Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 8891758, 17 pages
https://doi.org/10.1155/2021/8891758

Research Article

MSOM: Efficient Mechanism for Defense against DDoS Attacks in VANET

Mohammed Al-Mehdhara and Na Ruan

Department of Computer Science, Shanghai Jiao Tong University, Shanghai, China

Correspondence should be addressed to Mohammed Al-Mehdhara; almehdhar@sjtu.edu.cn

Received 16 June 2020; Revised 13 December 2020; Accepted 22 January 2021; Published 10 April 2021

Academic Editor: Oscar Esparza

The wireless nature of the Vehicular Ad Hoc Network (VANET), a technology that offers facilities such as traffic management and safety services, makes it vulnerable to distributed denial-of-service (DDoS) attacks that exploit network communications and reduce network reliability and performance. This paper proposes a design of a secure VANET architecture using a Software-Defined Networking (SDN) controller and Neural Network Self-Organizing Maps (SOMs). In the proposed design, we adopt the SDN architecture by using its separation of the control plane from the data plane and adding intelligent capabilities to the VANET. To resolve the drawbacks of standard SOMs and to enhance the SOM's efficiency, a Multilayer Distributed SOM (MSOM) model based on two levels of clustering and classification is used. Experimental results show that our solution can efficiently detect malicious traffic, prevent and mitigate DDoS attacks, and increase system security and recovery speed from the attacking traffic. Moreover, the proposed scheme achieves a high accuracy rate (99.67%). Simulation results demonstrate the effectiveness and efficiency of the MSOM regarding detection accuracy and other studied metrics.

1. Introduction

Vehicular Ad Hoc Network (VANET), an innovative network that uses different technologies for next-generation Intelligent Transportation Systems (ITS), offers an array of functional services to vehicles and roadside infrastructure, all of which result in better efficiency and safety of transport operations [1]. In this environment, vehicles communicate with each other through an ad hoc network—specifically, using the Vehicle-to-Vehicle (V2V) mode. Furthermore, vehicles can also communicate with roadside units (RSUs) through Vehicle-to-Infrastructure (V2I) communication, including Vehicle-to-Roadside-Units (V2RU) and Vehicle-to-Base-Stations (V2BT). Based on the appropriate input it receives, VANET can optimally manage traffic by providing safety information, traffic jam warnings, road maintenance, and intervehicle distance messages. However, the ad hoc nature of communication in VANET makes information sharing through it vulnerable to various types of security threats and privacy attacks [2] (see Table 1 for a summary). For instance, VANET is vulnerable to various attacks against confidentiality, such as eavesdropping, as well as to several types of attacks against integrity, such as masquerade, blackhole, and replay. Other possible attacks against VANET include jamming, denial of service (DoS), and distributed denial of service (DDoS) [3]. Among these attacks, the most severe threats to the VANET environment are DDoS attacks. While the main target of DDoS attacks is network availability, such attacks can also exhaust or destroy nodes and network resources. DDoS attacks also decrease the packet delivery rate so that the network throughput sustains a higher level of delay, control overhead, and overall network overload. Since real-time traffic is an important communication feature in VANET, an event causing a loss of regular transmissions, even for an instant of time, could be fatal [3]. Flooding attack is a critical DDoS attack category. The main idea of this type of attack is to turn the victim node inaccessible or to reduce the correspondence all the way by the network, thereby adversely affecting availability. Attacks of this kind result in network overloading. Figure 1 shows the most threats and attacks targeting VANET.

Table 1: Simulation parameters.

Parameter	Value
Simulators	NS-3
Simulation time	100 seconds (minimum time in network)
Communication range area	DSRC ~200 m
Packet size	1000 bytes
Wireless network	V-to-V and V-to-I = DSRC, V-to-eNBs = LTE/Wi-Fi
Data transmission speed	Wireless = 6 Mbps, wired = 100 Mbps, Internet = 100 Gbps
Packet intervals	Hello message = 3 seconds, status update message = 3 seconds
Routing protocol	OLSR

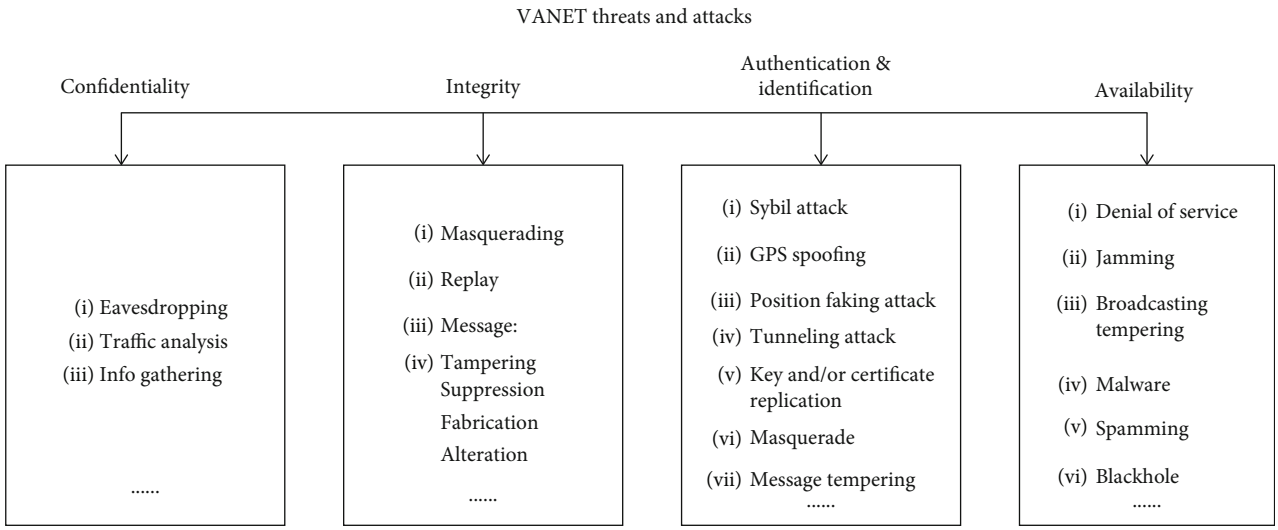

Figure 1: Classification of VANET threats and attacks.

Using VANET, vehicles may exchange messages such as periodic WAVE Short Messages (WSMs) with neighboring nodes by registering the nearest RSUs. In addition, vehicles may either create flooding WAVE Basic Service Set (WBSS) or send periodic WSMs to transmit their emergency public safety messages. Since all these critical service/application notifications or other important information from RSUs or vehicles are exchanged periodically, a malicious node can start a formidable flooding attack on VANET by simply synchronizing to the corresponding target periodic broadcast schedule.

Vehicular network security requirements can be addressed using Software-Defined Networking (SDN) [4] that brings a wide range of network features, such as adaptability, programmability, and centralized control with unified abstraction. All these features empower users to flexibly manage the network [5]. Due to the aforementioned capabilities of SDN, the SDN paradigm is now widely used in ad hoc networks [5, 6].

Furthermore, since malicious packets' headers imitate regular packets, an effective DDoS attack detection mechanism requires an exhaustive packet analysis, which may create more overhead on network utilization. Effective solutions to address this issue include machine learning techniques [7] and neural networks (NNs). The SDN programmability feature allows adding network solutions and lightweight add-ons implemented using machine learning algorithms (ML) and working in the real-time mode. In addition, ML techniques are widely used to gather audited data and compute patterns that predict the actual behavior of data input, all of which can be used to detect and track various DDoS attacks [8].

The self-organizing map (SOM) algorithm, a fundamental approach to NN introduced by Miljkovic [9], has been widely used in clustering problems and data investigation. A SOM is a neural network-based model that takes the form of a grid of neurons. In this paper, we aim to take advantage of the existing SOM model [9] to investigate the relationship between supervised and unsupervised learning. The main goal of developing a supervised layer with the standard SOMs is to improve the scope of data classification with this algorithm and to boost the learning process.

The centralized control plane architecture ensures superior control performance and resource management over the heterogeneous VANET states. However, this architecture has different vulnerabilities and is prone to various types of security threats, such as limited detection throughput and susceptibility to DDoS attacks. Likewise, due to the aggregation of traffic nodes into a single location, while active, a single SOM may be vulnerable to a performance bottleneck, which creates a target for a DDoS attack [10].

In the present study, seeking to overcome the aforementioned critical issues, as well as to enhance the robustness of VANET-based SDN networks, we propose a Multilayer Distributed SOM (MSOM) system—a new mechanism to tackle the performance bottleneck problem under DDoS attacks. Instead of using a single SOM in the upper plane as a security service, our method investigates the distribution of multilayered SOMs integrated into SDN controllers.

Due to their remarkable performance results, VANET networks, SDN technology, and VANET-based SDN have been extensively investigated in previous research. As argued in [11], Software-Defined Vehicular Network is a potential paradigm to systematically control networks. A comparison of the VANET's achieved performance with and without SDN support was investigated in [12], with the results suggesting that VANET and SDN networks can manage roadside controllers to achieve high performance. Furthermore, [13] proposed a software-defined vehicular network defense mechanism.

The defense mechanism is aimed at detecting a flooding attack by the time series analysis of packet flow and at mitigating the attack, creating thus a flow tree to establish the source of spoofed packets. In [14], a new DDoS detection method based on SDN architecture features, i.e., flow monitoring, was introduced. Using a SOM, the method proposed in [14] classifies network traffic as malicious or nonmalicious. Following [15], simplification of the security provision within the network can guarantee security against new SDN attacks aimed at the data and control planes. In [16], Flood Defender, which is protocol independent and elastic to mitigate the SDN-aimed DoSS attacks, was proposed. Furthermore, [17] discussed using the SDN paradigm to mitigate the distributed denial of service attack using OpenFlow protocol as a means to improve the legacy Remote Triggered Blackhole (RTBH).

