2018 IEEE 4th International Conference on Collaboration and Internet Computing

A Multilayer Perceptron-Based Distributed Intrusion Detection System for Internet of Vehicles

Ayesha Anzer and Mourad Elhadef, Ph.D.
College of Engineering
Abu Dhabi University
Abu Dhabi, UAE
1063572@students.adu.ac.ae, mourad.elhadef@adu.ac.ae

***Abstract*—Security of Internet of vehicles (IoV) is critical as it promises to provide with safer and secure driving. IoV relies on VANETs which is based on V2V (Vehicle to Vehicle) communication. The vehicles are integrated with various sensors and embedded systems allowing them to gather data related to the situation on the road. The collected data can be information associated with a car accident, the congested highway ahead, parked car, etc. This information exchanged with other neighboring vehicles on the road to promote safe driving. IoV networks are vulnerable to various security attacks. The V2V communication comprises specific vulnerabilities which can be manipulated by attackers to compromise the whole network. In this paper, we concentrate on intrusion detection in IoV and propose a multilayer perceptron (MLP) neural network to detect intruders or attackers on an IoV network. Results are in the form of prediction, classification reports, and confusion matrix. A thorough simulation study demonstrates the effectiveness of the new MLP-based intrusion detection system.**

Keywords-Internet of vehicles (IoV); Security Attacks; Multilayer perceptron; Intrusion detection.

I. Introduction

As technology is advancing day by day, there are many challenges researchers are facing to adapt wireless and mobile ad hoc networks (MANETs) in real life situations. People while driving are used to mobile technologies a lot to keep connected with their social life which leads to various accidents to happen. However, it can be mitigated by using the Internet of vehicles (IoV) as it can deal with events of driving like road accidents, fuel consumption, traffic jams, and pollutant emissions. An IoV is based on V2V (Vehicle to Vehicle) communication which makes those vehicles smarter with integrated sensors and embedded systems. These embedded systems gather data related to the situation on the road. The collected data can be information associated with a car accident, the congested road ahead, parked car, etc. This information exchanged with other neighboring vehicles on the road to promote safe driving. Therefore, it reduces driving time to reach the destination and minimize fuel consumption. IoV applications aim to provide safety to comfort for drivers and minimize the incidence of accidents by informing nearby vehicles about a situation. IoV decentralized nature anneals to manage each node within the network which exposes to a security vulnerability. Attackers exploit this vulnerability to gain access to the network and tamper with confidential data. It can lead to more accidents and change the meaning of safe driving. Attacks can disrupt the system functionality and attackers can also misuse the IoV for their purposes. For example, to switch a green light on, emergency braking, or free the fastest lane, etc. [1]. Hence, distributed intrusion detection systems (DIDS) are needed to monitor the functionality of the system by detecting network intruders.

Figure 1 shows the example scenario of IoV. Sensors of a car driving on a highway detect to use an emergency brake due to an accident. The active neighboring cars safety communication system broadcasts this ever to other vehicles driving on the same highway [22].

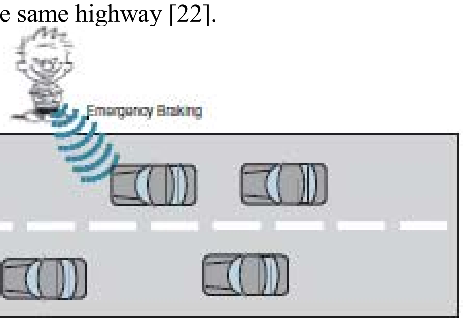

Figure 1. Example Scenario: Emergency Braking [22]

DIDS comprise multiple IDSs located in different IoV infrastructure components like Road Side Unit (RSU) and Onboard Unit (OBU) in a large network environment to detect unusual activities. These various IDSs communicate with a central server located in the host and with each other's [2]. Different artificial intelligence approaches have been used by other researchers to develop an intrusion detection system in IoV. Until now, the MLP based IDS is only used for MANETs. Also, Aminanto and Kim [12] claimed that machine learning algorithms applied to IDSs are difficult to train and set due to vast and complex input features required for training. Nevertheless, they used artificial neural networks for the feature selection for IDSs in the transport layer. Consequently, in this research, we implemented a distributed intrusion detection system by using a multilayer perceptron to help with classifying the suspicious and non-suspicious packets.

978-1-5386-9502-9/18/$31.00 ©2018 IEEE
DOI 10.1109/CIC.2018.00066

Authorized licensed use limited to: Wuhan University. Downloaded on January 08,2023 at 08:33:05 UTC from IEEE Xplore. Restrictions apply.

This paper makes the following contributions: i) a survey on how artificial networks have been addressed to the problem of network traffic attacks and how they have been applied to mobility networks so that they could be offered as a reliable solution for IoV security. Different approaches to artificial intelligence used by other researchers have been studied, ii) proposes a solution by using Multilayer perceptron to scrutinize the packets in the network.

The paper is organized as follows. Section II discusses related work. Section III describes the multilayer perceptron neural network used in intrusion detection systems to detect attacks in IoV. The last section is dedicated to the conclusion.

II. Related Work

Various approaches to the problem of detecting intruders to networks have been proposed. Liu et al. [3] proposed an anomaly-based IDS which uses cross-layer features to detect attacks and locate the source of the attack within one hop perimeter in an ad-hoc network. A compact feature set is suggested to merge intelligence from two different layers; Network layer and MAC layer with the purpose of profiling normal behavior of mobile nodes. A data mining anomaly-detection method is adapted to develop a collaborative detection scheme which allows the IDS to link global and local alerts. To validate their work, they used the NS2 simulator.

