hoc network. A network without preexisting infrastructure is called a mobile ad hoc network (MANET); it consists of mobile devices capable of establishing connections between arbitrary nodes. Ad Hoc On-Demand Distance Vector (AODV), Destination-Sequenced Distance-Vector Routing (DSDV), Optimized Link-State Routing (OLSR), and Dynamic Source Routing (DSR) protocols are used for routing at the network layer in MANET. One of the important areas in MANET is vehicular ad hoc networks (VANETs), which represent intelligent transport system where each vehicle is considered as a mobile node. Potential VANET applications include road condition warnings, collision alerts, accident alerts, road congestions, driver assistance systems, and infotainment systems. Each vehicle in VANET is equipped with a set of sensors and constantly exchanges crucial information with other nodes all over the network. These nodes may include fixed roadside units (RSUs), base station units (BSUs), trusted authority (TA) or control center (CC), and drones as mobile BSUs [2, 3]. It is very important to pay attention to security issues in VANET because the consequences of a network attack on the road can be unfortunate.

Complex research on VANET security issues may be divided into two directions. The first one is related to assistance of vehicle communication and vehicle privacy based on intelligent anonymous authentication and key agreement for 5G/beyond 5G (B5G) vehicular ad hoc networks. The second direction is related to machine learning algorithm applications in network threat detection and classification.

Modern trends in the creation of network connectivity of an increasing number of devices and the rise of the Internet of Things (IoT) and the Internet of Everything (IoE) required the development of new approaches to organizing network interaction. In cases where the network can contain several thousand devices, many of which are also intermediate for traffic transmission, traditional approaches can

be inefficient and slow. In the case of unstable links and nonstationary nodes, traditional mobile networking methods such as BATMAN-adv, OLSR, and AODV can cause losses, delays, rings, and instability of the entire network. Traditional software-defined networks (SDNs) can also be unstable under these conditions, especially in reactive mode. To implement intelligent transmission of information over such networks, for example, in the form of delayed packet transmission, a combination of controlled and autonomous approaches is required. Vehicular distributed software-defined networks (VDSDNs) are a combination of proactive SDN management for consistent precalculated routes, while local reactive mode is used in conjunction with neighbor detection methods through legacy protocols. General scheme of VDSDNs in VANET segment is presented in Figure 1.

SDNs are mainly used in VDSDNs in stable parts of the network and in virtualization infrastructure, especially for Network Function Virtualization (NFV) modules and edge computing. When an NFV module is used as an edge virtual machine or a container running on network equipment, the requirements for the selection and routing of network flows passing through this module can be implemented only by SDN infrastructures managed by the controller. Since traffic can pass through the balancing nodes and be routed to the endpoint through various communication channels, it is important to have complete information about all network flows to a specific destination. When using distributed networks, they can contain several controllers with state synchronization (for example, via KV-storages); in this case, separate synchronization of applications related to packet and flow analysis is required to intercept the maximum possible number of directions of flow vectors. At the same time, since the traffic volumes of modern applications can exceed the capabilities of their analysis in real time, selective or only header preliminary analysis of packets and consolidation of data from all distributed controllers into a single storage is required, which will be used by many streaming analyzers.

2. Related Work

In order to ensure secure communication between intelligently connected vehicles, a public key cryptography (PKC) mechanism was proposed. In [4], the traditional PKC was proposed to implement self-certified public key cryptography (SCPKC) for online registration of multiserver architecture and to ensure the security of various mobile service applications.

The traditional PKC mechanism can realize secure communication; however, the mechanism suffers from various drawbacks caused by managing a large number of user certificates. In [5], Shamir put forward the concept of ID-based PKC. In [6, 7], a bilinear pairing ID-based PKC mechanism was proposed to achieve the required privacy for vehicles. In [8], an improved scheme ID-based PKC mechanism without bilinear pairing was proposed. The improved scheme is not required to utilize bilinear pairing operations without lack of security and privacy protection, and the total computational cost of signature and authentication is constant for single message and n messages. In [9], an improved message authentication scheme together with a system secret key updating scheme was proposed to optimize the performance and security of V2V authentication process. Although the ID-based PKC avoids the problem of managing certificates brought by the traditional PKC mechanism, it still brings the problem of key escrow.

In 2013, Al-Riyami [10] first proposed the concept of certificateless PKC mechanism, which avoids the certificate management problems brought by traditional PKC mechanism and the key escrow problems caused by ID-based PKC mechanism. In [11], an anonymous authentication scheme based on certificateless PKC mechanism was proposed by using bilinear pairing operations. In [12, 13], all implement batch authentication without bilinear pairing based on certificateless PKC mechanism were proposed. In [14], a new authentication scheme without bilinear pairings was proposed. In [15], a reliable and efficient secure content sharing scheme for 5G-enabled VANETs was proposed. In [16], a lightweight and secure authenticated key agreement scheme for securing V2V and V2I communications simultaneously was proposed.

In order to ensure the efficiency and safety of VANETs, there are a large number of schemes for batch authentication of messages. In [17], Zhang et al. proposed a distributed aggregate batch authentication scheme. By dividing the received message into multiple subsets and then aggregating multiple subsets for batch authentication. In [18], an efficient batch authentication scheme based on elliptic curve cryptography was proposed. A proxy-based batch authentication scheme was proposed in [19], where some vehicles were selected as the proxy vehicles, whose message signatures were then verified by roadside units in batches.

Recently, there is a lot of research dedicated to machine learning methods' application in network threat detection. Montenegro J. et al. [20] applied machine learning techniques and trust model metrics to detect fake position attacks in VANETs. The same problem was solved by Singh P. K. et al. [21] using machine learning techniques on VeReMi dataset to detect false position information broadcast to the other vehicles. A Ghaleb F. et al. [22] used the random forest algorithm to train intrusion detection system classifiers on each vehicle node with the overall goal of reducing the communication overhead. Nandy T. et al. [23] also proposed a trust-based collaborative intrusion detection system with k-nearest neighbors nonlinear classifier to identify intruders in real time. To detect various malevolent attacks, Sharma S. et al. [24] proposed a Multicluster Head anomaly based intrusion detection system with Dolphin Swarm Algorithm optimization technique. Zhang T. et al. [25] in their research proposed a privacy-preserving machine learning based collaborative intrusion detection system for VANETs. Zhang D. et al. [26] proposed a software-defined trust-based deep reinforcement learning framework for VANET issues related to performance degradation. Belenko V. et al. [27] proposed approach to generate VANET dataset with various scenarios of cyber attacks for the ns-3 network simulator. Singh P. K. et al. [28]

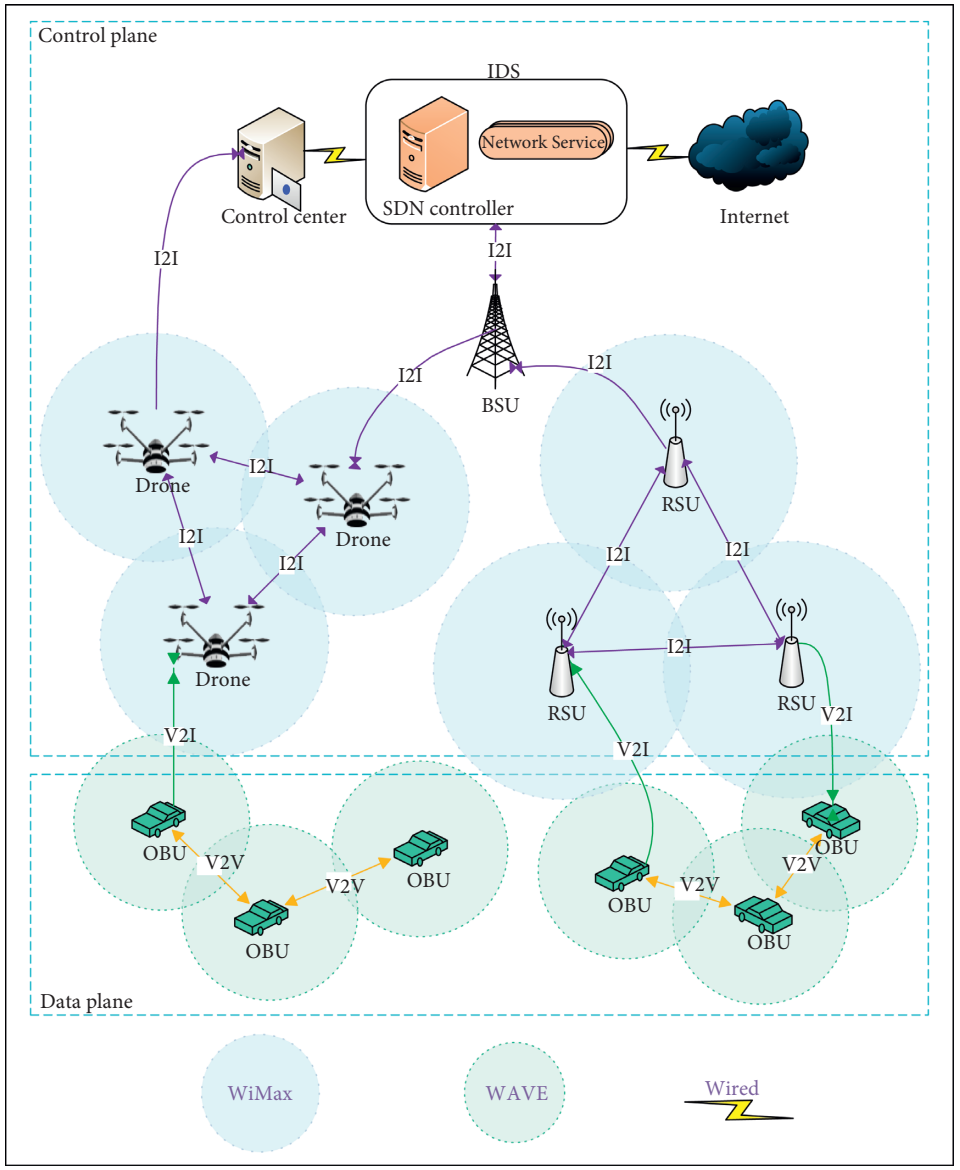

Figure 1: Scheme of VDSDNs in VANET segment.

also generated scenario of multihop communication on ns-3 network simulator to detect wormhole attacks in VANET using KNN and support vector machine models. Kumar S. et al. [29] presented a vehicular jamming system model with usage of CatBoost machine learning algorithm to predict the coordinates of jamming vehicle. Rehman A. et al. [30] described a novel approach to detect intrusion attacks on the CAN bus using convolutional neural network and attention-based gated recurrent unit. Jhaveri R. et al. [31] proposed a bandwidth contract-based framework to provide resilience to violation of the bandwidth requirements of the traffic flows in vehicular ad hoc networks.