In [18], a data mining methodology to detect known attacks and discover other unknown attacks in VANETs was presented. Overall, this solution has the following three main advantages: (1) a decentralized vehicle network with scalable communication and data available about the network, (2) the use of two data mining models to show feasibility for an IDS in VANETs, and (3) finding new patterns of unknown intrusions. In the proposed system, the network is divided into a cell grid where each cell has a transmission tower enabling communication with other cells and the Internet. Each cell will run its data mining models and rules, detecting new attacks and thereby allowing the IDS to create new rules of transmission for each subnetwork. In [19], a multiqueue SDN controller scheduling algorithm to mitigate DDoS attacks in SDN was proposed. This algorithm tries to protect the normal switches during a DDoS attack by scheduling the flow request processing through different switches. This method uses multiple time-slicing plans depending on the DDoS attack severity. Furthermore, [20] proposed a solution to protect the VANET availability from DDoS and DoS attacks based on various severity levels. Seeking to minimize the number of messages at any stamp of time to be received by any node in the network, the aforementioned solution uses evocation techniques and a Dedicated Short-Range Communication (DSRC) channel. Another relevant study investigated the DoS attacks in IEEE 802.11 network detection in real time [21]. The proposed detector continuously monitors the events occurring in the wireless channel and inspects each collision using probabilistic computation. The main goal of this method is the basic mode of IEEE 802.11 with arbitrary unicast traffic. To this end, the binary exponential back-off algorithm to trigger the retransmission is employed. In [22], the Multivariant Stream Analysis (MVSA) approach was proposed to feature the numerous phases of detecting a DDoS attack in Vehicular Ad Hoc Networks.

Another approach to detect and locate DDoS attacks in VANET is Stream Position Performance Analysis (SPPA) proposed in [8, 23]. By calculating and monitoring several metrics, such as conflict field, conflict data, and attack signature sample rate (CCA), this approach inspects the position of any field station in sending the information to launch a DDoS attack. Using the calculated metrics, CCA distinguishes malicious and trustworthiness packets. Its DDoS detection performance was reported to be remarkable [8, 23].

Furthermore, [22] proposed the Multivariant Stream Analysis (MVSA) method I to detect and mitigate DDoS attacks on VANET using NS2 simulation. MVSA provides V2V communication through RSU, by determining an average payload rate, frequency at different times, and the time to live per vehicle for each strike class. The MVSA method inspects the trace files to identify the DDoS. Then, MVSA decides on the stream weight, which is followed by the classification of stream packets as either legitimate or malicious. However, while the MVSA method demonstrated stability and good performance, its drawback is that the reduction in the packet delay is not assured to detect the malicious node.

In another relevant study, Periodic Self-Organizing Maps (PSOM) based on an unsupervised method to detect aberrations in periodic time series were introduced [24]. Similarly, [25] proposed an anomaly detection approach based on the growing hierarchical SOM. This approach relies on the following two stages: (1) the mining stage and (2) the identifying stage. Converting high-dimensional data to two-dimensional data while keeping the relationships between clustering and topological relation, the proposed method is used with off-line intrusion detection assessment, and the detection knowledge is used with an IDS.

Based on the evidence briefly reviewed above, it can be concluded that previous studies have attempted to solve the issue of DDoS in the SDN and VANET. However, although the SOM technique has been applied in various studies on SDN, to the best of our knowledge, none of the previous studies focused on the domain of VANET. Moreover, most of the proposed solutions used only the traditional SOM mechanism and thus may not have overcome the limitations of the centralized SDN design. Furthermore, while all available solutions to prevent DDoS attacks—including both those used in conjunction with the eNBC and those integrated with RSU or edge devices present in the vehicles—hold much promise, the information-based metrics and verification algorithms have various drawbacks, which may cause delays in attack detection. Furthermore, several of the proposed have focused only on the management of privacy issue

solutions. In addition, while several DDoS detection mechanisms work by periodically triggering the detection techniques, they have limitations in terms of the time needed to activate the detection time window.

Another limitation of previously proposed solutions DDoS mitigation approaches is that they use centralized solutions by implementing systems at the node location, on the cloud, or at the data center. While centralized solutions are indeed capable of mitigating DDoS attacks, they are not sufficiently effective with regard to the delay issues and required resources. In a centralized SDN VANET-based architecture, the controller is typically removed from the vehicles, which results in a significant latency of the design's operations [26]. This impact of this latency impact gets particularly noticeable in collecting flow information, which gets very time-consuming [27]. Consequently, the centralized approach adversely affects routing, and the route setup time gets considerably extended. Furthermore, while a centralized approach offers a global view of the entire underlying network, it may consider as a single point of network failure.

The present study is inspired by the concept of distributed multilayer SOM-based detection, which may be the most efficient currently available defense method to detect DDoS flooding attacks. In order to protect the network from such attacks, we need a bulletproof solution that would effectively overcome the shortcomings of the available solutions. The approach proposed in the present study differs from the approaches briefly reviewed above in the following several aspects. First, our architecture works in a distributed method to solve the central point of failure and to achieve load balancing. Second, we use a multilayered distributed SOM to enhance dictation performance with low overhead. Third, the proposed approach is designed to provide a smart system to not only detect the attack but also mitigate and prevent future attacks. Finally, as will be demonstrated in Section 4, the scenario results collected with our MSOM approach are very efficient. The main contributions of the present study can be summarized as follows:

(i) We propose integrating novel multilayer self-organizing map (MSOM) security modules into the distributed SDN controllers. In this module, we propose a multilayered SOM using an unsupervised and supervised learning algorithm

(ii) We propose a distributed real-time VANET-SDN-based detection and mitigation mechanism that offers reliable and accurate classification to detect and mitigate DDoS flooding attacks

(iii) In order to demonstrate the effectiveness of our architecture in terms of securing applications built for VANET, extensive simulations are performed

The remainder of the paper is organized as follows. Section 2 describes the preliminary work. Section 3 presents the architecture design. A detailed explanation of the experiments and a summary of the results are provided in Section 4. Finally, Section 5 draws conclusions and outlines directions of future research.

2. Preliminary Work

A self-organizing map (SOM), a type of the NN technique, is an unsupervised learning algorithm first introduced by Adhikary [8]. Due to its capability to transmit nonlinear relationships between high-dimensional data into a lower-dimensional geometric relationship of a regular two-dimensional map, SOM can be used to classify and visualize high-dimensional data.

The process of the SOM algorithm consists of the following four steps:

(i) *Initialization*. In this step, the algorithm selects random numbers for the initial weight vectors. Then, a learning rate is assigned via a parameter value

(ii) *Activation*. The input of Xi activates the SOM and searches for similarities of the Best Matching Unit (BMU) neuron Xi at iteration p. The measure in Equation (1) is calculated using the norm of Euclidean distance

$$E = \min \left\| X - W_{ij}(p) \right\| = \sqrt{\sum_{i=1}^{n} X_i - W_{ij}(p)\big]^2} \quad (1)$$

(iii) *Updating*. This step involves applying the weight update function. Equation (2) describes the functionality of the update step where Θ is a restraint function. Since Equation (2) has a distance from the BMU, it is also known as the neighborhood function. The function refers to the learning rate of the SOM which, in turn, refers to the weight repairing and its iteration

$$W_{ij}(p+1) = W(p) + \Theta(p)\alpha(p)\big(X(p) - W_{ij}(p)\big) \quad (2)$$

(iv) *Continuation*. In this final step, the second step is repeated until the map is adequately organized. Noticeable changes in the map become zero. Upon completion, the data become organized and are maintained in a similar region. Winning nodes will handle data from similar regions [8]

3. The Proposed MSOM Architecture with the Multilayer Distributed Mechanism

In this section, we describe the architecture of our system and how integrating SDN on top of VANET networks can enhance the security against DDoS attacks while using Multilayer Distributed SOM (MSOM).

3.1. Architecture Design. As mentioned in Section 2, when deploying the SDN VANET architecture, the controllers and data plane nodes should be located as close as possible to each other to decrease the latency encountered by the vehicles. Accordingly, recent research on the SDN VANET is aimed at maximally minimizing the amount of communication between the control and data planes [28]. The standard VANET architecture features fixed RSUs, which makes it

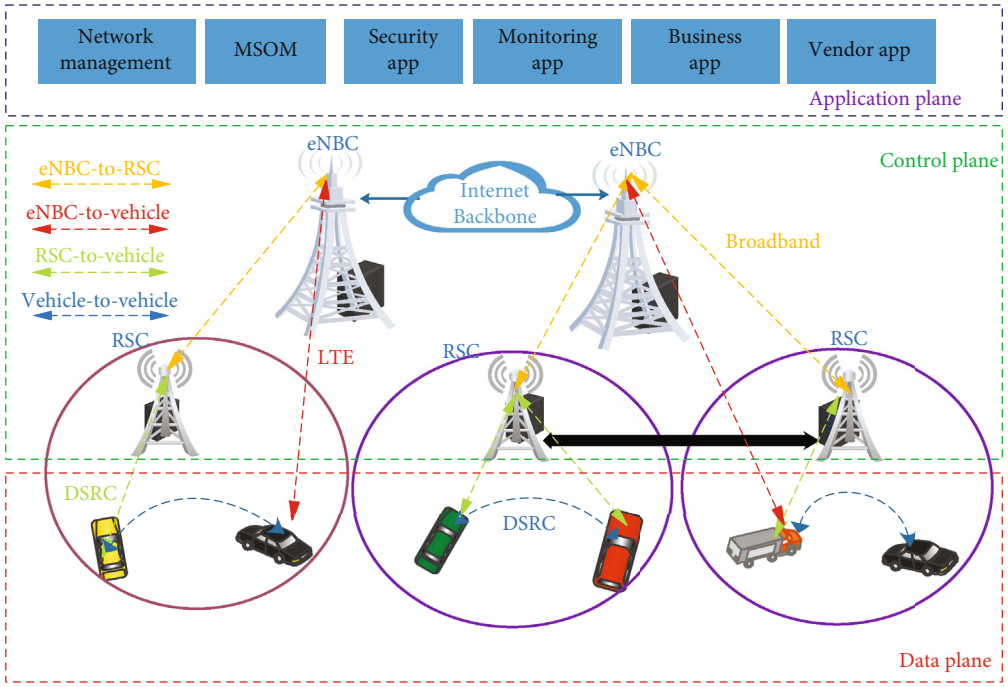

Figure 2: VANET-based SDN components of the communication route.

possible to move the control plane from the Internet to the RSU level [29]. This will reduce connecting to the controller and enables the use of the existing DSRC wireless technology to communicate with controllers, instead of using high-cost cellular links. The SDN controller communicates with the roadside controller by using the OpenFlow protocol, while the roadside controller connects with vehicles within its coverage area. To this end, it relies on the capability extension of the OpenFlow standard through the supporting status update of each vehicle. In order to tackle the issue of load balance and to distribute the workload among controllers, our architecture was designed with distributed controller placement.