In [4], Besson et al. deployed a distributed intrusion detection system in an AWISSENET. The focus of the research is on security and AWISSENET to select a secure service discovery and trusted path. Flexible and efficient results have been achieved through the proposed DIDS which can be used for a different wireless network. It can be validated through a heterogeneous testbed.

Lauf et al. [5] proposed a 2-stage intrusion detection system for MANETs. The type of IDS used was an anomaly based. It analyzed by the context of application-level interactions of networked nodes. The set of attacks behavior identified and monitored dynamically in the network. Density functions, global maxima, and local maxima of behavior are used to detect the behavior of the network. The cross-correlative component is used to distinguish different threats. To identify various threats cross-correlative component is used. The proposed approach dispenses IDSs in all connected nodes in the network. Each node permitted to recognize potential threats exclusively. As a result, they reported a decrease in computing needs to adjust optimally to a sensor and embedded devices in the ad-hoc network.

Bankovic et al. [6] developed a DIDS organized as a reputation system in a wireless sensor network (WSN). For each network node, reputation is allocated by SOM (Self-Organizing Maps) trained to detect intrusions. SOM algorithm is implemented by using energy-efficient SORU co-processor. The proposed solution comprises various benefits like the ability to detect unspecified attacks or fast response to adversarial activities. The result indicates that proposed IDS isolates the malicious nodes from the network and restraint to transmit malicious activities, e.g. Sybil attack to other nodes [26].

In [7] the authors implemented a DIDS using TMote sky wireless sensor. It is used for simulation of the set of parameters that includes energy consumption, response time, accuracy, and detection. Mobile Agents (MA) are integrated for efficient monitoring and improvement of intrusion detection and prevention. Kumar et al. [8] worked with DIDSs for wireless sensor networks. It classified DIDSs into hierarchical, hybrid IDSs, and mobile agent-based IDSs. DIDSs embedded inside intelligent agents employed in an extensive network. An agent shares partial views which decide on the identification of the source and depict it. These IDSs based agents distributed through the network collaborates with each other and make the system adaptive and scalable. This DIDS encompasses different components like NbrPerimeter, Local Packet Monitoring, Alert Region, local Detection, Key Management, Voting, and Local Response. According to their results, the authors claim their DIDS architecture for wireless sensor networks is accurate, scalable, and robust.

Cho et al. [9] proposed a partially DIDS mechanism for wireless sensor networks (WSN). A WSN comprises various small sensor nodes for hardening of each node. The proposed DIDS showed low memory and power demands with a bloom filter. This bloom filter reduced signature code size. This mechanism was able to detect potential DoS attacks in a simulation and validation environment.

A distributed IDS with an intelligent method to detect attacks in a WSN reported in [10], with two different IDS algorithms used at various levels of different learning mechanism like supervised and unsupervised learning. Supervised learning is used on sensor node whereas unsupervised learning is used on a base station and sink node. A structure of detection rules formed as a binary tree. The author worked with only 10% of the data for training the IDS. The output of the algorithms showed high detection accuracy using a few selected features. It also reduces the processing time and complexity of detection. In this research, the J48 classification algorithm also enhanced which shrink in the size of a decision tree of algorithms. Agrawal et al. [11] proposed a DIDS using different concepts like Bayesian learning and Apache Mahout. The purpose of IDS was to reduce limitation of IDS to detect attacks and system failure.

Vehicle ad-hoc networks (VANETs) are decentralized. The VANET fully controls each node. Hence, the system is prone to attacks like misuse of the vehicular ad hoc communication and disruption of system functionality, e.g. changing the traffic light red to green or give wrong signals to free the fastest lane on a highway, etc. Maglaras [1] combined the dynamic agents and static detection to design a DIDS for VANET. Anomaly-based detection and signature-based are used on CAN protocol to conjecture attacks. Signature-based detection is used to compare network traffic with a known signature of attacks whereas anomaly detection is used to identify the normal communication behavior of VANETs.

Artificial neural networks (ANNs) and a deep learning popular algorithm like Stacked Auto Encoder (SAE) [13], [14] have been also used. MATLAB R2016a is used on Intel Xeon CPU with 32GB of RAM. The KDD99 dataset is used

Authorized licensed use limited to: Wuhan University. Downloaded on January 08,2023 at 08:33:05 UTC from IEEE Xplore. Restrictions apply.

for training the network. The method for labeling classes followed by the functional characteristics of the TCP/IP stack. The six types of attacks are selected related to the transport layer. The proposed SAE architecture encompasses two hidden layers that represent two encoders. The output of the second encoder is the input for a SoftMax regression function. This procedure helped to produce a lightweight IDS [25]. Even that this application is not directly intended to VANETs, it used a protocol for testing experiments. For example, sending video frames in a UDP connection among vehicles. Therefore, this type of IDS might be extended for wireless networks and to VANETs.