Different types of simulation tools can be used to generate reliable VANET traffic and experiment over many types of scenarios within intelligent transport system. Akhtar et al. [32] presented simulation model of microscopic mobility VANET segment by using SUMO [33] traffic simulation package and Freeway Performance Measurement System database. Michaeler et al. [34] presented 3-dimensional driving simulator based on Open-StreetMap data, which integrates VANET communication capabilities. Buse et al. [35] proposed event-driven simulator with a city-scale VANET simulation for the advanced driver assistance system development. To ensure reliable driver assistance systems, Obermaier et al. [36] presented an approach for testing VANET devices and the applications in hardware in the Loop environment using OMNeT++ simulation tool [37] and the VANET model Artery. Fahad et al. [38] proposed a new scheme based on compressed fuzzy logic method to enhance AODV routing decisions in VANET. Maratha et al. [39] conducted performance study of AODV, DSDV, and DSR MANET protocols using NCTUns 6.0 network simulator [40]. Raj et al. [41] simulated various routing protocols using ns-3 simulator [42] and SUMO package and studied performance metrics such as Packet Delivery Ratio, Throughput, and End-to-End Delay.

Table 1 contains the coverage of machine learning methods' applications of network attack detection in some recent publications, including Naive Bayes (NB), logistic regression (LR), support vector machine (SVM), random forest (RF), CatBoost (CB), AdaBoost (AB), and gradient boosting (GB). As you can see, existing works are limited to the usage of individual classifiers only. In our current research, we will make a comparison of the most up-to-date classifiers and ensemble methods for the multiclass classification problem of distributed denial-of-service attack detection using simulated VANET environment.

To study security issues in vehicular ad hoc networks and detect DDoS attacks, we make the following contributions:

Authentication method: we proposed an anonymous authentication scheme based on elliptic curve encryption to meet the security requirements of vehicular ad hoc networks providing the least time cost on signing and verifying a message.

Simulation of VANET dataset: using OMNeT++ simulation tool we simulated segment of VANET and implemented three types of popular network attacks which degrade overall performance of intelligent transport system by flooding vehicles with great amount of generated messages.

Experimentation and evaluation: we recorded network flows information from the nodes in simulated segment of VANET and conducted experiments on multiclass classification using the most advanced classifiers and ensembles of classifiers.

The structure of the rest of the paper is as follows: Section 3 introduces the intelligent anonymous authentication scheme. In Section 4, we describe the proposed generation of VANET segment traffic in OMNeT++ simulation tool and present results of multiclass classification of DDoS attacks. Section 5 gives the conclusion.

3. Intelligent Authentication Methods for Vehicular Ad Hoc Networks

3.1. System Model. The considered system consists of three parts as shown in Figure 2, which can be divided into three layers. The first layer contains TA which communicates with EA and vehicles over a secure channel by wired connections. The second layer includes EA, which communicates with TA and vehicles by the secure channel. The third layer involves multiple vehicles which mainly communicate with cluster head (CH) by DSRC protocol. The definitions of the roles involved in the considered model are as follows.

TA: It is a trusted authority that consists of a Key Generation Center (KGC) and a Trace Authority (TRA) in the practical environment. The KGC initializes the system and generates all public parameters and private keys. The TRA extracts the real identities of malicious vehicles by pseudo identities in the case of controversial traffic events.

EA: It is an edge authority that contains multiple mobile devices (e.g., mobile phones, laptops, and other electronic devices). The EA contains enough computing and storage sources which are used to handle reputation update and reveal vehicles real identities by pseudo identities in the disputed traffic environment.

Vehicles: We assume that multiple vehicles can form a cluster in the same area. In order to avoid repeated calculations of multiple vehicles in the cluster, the EA selects the vehicle with good network resources (e.g., high network bandwidth, constant speed, and suitable location) as the CH vehicle. The CH predownloads the road condition related data required by the vehicle; then, the remaining vehicles and the CH authentication reduce the redundancy calculation.

3.2. Our Proposed Authentication Scheme. In order to meet the security requirements of VANETs, we design an anonymous authentication scheme using an online certificateless signature technology based on prioritization. Our proposed scheme consists of several algorithms including setup, pseudo identity generation/partial key extraction, sign, batch verification, and revocation/update of revocation list. The scheme details can be shown as follows.

3.2.1. Setup. TA initializes the system, generates public parameters for the system, and then sends related parameters to EA by security channel. The details are as follows.

(1) The TA chooses a cyclic addition group G_1 with order q generated by P. Let E: $y^2 = x^3 + ax + b(\text{modn}), a, b \in F_n$, be an elliptic curve over the finite field F_n, where n indicates a large prime number. All the points of E and an infinity point O are in the group G_1.

(2) The KGC selects two random numbers as x, y and calculates $PK_{TA} = yP$. The KGC generates public-private key pairs for EA: $sk_{EA} = s, pk_{EA} = sP$.

(3) The TA selects two one-way hash functions: $h_1\{0,1\}^* \longrightarrow \{0,1\}^*, h_2$: $\{0,1\}^* \longrightarrow Z_q$.

(4) The TA publishes $\{P, PK_{TA}, pk_{EA}, h_1, h_2\}$ as the system parameter and sends (x, sk_{EA}) to EA through the secure channel.

3.2.2. Pseudo Identity Generation and Partial Key Extraction. The EA communicates with vehicles online to generate pseudo identities for vehicles, and the TA generates a partial key for the vehicle through its KGC. The details are as follows:

(1) The vehicle calculates partial public-private key pairs for itself: $sk_{i,1} = s_1$, $pk_{i,1} = s_1 P$.

(2) Vehicle i encrypts $(RID_i, pk_{i,1})$ with its own private key and EA's public key and sends $\{Enc_{sk_{i,1}, pk_{EA}}(RID_i, pk_{i,1}), TS_i\}$ to the EA, where Enc(.) is the asymmetric encryption method, RID_i is the real identity of vehicle i, and TS_i represents the current timestamp.

Table 1: Machine learning classifiers' application in network attack detection.

	So et al. [43]	Grover et al. [44]	Zeng et al. [45]	Singh et al. [21]	Montenegro et al. [20]	Sharshembiev et al. [46]	Tama et al. [47]	The proposed scheme
IBK		+						
J-48		+					+	
NB		+						
LR				+		+		
SVM	+		+	+				
KNN	+				+	+	+	+
RF		+					+	+
CB								+
XGB							+	+
LGBM								+
AB		+						+
GB							+	+

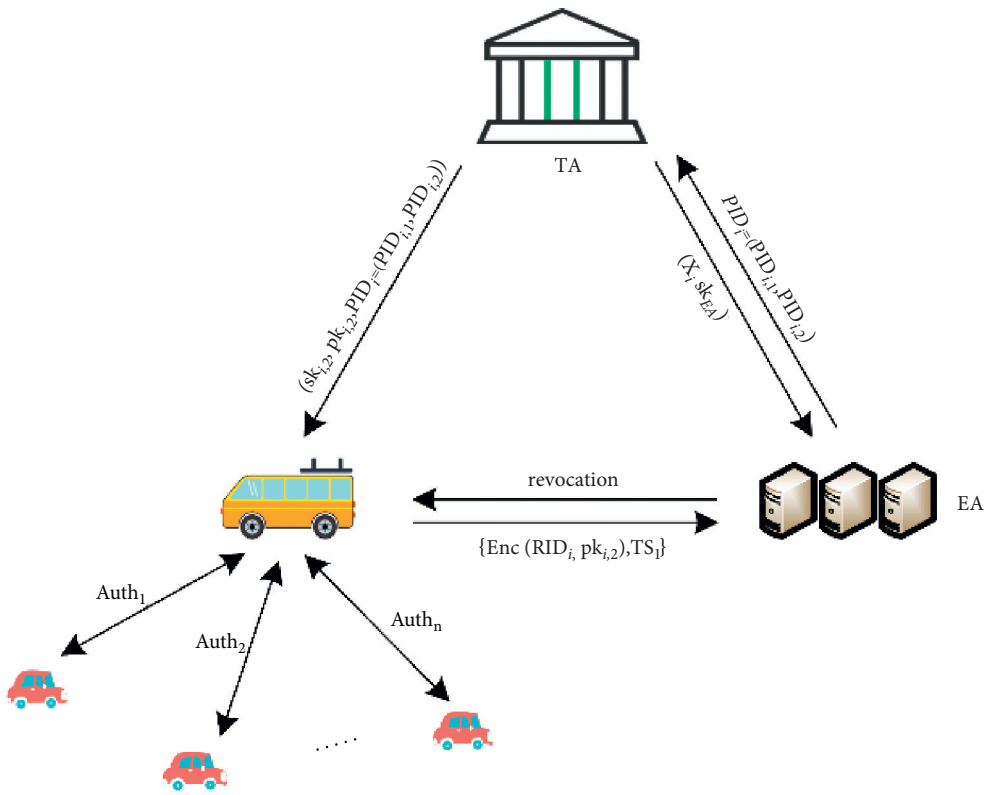

Figure 2: The system model.

(3) The EA checks the timestamp for freshness by determining whether $|TS - TS| \le \Delta T_i$ holds or not. If not, the EA stops. Otherwise, it utilizes its own private key and vehicle i's public key to decrypt $Dec_{sk_{i,1},pk_{EA}}(RID_i, pk_{i,1})$, calculates the pseudo identity $PID_{i,1} = xP$, $PID_{i,2} = RID_i \oplus h_1(xpk_{i,1})$, and then sends $PID_i = (PID_{i,1}, PID_{i,2})$ to the TA.