Specifically, the proposed architecture was designed to delegate traffic processing tasks among distributed controllers over RSUs to decrease the workload and to process a small amount of traffic in the case of policy checking. In addition, in our architecture, each roadside unit handles only the traffic entering its ports from outside networks. Therefore, as compared to the single controller case, the load on the RSU is not very significant.

Figure 2 shows the distribution of the architecture components. Overall, the proposed architecture model contains the following modules.

(i) *Application Module*. This module is placed in the vehicle and contains an application (software) to communicate with other application modules via wireless technology. Application module communication is facilitated by Open vSwitch (OVS) and Open vSwitch Database (OVSDB) [30]. This communication works by exchanging data with roadside controllers, node base controllers, and other vehicles. Each vehicle will include the local agent controlled by SDN controllers in the roadside unit and base node stations.

(ii) *Roadside Controller (RSC)*. This is an SDN controller positioned at RSUs and used along the road to enable direct connections with vehicles. The direct connection offers high-level management of roadside controllers with fast data processing, better security mechanisms, and a data plan control. The RSCs work in a distributed way to share load balance, traffic management, and security insurance with the base station controller

(iii) *Evolved Node Base Controller (eNBC)*. This SDN controller is a central intelligence unit in VANET and is responsible for security enforcement, consistency establishment, traffic management, and policy control. The eNBC is an SDN integration with the evolved base stations

In the proposed architecture, we take advantage of the standard VANET architecture's use of different wireless technologies to resolve load balancing, cost, and security concerns. For the control plane, LTE/Wi-Fi is used. The eNBC controllers are connected via the Internet backbone. The down layer accommodates RSU controllers connected via the RSU backbone through Ethernet. Each vehicle is facilitated by both the DSRC interface and the long-term evolution (LTE) interface, which will only be used in cases of the absence of the RSUC coverage. Although the control plane requires low bandwidth and highly secure communication, while the data plane needs higher bandwidth and sufficiently secure communication, the proposed architecture is not

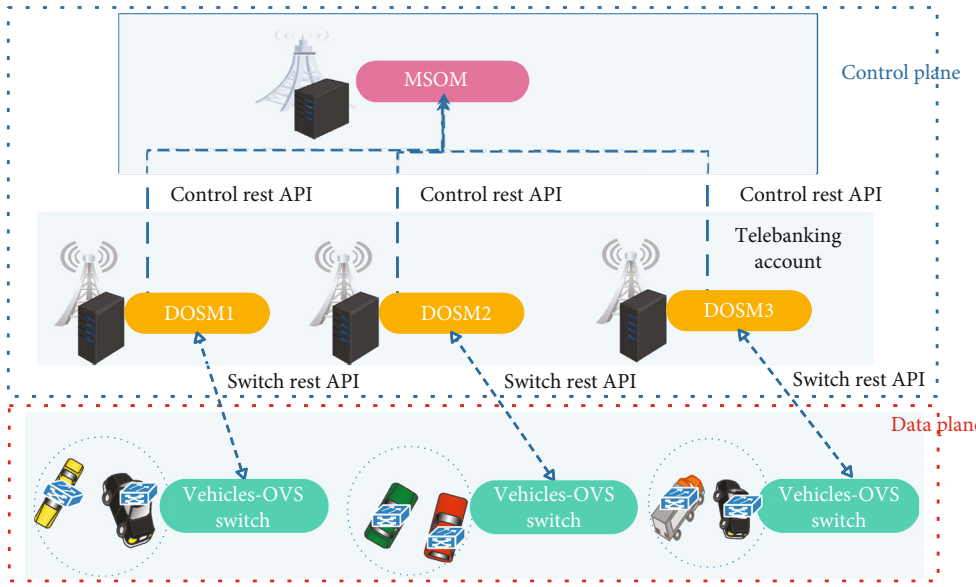

Figure 3: The proposed multidistributed SOM system.

restricted to these methods but can be used with any communication technologies.

Our design works by distributing the workflow among RSCs and vehicles' local agents while maintaining synchronization with a base controller [31]. Figure 3 shows the different communication technologies between the components of the designed architecture. The communication channels are as follows:

(i) *eNBC2RSC*. A channel that transfers control plane rules and installed policies

(ii) *Vehicles2RSC*. Two data plane channels between the vehicles and RSCs. The first and the second channels transfer the data and security rules/flow management, respectively

(iii) *eNBC2Vehicles*. A channel that connects the eNBC to the vehicles when the vehicles cannot directly connect to RSCs

(iv) V*ehicles2Vehicles*. A data plane channel between vehicles. The data exchanged between vehicles is controlled by flow rules and policies installed and administered by RSCs and the eNBC

To enable communication between the controller and other SDN planes, the architecture incorporates the following three interfaces: (1) southbound API, (2) northbound API, and (3) east/westbound API. The southbound API facilities the communication between the controller and the data plane entities based on one of the Application Interfaces (APIs), as the OpenFlow protocol is the standard protocol in the SDN environment that regulates the data communication between the data plane and the control plane via the interconnecting interfaces in the network. The controller can communicate with the application plane applications through the northbound interface (NBI), which mainly provides abstract network views and enables programming and network management. In a distributed SDN architecture, a third interface is required that would act as an interface between multiple controllers. To this end, east/westbound interfaces, which are specialized interfaces that work only in distributed SDN controllers, are used. Each controller should have its own east/westbound API to communicate with each other. Some of the main functions of these interfaces include transferring reachability information between controllers to facilitate inter-SDN routing, synchronizing among each other, monitoring/notification capabilities, getting up-to-date topological information, and coordinating the flow setup from the applications.

The proposed design ensures communication between the roadside controller and the vehicle in a strict way. The software-defined agent located in the vehicle links with the respective device using a v-port. Since the SDN agent has one ID, it can connect via Open v-switch. Considering that the controller includes the roadside controller and evolved NBC, the critical part of the architecture contains a security module placed on the controller side. The controller is accountable for addressing various issues, including the analysis of events and alarms.

3.2. The Proposed Multilayer Distributed SOM Algorithm. To detect and minimize DDoS attacks, the proposed system uses the MSOM algorithm. This algorithm is cast-off to generate a detailed description of the attack and its type, which can be detected through an anomaly detector. Figure 3 presents an overview of the proposed system consisting of MSOM and DSOM modules, and a vehicle switch located in the eNBC, RSCs, and vehicles. In our architecture, a DSOM module is integrated into an RSU controller by adding extension modules, and the MSOM module that controls the complete system operation is placed in the eNBC.

To facilitate independent performance upon detection, the Multilayer Distributed SOM algorithm is installed on the roadside and node base controllers. For VANET

communication media, network interruptions during overcrowding must be determined.

The methodology underlying the proposed system is based on a distributed multilayer self-organizing map constructed by a distributed SOM (DSOM) with a classification SOM (CSOM). The system is divided into the following six phases:

(i) *DSOM Training*. The process starts from training the DSOM, which is performed similarly to the SOM initialization outlined in Section 2. Each RSC is trained with the initial training dataset sent by the eNBC, and the outcome of this process is the input data organized into regions based on their similarity

(ii) *CSOM Layer*. The process continues with training SOM as supervised NN with a class label to improve the accuracy of our system. However, this time, the similarity function, the neighborhood update, and the class label of each cluster are already known from the DSOM. The drawback of the traditional SOM algorithm is that it has a standard deviation (SD) that distorts its performance. To overcome this limitation, we replace the equation of the Euclidean distance with Pearson squared regression [16]. This approach works well with the standard deviation (SD) and yielded excellent results. The Pearson squared distance computes the correlation between two profiles, as well as apprehends reversed relationships

$$\begin{aligned} d &= 1 - r, \\ r &= z(x).\frac{z(w)}{n}, \\ (x), z(w) &= Z - \text{Score } S \text{ tan dardization}. \end{aligned} \tag{3}$$

Z-score standardization, an extensively used statistical analysis technique, takes the difference between the field value and the field mean value and scales this difference by the standard deviation of the field values (see Equation (4)). Z-score is the most common method used to generate the SD to all attributes x and weight value w. Specifically, Z -score transformation reforms each attribute value to the standard scores with a zero mean and a unit.

$$\text{SD} = \frac{\sqrt{\sum x - (x - \bar{x})^2}}{n - 1}, \tag{4}$$

where mean

$$X_i = \frac{x - \bar{x}}{\text{SD}(x)}. \tag{5}$$

In addition to the set of elements in every sequence, w contains additional blocks to store the class counter $\{c\}z$. For this reason, it will be trained with CSOM which will be used in the training phase, where Z is the number of classes in the input data. To avoid significant differences between the used attributes, normalization makes all values equal, thereby enhancing the numerical precision of distances of each attribute's input dataset Xi and SD via subtracting the mean and dividing by the standard deviation for individual attribute

(iii) *Feedback*. After training the CSOM, the CSOM results are fed back into the DSOM to deduce the optimal classification features

(iv) *Merging (DSOM and CSOM)*. If each RSC was to be allowed to work independently, the variations in the mapping could become sufficiently large for each MSOM to generate disparate results even for the same input data. Toprevent such map deviation, the individual MSOMs are periodically consolidated into a single MSOM in a weighted sum method (see Equation (6)).

$$\text{MSOM} = \sum_{i=n}^{j=1} \frac{W_j}{\sum_{i=1}^{j=n} W_i} \times (\text{MSOM})_j \tag{6}$$

(v) *Updating*. After merging the MSOMs, the merged map is sent to the RSCs by the eNBC. To ensure that the classification continues at the switch, the existing DSOM map is substituted by the merged map

(vi) *Classification*. Finally, the MSOM is used to perform classification based on the inputs, and the output is transmitted to the controller for further processing

We set up our system as illustrated in Algorithm 1 (see below) and conduct a test of the DSOM & CSOM system.