Perakovic et al. [15] worked on the problem of modeling network traffic classification using artificial neural networks. MATLAB R2016a and the toolbox Neural Pattern Recognition is used to detect distributed denial of service. UDP protocol used for DDoS to send a massive amount of UDP packets with spoofed IP addresses. Incoming traffic separated as legitimate and illegitimate. Four labels have been used comprise of UDP, DNS, CharGen, and Other. This classification of the incoming traffic is challenging for the many characteristics that must be considered for getting actual network traffic labeled as legitimate or not. The model obtained got 82% accuracy due to some similarities in the values of legitimate traffic and UDP DDoS attack parameters. Based on the results, the conclusion was that adding more parameters to the dataset will make the model learn to classify with more accurately.

Kwon et al. [16] presented various deep learning algorithms used by computers to enhance their performance and to test intrusion detection for anomaly-based techniques. Such technologies are mostly used for data reduction and classification. They used fully connected network (FNC) experimentation to test the effectiveness of various deep learning techniques as a way of analyzing the network traffic. This FNC model is a highly effective deep learning technique for finding the anomaly as compared to other similar models such as Adaboosting, random forest, and SVM [14]. Pavani et al. [21] proposed a neural network-based on multi-layer perceptron to detect normal and attacked behavior in MANETs. The test was performed on a grey hole and black hole attacks [20] which were implemented using the NS2 simulator. Their technique successfully detected the attacks. Leinmueller et al. [22] found that classical approaches of intrusion detection system are no longer effective, and they proposed a new system named as modular cross layered intrusion detection which utilizes various sources of information including the layers to gather data. Their system uses another abstract language to access and check events occurring at different layers, and the same language is used to explain the dependencies of such events. Also, different layers are evaluated using such detection techniques which are context-supported. This system is supposed to find out warnings related to safety issues and in the detection of various malicious nodes.

In [23], Alheeti et al. found that VANETs are more exposed to security attacks because they have no traditional security infrastructure, high dynamic topology, and mobility. The researchers build an advanced IDS which uses the Proportional Overlapping Score (POS) technique. This intelligent IDS uses the information received from the trace file of VANETs to reduce the additional features in order to improve the performance and security of self-driving vehicles. The features extracted from the trace file helped to differentiate the normal vehicles form abnormal ones. They used artificial neural networks (ANNs) to retrieve the data in an auditable format and further uses it to track the attacks of blackholes. Moreover, the study also proposed that the use of hybrid IDS and fuzzy data can help to detect the blackholes better. In addition, Alheeti et al. [24] designed another IDS against malicious attacks in driverless cars such as Google driverless cars. VANETs technology has gained popularity during the recent years, but these networks are more exposed to security attacks as compared to other similar networks such as wired networks. Also, VANETs autonomously collects the data of mobile vehicles in the absence of security infrastructure and dynamic topology and all these features collectively make it a vulnerable system. That is why Alheeti et al. designed a new IDS mechanism with the help of ANNs to find the possible attacks by making the security system stronger. It mainly works by collecting the data formed by trace files and use them to detect malicious attacks later.

III. An MLP-based intrusion detection system

A multilayer perceptron is used to determine the flow of data transfer during communication between vehicles to be legitimate or illegitimate. In order to generate traffic, we used NS3 and due to a limitation of the dataset we used KDD Cup 1999 Data to complete the research. Anaconda is used as a framework for training the model. We are interested in network attacks as many common network protocols have been ported to IoV from wireless networks, and not all of them have been standardized to IoV. Given that network, attacks are intended to find vulnerabilities within the network structure. It is crucial to detect and mitigate them to avoid damages for network users. Particularly in IoV services where they might be focused on road and traffic status or applications programs for human leisure, i.e. video chat between a group of vehicle users or sharing geo-referencing data each other. An attack might occur at the application layer in the form of a DoS attack [28] that affects the application protocols and related services directly. In this way, network resources could reflect a limited performance for an application. In contrast, this attack is heavily targeted and looks for being not detected after it is happening, so they gain a stealth feature and become successful tools for network intruders.

Figure 2 depicts a UML diagram of the process. The most challenging processes are the selection of enough features for traffic attacks. This selection is directly related to training, validation, and testing processes.

Authorized licensed use limited to: Wuhan University. Downloaded on January 08,2023 at 08:33:05 UTC from IEEE Xplore. Restrictions apply.

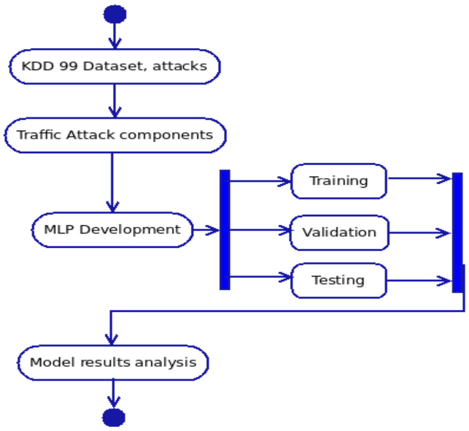

Figure 2. Flowchart of the experiment

Many intrusion detection techniques are based on artificial intelligence. These learnings based on assumptions on patterns of attack packets as compared to normal packets. In learning techniques, several label data are required to be set for training purpose. We set the last column "provisore" as a label. There are many modules of IDS which gather and analyze a huge amount of data packets. The IDS module includes a monitoring module and a profiling module. The monitoring model detects the type of incoming packets after feature extraction. Profiling module comprises features which trained off-line. Any feature which is detected by the monitoring module will be stored in the database by profiling module for upcoming packets as shown in Figure 3 [17].