(4) The TA selects a random number s_2 and calculates partial public key $pk_{i,2} = s_2 P$ and partial private key $sk_{i,2} = s_2 + h_2(PID_i, pk_{i,2})y$, and then the TA sends partial public-private key pairs and pseudo identities $\{pk_i = (pk_{i,1}, pk_{i,2}), sk_i = (sk_{i,1}, sk_{i,2}), PID_i\}$ to the vehicle.

3.2.3. Sign. Each vehicle calculates the signature using the following steps:

(1) Vehicle i calculates the signature:

$$\begin{aligned} S_i &= sk_{i,2} h_1\left(PID_i \| m_i \| TS_i\right) + sk_{i,1}, \\ RV_i &= \left(S_i ID_i \| m_i \| TS_i\right) \oplus h_1\left(RU_j\right), \end{aligned} \tag{1}$$

where RU_j is used to check whether vehicle i is in the revocation list.

(2) Vehicle i sends $\{S_i, PID_i, m_i, TS_i, RV_i, h_1(RU_i)\}$ to the CH.

3.2.4. Batch Verification. The CH batch verifies the vehicles in the cluster using three steps:

(1) It checks whether vehicle i is in the revocation list:

$$RV_i \oplus h_1\left(RU_j\right) == m_i \| S_i \| PID_i TS_i. \quad (2)$$

If the equation is true, which indicates that vehicle i is not in the revocation list, CH continues to execute the next step; otherwise, it stops making the remaining steps.

(2) It checks if the signature is valid or not by checking the equation $|TS - TS_i| \leq \Delta T$. If so, the timestamp is fresh, and the CH continues to execute the next step; otherwise, it stops making the remaining steps.

(3) Batch verification: the batch verification method is as follows:

(1) The CH calculates $h_1(PID_i \| m_i \| TS_i)$ for the vehicle i.

(2) The CH verification for the vehicle i is as follows:

$$S_i P = \left(pk_{i,2} + h_1\left(PID_i || m_i || TS_i\right) PK_{TA}\right) \cdot h_1\left(PID_i || m_i || TS_i\right) + pk_{i,1}. \quad (3)$$

(3) The CH performs the following check for every vehicle i, $i \in [1, n]$:

$$\sum_{i=1}^{n} S_i P == \sum_{i=1}^{n}\left(pk_{i,2} + h_1\left(PID_i || m_i || TS_i\right) PK_{TA}\right) \cdot \cdot h_1\left(PID_i || m_i || TS_i\right) + \sum_{i=1}^{n} pk_{i,1}. \quad (4)$$

If the equation is true, it means that the vehicle is verified, and the relevant data can be obtained from the CH; otherwise, the next step is continued.

3.2.5. Revocation/Update of Revocation List. If the CH verification equation does not hold, we need to trace the specific vehicle and update the revocation list. We assume the vehicle i is a malicious vehicle, then revoke it, and update revocation, which includes three steps:

(1) The CH sends (PID_i, pk_i) to the EA; then, the EA extracts the real identity of the vehicle i by calculating $RID_i = PID_{i,2} \oplus h_1(xpk_{i,1})$ and updating the revocation list.

(2) If the EA has a single point of failure, the CH sends (PID_i, pk_i) to the TA. The TA calculates $RID_i = PID_{i,2} \oplus h_1(xpk_{i,1})$, then the real identity is sent to the EA, and the revocation list is updated.

(3) If the vehicle is not on the revocation list, the EA calculates $H_u = h_1(RU_j) \oplus h_1(xpk_{i,1})$ for the vehicle. After the vehicle gets H_u, $h_1(RU_j)$ is obtained by calculating $h_1(RU_j) = H_u \oplus h_1(sk_{i,1}, PID_{i,1})$.

3.3. Security Analysis. In this subsection, we discuss the security requirements for our proposed scheme.

(1) Message authentication: In our proposed scheme, the vehicles are authenticated by the CH and thus obtain the relevant data. Multiple vehicles send messages to be verified to the CH with $\{S_i, PID_i, m_i, TS_i, RV_i, h_1(RU_i)\}$. Using the batch verification method, the message receiver can verify the legality of the message.

(2) Conditional privacy: Our scheme achieves conditional privacy protection. If a malicious vehicle i appears during verification, the CH can send (PID_i, pk_i) to the EA; then, the EA extracts the real identity of the vehicle i by calculating $RID_i = PID_{i,2} \oplus h_1(xpk_{i,1})$. The proposed scheme achieves double trace. When the EA has a single point of failure, the CH sends (PID_i, pk_i) to the TA. The TA calculates $RID_i = PID_{i,2} \oplus h_1(xpk_{i,1})$.

(3) Identity privacy preserving: In the proposed scheme, CM communicates with CH using pseudo identities. The EA calculates the pseudo identities for each vehicle: $PID_{i,1} = xP$, $PID_{i,2} = RID_i \oplus h_1(xpk_{i,1})$. Since solving $CDHP$ is difficult, it is not feasible for any vehicle to extract the real identity of another vehicle via pseudo identity, except for the trusted authority.

(4) Strong privacy preserving: Our proposed scheme achieves strong privacy preserving. Since the EA only knows the secret value x and does not know y, even if the EA is compromised, the adversary cannot obtain the privacy information of any vehicle.

3.4. Performance Analysis. In this subsection, we evaluate the computational overhead and communication load of the proposed scheme and compare it with three related schemes [17, 18, 48]. In [48], a distributed aggregation privacy protection authentication scheme (DAPPA) is proposed. All the signatures are divided into multiple subsets, and the aggregated signature is verified. In [17], an efficient certificateless batch authentication scheme without pairing (ECLA), which utilizes the elliptic curve cryptography to achieve batch authentication, is proposed. In [18], a new

identity-based message authentication scheme (ID-MAP) is proposed; it uses the message authentication of proxy vehicles to significantly reduce the computational cost of roadside units. The specific comparison process is described in the next two sections.

In this section, we analyze the computational overhead of the proposed scheme and compare it with [17, 18, 48]. We choose a bilinear pairing, e: $G_1 \times G_1 \longrightarrow G_2$, to achieve the security level of 80 bits, where G_1 is an additive group generated by a point p^* with the order q^* on the super singular elliptic curve E: $y^2 = x^3 + ax + b \pmod{p^*}$, p^* is a 512-bit prime number, and q^* is a 160-bit Solinas prime number. Meanwhile, we choose a nonsingular elliptic curve E: $y^2 = x^3 + ax + b \pmod{p}, a, b \in F_p$, where all points on the elliptic curve (including an infinity point) are on an addition group G whose generator is P, and p, q are 160 bits to achieve security level of 80 bits. To simplify the expression, we predefine the following symbols:

T_{bp}: the time to perform a bilinear pairing operation.

T_{bp}^{sm}: the time to perform a scale multiplication operation related to pairing-based cryptography (PBC).

T_{bp}^{pa}: the time to perform a point addition operation of the bilinear pairing.

T_{mtp}: the time to perform a map-to-point hash function operation related to PBC.

T_{ecc}^{sm}: the time to perform a scale multiplication operation related to ECC.

T_{ecc}^{pa}: the time to perform a point addition operation related to ECC.

We can obtain these cryptographic operations' execution time using MIRACL library [19], with the platform of 3.4 GHZ i7-4770. The execution time of the above cryptographic operations is $T_{\text{bp}} = 4.211ms, T_{bp}^{sm} = 1.709ms$, $T_{\text{bp}}^{pa} = 0.007ms$, $T_{mtp} = 4.406ms, T_{ecc}^{sm} = 0.442ms, T_{ecc}^{pa} = 0.0018$ms. Next, we analyze the details of computational overhead for DAPPA, ECLA, ID-MAP, and the proposed scheme.

The DAPPA scheme was proposed in [48] based on bilinear pairing. In this scheme, signing a message requires the signer to perform five scale multiplication operations of the bilinear pairing, one point addition operation of the bilinear pairing, and two map-to-point hash function operations: $5T_{bp}^{sm} + T_{bp}^{pa} + 2T_{mtp} = 17.364$ms. When the verifier receives n messages, it first divides the n messages into n^* subsets; each subset includes n/n^* messages to be verified. To verify n messages, the verifier needs to perform two bilinear pairing operations, n scale multiplication operations of the bilinear pairing, $(n - n^*)$ point addition operations of the bilinear pairing, and $2n$ map-to-point hash function operations (in this case, we take $n^* = 1$): $2T_{bp} + nT_{bp}^{sm} + 2T_{bp}^{pa} + 2nT_{mtp} = (10.521n + 8.436)$ms.

The ECLA scheme was proposed in [17] based on elliptic curve encryption. Signing a message in this scenario requires the signer to perform two scale multiplication operations of the elliptic curve cryptography: $2T_{ecc}^{sm} = 0.884$ms. When the verifier receives n messages, the verification of n messages requires the verifier to perform $5n$ scale multiplication operations of the elliptic curve cryptography and $3n$ point addition operations of the elliptic curve cryptography: $(5T_{ecc}^{sm} + 3T_{ecc}^{pa})n = (2.2154n)$ms.

The ID-MAP scheme was proposed in [18] based on elliptic curve encryption. In this scenario, the proxy vehicle needs to perform $(l + 6)$ scale multiplication operations of the ECC: $(l + 6)T_{ecc}^{sm} = 7T_{ecc}^{sm} = 3.094$ms, where l represents the number of proxy vehicles, and we set it as 1. When the roadside unit receives n messages, it needs to perform $5n/l$ scale multiplication operations of ECC to verify n messages: $5n/lT_{ecc}^{sm} = 5nT_{ecc}^{sm} = (2.21n)$ms. For the proposed OAAS based on elliptic curve encryption, to implement the signature of a message in this scheme, the CM should perform one scale multiplication operation of ECC: $T_{ecc}^{sm} + T_{ecc}^{pa} = 0.4438$ms. When the CH receives N messages to be verified, where there are n messages in the sequence with high priority, the CH verifies that n messages need to perform $2n$ scale multiplication operations of the elliptic curve cryptography and $(n + 1)$ point addition operations of ECC: $2nT_{ecc}^{sm} + (n + 1)T_{ecc}^{pa} = (0.8858n + 0.0018)$ms.