3.3. The Proposed SDN Controller Architecture. The SDN controller at the eNBC level does not retain full control; instead, it can deputize control of traffic management and control details to other RSCs and local agents in the vehicles. In this way, traffic control and security are shared between all SDN controllers.

The proposed architecture was designed to fulfill the requirements of an adequate DDoS defense system with multiple levels. Further detail is provided below.

3.3.1. eNBC Level. On the eNBC level, control of our scheme incorporates both the control plane and the application layer. The eNBC controller is assisted by the OpenFlow controller module that runs applications in the control plane. The proposed architecture is divided into the MSOM module, eNBC agent, global database, and security module (Figure 4). The northbound interface abstraction layer (i.e., REST API) is in charge of maintaining an open communication channel between the OpenFlow controller and the eNBC agent. Furthermore, the MSOM module is responsible for the unified MSOM map creation process, while the eNBC agent controls communication among all modules. Since our architecture contains different types of connections, each module employs separate socket connections. Additionally, the agent also manages the policy module and the mitigation engine,

1 ***while*** *(True)* ***do***
2 *Randomly initialize DSOM weights* w_{ij}
3 *Sequentially input data from initial dataset* X_i
4 ***while*** *(criterion DSOM)* ***do***
5 *Compute the Euclidean distance between* X_i *and* W_{ij}
6 *Update the weights of the winning neuron and its neighborhood of BMU*
7 *Input into CSOM*
8 *Compute the Pearson squared distance between* X_i *and* W_{ij} *with* C_i
9 *Start training*
10 *Merge DSOMs sent from all RSCs to eNBC*
11 *Distribute the merged map (DSOM) to all RSCs*
12 *Classify incoming traffic as normal or attack*

Algorithm 1: Multilayer Distributed SOM.

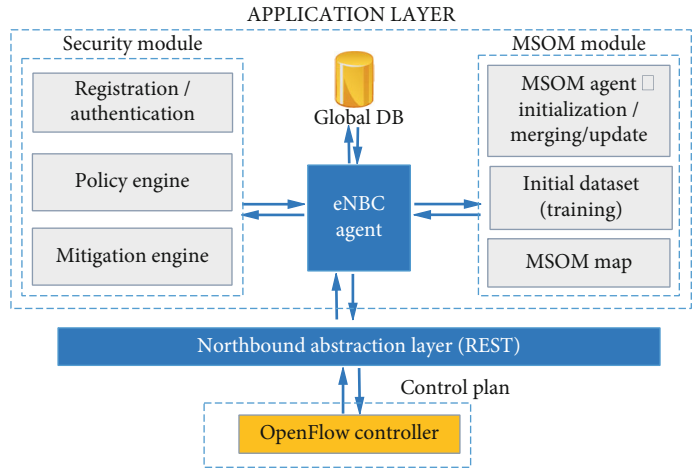

Figure 4: The evolved node base station SDN controller architecture (eBNC).

while the MSOM engine manages and executes the steps of initialization, merging, and updating discussed in Section 2.

During the initialization step, the MSOM module retrieves the information of the registered RSCs and vehicles stored in the eNBC's global database via an eNBC agent. The agent controls the MSOM steps performed in the eNBC, as well as manages the initial dataset and map. Upon completion of the RSC registration process at the authentication and registration unit, the MSOM agent sends the initial training dataset to all registered RSCs. Finally, the initialization step is completed at the eNBC after all the RSCs received the initial training dataset.

The MSOM merging process starts with the MSOM agent messaging the registered RSCs to collect the DSOM maps, and all DSOM map information received by the eNBC agent is immediately sent to the MSOM agent. After verification of all information from the RCSs with the help of the security module, the MSOM agent stacks the resulting data for subsequent merging. Upon completion of the verification procedure, the MSOM agent starts the merging process described above. Subsequently, the update procedure is started by sending the merged map to all registered RSCs and updating the MSOM map managed by the MSOM module.

3.3.2. The RSC Level. The RSC architecture has the same modules (MSOM and security), but with different functions. Figure 5 shows the RSC architecture components and the communication with OpenFlow switches. The RSC agent orchestrates the functions and manages the workflow and communication both among the modules themselves and with other parties within the system.

The MSOM module in the RSC level is responsible for the initialization, feedback, and classification steps. The training process starts after all registered RSCs receive the initial dataset sent by the eNBC. Then, the RSC agent forwards the dataset to the MSOM module to save it in the initial database, which then becomes the input for the initialization step. The MSOM agent creates the map by training the initial dataset using the steps described in the proposed algorithm, including initialization, CSOM, and feedback. Upon creation of the map, the MSOM will save it in the MSOM database. Moreover, the eNBC will update the map by the merged map, a new periodic call for the training procedure, or any update that occurs during the feature extractor procedure. This periodic training ensures that the map always adapts to any new incoming traffic and advances each local MSOM's RCS classification performance.

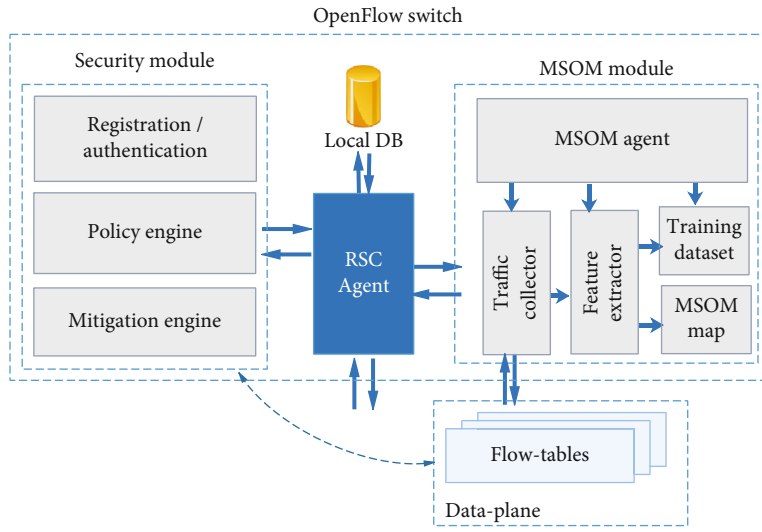

Figure 5: Roadside unit SDN controller architecture (RSC).

As discussed previously, the main aim of the present study is to evaluate the feasibility of using an MSOM against DDoS flooding attacks. Overall, the packet information plays an important role in the DDoS detection process. The traffic collector forwards all packets sent by the vehicles registered with a single RSU to the feature extractor that, in turn, extracts each packet feature for every IP address source. To classify the traffic, the feature extractor uses the following six characteristics:

(i) *Number of Flows*. An ICMP flooding attack uses only one flow, while a TCP flooding attack can use an almost unlimited number of flows

(ii) *Number of Packets/Flow*. An ICMP attack uses a massive number of packets per flow, while a TCP attack uses only several (1 to 3) packets per flow

(iii) *Number of Bytes/Packet*. An ICMP attack leads to a large number of bytes per flow, while a TCP attack uses a reasonable small number of bytes per flow

(iv) *Protocol*. Protocol is a characteristic feature that performs a fundamental function in the classification of the type of the flooding. Different protocols can be represented as (unique) integers, e.g., TCP 1 and ICMP 0

(v) *Client Ports' Growth*. The attacker port number does not change during an ICMP attack; however, a TCP attack yields many service ports where the attacker uses multiple ports to send SYN packets

(vi) *Duration*. In an ICMP attack, the attackers are connected to the server for a considerably longer time than in a TCP attack

The classification step forms input vectors using the features obtained in the extraction step. Similar to the standard SOM, the MSOM map classifies traffic into malicious and nonmalicious. In order to make the final determination, we introduce a clustering algorithm to partition n patterns into K groups. When an attack is classified, the MSOM agent forwards the attack information to the security module via the RSC agent. Further detail is provided in Section 3.3.2.

3.3.3. Security Module. The security module distributes the workload between the multiple controller layers. The distributed controller layer improves the reliability and scalability of the proposed solution to match the needs of VANET-based SDN networks. The proposed security module workflow and its components are as follows:

(i) *Registration and Authentication*. In the proposed system, the eNBC is considered as the trusted component and the central authentication party. In addition, the RSU must be registered and authenticated by the eNBC for one time. After its successful registration, each RSC will initially grant a security certification, and this certificate will be saved in the local database for RSCs. However, the vehicle's registration consists of the following two phases. In the first phase, each vehicle attempts to join the system by sending a registration request to the nearest RSC. In the second phase, unless the vehicle is already known, the received request is escalated to the eNBC. Once a vehicle is authenticated by the eNBC, it is granted a certificate and an ID that will be saved in the global eNBC database, local RSC database, and the car itself. All certified cars will be assigned a key by RSC or eNBC each time they connect to the system

(ii) *Policy and Mitigation Engine*. While the policy engine defines attack detection and mitigation policy, policy and mitigation engines determine the defense strategy. Our design supports several types of mitigation plans, such as dropping packets, traffic redirection, and blocking ports. The policy and the mitigation engines apply different rules within the different components

The policy engine in the RSC makes local policy decisions on traffic flows and controlling policies installed by the eNBC. The policy engine communicates with the global

policy engine at the eNBC to receive policies via REST API. In order to manage all incoming packets collected by the traffic collector, the policy engine directly inserts rules into the flow tables. In the event that an attack is detected, or if changes to the old policy are requested via a REST API command, the global policy engine will update all RSCs, the vehicle policy engine, and the flow table. The global policy engine feeds the RSCs and vehicles with new rules and provides updates of all rule violations. Whenever an attack is detected by one or more RSCs, the eNBC policy engine sends a REST API command with the new policy, and the policy module makes the decision for the whole network based on the global view of the network.

The mitigation module in the RSCs works with the security module in the eNBC. This process starts after the security module receives an attack alarm sent by the MSOM agent. Upon successful verification, the mitigation engine module informs the policy engine module to locally update the policies and forwards the attack definitions to eNBC. The rules are pushed to the vehicles via PUSH-based REST API. The vehicles' flow table is updated based on the flow rules provided by the RCS. The traceback procedure is activated on the RSC level by inserting the vehicle source ID of the spoofed packets that do not match the installed rules to a signature IP.