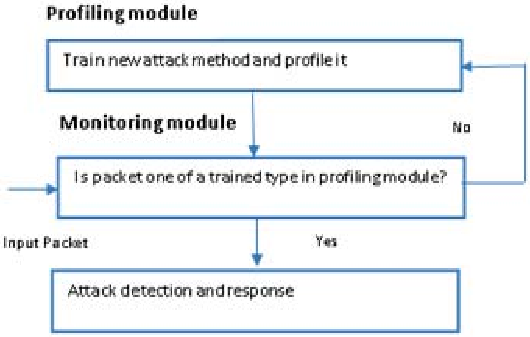
Profiling module
Train new attack method and profile it
Monitoring module
No
Is packet one of a trained type in profiling module?
Input Packet
Yes
Attack detection and response

Figure 3. IDS architecture based on machine learning [17]

As we expected to get a model for classifying incoming network traffic in the form of an attack, we used the multilayer perceptron (MLP) algorithm. MLP model inserted as a command that an IoV OBU will run. A multilayer perceptron is a supervised learning algorithm which learns functions through training based on dataset [18]. Through some features and target, the non-linear function learned as classification or regression. Figure 4 given below shows one hidden layer of MLP with a scalar output. It is different from logistic regression as one or more than non-linear layers among the input layer and an output layer. The leftmost layer is input later comprise of a set of neurons which represents the input features. In the hidden layer, each neuron transforms values of the prior layer with a weighted linear summation followed by non-linear activation function (hyperbolic tan function). The output layer receives the values from the last hidden layer then transform into output values. This model provides with the capability to learn a model in real-time using partial fit and non-linear models [18].

We took KDD Cup 1999 Data as input for a multilayer perceptron. This layer deal with the features, we need to extract from a network attack, as we described before there are many attacks in that dataset as shown in Table 1. The dataset has been preprocessed to load into the model without any error. To preprocess the dataset, data type conversion and additional columns deletion has been done. The dataset selected comprised of 43 features. We labeled "provisore" to predict classification. The number of records in the dataset is 1048576. Initially, basic normalization has been done then data normalized with zero mean and zero standard deviation. After reading the data, data separated to input data (features), and output data (label) as MLP trains on two arrays: x_data (input data) and y_data (output data). Input data holds training the samples signifies floating-point feature vectors and output data holds the class labels for training samples [18].

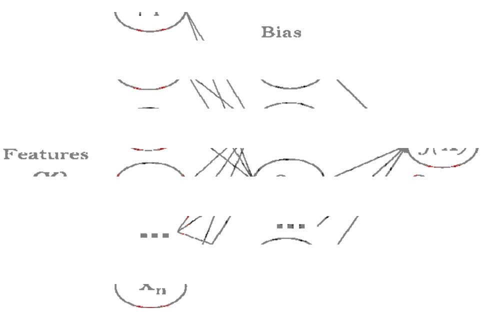

Figure 4. One hidden layer of Multilayer perceptron [18]

Standard scale normalization has been applied on input data only then split into training data and testing data. Class MLPClassifier is used to implement an MLP algorithm which trains using backpropagation. The different hidden layer sizes are used. The solver is weight optimization. There are many different solvers used in MLP. In this experiment, we used Adam which is stochastic gradient-based optimizer used well on large datasets in regard to validation score and training time. Activation function [18] used is logistics. The logistic is a sigmoid function used as activation function in MLP. It returns function as:

$$f(x) = 1 / (1 + exp(-x))$$

The default alpha 0.0001 is used. The batch size is set to auto which is equal to min (200, n_samples). The default beta_1 and beta_2 are used as solver used is 'Adam'. The early stopping approach is used to avoid overfitting. The output shows after running MLPClassifier that the network converged for a global minimu. The default epsilon used for numerical stability in Adam. Learning rate is set to a constant given by learning_rate_init MLP attribute. The learning rate schedule for weight updates. The

Authorized licensed use limited to: Wuhan University. Downloaded on January 08,2023 at 08:33:05 UTC from IEEE Xplore. Restrictions apply.

learning_rate_init is used 0.001 which controls the step-size to update the weights. The maximum number of iterations used is 100000 in Adam solver it shows the number of epochs which indicates how many times each data point can be used. The momentum is 0.9 which is for the gradient descent. The Nesterov momentum is set to true. It calculates the gradient not for current weights instead for current weight plus momentum constant times the previous delta [19]. An exponent for inverse scaling learning rate (power_t) is set to 0.5. Random state number is none, so random number generator is a random state instance used by np. random. Shuffle is used to shuffle samples in each iteration. Tolerance is 0.0001 for optimization as when a loss is improving at least for two consecutive iterations. For early stopping, the proportion of training data is set aside as a "validation set".

TABLE I. ATTACK TYPES IN KDD DATASET

S/N	Name	Type
1)	Back	DoS
2)	land	DoS
3)	neptune	DoS
4)	pod	DoS
5)	smurf	DoS
6)	teardrop	DoS
7)	Buffer overflow	U2R
8)	Load module	U2R
9)	perl	U2R
10)	ftp write	R2L
11)	Guess passwd	R2L
12)	imap	R2L
13)	multihop	R2L
14)	phf	R2L
15)	warezmaster	R2L
16)	10_write	R2L
17)	ipsweep	Probe
18)	nmap	Probe
19)	portsweep	Probe
20)	satan	Probe

IV. EXPERIMENT RESULTS AND DISCUSSION

MLPClassifier trains iteratively at each time loss function is computed to update model parameters. There is also regularization which is added to lose function which shrinks model parameters to prevent overfitting. A different number of iterations and loss achieved when a different number of neurons are used in the 2nd layer. To optimize the model's results, a different number of neurons have been tested in the 2nd layer. In the first layer, we used 43 for each result. We selected five best results based on loss and number of iterations.