In order to more intuitively observe the computational performance of our proposed scheme, Figure 3 compares the total cost of the four schemes. As can be seen from Figure 3, our proposed scheme has the least time cost in signing and verifying a message.

Next, we analyze the computational complexity of the main steps of the proposed authentication scheme. Suppose that the security parameter is K. The most computationally expensive operations in the authentication scheme mainly include the scale multiplication operations of ECC and the multiplication operation of two big numbers, whose computational complexities are $O(\log_2 K)$ and $O(K)$, respectively. Other operations such as hash and XOR operations have $O(1)$ computational complexity. Therefore, the computational complexities of setup, pseudo identity generation and partial key extraction, sign, batch verification, and revocation/update of revocation list steps are $O(\log_2 K)$, $O(K)$, $O(K)$, $n\,O(\log_2 K)$, and $O(K)$, respectively, where n denotes the number of vehicles.

4. Intelligent Algorithm for DDoS Attack Detection in VANET Dataset

4.1. Simulation of VANET Segment in OMNeT++. Another problem of our research is to develop methods for detecting distributed denial-of-service attacks in software-defined vehicular ad hoc networks. To solve this problem we decided to generate VANET dataset which is suitable for our research purposes. VANET dataset generation includes the following steps:

(1) Simulation of VANET segment in different scenarios.

(2) Getting simulation results in the form of PCAP files.

(3) PCAP files processing and traffic flow feature extraction.

(4) Formation of single .csv dataset with the obtained features.

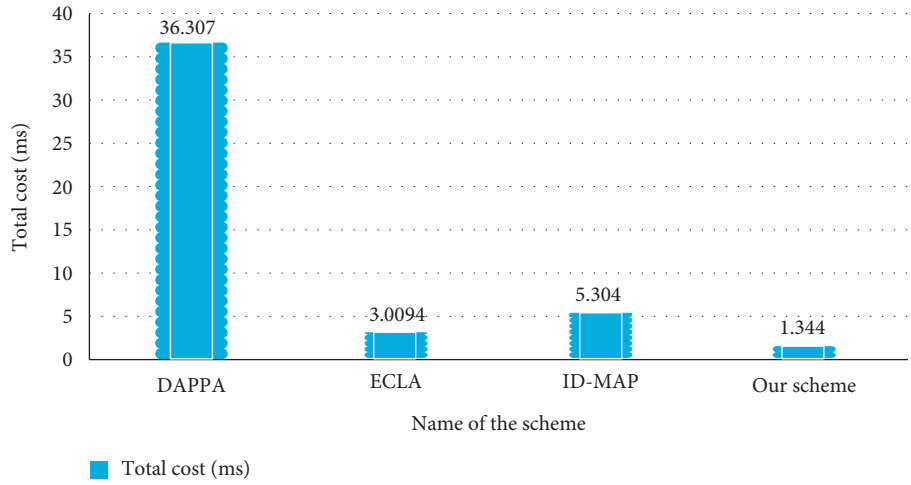

Figure 3: Total computation cost of the four schemes.

Typical solution of VANET simulation includes OMNeT++, INET framework [49], SUMO, and Veins framework [50]. As a first approximation, we will consider OMNeT++ and INET solution to build small segment of mobile ad hoc network with DoS and DDoS attack implementation scenarios. Our solution is based on MANET routing protocols, in particular AODV showcase.

Generally, simulation in OMNeT++ consists of three steps:

(1) Set network elements in .ned file to describe network model.

(2) Set general settings and each element's settings in omnetpp.ini file.

(3) Perform simulation and record the results.

5. Perform Simulation and Record the Results

The general settings of our VANET test segment are presented in Table 2.

We will consider the following simulation case: we have an immobile vehicle (source node) which had an accident and is trying to send information to the base station unit (destination node) using other vehicles (up to ten) as relay nodes. According to this, the elements of our VANET test segment are presented in Table 3.

Ten vehicles are freely moving across the given square area in random directions. To simulate this, we need to set vehicle movement to linear mobility type (see Table 4). We can also change type of node mobility to the VehicleMobility; in this case, it is necessary to set waypointFile for each vehicle with pairs of coordinates of each movement route around the designed area.

Our goal is to obtain the PCAP file of traffic flow for each node type, so we need to apply the PCAP recorder settings provided in Table 5. To use tools for network traffic flow feature extraction, we need to record data of sending and receiving packets.

The number of PCAP recorders is set to 4 because we will record information from source node, destination node, malicious node (in scenario of DoS attack), and arbitrary relay vehicle (let us say node 5). Option pcapLinkType = 101 allows us to record raw IP information of corresponding node.

AODV protocol operates with three types of messages: RREQ, RREP, and RERR. Source node is sending request

Table 2: General setting of VANET segment.

WLAN bitrate	24 Mbps
Transmitter power	1 mW
Area size	800 m × 800 m
Routing protocol	AODV
Link type	IEEE 802.11 wireless LAN
Simulation time	1000 s

Table 3: VANET segment elements and their types.

Element	Type
Source	ManetRouter
Destination	ManetRouter
Nodes 1–10	ManetRouter
radioMedium	Ieee80211ScalarRadioMedium
Visualizer	IntegratedMultiVisualizer
Configurator	Ipv4NetworkConfigurator
PcapRecorder	PcapRecorder

Table 4: Mobile node movement settings.

Type	Linear mobility
Initial movement heading	Uniform (0 deg, 360 deg)
Speed	Uniform (23 mps, 24 mps)

Table 5: Traffic flow PCAP recorder settings.

pcapLinkType	101 # raw IP
pcapFile	Results/∗∗∗.pcap"
moduleNamePatterns	"ipv4"
dumpProtocols	"ipv4"
numPcapRecorders	4

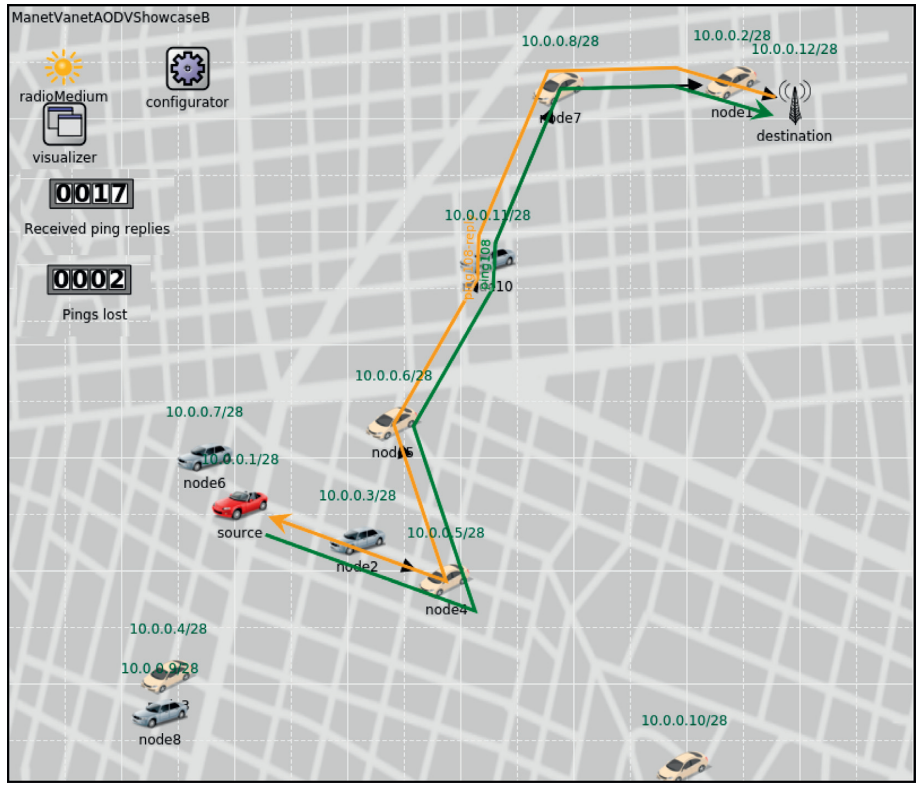

Figure 4: Demonstration of AODV protocol operation in OMNeT++.

message RREQ into the network, and relay nodes forward this message further, causing the building of temporary routes to the destination node. When destination node receives request, it sends RREP message back to the source node using a built temporary route. In case when a destination node is unreachable, a RERR message is used to notify other nodes in the network segment. In Figure 4, you can see an example of successful route building between source node and destination node using six relay vehicles. "Ping108" and "ping108-reply" lines are used for visual indication of established bidirectional connection between two nodes.

To implement denial-of-service attack, we will add single MaliciousNode element with ManetRouter type and the settings in Table 6.

MaliciousNode performs simple DoS attack of PingApp type, constantly sending dozens of packets to the source node. MaliciousNode is immovable and is placed closer to the source node in the area. In case of DDoS attack implementation, we will set five moving malicious nodes of ManetRouter type which inherit other vehicles movement settings mentioned earlier in the paper.

To prove the efficiency of implemented network attacks, we performed simulation in three scenarios (without attacks, with DoS attack, and with DDoS attack) and calculated the number of received pings (successful establishment of connection between source node and destination node) and the number of lost pings. Experiment results are presented in Table 7.

You can see that implementation of DoS attack with single malicious node interfered with network routing, and some connections were lost. Implementation of DDoS attack with five moving malicious nodes completely "paralyzed" the network segment, and no connections were established.

Table 6: Malicious node settings.

Attack type	PingApp
destAddr	"Source"
Start time	5 s
Sending interval	0.01 s
Radio transmitter power	200 mW
Packet size	0.5 Mb

5.1. Traffic Flows Feature Extraction. After obtaining PCAP files with captured traffic flows, we need to extract the set of features which will be used in machine learning methods. Any unusual behavior in traffic flow patterns and rapidly increased/decreased values of features can be qualified as possible network threat. Each PCAP file contains basic features such as source IP and number of port, destination IP and number of port, number of used protocols, and flow duration in seconds. Other features (such as number of forward/backward packets, average length of forward/backward packets, and average length of forward/backward packets headers) can be calculated manually or obtained using some software. CICFlowMeter [51] is a powerful tool to extract up to 84 network traffic features from PCAP files. The final segment of the network includes 15 relay vehicles, 2 RSUs, 1 BSU, and 6 malicious nodes with three types of

Table 7: AODVroute building results.