Depending on the attack severity information extracted from the alert message, the mitigation module includes the following three filters: (1) LR (least reactive), (2) IR (intermediate reactive), and (3) HR (high reactive) filter. These filters are used to defend against the attacks by blocking ports, dropping packets, and reducing the traffic sent between any two systems. Once the mitigation policy is successfully installed, tremendous volumes of malicious flow entries may be created by the attack. Since this flow of malicious data can consume storage space, upon performing mitigation action against the attack traffic, the attacker flow entries will be deleted to clear the occupied storage space in the vehicle agent switch.

4. Performance Evaluation

This section reports the results of the experiments described in Section 3 and demonstrates the effectiveness of the proposed method.

As concerns the training datasets used for the multilayer distributed SOM, initially, the MSOM was trained using the datasets of DDoS attacks and normal traffic. The DDoS-attack training sets were acquired from the following three datasets: (1) CAIDA-Dataset 2015 and 2007 [32], (2) NSL-KDD Dataset for Network-based Intrusion Detection Systems 2009 [33], and (3) LANDER DARPA 2009 Intrusion Detection [34].

In addition, to generate DDoS traffic attacks in the test bed, BoNeSi [35], a flooding tool, and a DDoS Botnet Simulator were used. We examined two types of attacks: (1) the TCP flooding attack and (2) the ICMP flooding attack. For the implementation of the proposed approach, the BoNeSi flooding attack tool offers an efficient way to examine our solution's performance against the DDoS attack traffic.

Table 2: Training and testing sample parameters.

Approach	Init.	Training	Testing datasets		
			CAIDA	NSL-KDD	DARPA
SOM	4000	6000	3000	3000	3000
DSOM	4000	6000	3000	3000	3000
MSOM	4000	6000	300	3000	3000

4.1. Experiments. To prove the efficiency and accuracy of the logic in the proposed architecture and evaluate its performance, we first independently implemented and tested each module. Second, we tested the connectivity layers between the RSCs and the agent from one side, and the MSOM module and the eNBC from the other side. Furthermore, to simulate the controller on the RSC and eNBC, we implemented the MSOM modules as an application in the SDN controller. The setup was run natively on a Linux machine with Ubuntu 15.04 in a virtual machine using VirtualBox. In this experiment, the SDN controller that we used was Floodlight. We also used Mininet-Wi-Fi [33], the standard network emulation tool for SDN with wireless support, as the network emulator. A customized Floodlight controller in RSC and eNBC and Open Virtual Switch (OVS) [30] was used for the vehicles.

Finally, the architecture was examined with the NS-3 simulator. To evaluate the performance, we considered a 4-junction road. The simulation area spread-over was 1000×1000 meters. In general, the vehicle can initiate a request for its attentive data. However, in our simulation, we used 60 vehicles configured with SDN switches. The vehicles were located randomly within the margins, 7 RSCs, and one NBC was located in the middle of the architecture area to connect with all RSCs. The time for the simulation was set at 100 seconds, and the number of attackers varied from 5 to 40. Optimized Link State Routing (OLSR) [36], which is widely applied in ad hoc networks [37–39], was used. Many features of OSLR—including short packet transmission delay, fast adjusting of topology changes, simple operations, and easy integration with different kinds of devices—make its use optimal for VANET. Furthermore, many previous studies showed the superiority of the OLSR over other routing protocols [40–42]. It should be noted, however, that our solution can be directly used with any other protocol. The simulation parameters and configuration used for evaluation are summarized in Table 1.

Next, we evaluated the performance of the proposed model as compared to single SOM and DSOM in terms of DDoS attack detection and mitigation. A SOM filter was implemented at the eNBC, and all vehicle traffic was forwarded to the controller for further analysis as per the described standard SOM using the flow collector, feature extractor, SOM map, and policy engine functions. In the DSOM scenario, all RSC trainings were first mapped and then merged by the eNBC and then sent back to all RSCs. The last scenario was our proposed MSOM that included all aforementioned steps and functionality. In all scenarios, we used the same dataset parameters (see Table 2). The BoNeSi tool produced different attack levels with traffic rates (50, 100, 200, and 300 Mbps) in all scenarios.

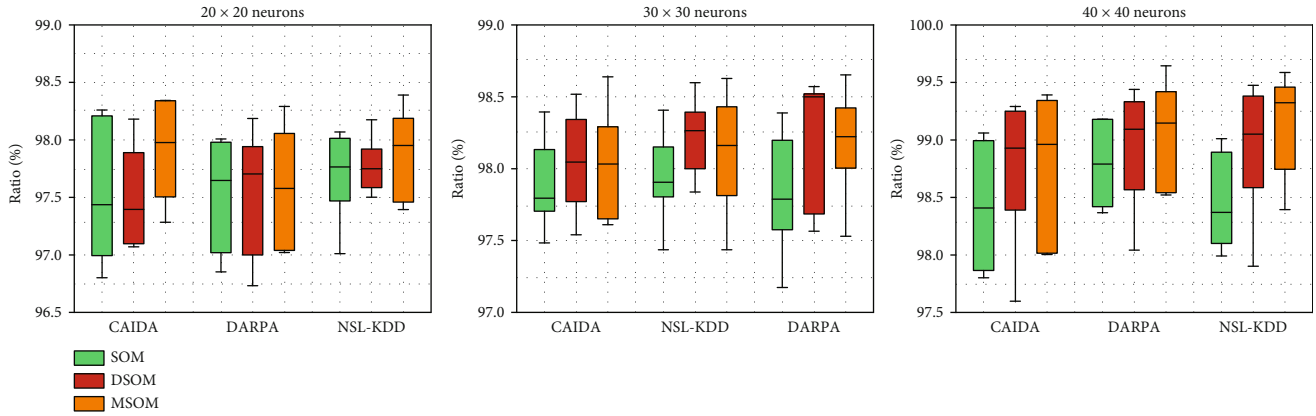

Figure 6: The detection rate in classifying abnormal traffic with three different neuron numbers and three datasets.

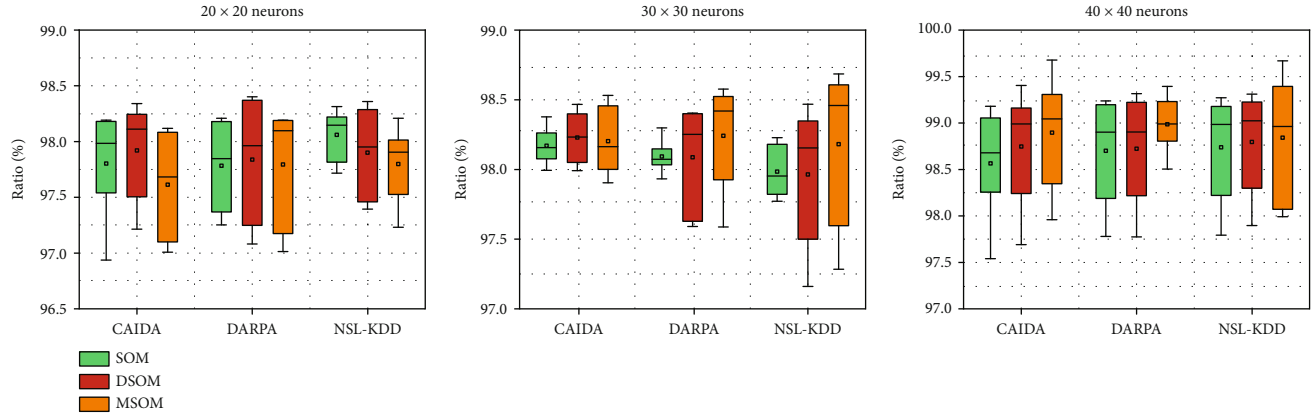

Figure 7: The accuracy of the SOM maps with three different neuron numbers and three datasets.

The mechanism revealed an interesting property if the input patterns were not uniformly distributed over the input space. Specifically, the MSOM on the second layer was a supervised mechanism and distributed more neural units where the input patterns had a higher density—an effect known as the magnification factor. The first layer was an unsupervised mechanism; the network passed the input pattern if they were more frequent; this trend was unaffected by the density of the input patterns.

We focused on the vehicle to infrastructure communication—specifically, on the traffic from vehicles and RSU. Since the proposed solution does not prioritize the use of any specific protocol, it is equally applicable to all protocols. Various data packets were transmitted in vehicular networks, such as position, average speed, and road condition. Since the vehicles may not have had the same features and applications and may have transmitted various packets via the network, we did not define the type of data for our detection and mitigation model for the sake of generality.

The initialization, training, and testing processes were performed for SOM, DSOM, and MSOM. A total of 4000 samples from each dataset (NSL-KDD, CAIDA, and DARPA) were used for the initialization process, while 6000 samples were used for training. The testing process was conducted with 3000 samples from each dataset and 6000 samples from the BoNeSi tool. We ran the testing procedures multiple times with different neuron patterns in the SOM map (20×20, 30×30, and 40×40).

In addition, seeking to test the efficiency of the proposed method not only with different datasets but also with other classifiers, we ran an additional test to compare the proposed classifier detection with other classifiers [43–45] (Naive, Bayesian, ML-Perceptron, and C4.5, respectively).

Accuracy was defined as the number of correctly classified cases as compared to the total number of cases presented to the system. Detection rate (DR) was computed as the percentage of True Positives (TP) as compared to the total number of cases classified as positive events. The detection rates and accuracy were computed as shown in Equations (7) and (8).

$$\text{Accuracy} = \text{TP} + \text{TNTP}{+}{+}\text{TNFP} + \text{FN} \times 100, \tag{7}$$

$$\text{DR} = \text{TPTP} + \text{FN}, \tag{8}$$

where TN are true negatives, FP are false positives, and FN are false negatives. Finally, the SDN VANET architecture performance with the proposed method and SOM and DSOM was measured in terms of throughput, average end-to-end delay, and packet delivery ratio (PDR).

Table 3: Comparing training and testing procedures.