Figure 5 shows the loss achieved. In the 1st layer, 43 neurons are used, and in the 2nd layer, 100 neurons are used. The number of iteration achieved here is 11.

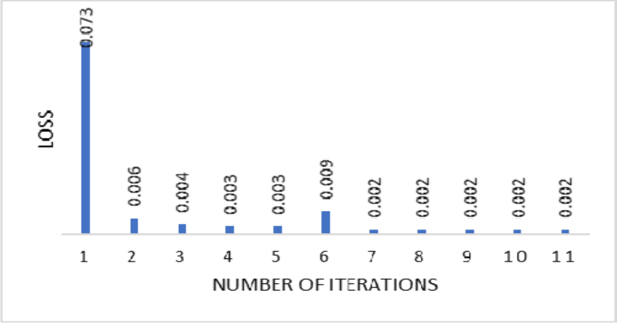

Figure 5. Use of 100 neurons in a 2nd layer.

Figure 6 shows the loss achieved. In the 1st layer, 43 neurons are used, and in the 2nd layer, 43 neurons are used. The number of iteration achieved here is 12.

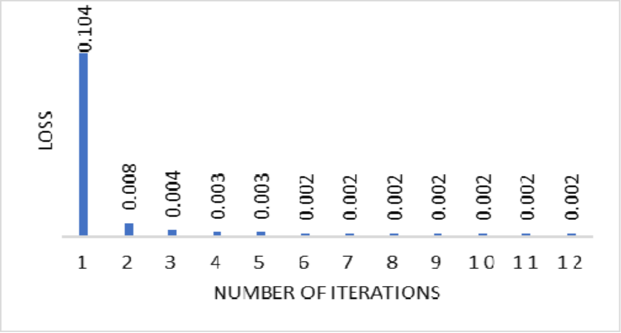

Figure 6. Use of 43 neurons in a 2nd layer.

Figure 7 shows the loss achieved. In the 1st layer, 43 neurons are used, and in the 2nd layer, 35 neurons are used. The number of iteration achieved here is 11.

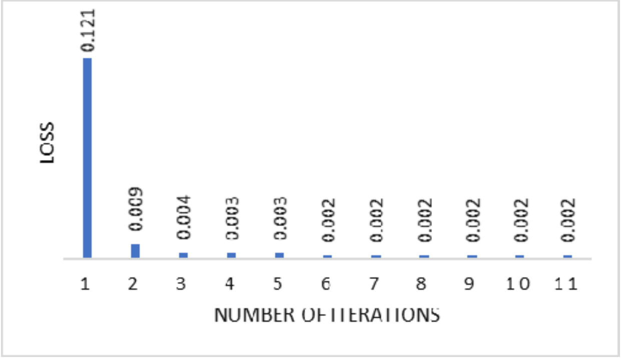

Figure 7. Use of 35 neurons in a 2nd layer.

Figure 8 shows the loss achieved. In the 1st layer, 43 neurons are used, and in the 2nd layer, 25 neurons are used. The number of iteration achieved here is 15.

Authorized licensed use limited to: Wuhan University. Downloaded on January 08,2023 at 08:33:05 UTC from IEEE Xplore. Restrictions apply.

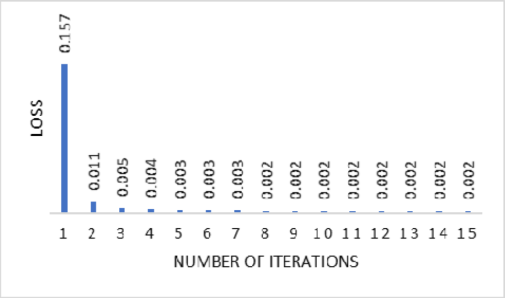

Figure 8. Use of 25 neurons in a 2nd layer.

Figure 9 shows the loss achieved. In the 1st layer, 43 neurons are used, and in the 2nd layer, 15 neurons are used. The number of iteration achieved here is 13.

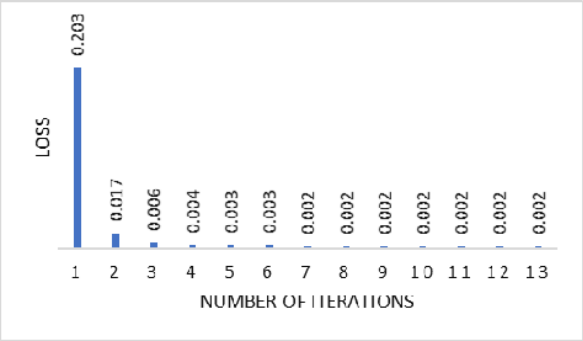

Figure 9. Use of 15 neurons in a 2nd layer.

After experimenting with a different number of neurons, we concluded that best results we achieved are for 25 neurons in layer 2 as the number of iterations received are 15 which are maximum we received, and loss achieved is less as compare to all results achieved. For each hidden layer size used during training, we make a prediction on test data, generate classification report to analyze the performance and plot the graph based on confusion matrix to visualize the performance of the MLP model. The prediction and classification report for each hidden layer sized are same. However, the difference has been noticed in the confusion matrix.