Route reply message	Scenario 1, without attacks	Scenario 2, DoS attack	Scenario 3, DDoS attack
Received pings replies	18	8 (partial connection)	No connection
Pings lost	2	1 (partial connection)	No connection

Table 8: Malicious nodes' behavior in scenario 4.

No.	App	Target node	Operation time interval	Packet size interval	Sending frequency	Mobility option
1	PingApp	Source	200–400 s	40–50 KB	0.01 s	Fixed
2	PingApp	Destination	400–700 s	50–60 KB	0.01 s	Fixed
3	PingApp	RSU1	250–450 s	10–30 KB	0.001 s	Linear
4	PingApp	RSU2	850–1450 s	20–40 KB	0.001 s	Linear
5	UDPBasicApp	Source	2600–3650 s	30–50 KB	0.01–0.05 s	Fixed
6	UDPBasicApp	Destination	3000–3850 s	40–60 KB	0.01–0.03 s	Fixed

(1) Calculate correlation matrix of 58 features.
(2) if (corr_value > 0.05):
(3) Select best features in correlation with "Label" column.
(4) Build scatter plot matrix of the best selected features.
(5) Carry out data preparation (data balancing).
(6) Perform multiclass classification by separate classifiers.
(7) Perform multiclass ensemble classification:
(8) Bagging Classifier usage.
(9) Voting Classifier usage.
(10) Stacking Classifier usage.

Algorithm 1: Multiclass classification of DDoS attacks.

behavior. Settings of each malicious node are shown in Table 8. Values in each given interval are distributed uniformly. To obtain more data, we increased simulation time to 5000 s and performed scenario 4 with three types of network attacks.

Recorded PCAP files were processed with CICFlowMeter; in general, 58 features of traffic flows were calculated in the form of .csv dataset. Each processed dataset contains "Label" column which is filled manually depending on the node type. Regular traffic flows were labeled "Benign," and malicious traffic flows were given three labels: "DDoS," "Intensive DDoS," and "UDP Flood" according to the type of network attack behavior. All processed files were concatenated into single dataset.

5.2. DDoS Attack Detection Using Machine Learning Methods. The final dataset of VANET segment contains 11212 rows: 8720 records of normal traffic, 965 records of DDoS attack, 820 records of Intensive DDoS attack, and 707 records of UDP Flood attack. In the first step, we dropped the following columns: "Src Port," "Dst Port," and "Protocol", which were used in traffic flows labeling. We also dropped the columns with constant values. Since our traffic flows are categorical features, we performed label encoding from ["Benign", "DDoS", "Intensive DDoS", "UDP Flood"] to [0, 1, 2, 3]. General approach to multiclass classification of network attacks includes the steps presented in Algorithm 1.

In the first three steps, we performed correlation analysis of our generated VANET dataset. The correlation matrix is shown in Figure 5. The seven best features were selected according to the set threshold corr_value (see Table 9 for details). "Active Max" feature with highest correlation is corresponding to the maximum value of time when traffic flow was active before becoming idle during simulation.

In step 4, we built plot pairwise relationships (scatter plot matrix) of the best selected features as is shown in Figure 6. It can be concluded that the distribution of the values of the best selected features by classes is visually distinguishable, and further multiclass classification is justified.

In step 5, we formed two datasets based on the original one: imbalanced dataset, Data; balanced dataset, DataSMOTE. To obtain a balanced dataset, the algorithm SMOTE [52] (Synthetic Minority Oversampling Technique) was used to synthesize new examples for the minority class (class with three types of network attacks) without duplication of data. The resulting dataset DataSMOTE contains 7831 records of normal traffic, 8174 records of DDoS attack, 8167 records of Intensive DDoS attack, and 8232 records of UDP Flood attack. Dataset Data remained intact. For both Data and DataSMOTE datasets, we performed a split into training and test subsets with default proportion of 3 : 1.

In step 6, we performed multiclass classification with 7 well-known classifiers: KNeighborsClassifier, RandomForestClassifier, CatBoostClassifier, XGBClassifier, LGBMClassifier, AdaBoostClassifier, and

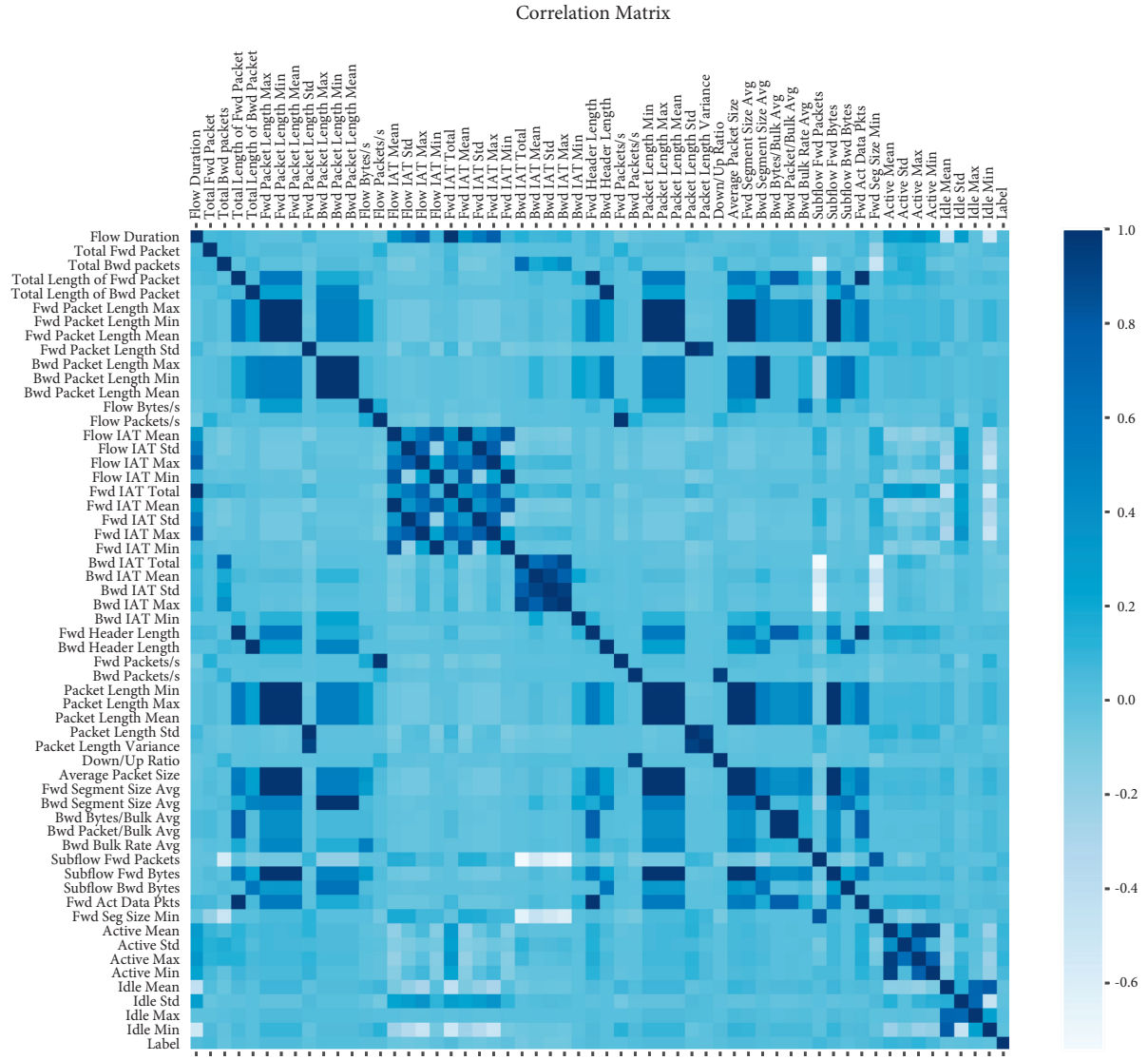

Figure 5: Correlation matrix of generated VANET dataset.

GradientBoostingClassifier. KNeighborsClassifier is implementation of k-nearest neighbors algorithm used for classification and regression problems. RandomForestClassifier is implementation of random forest meta-estimator algorithm. CatBoostClassifier is implementation of gradient boosting on decision trees algorithm developed by Yandex Company. XGBClassifier is implementation of parallel tree boosting algorithm. LGBMClassifier is implementation of gradient boosting model developed by Microsoft Company. AdaBoostClassifier is implementation of AdaBoost-SAMME algorithm. GradientBoostingClassifier is implementation of decision trees algorithms. Bentéjac C. et al. [53] performed comparative analysis of the family of gradient boosting algorithms in terms of speed and accuracy metrics. They concluded that LightGBM is the fastest classifier, but CatBoost shows the best results in generalization accuracy.

For each machine learning method, the optimal parameters were selected using the function GridSearchCV of the Python module sklearn. We built confusion matrices and calculated the following statistical metrics for all 7 classifiers: precision, recall, F1-score, and accuracy (see Figures 7 and 8). LightGBM classifier shows the best classification results for the balanced dataset DataSMOTE with accuracy of 0.9180; its confusion matrix is shown in Figure 9.

In steps 6–10, to improve the accuracy of network attack detection, we built ensembles of the best models using the bagging, voting, and stacking methods.

Table 9: Best selected features.