No. of neurons		Single SOM	DSOM	MSOM
Map (20 ∗ 20)	Classification (ms)	.963 s	.920 s	.919 s
	Processing (s)	8.21	7.22	7.11
Map (30 ∗ 30)	Classification (ms)	1.93	1.82	1.74
	Processing (s)	17.78	14.4	12.9
Map (40 ∗ 40)	Classification (ms)	5.23	4.09	3.44
	Processing (s)	32.44	29.31	26.22

5. Results

Figure 6 shows the detection rate of the three models with three different datasets. According to the results, due to the variation in the distribution for each dataset, both the MSOM model and the DSOM model showed different detection rates. Moreover, in our calculation of the precision, which is a standard classification measure to indicate false positives, the proposed model achieved 0.9924, SOM 0.9135, and DSOM 0.9414. Experimental results also showed that a higher number of neurons in the model results in a stronger detection performance.

Figure 7 shows that the accuracy of the three tested models always tended towards high ratios. While, in the case of 400 neurons, the lowest rate of accuracy was around 98.5%, in the case of 1600 neurons, it was 99.62%. The results also suggested that the detection rate with the MSOM model was higher than with both the SOM and the DSOM models. This finding can be explained by the fact that the proposed model has more layers, which results in a stronger feature learning ability as compared to the other tested models; in addition, the proposed model can obtain an extensive feature presentation and higher detection ability.

In testing the effectiveness of the proposed method, the classifier plays the primary role in an artificial neural network. Accordingly, we experimentally compared the detection and accuracy results of MSOM with the existing techniques using the NSL-KDD dataset (Table 3).

As suggested by the results shown in Table 3, the proposed model yielded a remarkably better performance in terms of detection rates and accuracy as compared to those of other classifiers. Specifically, both detection rate and accuracy achieved with the proposed model were higher than those attained using the other methods. This finding can be explained by the fact that the MSOM maps in the MSOM agents were individually trained by different local RSU traffic. With a fixed and limited number of neurons in the agent, or when there are many traffic types trained for an MSOM map or several merging times, the weights of each neuron in the map will change considerably. Moreover, the attack is detected by different RSCs either at the same time or with different time margins. All these reasons may have led to the variance in detection rates and accuracy achieved using the three methods.

Table 4: Processing and classification time.

Classifier	Training	Testing	Detection rate (%)	Accuracy (%)
Naive Bayesian	2000	3000	88.80	92
ML-Perceptron	2000	3000	92	94
C4.5	2000	3000	85	87.50
MSOM	2000	3000	99.10	99.67

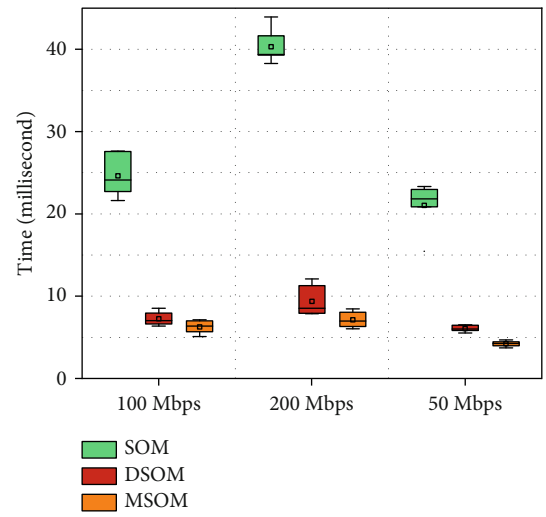

Figure 8: Reaction time to various attack levels.

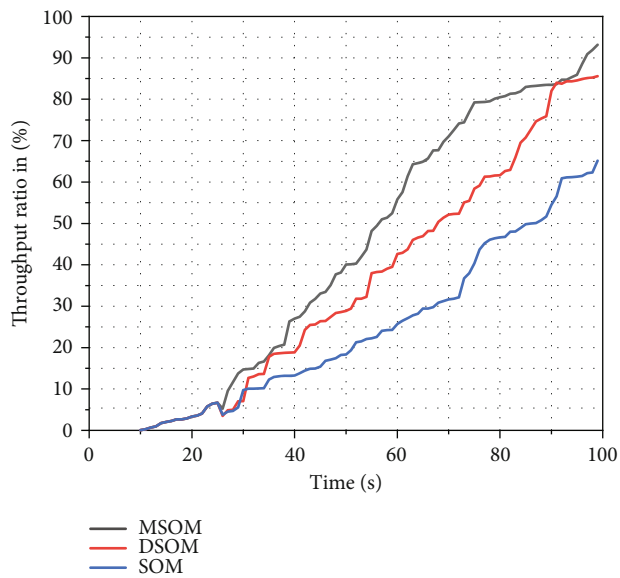

Figure 9: Throughput comparison.

As can be seen in Table 4, the MSOM and DSOM achieved faster processing times, specifically in the case of large maps. Whereas, due to the training process using a huge input dataset, the single SOM took more time, both MSOM and DSOM were trained with the initial dataset sent by the central controller. Moreover, the MSOM achieved the shortest classification with the three different map sizes. This result can be attributed to the extra layer (CSOM) of MSOM

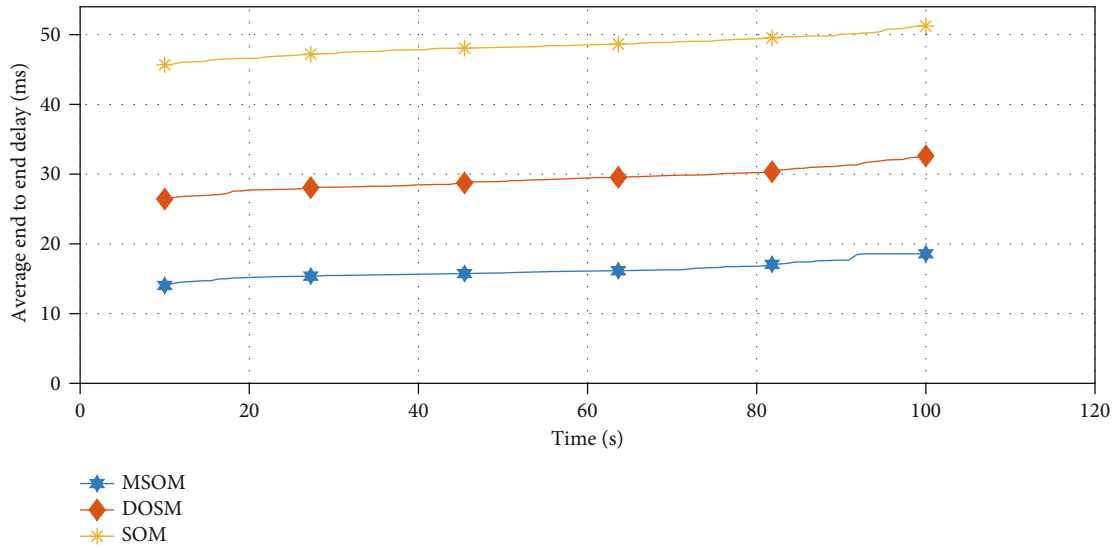

Figure 10: Average end-to-end delay comparison.

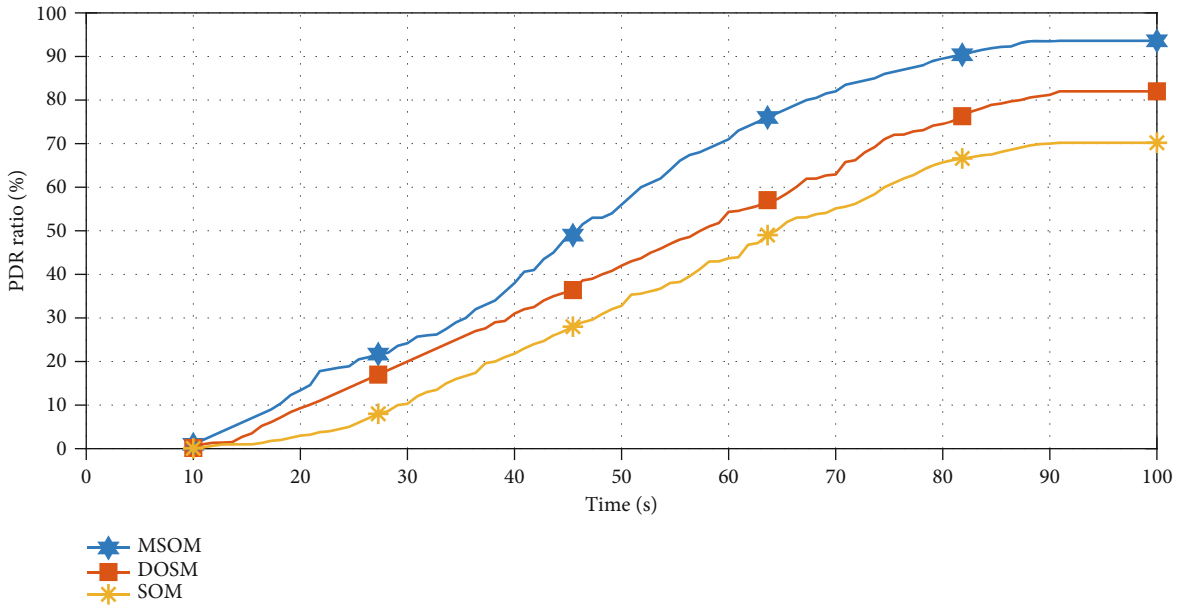

Figure 11: Packet delivery ratio comparison.

that increases the clustered profiles, which results in fewer iterations and a stronger feature learning capability that, in turn, enhances feature recognition.

The next evaluation criterion we used to evaluate our system and to compare its performance with that of other methods was attack reaction time. Figure 8 shows three different attack traffic levels (50 Mbps, 100 Mbps, and 200 Mbps). The results suggest that, in general, the proposed method yielded the best results in all test scenarios. In the single SOM scenario where the RSUs had to send all traffic to the single SOM for attack examination, there was a substantial delay due to the centralized point. Then, policies were sent back to the RSU, and the new policies were forwarded to vehicles to install a flow rule in the switch flow table. The DSOM partially suffered from a similar delay when an attack was detected by the RSU and then forwarded the SOM map to the eNBC. The eNBC responded with new rules to handle the attack with the new map. Accordingly, since the policy engine would immediately react to the detected attack, the time needed for the MSOM framework solution to react to attack patterns was shorter for all traffic levels. When needed, the RSU policy engine sent the new rules to the vehicles and attack information to the eNBC.