After training MLP to learn the pattern, MLP predicts test data as shown in table 2. It can be noticed that the model predicted three attacked nodes.

TABLE II. PREDICTION ON TEST DATA

Prediction	Attack type
Smurf	DoS
Normal	-------
Neptune	DoS
Neptune	DoS
Normal	--------

The table 3 shows the classification report generated using the MLP algorithm. The classification report is used to visualize precision, recall, F1-score, and support [27] scores for the MLP. We used it to compare the classification model to select the model which comprise of stronger classification metrics or balanced. The metrics used defined as true and false positives and true and false negatives. Precision shows for all instances classified positive what percent it is correct. Recall shows the percentage for all instances which are classified correctly. Precision and recall for each provisore given which is accurate as it is between 0-1. F1-score is the weighted mean of precision and recall values.

For example; the worst score is 0.0, and the best score is 1.0. As per results achieved, it has been noticed that best scores are for normal, smurf (DoS), neptune (DoS), back (DoS), satan (probe), portsweep (probe), and ipsweep (probe). The worst scores are phf (r2l), 10_write (r2l), multihop (r2l) and ftp_write (r2l). Rest of the values are in between 0 to 1. The overall average of the classification report is 1.00 so its best score we achieved. The number of actual occurrences of class in kdd dataset shown in support. If support is imbalanced in training data, it indicates structural weaknesses in reported scores of the classifier which required the use of rebalancing and stratified sampling, but in results, we achieved balanced support.

TABLE III. CLASSIFICATION REPORT OF DOS ATTACK IN IOV

Provisore	Precision	Recall	F1-score	Support
normal	1.00	1.00	1.00	148960
land	1.00	0.33	0.50	3
smurf	1.00	1.00	1.00	56771
neptune	1.00	1.00	1.00	51377
loadmodule	0.00	0.00	0.00	1
back	0.99	1.00	1.00	504
imap	0.50	0.50	0.50	2
satan	1.00	0.99	1.00	1300
phf	0.00	0.00	0.00	1
portsweep	1.00	0.99	1.00	701
10_write	0.00	0.00	0.00	5
nmap	0.99	0.98	0.99	557
warezmaster	0.75	0.86	0.80	7
teardrop	1.00	0.98	0.99	51
buffer_overflow	0.33	1.00	0.50	1
pod	0.88	1.00	0.93	7
ipsweep	1.00	1.00	1.00	1883
multihop	0.00	0.00	0.00	3
guess_passwd	0.88	0.88	0.88	8
ftp_write	0.00	0.00	0.00	2
Average/ Total	**1.00**	**1.00**	**1.00**	**262144**

The confusion matrix is used to visualize the performance of the MLP algorithm. The confusion matrix is also called an error matrix. It comprises of a contingency table with two proportions "actual class" and "predicted class". We trained classification system to distinguish between traffic generated to identify is it compromised or normal packet. The confusion matrix is used to recapitulate

Authorized licensed use limited to: Wuhan University. Downloaded on January 08,2023 at 08:33:05 UTC from IEEE Xplore. Restrictions apply.

the results achieved after testing the MLP algorithm for further analyzing.

Figures given below shows the actual class in the x-axis (top-horizontal) and predicted class in the y-axis (top-vertical). All correct prediction in diagonal colored with yellow. The values in different colors and outside of the diagonal are prediction errors. These prediction errors can be overfitting or noise. The best result we received is for 43 neurons as shown in Figure 11.

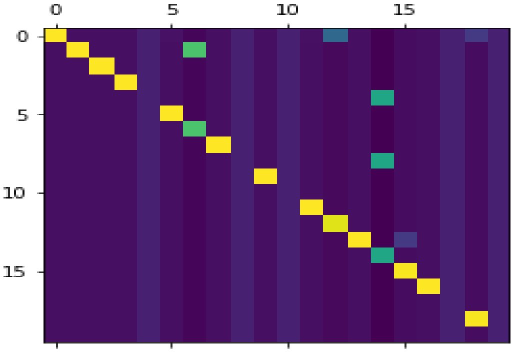

Figure 10. Confusion matrix plotted for 100 neurons in a 2nd layer.

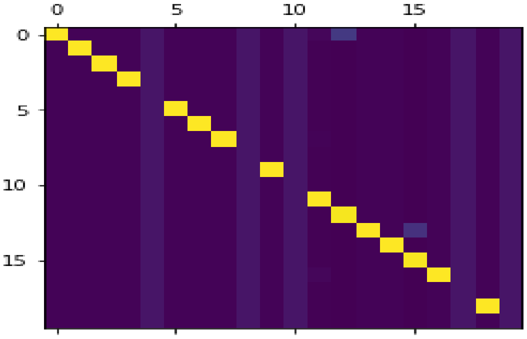

Figure 11. Confusion matrix plotted for 43 neurons in a 2nd layer.

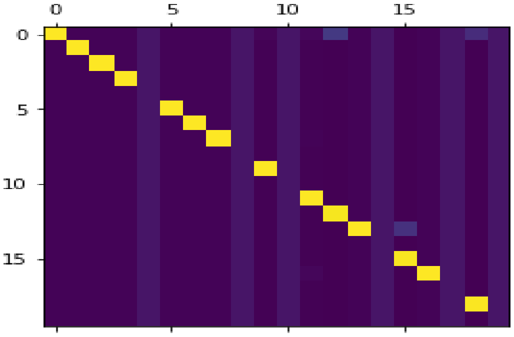

Figure 12. Confusion matrix plotted for 35 neurons in a 2nd layer.

