TS-VNIDD: A Novel Vehicular Network Traffic Dataset for Machine Learning based Intrusion Detection Systems

Xun He, Zhen Yang, Yongfeng Huang

Abstract: This paper published a large-scale dataset for vehicular network intrusion detection system named TS-VNIDD (Tsinghua-SAIC Vehicular Network Intrusion Detection Dataset). Existing datasets, like KDD CUP 1999, CICIDS2017, and MAWILab, have significant drawbacks. First of all, a common shortcoming of these three datasets is that they are not for the specific scene of the vehicular network, but are collected from the general Internet environment, and therefore cannot fully reflect the characteristics of the data traffic of the vehicular network. It is also worth noting that KDD cup 1999 exists a long time, so it is not as representative as it was at the beginning. In order to overcome these problems, the TS-VNIDD published in this paper has the following characteristics. Firstly, the attack behaviors of the three public datasets mentioned above are analyzed, and we generate and capture traffic between vehicle terminal and server according to the attack behavior which can be implemented in the vehicular network, so that the attacks included in the traffic are more targeted. Secondly, our dataset is designed based on TSP&OTA protocol, which is the communication bridge between vehicle and server. For evaluation, this paper benchmarked the performance of several state-of-the-art machine learning based intrusion detection methods on CICIDS2017, MAWILab and TS-VNIDD. The experimental results show that different methods' different performance on our proposed TS-VNIDD and future research direction of vehicular network intrusion detection systems.

Key words: vehicular network; intrusion detection; TS-VNIDD; machine learning

1 Introduction

Recently, vehicular network becomes increasingly popular in both academic and industrial fields. Vehicles in the vehicular network can get necessary information support to enable automated driving and some other intelligent applications. However, network brings new information security problems for vehicles, in which most worrying one is the traffic intrusion of vehicles. Once vehicle is intruded, not only vehicle's information security could be harmed but also vehicle passengers' physical safety could be threatened.

To resist network traffic intrusion, intrusion detection system has been proposed since the 1980s [23]. In the field of intrusion detection, there are many standards for the classification for it. We refer to one of the standards here [24] to divide it into two categories, one is misuse detection and the other is anomaly detection. For the former, it can also be called a rule-based intrusion detection system, which uses rules to describe the characteristics of intrusion behavior. Therefore, this method can be used to detect all known attacks. Its disadvantage is that it requires artificially establishing huge rule set, and discriminates intrusion by parsing and matching this set when detecting traffic, accounting for the large overhead of system. The latter is based on machine learning and statistical models. It is currently popular to use machine learning methods to analyze traffic through model training. From traditional machine learning, such as Naive Bayes Classification [25], Genetic Algorithm [26], SVM [27], Decision Tree [28] to deep learning network [29], machine learning has been widely used in intrusion detection. Compared with general rule-based detection systems, machine learning can predict new unknown attacks well based on existing attack behaviors, avoid the establishment of rule set and reduce the cost of manual rule construction. Compared with the ordinary machine learning method, deep learning can better extract the subtle changes of abnormal network activities, making the learning ability of the system more powerful. Thanks to the improvement of hardware performance, deep learning applied to vehicular network intrusion detection is acceptable. Considering the prospect of deep learning for the possible improvement of performance, deep learning-based intrusion detection method is mainly used in this paper.

In this paper, we mainly focus on traffic security in the vehicular network environment. Our goal is by some effective means to detect the abnormal behavior hidden in the data traffic sent to vehicle to ensure information security of the vehicle. However, the realization of a machine learning based intrusion detection system in the vehicular network has two challenges. The first challenge is that the rule-based intrusion detection technology in the

- X. He is with the Tsinghua NGN Lab, Institute for Network Sciences and Cyberspace, Tsinghua University, Beijing 100084, China (e-mail: hx18@mails.tsinghua.edu.cn).
- Z.Yang is with the Tsinghua NGN Lab, Department of Electronic Engineering, Tsinghua University, Beijing 100084, China (e-mail: yangzhenyz@mail.tsinghua.edu.cn)
- Y. Huang is with the Tsinghua NGN Lab, Department of Electronic Engineering, Tsinghua University, Beijing 100084, China (e-mail: yfhuang@mail.tsinghua.edu.cn)