Furthermore, the results showed that the MSOM system with the proposed architecture succeeded in outperforming other models in terms of throughput, average delay, and packet delivery ratio at all times. As can be seen in Figure 9, the MSOM system consistently outperformed the DSOM and SOM approaches. This significant throughput improvement observed in the MOSM method was due to its

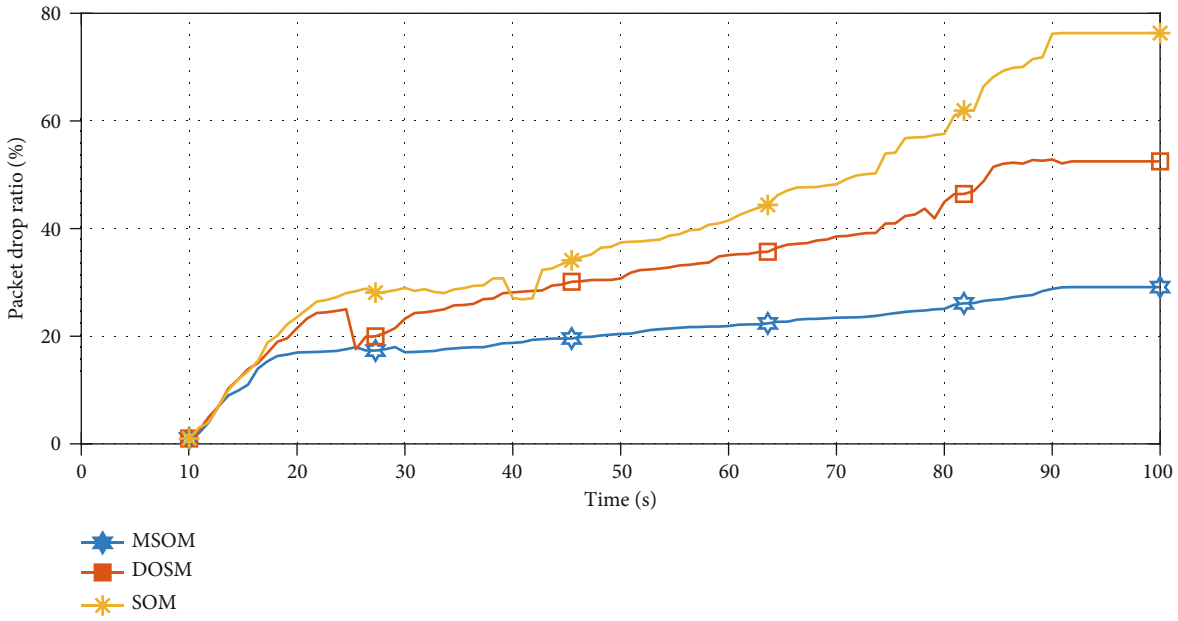

Figure 12: Packet drop ratio comparison.

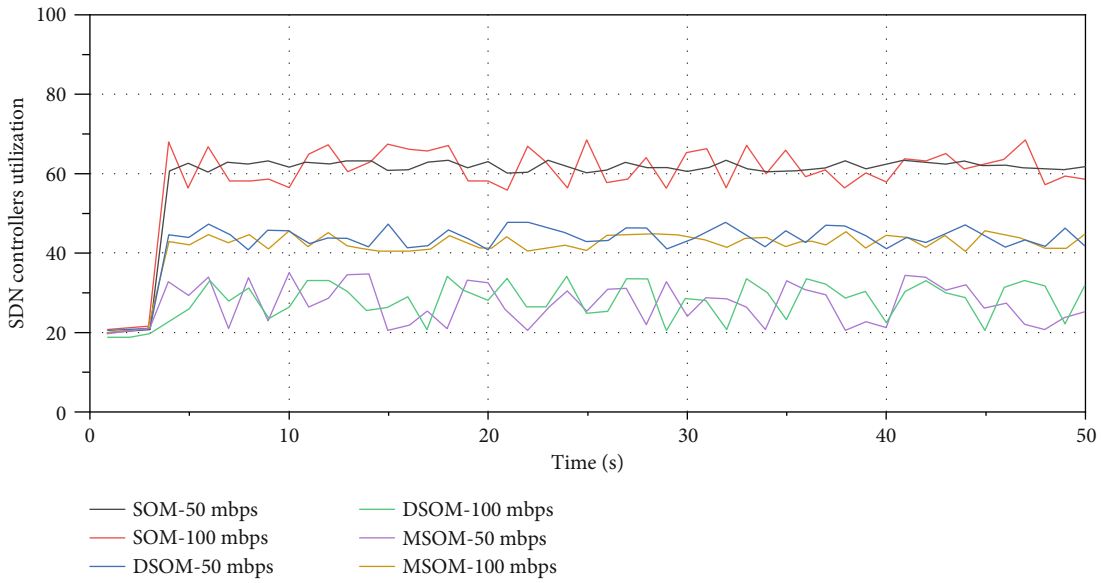

Figure 13: CPU utilization in controllers under DDoS attacks.

simplicity and effectiveness in detecting DDoS attacks. Furthermore, with regard to the correlation between performance of the tested approaches and time, Figure 10 shows that delay range in our method was 15-20 ms, which can be practiced in all the VANET applications with significant QoS. This result suggests that, owing to faster communication and more accurate route information, the proposed architecture minimizes the delay. We also examined the packet delivery ratio (PDR), one of the major metrics for any network model, which is the ratio between the total number of sent and received packets. In this respect, the MSOM model was found to have a higher PDR than the DOSM and SOM methods (Figure 11), which can be attributed to a higher accuracy of MSOM in detecting malicious nodes and eliminating their traffic via blocking or dropping mechanisms. This allows the proposed method to drop redundant routing packets on the network, which results in a higher PDR stability [46]. Finally, stability in delay and increase in PDR increase throughput.

Furthermore, we also found that MSOM consistently outperformed DSOM and SOM models in terms of the packet drop ratio as a function of time (Figure 12), which arose as a result of MSOM's capacity to offer faster communication and more accurate route information to the vehicles, which minimizes the delay. Furthermore, the fast and efficiency of detecting the attack and blocking the source of the

huge amount of traffic empowered the MSOM model to achieve a lower dropping ratio as compared to that of DSOM and SOM that required more time to detect the attack and prohibit the attack source. Of note, however, the performance of all three evaluated methods improved with time. On the other hand, packet drop can occur when a vehicle gets far from the neighbor vehicle or the RSUs.

Finally, we examined the overall CPU resource consumption of the proposed system in the case of DDoS attacks. To this end, we measured the CPU usage of all controllers and calculated the average system resource. The CPU usage of MSOM, DSOM, and SOM was tested for 50, 100, and 200 Mbps (see Figure 13). As shown by the results, due to the centralization mechanism, the DSOM method consumes the resources in all cases. The single SOM needs to communicate with all vehicles to collect traffic and process load, all of which is done in a single controller. Furthermore, in the MSOM and DSOM schemes, the traffic collection and the process load are distributed between all controllers. Accordingly, each controller is limited to collecting and processing a rather small amount of local traffic. In contrast, to the multilayer scheme that provides a stronger feature learning ability, more in-depth feature, and less iteration the MSOM yielded better results as compared with DSOM. All aforementioned factors also led to less processing time and lower resource consumption in the proposed system.

In summary, based on the results reported in this section, we can conclude that the proposed multilayer distributed SOM method outperforms the single SOM and the distributed SOM model in a VANET-based SDN network and efficiently solves the problems associated with DDoS flooding attacks. Specifically, we found that, by showing the clustering between attacks, the MSOM can efficiently classify the DDoS network attack (as contrasted to the normal traffic) in a graphic way. In the experimental results on both single and distributed SOM, we observed that using MSOM enabled effective detection of the attack mechanism.

6. Conclusion

In this paper, aiming to address security issues caused by flooding attacks using a distributed system, we proposed a multilayer distributed self-organizing map as a DDoS prevention system for VANET-based Software-Defined Networks. To test the effectiveness of the proposed system, we conducted our experiments with three datasets and used several evaluation tests. Experimental results showed that, due to the efficient adaptation to local traffic in the MSOM system, it had a rapid system reaction to attacks which, in turn, enhanced the detection rate and accuracy (to 99.10 and 99.67, respectively). Overall, the SDN capabilities offer many benefits, such as the decoupling of the data plan and control, which makes it possible to use different communication channels. Furthermore, the programmability feature of the proposed system enables users to implement the security module in the DDoS detection method. In addition, easy management feature enables efficient construction of appropriate mitigation and recovery mechanisms. We will report on the SDN communication and policy details in our future work. Moreover, the distributed architecture and security modules of the proposed system make it possible to avoid the single point of failure occurring in DDoS attacks, as well as enable a reduction of resource consumption with the concerned CPU usage as compared to other methods. In summary, the MSOM proposed and tested in the present study can be concluded to an efficient and feasible security framework for a VANET-SDN-based environment.

Data Availability

The network traffic data used to support the findings of this study are available from the corresponding author upon request. The project is ongoing for further work expansion.

Conflicts of Interest

The authors declare no conflicts of interest.

References

[1] S. Al-Sultan, M. M. Al-Doori, A. H. Al-Bayatti, and H. Zedan, "A comprehensive survey on vehicular ad hoc network," *Journal of Network and Computer Applications*, vol. 37, pp. 380–392, 2014.

[2] H. Khelifi, S. Luo, B. Nour, and C. S. Shah, "Security and privacy issues in vehicular named data networks: an overview," *Mobile Information Systems*, vol. 2018, Article ID 5672154, 11 pages, 2018.

[3] M. Arif, G. Wang, O. Geman et al., "SDN-based VANETs, security attacks, applications, and challenges," *Applied Sciences*, vol. 10, no. 9, p. 3217, 2020.