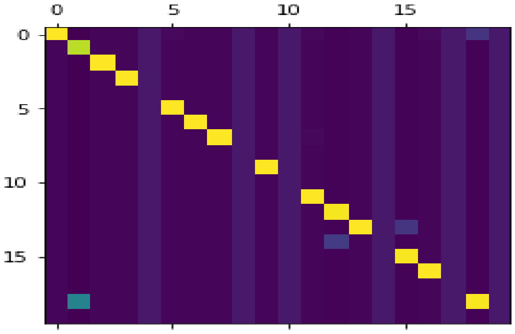

Figure 13. Confusion matrix plotted for 25 neurons in the 2nd layer.

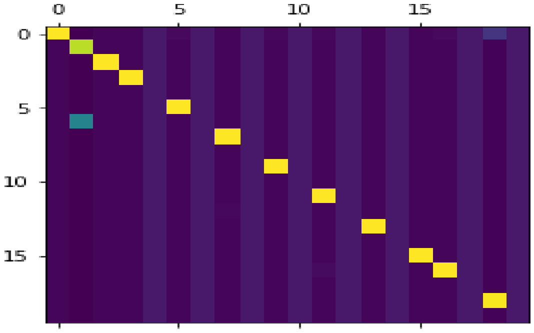

Figure 14. Confusion matrix plotted for 15 neurons in the 2nd layer.

V. Conclusion

Computational techniques for artificial intelligence require a phase for setting a good dataset and techniques. Many of the techniques found claims that algorithms could be computing challenging meanwhile an IDS should be the first input for a decision making of how network infrastructure built against network attacks. Perhaps multilayer perceptron (MLP) could be successful in getting a zone activity where an attack has been present, but how to broadcast communication between different nodes in the network may rely on the hop by hop communication or the use of the fixed unit, in the case of IoV the candidate is a Road Side Unit (RSU). IoV is receptive to security attacks due to its properties like decentralized nature, node mobility, and limited bandwidth. The risk of attacks can be eliminated through different security mechanisms such as authentication and encryption, but risk cannot be reduced completely. Hence, Intrusion detection system is proposed using MLP to detect normal and suspicious traffic. This technique tested on different attack types like DoS, probe, R2l, and U2R. A different number of neurons are tested to predict the attack types and produce a classification report and a confusion matrix. The method successfully detected attacks and presents a graphical representation of a confusion matrix. The result of the MLP algorithm showed high detection accuracy using a few selected features. It also reduces the processing time and complexity of detection.

The limitation of this approach is to deal with a problem of how deep network must be to achieve better results. The

Authorized licensed use limited to: Wuhan University. Downloaded on January 08,2023 at 08:33:05 UTC from IEEE Xplore. Restrictions apply.

KDD Cup 1999 Data used is commonly used for networking issues, and we used it for IoV without considering mobility, protocols involved, and time to establish communications or attacks.

In the future, deep learning will be used to study a cascade of multiple layers for intrusion detection in IoV. It will be helpful in accurately identifying various attacks in networks based on several neurons, different learning rates, and impact of the performance. These results will be compared with other models used to know the accuracy of each model and provide a new research method for intrusion detection.

ACKNOWLEDGMENT

This work is supported by ADEC Award for Research Excellence (A2RE) and Office of Research and Sponsored Programs (ORSP), Abu Dhabi University.

REFERENCES

[1] Leandros A. Maglaras, "A Novel Distributed Intrusion Detection System for Vehicular Ad Hoc Networks," (IJACSA) Int. Journal of Advanced Computer Science and Applications, vol. 6(4), pp. 101-106, 2015.

[2] Nathan Einwechter, "An Introduction to Distributed Intrusion Detection Systems," Symantec, 8 January 2002. [Online]. Available: https://www.symantec.com/connect/articles/introduction-distributed-intrusion-detection-systems. [Accessed 7 July 2017].

[3] Yu LIU, Yang LI, and Hong MAN, "A distributed cross-layer intrusion detection system for ad hoc networks," Annales Des Telecommunications, vol. 61, no. 3-4, pp. 357-378, April 2006.

[4] Lionel Besson and Philippe Leleu, "A distributed intrusion detection system for ad-hoc wireless sensor networks The AWISSENET Distributed Intrusion Detection System," 2009 16th Int. Conference on Systems, Signals, and Image Processing, pp. 1-3, 18-20 June 2009.

[5] A. P. Lauf, Richard A. Peters, and William H. Robinson, "A distributed intrusion detection system for resource-constrained devices in ad-hoc networks," Ad Hoc Networks, vol. 8(3), pp. 253-266, May 2010.

[6] Z. Bankovic, D. Fraga, J. M. Moya, and J. C. Vallejo, "Distributed intrusion detection system for wireless sensor networks based on a reputation system coupled with kernel self-organizing maps," Integrated Computer-Aided Engineering, vol. 17, no. 2, pp. 87-102, 14 April 2010.