- The TS-VNIDD dataset with the code are available at https://github.com/XunMrh/TS-VNIDD

industry is more mature, such as the famous open source intrusion detection system named Snort[1], which has become an industry standard. In contrast, machine learning is still an emerging technology in the industry. Another challenge is that the vehicular network traffic used to train machine learning is quite related to privacy and interests, therefore, the vehicle manufacturers pay great attention to the confidentiality of these data, and are unwilling to publish it. These two challenges lead to the lack of traffic data in the vehicular network scenario. So far, the large-scale dataset for the vehicular network in the true sense has not yet.

The motivation for this paper is to design an intrusion detection dataset TS-VNIDD for the vehicular network. So far, due to some reasons mentioned above, there is no large-scale dataset for intrusion detection of the vehicular network in real sense, and the dataset that has been disclosed for general network intrusion detection has some obvious defects. First of all, datasets such as KDD CUP 1999[2], CICIDS2017 [17], and MAWILab [18] are all collected data traffic on the general Internet. They are not representative in the scene of vehicular network, and cannot reflect the essential characteristics of the vehicular network traffic. KDD CUP 1999 is a dataset used by the KDD competition in 1999. It collects 9 weeks of TCP network connectivity and system audit data by simulating various user behaviors and attack methods. CICIDS2017 is sampled from near-real background traffic, which is generated by the Canadian Cyber Security Institute using the B-Profile system to describe human interactions [2], based on HTTP, HTTPS, FTP, SSH and Email protocols. MAWILab is generated by the Japanese Fukuda laboratory to sample the traffic in a real network of a Japanese-US trans-Pacific link called WIDE, then an abnormal label is created for the obtained data traffic. Second, some datasets are quite old, and the typical representative is KDD CUP 1999. With the rapid changes of the network, many previous attacks are no longer applicable, meanwhile, many new types of attacks continue to emerge, which requires datasets used to train intrusion detection to maintain an appropriate timeline.

Taking into account the problems of the existing dataset mentioned above, this paper establishes a new dataset called TS-VNIDD for training vehicle intrusion detection system based on machine learning. This dataset contains abnormal traffic of around 40GB and normal traffic of around 60GB. The network traffic is generated and captured between vehicle terminal and server based on TSP&OTA protocol in co-operation of Tsinghua University and Shanghai Automotive Industry Corporation (SAIC). Compared with existing datasets, this dataset has these differences: First, the traffic data is closer to the vehicular network environment. Second, the attack behavior is comprehensive. Third, the data is sufficiently guaranteed in timeline. Therefore, this dataset can provide a more realistic and reliable benchmark in vehicular network intrusion detection based on machine learning. On these three datasets of TS-VNIDD, CICIDS2017, MAWILab, this paper tests benchmark of several machine learning methods. We use ROC, F-score, precision, accuracy and other indicators to evaluate the results.

The following chapters are as follows. In the second section, we outline the related work and background of the intrusion detection and the vehicular network. In the third section, we briefly introduce the dataset TS-VNIDD. In the fourth section, we introduce the framework of vehicular network intrusion detection system based on deep learning proposed in this paper. In the fifth section, we introduce the benchmark method and evaluation indicators. In the sixth section, we analyze our experimental results. In the last section, we summarize the article and introduce our future work.

2 Related Work

2.1 Intrusion Detection Methods

Signature-based methods. The core of the signature-based approach is to construct a large-scale rule set, similar to the blacklist mechanism, by analyzing the known attack behaviors and manually describing the corresponding rules. Traffic packet which matches a rule in the rule set, is determined to be abnormal traffic. Its representative is the well-known open source rule-based intrusion detection system Snort, in which the rule format defined has become the actual industry standard for various intrusion detection systems. Suricata and Zeek are inherited from Snort and later developed and improved. Suricata is similar to Snort and is dedicated to intrusion detection. It can only filter the traffic through the rules by building rule set. Zeek can be used not only for intrusion detection but also for other functions. It defines a script language that is flexible to use and provides anomaly detection besides rule-based detection.