Feature	Correlation value
Total Fwd Packets	0.060131
Subflow Fwd Packets	0.078293
Fwd Seg Size Min	0.060144
Active Mean	0.092884
Active Std	0.084663
Active Max	0.104269
Active Min	0.066607

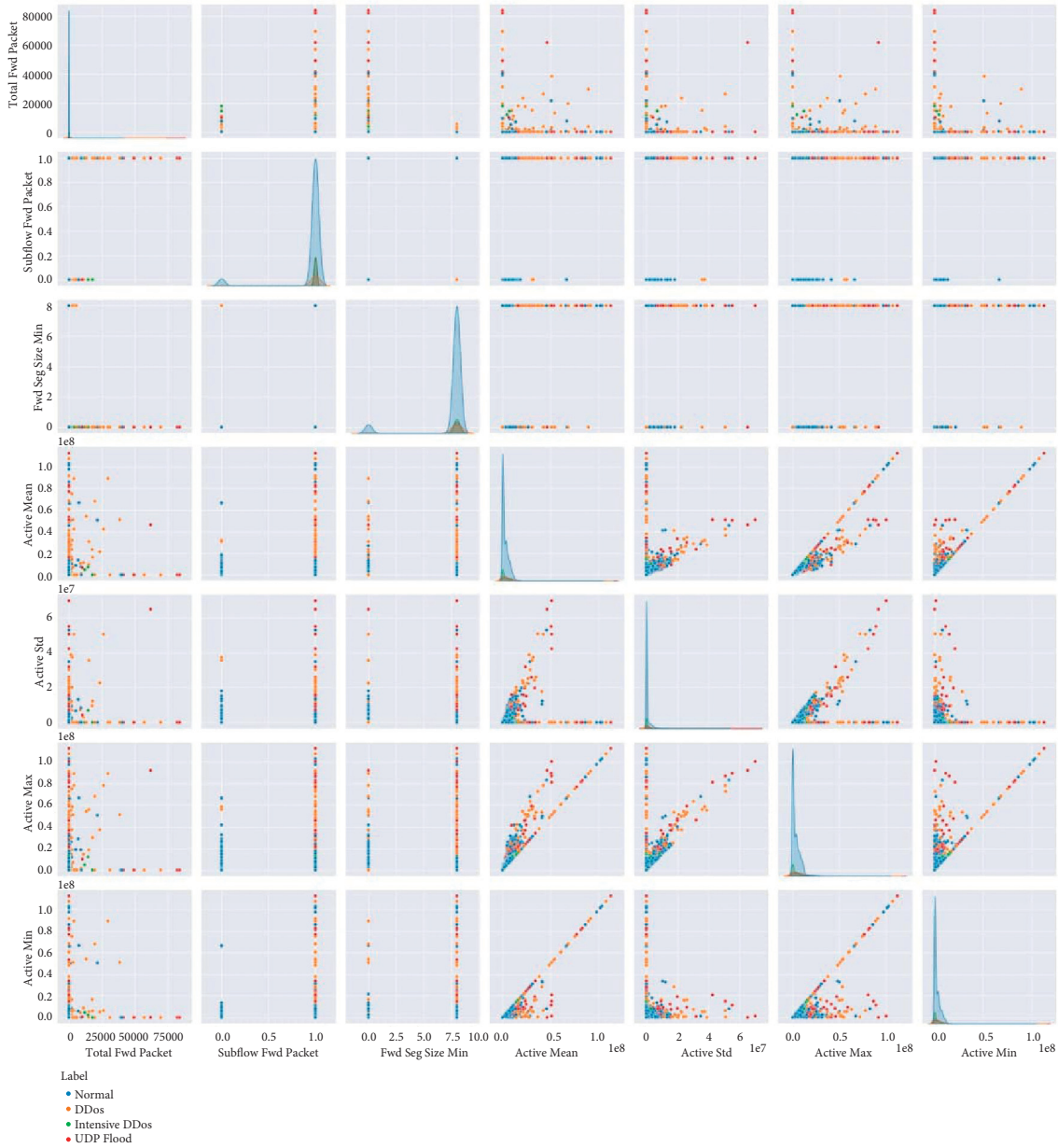

Figure 6: Pairwise relationships of the seven best selected features.

The bagging (or Bootstrap Aggregating) method is used to reduce variance and helps to avoid overfitting of machine learning algorithms of multiclass classification. We used all 7 classifiers and concluded that LightGBM classifier shows the best classification results with accuracy of 0.8942.

For the voting method, we selected 5 best classifiers (CatBoostClassifier, XGBClassifier, LGBMClassifier, AdaBoostClassifier, and GradientBoostingClassifier) and considered two voting methods: soft voting with obtained accuracy of 0.8773 and soft_weight voting with obtained accuracy of 0.8933. Soft voting is responsible for the simple averaging of the classifier values in the dataset and the output of average value of the class label. We first evaluated the classifiers separately, and we know their accuracy; from the obtained accuracy, we know the confidence levels of the classifiers. The confidence level is the weight coefficient; the higher it is for the classifier, the more influence it has on the weighted average value; therefore, the following weights were chosen for the 5 best classifiers: weights = [2, 2, 3, 2, 1].

For the stacking method, we considered the same 5 best classifiers and selected DecisionTreeClassifier and RandomForestClassifier as metaclassifiers. Calculated metrics of ensemble classifiers for both balanced and imbalanced datasets are presented in Figures 10 and 11. RandomForestClassifier showed the best classification results across all researched methods with accuracy of 0.9256; its confusion matrix is shown in Figure 12.

Due to the fact that we use our personal generated dataset, we cannot make a direct comparison with existing

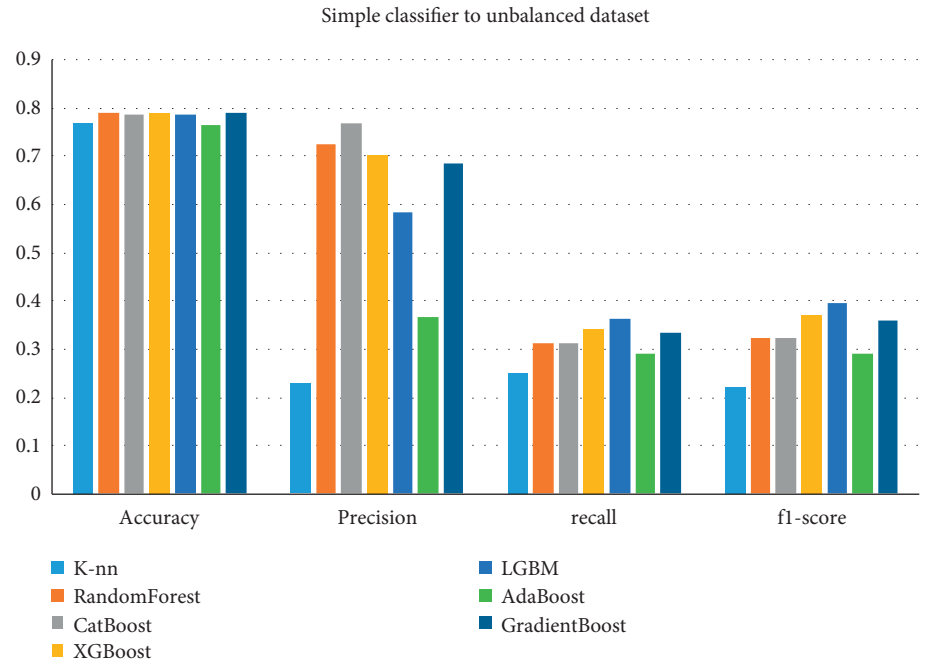

Figure 7: Separate classifiers' metrics comparison on the imbalanced dataset Data.

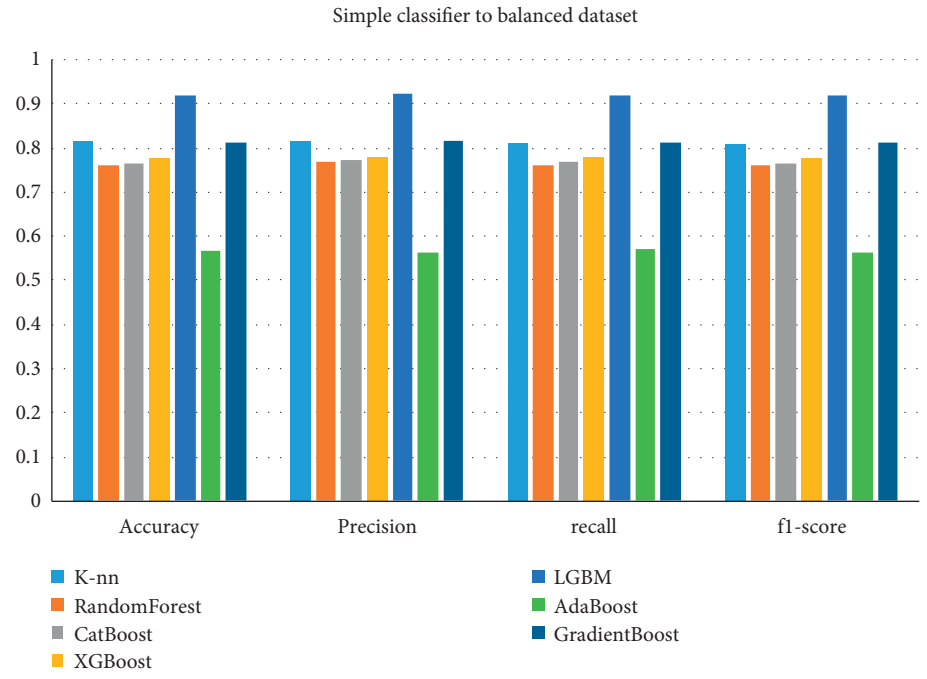

Figure 8: Separate classifiers' metrics comparison on the balanced dataset DataSMOTE.

results in DDoS attack detection. The primary goal of this paper was to demonstrate in detail the process of researching VANET network security issues, from generation of a network segment to usage of up-to-date classifiers and ensembles of classifiers for the multiclass classification problem.

The approach applied in this work to the formation of intelligent models for identifying network attacks can be replaced by an end-to-end pipeline machine learning system. The problem of automated application of machine learning methods to real-world problems is called AutoML. The AutoML [54] pipeline typically includes the

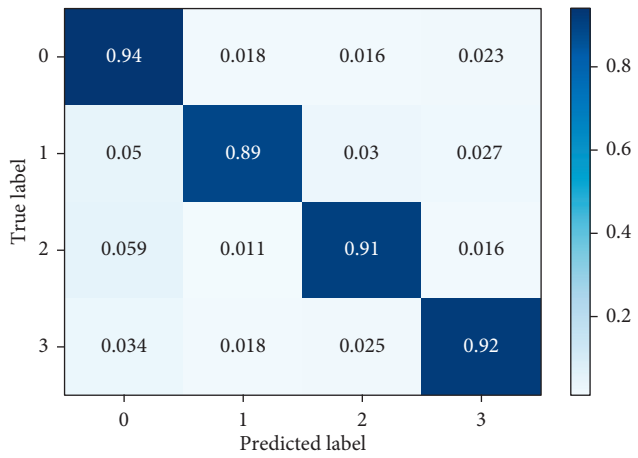

Figure 9: Confusion matrix of LightGBM classifier on the balanced dataset DataSMOTE.