[4] T. Huang, F. R. Yu, C. Zhang, J. Liu, J. Zhang, and Y. Liu, "A survey on large-scale software defined networking (SDN) testbeds: approaches and challenges," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 2, pp. 891–917, 2017.

[5] C. Sun, J. Bi, H. Chen et al., "Sdpa: toward a stateful data plane in software-defined networking," *IEEE/ACM Transactions on Networking*, vol. 25, no. 6, pp. 3294–3308, 2017.

[6] S. Jain, A. Kumar, S. Mandal et al., "B4," *ACM SIGCOMM Computer 13 Communication Review*, vol. 43, no. 4, pp. 3–14, 2013.

[7] T. M. Nam, P. H. Phong, T. D. Khoa et al., "Self-organizing map-based approaches in ddos flooding detection using sdn," in *2018 International Conference on Information Networking (ICOIN)*, pp. 249–254, Chiang Mai, Thailand, 2018.

[8] K. Adhikary, S. Bhushan, S. Kumar, and K. Dutta, "Hybrid algorithm to detect DDoS attacks in VANETs," *Wireless Personal Communications*, vol. 114, no. 4, pp. 3613–3634, 2020.

[9] D. Miljkovic, "Brief review of self-organizing maps," in *40th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO 2017)*, Opatija, Croatia, 2017.

[10] K. Choksi, B. Shah, and O. Kale, "Intrusion detection system using self organizing map: a surevey," *International Journal of Engineering Research and Applications*, vol. 4, no. 12, pp. 11–16, 2014.

[11] K. Liu, J. K. Ng, V. Lee, S. H. Son, and I. Stojmenovic, "Cooperative data scheduling in hybrid vehicular ad hoc networks: Vanet as a software defined network," *IEEE/ACM Transactions on Networking*, vol. 24, no. 3, pp. 1759–1773, 2016.

[12] M. Kalinin, P. Zegzhda, D. Zegzhda, Y. Vasiliev, and V. Belenko, "Software defined security for vehicular ad hoc networks," in *Information and Communication Technology Convergence (ICTC)*, pp. 533–537, Jeju, South Korea, 2016.

[13] G. de Biasi, L. F. Vieira, and A. A. Loureiro, "Sentinel: defense mechanism against ddos flooding attack in software defined vehicular network," in *2018 IEEE International Conference on Communications (ICC)*, pp. 1–6, Kansas City, MO, USA, 2018.

[14] Y. Xu and Y. Liu, "Ddos attack detection under sdn context," in *INFOCOM 2016-the 35th Annual IEEE International Conference on Computer Communications*, pp. 1–9, San Francisco, CA, USA, 2016.

[15] R. Kloti, V. Kotronis, and P. Smith, "Openflow: a security analysis," in *Network Protocols (ICNP)*, pp. 1–6, Goettingen, Germany, 2013.

[16] G. Shang, P. Zhe, X. Bin, H. Aiqun, and R. Kui, "Flooddefender: protecting data and control plane resources under sdn-aimed dos attacks," in *INFOCOM 2017-IEEE Conference on Computer Communications*, pp. 1–9, Atlanta, GA, USA, 2017.

[17] K. Giotis, C. Argyropoulos, G. Androulidakis, D. Kalogeras, and V. Maglaris, "Combining openflow and sflow for an effective and scalable anomaly detection and mitigation mechanism on sdn environments," *Computer Networks*, vol. 62, pp. 122–136, 2014.

[18] X. Liu, G. Yan, D. B. Rawat, and S. Deng, "Data mining intrusion detection in vehicular ad hoc network," *IEICE Transactions on Information and Systems*, vol. E97.D, no. 7, pp. 1719–1726, 2014.

[19] Q. Yan, Q. Gong, and F. R. Yu, "Effective software-defined networking controller scheduling method to mitigate DDoS attacks," *Electronics Letters*, vol. 53, no. 7, pp. 469–471, 2017.

[20] A. Sinha and S. K. Mishra, "Preventing VANET from DOS & DDOS attack," *International Journal of Engineering Trends and Technology*, vol. 4, 2013.

[21] A. L. Toledo and X. Wang, "Robust detection of MAC layer denial-of-service attacks in CSMA/CA wireless networks," *IEEE Transactions on Information Forensics and Security*, vol. 3, no. 3, pp. 347–358, 2008.

[22] R. Kolandaisamy, R. M. Noor, I. Ahmedy et al., "A multivariant stream analysis approach to detect and mitigate DDoS attacks in vehicular ad hoc networks," *Wireless Communications and Mobile Computing*, vol. 2018, Article ID 2874509, 13 pages, 2018.

[23] R. Kolandaisamy, R. M. Noor, I. Kolandaisamy et al., "A stream position performance analysis model based on DDoS attack detection for cluster-based routing in VANET," *Journal of Ambient Intelligence and Humanized Computing*, pp. 1–14, 2020.

[24] S. Zhang, C. Fung, S. Huang, Z. Luan, and D. Qian, "Psom: periodic self-organizing maps for unsupervised anomaly detection in periodic time series," in *Quality of Service (IWQoS), 2017 IEEE/ACM 25th International Symposium on, IEEE*, pp. 1–6, Vilanova i la Geltru, Spain, 2017.

[25] S.-Y. Huang and Y.-N. Huang, "Network dependable systems and networks (DSN)," in *2013 43rd annual IEEE/IFIP international conference on, IEEE*, pp. 1-2, Budapest, 2013.

[26] D. J. Deng, S. Y. Lien, C. C. Lin, S. C. Hung, and W. B. Chen, "Latency control in software-defined mobile-edge vehicular networking," *IEEE Communications Magazine*, vol. 55, no. 8, pp. 87–93, 2017.

[27] H. Li, M. Dong, and K. Ota, "Control plane optimization in software-defined vehicular ad hoc networks," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 10, pp. 7895–7904, 2016.

[28] M. Yu, J. Rexford, M. J. Freedman, and J. Wang, "Scalable flow-based networking with DIFANE," *ACM SIGCOMM Computer Communication Review,*, vol. 40, no. 4, pp. 351–362, 2010.

[29] M. A. Salahuddin, A. Al-Fuqaha, and M. Guizani, "Software-defined networking for RSU clouds in support of the internet of vehicles," *IEEE Internet of Things Journal*, vol. 2, no. 2, pp. 133–144, 2015.

[30] Openvswitch, "Openvswitch version 2.3.0, (2.3.0)," http://openvswitch.org.

[31] A. Hawbani, E. Torbosh, W. Xingfu, P. Sincak, L. Zhao, and A. Y. Al-Dubai, "Fuzzy based distributed protocol for vehicle to vehicle communication," *IEEE Transactions on Fuzzy Systems*, 2019.

[32] "Caida dataset," 2015, https://data.caida.org/datasets/passive–2015.

[33] "Caida datasetnsl-kdd data set for networkbased intrusion detection systems," 2009, http://iscx.cs.unb.ca/NSL-KDD/CAIDADataset.DDoSAttack2007.

[34] "Nsl-kdd data set for network-based intrusion detection systems," 2009, https://ant.isi.edu/datasets/readmes/DARPA-2009-DDoS-attack-20091105.README.txt.

[35] Bonesi, "The ddos botnet simulator," https://github.com/Markus-go/bonesi.

[36] T. Clausen and P. Jacquet, "Optimized link state routing protocol (OLSR)," 2003, IETF RFC 3626, United States, http://www.ietf.org/rfc/rfc3626.txt.

[37] R. Mehra and R. Dudeja, "Secure olsr routing protocol based on hash chain for efficient clustering in vanet," *Journal of Natural Remedies*, vol. 21, no. 2, pp. 113–120, 2020.

[38] K. Prakash, P. C. Philip, R. Paulus, and A. Kumar, "A packet fluctuation-based OLSR and efficient parameters-based OLSR," in *Recent Trends in Communication and Intelligent Systems*, p. 79, Proceedings of ICRTCIS 2019, 2020.

[39] M. Usha and B. Ramakrishnan, "An enhanced MPR OLSR protocol for efficient node selection process in cognitive radio based VANET," *Wireless Personal Communications*, vol. 106, no. 2, pp. 763–787, 2019.

[40] E. Spaho, M. Ikeda, L. Barolli, F. Xhafa, M. Younas, and M. Takizawa, "Performance of olsr and dsdv protocols in a vanet scenario: evaluation using cavenet and ns3," in *2012 seventh international conference on broadband, wireless computing, communication and applications. IEEE*, Victoria, BC, Canada, 2012.

[41] T. S. Chouhan and R. S. Deshmukh, "Analysis of DSDV, OLSR and AODV routing protocols in VANETS scenario: using NS3," in *2015 International Conference on Computational Intelligence and Communication Networks (CICN). IEEE*, Jabalpur, India, 2015.

[42] E. Spaho, M. Ikeda, L. Barolli, F. Xhafa, M. Younas, and M. Takizawa, "Performance evaluation of OLSR and AODV protocols in a VANET crossroad scenario," in *2013 IEEE 27th International Conference on Advanced Information Networking and Applications (AINA). IEEE*, Barcelona, Spain, 2013.

[43] R. R. Fontes, S. Afzal, S. H. B. Brito, M. A. S. Santos, and C. E. Rothenberg, "Mininet-wifi: emulating software-defined wireless

networks," in *2015 11th Int. Conf. on Network and Service Management (CNSM)*, pp. 384–389, Barcelona, Spain, 2015.

[44] A. Vehtari, A. Gelman, and J. Gabry, "Practical Bayesian model evaluation using leave-one-out cross-validation and waic," *Statistics and Computing*, vol. 27, no. 5, pp. 1413–1432, 2017.

[45] D. W. Ruck, S. K. Rogers, M. Kabrisky, M. E. Oxley, and B. W. Suter, "The multilayer perceptron as an approximation to a bayes optimal discriminant function," *IEEE Transactions on Neural Networks*, vol. 1, no. 4, pp. 296–298, 1990.

[46] S. Ruggieri, "Efficient C4.5 [classification algorithm]," *IEEE Transactions on Knowledge and Data Engineering*, vol. 14, no. 2, pp. 438–444, 2002.