[7] S. I. Eludiora, O.O. Abiona, A. O. Oluwatope, S. A. Bello, M.L Sanni, D. O. Ayanda, and C.E. Onime E. R., "A Distributed Intrusion Detection Scheme for Wireless Sensor Networks," 2011 IEEE International Conf. on Electro/Information Technology, pp. 1-5, 15-17 May 2011.

[8] Aravendra Kumar Sharma, Sushil Kumar Saroj, and Prashant Kumar, "Distributed Intrusion Detection System for Wireless Sensor Networks," IOSR Journal of Computer Engineering, vol. 14(1), pp. 61-70, 2013.

[9] Eung Jun Cho, Choong Seon Hong, Sungwon Lee, and Seokhee Jeon, "A Partially Distributed Intrusion Detection System for Wireless Sensor Networks," Sensors, no. 13, pp. 15863-15879, 2013.

[10] [Karen Medhat, Rabie A. Ramadan, and Ihab Talkhan, "Distributed Intrusion Detection System for Wireless Sensor Networks," 2015 9th International Conference on Next Generation Mobile Applications, Services and Technologies, pp. 234-239, 9-11 September 2015.

[11] Ronak Agrawal, Ganesh Talekar, Akshay Singh Chandel, Shriganesh Munde, and Deep Lakshmi Zingade, "Distributed Intrusion Detection System using Bayesian learning and Apache Mahout," Int. Journal of Advanced Research in Computer and Communication Engineering, vol. 5, no. 1, pp. 361-364, January 2016.

[12] Aminanto, M. E., & Kim, K. (2016). Deep learning-based feature selection for intrusion detection system in the transport layer.

[13] L. Deng, "A tutorial survey of architectures, algorithms, and applications for deep learning." Transactions on Signal and Information Processing 3: e2, Cambridge Univ Press, 2014.

[14] A. Anzer and M. Elhadef, "Deep Learning-Based Intrusion Detection Systems for Intelligent Vehicular Ad Hoc Networks", Proc. of the 12th International Conference on Ubiquitous Information Technologies and Applications, Taichung, Taiwan, Dec. 18-20, 2017.

[15] Peraković, D., Periša, M., Cvitić, I., & Husnjak, S. (2017). Model for detection and classification of DDoS traffic based on the artificial neural network. Telfor Journal, 9(1), 26.

[16] D.Kwon, H. Kim, J. Kim, S.C. Suh, I. Kim, and K.J. Kim, "A survey of deep learning-based network anomaly detection", Springer US, pp. 1-11, September 2017.

[17] Min-Joo Kang, Je-Won Kang, "Intrusion Detection System Using Deep Neural Network for In-Vehicle Network Security," PLOS ONE, vol. 11, no. 6, 7 June 2016.

[18] Scikit-learn, "Neural network models (supervised)," 2017. [Online]. Available: http://scikit-learn.org/stable/modules/neural_networks_supervised.html. [Accessed 24 August 2018].

[19] James Mccaffrey, "Neural Network Nesterov Momentum," 24 July 2017. [Online]. Available:

[20] https://jamesmccaffrey.wordpress.com/2017/07/24/neural-network-nesterov-momentum/. [Accessed 28 August 2018].

[21] W. Ahmed and M. Elhadef, "Securing Intelligent Vehicular Ad Hoc Networks: A Survey", Proc. of the 12th Int. Conf. on Future Information Technology (FutureTech 2017), Seoul, Korea, May 2017.

[22] K. Pavani and A. Damodaram, "Intrusion detection using MLP for MANETs," Third International Conference on Computational Intelligence and Information Technology, Mumbai, 2013, pp. 440-444.

[23] T. Leinmüller, A. Held, G. Schafer, A. Wolisz, "Intrusion Detection in VANETs", Proc. 12th IEEE International Conference on Network Protocols (ICNP 2004) Student Poster Session. IEEE ICNP 2004, 2004.

[24] K. M. A. Alheeti, A. Gruebler and K. D. McDonald-Maier, "An Intrusion Detection System against Black Hole Attacks on the Communication Network of Self-Driving Cars," 2015 Sixth International Conference on Emerging Security Technologies (EST), Braunschweig, 2015, pp. 86-91.

[25] K. M. A. Alheeti, A. Gruebler and K. D. McDonald-Maier, "An intrusion detection system against malicious attacks on the communication network of driverless cars," 2015 12th Annual IEEE Consumer Communications and Networking Conference (CCNC), Las Vegas, NV, 2015, pp. 916-921.

[26] S. Zaman and F. Karray, "Lightweight IDS based on features selection and IDS classification scheme." Computational Science and Engineering,2009. Int. Conference on. Vol. 3. - CSE 2009, IEEE.

[27] A. Muhamad and M. Elhadef, "Sybil Attacks in Intelligent Vehicular Ad hoc networks: A Review", Proc. of the 9th Int. Conf. on Computer Science and its Applications, Taiwan, Dec. 18-20, 2017.

[28] Scikit-yb, "Classification Report," 2016. [Online]. Available: http://www.scikit-yb.org/en/latest/api/classifier/classification_report.html. [Accessed 29 August 2018].

[29] W. Ahmed and M. Elhadef, "DoS Attacks and Countermeasures in VANETs", In Proc. of the 12th Int. Conf. on Ubiquitous Information Technologies and Applications, Taichung, Taiwan, Dec. 18-20, 2017

Authorized licensed use limited to: Wuhan University. Downloaded on January 08,2023 at 08:33:05 UTC from IEEE Xplore. Restrictions apply.