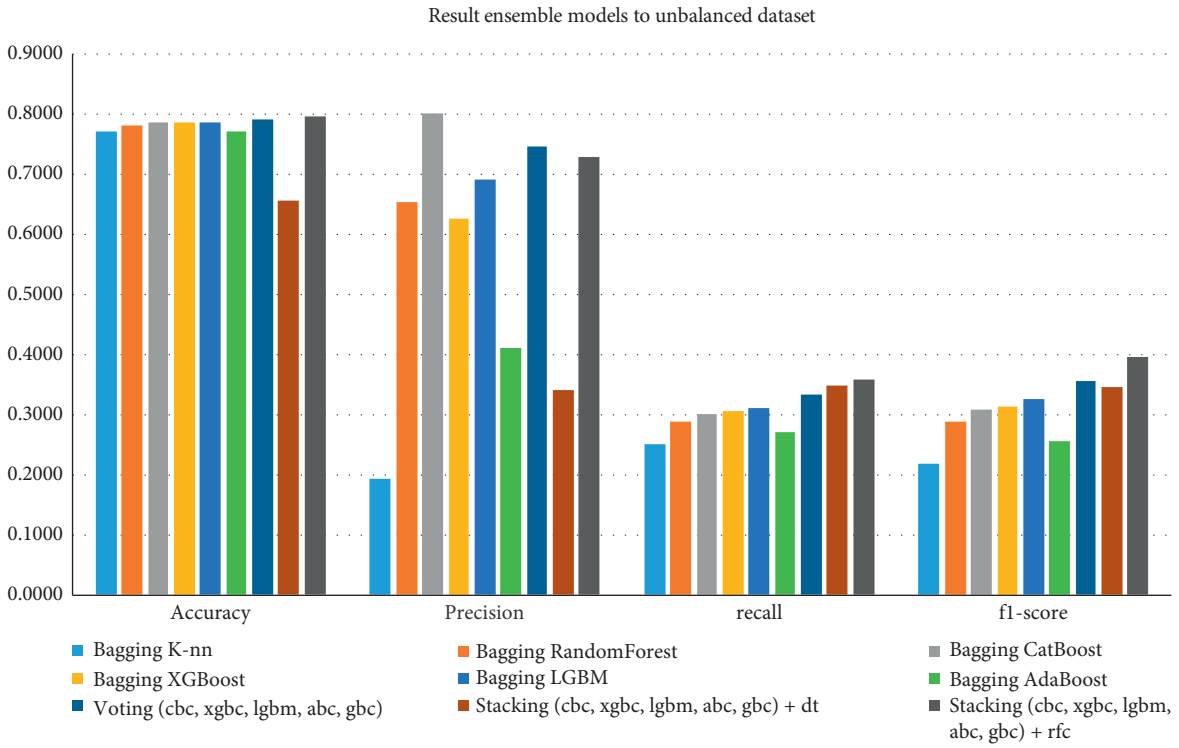

Figure 10: Ensemble classifiers' metrics comparison on the imbalanced dataset Data.

steps of data preparation, feature construction, model generation, and model evaluation. At the moment, there are many libraries that partially solve this problem; however, working with various kinds of raw data introduces great uncertainty into the assessment of their effectiveness.

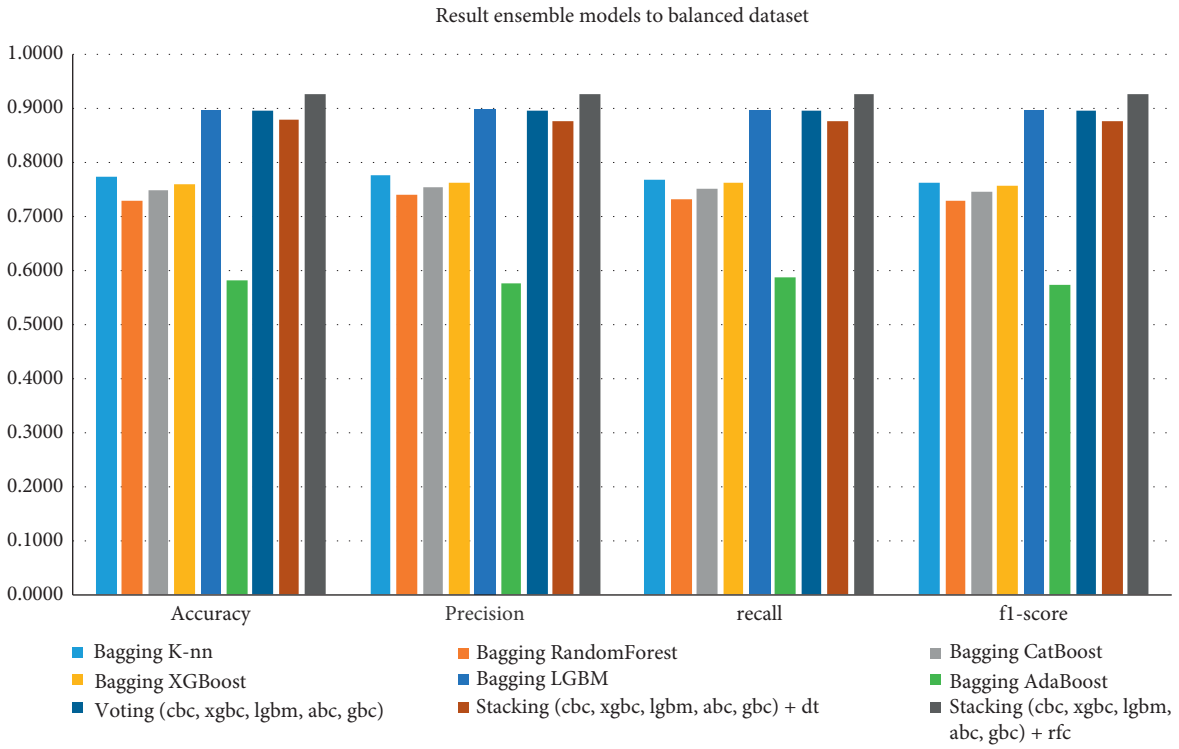

Figure 11: Ensemble classifiers' metrics comparison on the balanced dataset DataSMOTE.

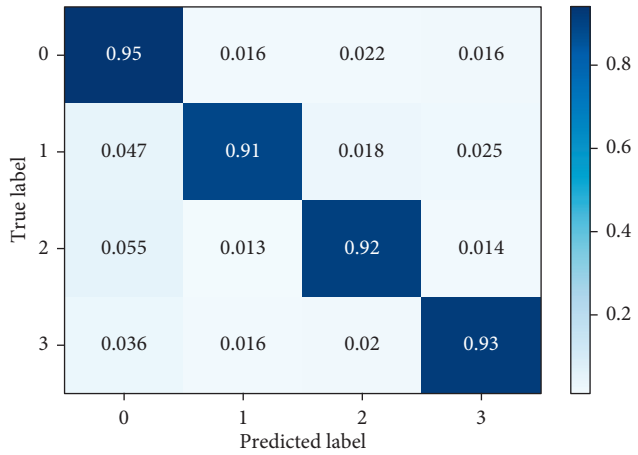

Figure 12: Confusion matrix of RandomForestClassifier on the balanced dataset DataSMOTE.

6. Conclusion

OMNeT++ software and INET framework are powerful tools for simulation of mobile ad hoc networks and implementation of various network threats. Modeling of VANET segment was conducted using OMNeT++, and the resulting VANET dataset contains 11212 network traffic flows with 58 extracted features and 3 network attacks behaviors implementations. The obtained dataset is imbalanced; therefore, for further research, balanced dataset was built using SMOTE technique.

An experimental comparison of the quality of modern machine learning methods of multiclass classification on the original and balanced datasets was carried out. The best results with accuracy of 0.9256 were shown by the stacking technique of classifiers with random forest as a metaclassifier.

In future research, we plan to build more simulation scenarios with increased number of vehicles, new mobility movements options, and different network attacks implementation. Our main goal is to increase efficiency of well-known multiclass classification algorithms on arbitrarily generated VANET datasets.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

The work was supported by the National Natural Science Foundation of China (Nos. 61872001 and 62011530046), the Cooperation and Exchange Project between NSFC and RFBR (No. 20-57-53019), and the grant of the President of the Russian Federation (MK-2959.2021.1.6), as well as scholarships of the President of the Russian Federation to young scientists and postgraduates (SP-3652.2021.5), the Open Fund of Key Laboratory of Embedded System and Service Computing (Tongji University), Ministry of Education (No. ESSCKF2018-03), the Open Fund for Discipline Construction (Institute of Physical Science and Information Technology, Anhui University), and the Excellent Talent Project of Anhui University.

References

[1] C.-X. Wang, M. D. Renzo, S. Stanczak, S. Wang, and E. G. Larsson, "Artificial intelligence enabled wireless networking for 5G and beyond: recent advances and future challenges," *IEEE Wireless Communications*, vol. 27, no. 1, pp. 16–23, 2020.

[2] J. Cheng, G. Yuan, M. Zhou et al., "Accessibility analysis and modeling for IoV in an urban scene," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 4, pp. 4246–4256, 2020.

[3] H. Zhong, J. Ni, J. Cui, J. Zhang, and L. Liu, "Personalized location privacy protection based on vehicle movement regularity in vehicular networks," *IEEE Systems Journal*, 2021.

[4] D. He, S. Zeadally, N. Kumar, and W. Wu, "Efficient and anonymous mobile user authentication protocol using self-certified public key cryptography for multi-server architectures," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 9, pp. 2052–2064, 2016.

[5] A.. Shamir, "Identity-based cryptosystems and signature schemes," *Workshop on the Theory and Application of Cryptographic Techniques*, Springer, Berlin, Heidelberg, Germany, 1984.

[6] J. Jinyuan Sun, C. Zhang, Y. Zhang, and Y. Fang, "An identity-based security system for user privacy in vehicular ad hoc networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 21, no. 9, pp. 1227–1239, 2010.

[7] S. F. Tzeng, S. J. Horng, T. Li, X. Wang, P. H. Huang, and M. Khurram, "Enhancing security and privacy for identity-based batch verification scheme in VANETs," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 4, pp. 3235–3248, 2015.

[8] X. Hu, J. Wang, H. Xu, Y. Liu, and X. Zhang, "Secure and pairing-free Identity-based batch verification scheme in vehicle ad-hoc networks," in *Proceedings of the International Conference on Intelligent Computing*, July 2016.

[9] L. Wei, J. Cui, Y. Xu, J. Cheng, and H. Zhong, "Secure and lightweight conditional privacy-preserving authentication for securing traffic emergency messages in VANETs," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 1681–1695, 2020.

[10] A. Riyami, S. Sattam, and K. G. Paterson, "Certificateless public key cryptography," in *Proceedings of the International Conference on the Theory and Application of Cryptology and Information Security*, vol. 2894, Novenmber 2003.

[11] A. Malip, S. L. Ng, and Q. Li, "A certificateless anonymous authenticated announcement scheme in vehicular ad hoc networks," *Security and Communication Networks*, vol. 7, no. 3, pp. 588–601, 2014.

[12] J. Cui, J. Zhang, H. Zhong, R. Shi, and Y. Xu, "An efficient certificateless aggregate signature without pairings for vehicular ad hoc networks," *Information Sciences*, vol. 451-452, pp. 1–15, 2018.

[13] Y. Ming and X. Shen, "PCPA: a practical certificateless conditional privacy preserving authentication scheme for vehicular ad hoc networks," *Sensors*, vol. 18, no. 5, p. 1573, 2018.

[14] J. Cui, J. Zhang, H. Zhong, and Y. Xu, "SPACF: a secure privacy-preserving authentication scheme for VANET with cuckoo filter," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 11, pp. 10283–10295, 2017.

[15] J. Cui, J. Chen, H. Zhong, J. Zhang, and L. Liu, "Reliable and Efficient Content Sharing for 5G-Enabled Vehicular Networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, 2020.

[16] L. Wei, J. Cui, H. Zhong, Y. Xu, and L. Liu, "Proven secure tree-based authenticated key agreement for securing V2V and V2I communications in VANETs," *IEEE Transactions on Mobile Computing*, vol. 14, 2021.

[17] N. B. Gayathri, G. Thumbur, P. V. Reddy, and M. Z. Ur Rahman, "Efficient pairing-free certificateless authentication scheme with batch verification for vehicular ad-hoc networks," *IEEE Access*, vol. 6, pp. 31808–31819, 2018.

[18] M. R. Asaar, M. Salmasizadeh, W. Susilo, and A. Majidi, "A secure and efficient authentication technique for vehicular ad-

hoc networks," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 6, pp. 5409–5423, 2018.

[19] Miracl, "Multiprecision integer and rational arithmetic cryptographic (MIRACL) library," 2019, https://github.com/miracl/MIRACL.

[20] J. Montenegro, C. Iza, and M.. Aguilar Igartua, "Detection of position falsification attacks in VANETs applying trust model and machine learning," in *Proceedings of the 17th ACM Symposium on Performance Evaluation of Wireless Ad Hoc*, November 2020.

[21] P. K. Singh, S. Gupta, R. Vashistha, S. K. Nandi, and S. Nandi, "Machine learning based approach to detect position falsification attack in vanets," in *Proceedings of the International Conference on Security & Privacy*, April 2019.

[22] F. A. Ghaleb, F. Saeed, M. Al-Sarem et al., "Misbehavior-aware on-demand collaborative intrusion detection system using distributed ensemble learning for VANET," *Electronics*, vol. 9, no. 9, p. 1411, 2020.

[23] T. Nandy, R. M. Noor, M. Y. I. B. Idris, and S. Bhattacharyya, "T-BCIDS: trust-based collaborative intrusion detection system for VANET," in *Proceedings of the National Conference on Emerging Trends on Sustainable Technology and Engineering Applications (NCETSTEA)*, February 2020.

[24] S. Sharma and A. Kaul, "Hybrid fuzzy multi-criteria decision making based multi cluster head dolphin swarm optimized IDS for VANET," *Vehicular Communications*, vol. 12, pp. 23–38, 2018.

[25] T. Zhang and Q. Zhu, "Distributed privacy-preserving collaborative intrusion detection systems for VANETs," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 4, no. 1, pp. 148–161, 2018.

[26] D. Zhang, F. Richard Yu, and R. Yang, "A machine learning approach for software-defined vehicular ad hoc networks with trust management," in *Proceedings of the IEEE Global Communications Conference (GLOBECOM)*, December 2018.

[27] V. Belenko, V. Krundyshev, and M. Kalinin, "Synthetic datasets generation for intrusion detection in VANET," in *Proceedings of the 11th International Conference on Security of Information and Networks*, Cardiff, UK, September 2018.

[28] P. K. Singh, R. R. Gupta, S. K. Nandi, and S. Nandi, "Machine learning based approach to detect wormhole attack in VANETs," in *Proceedings of the Workshops of the International Conference on Advanced Information Networking and Applications*, March 2019.

[29] S. Kumar, K. Singh, S. Kumar, O. Kaiwartya, Y. Cao, and H. Zhou, "Delimitated anti jammer scheme for Internet of vehicle: machine learning based security approach," *IEEE Access*, vol. 7, pp. 113311–113323, 2019.

[30] A. Rehman, S. Rehman, M. U. Khan, M. Alazab, and T. Reddy, "CANintelliIDS: detecting in-vehicle intrusion attacks on a controller area network using cnn and attention-based gru," *IEEE Transactions on Network Science and Engineering*, vol. 8, no. 2, 2021.

[31] R. Jhaveri, S. V. Ramani, G. Srivastava, T. R. Gadekallu, and V. Agarwall, "Fault-resilience for bandwidth management in industrial software-defined networks," *IEEE Transactions on Network Science and Engineering*, vol. 8, 2021.

[32] N. Akhtar, S. C. Ergen, and O. Ozkasap, "Vehicle mobility and communication channel models for realistic and efficient highway VANET simulation," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 1, pp. 248–262, 2014.

[33] Sumo, "simulation of urban MObility," 2001, http://sumo.sourceforge.net.

[34] F. Michaeler and C. Olaverri-Monreal, "3D driving simulator with VANET capabilities to assess cooperative systems: 3DSimVanet," in *Proceedings of the 2017 IEEE Intelligent Vehicles Symposium (IV)*, June 2017.

[35] D. S. Buse, "Christoph Sommer, and Falko Dressler. Demo abstract: integrating a driving simulator with city-scale VANET simulation for the development of next generation ADAS systems," in *Proceedings of the 2018-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, April 2018.

[36] C. Obermaier, R. Riebl, and C. Facchi, "Fully reactive hardware-in-the-loop simulation for vanet devices," in *Proceedings of the 2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, November 2018.

[37] OMNeT, "OMNeT++Discrete event simulator," 2020, https://omnetpp.org/.

[38] T. O. Fahad and A. A. Ali, "Compressed fuzzy logic based multi-criteria AODV routing in VANET environment," *International Journal of Electrical and Computer Engineering*, vol. 9, no. 1, p. 397, 2019.

[39] B. P. Maratha, T. R. Sheltami, and K. Salah, "Performance study of MANET routing protocols in VANET," *Arabian Journal for Science and Engineering*, vol. 42, no. 8, pp. 3115–3126, 2017.

[40] Network world, "NCTUns 6.0 Network Simulator and Emulator," 2017, http://nsl.cs.nctu.edu.tw/NSL/nctuns.html/.

[41] C. Raj, T. Makwana's, U. Upadhayaya, and P. Mahida, "Simulation of VANET using ns-3 and SUMO," *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 4, p. 4, 2014.

[42] NS-3, "Discrete-event network simulator," 2010, https://www.nsnam.org/.

[43] S. So, P. Sharma, and J. Petit, "Integrating plausibility checks and machine learning for misbehavior detection in VANET," in *Proceedings of the 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, December 2018.

[44] J. Grover, N. K. Prajapati, V. Laxmi, and M. S. Gaur, "Machine learning approach for multiple misbehavior detection in VANET," in *Proceedings of the International Conference on Advances in Computing and Communications*, July 2011.

[45] Yi Zeng, Z. Ming, and M. Liu, "Senior2local: a machine learning based intrusion detection method for vanets," in *Proceedings of the International Conference on Smart Computing and Communication*, December 2018.

[46] K. Sharshembiev, S. M. Yoo, and E. Elmahdi, "Protocol misbehavior detection framework using machine learning classification in vehicular Ad Hoc networks," *Wireless Networks*, vol. 27, pp. 1–16, 2021.

[47] B. A. Tama and S. Lim, "Ensemble learning for intrusion detection systems: a systematic mapping study and cross-benchmark evaluation," *Computer Science Review*, vol. 39, Article ID 100357, 2021.

[48] L. Zhang, Q. Wu, J. D. Ferrer, B. Qin, and C. Hu, "Distributed aggregate privacy-preserving authentication in VANETs," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 3, pp. 516–526, 2016.

[49] INET, "INET Framework," 2002, https://inet.omnetpp.org/.

[50] C. Sommer, R. German, and F. Dressler, "Bidirectionally coupled network and road traffic simulation for improved IVC analysis," *IEEE Transactions on Mobile Computing*, vol. 10, no. 1, pp. 3–15, 2010.

[51] L. A. Habibi, G. D. Gil, M. Mamun, and A. A. Ghorbani, "Characterization of tor traffic using time based features," *ICISSp*, 2017.
[52] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, 2002.
[53] C. Bentéjac, C. Anna, and G. Martínez-Muñoz, "A comparative analysis of gradient boosting algorithms," *Artificial Intelligence Review*, vol. 54, pp. 1–31, 2020.
[54] Autogluon, "AutoML for text, image, and tabular data," https://github.com/awslabs/autogluon.