Machine learning method. The machine learning method is to train a model on the data traffic, and use the obtained model to discriminate the traffic. When extracting features, [1] used the deep learning method of Self-taught Learning (STL) to train NIDS on the NSL-KDD dataset. [2][4][5] used a neural network based on an automatic encoder for network intrusion detection. [3] proposed a new method of SCDNN for sensor networks, which combines spectral clustering (SC) and deep neural network (DNN) algorithms. In [6][12], an intrusion detection system for network traffic was built using DBN, and a logistic regression classifier was used to make prediction result. In [7], A new deep learning neural network CDBN combining CNN and DBN is proposed. In [8], A deep learning based distributed attack detection system for distributed IoT applications is designed. [9] [10] proposed an intrusion detection system using deep

1.https://www.snort.org/

2. http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.

Table 2-1 KDD CUP 1999 dataset`s attacks in classification

Behavior	Defination	Specific Type
Probe	scan the ports which are active	Ipsweep,mscan,nmap,portsweep,saint,satan
Dos	Send amounts of packet to waste host resource	Smurf,udpstorm;Apache2,back,mailbomb,Neptune,processtable; Land,pod,teardrop;
U2R	Inject shellcode to get root	Buffer_overflow,httptunnel,loadmodule,perl,ps,rootkit,sqlattack,xterm
R2L	Override access to the host by connecting to it	ftp_write,guess_passwd,imap,multihop,named,phf,sendmail, snmpgetattack,snmpguess,spy,warezclient,warezmaster,worm,xlock, xsnoop

Table 2-2 Classification of attack behavior·in CICIDS2017

Behavior	Defination	Specific Type
Brute Force	Get password brutely	FTP-Patator,SSH-Patator
DoS/DDoS	-	Slowloris,Slowhttptest,Hulk, GoldenEye,LOIT
Web Attack	Attack aiming at web server	XSS,SQL Injection
Infiltration	Application exploit	Dropbox download,Cool disk
Meta exploit Win Vista	Windows exploit	-
Botnet ARES	IOT device exploit	-
Heartbleed Port 444	Open-SSL exploit	-
Port Scan	-	-

Table 2-3 Classification of attack behavior·in MAWILab

Behavior	Specific Type	
Denial_of_ Service	Distributed	distributed_denial_of_service, distributed_denial_of_service_ICMP, distributed_denial_of_service_SYN, distributed_denial_of_service_SYN_ACK_response, distributed_denial_of_service_UDP
	Point_to_point	point_to_point_denial_of_service, point_to_point_denial_of_service_ICMP, point_to_point_denial_of_service_SYN, small_point_to_point_denial_of_service_ICMP, small_point_to_point_denial_of_service_SYN
Scan	Network_scan	distributed_network_scan, distributed_network_scan_ICMP_ecrq, distributed_network_scan_SYN, network_scan_ICMP, network_scan_ICMP_ecrq, network_scan_ICMP_netmask, network_scan_ICMP_timestamp, network_scan_ACK, network_scan_RST, network_scan_UDP
	Network_scan_response	network_scan_ICMP_ecrq_ICMP_du_response,. network_scan_ICMP_ecrq_ICMP_ecrp_du_response, network_scan_ICMP_ICMP_response, network_scan_TCP_RST_ACK_ICMP_response, network_scan_UDP_UDP_response
	Port_scan	point_to_point_port_scan_UDP, port_scan_FIN, port_scan_FIN_RST, port_scan_response, port_scan_RST_SYN, port_scan_UDP_ICMP_response, port_scan_SYN_ACK
Alpha flow		Alphfl,malphfl,salphfl,point_to_point,heavy_hitter
IPv6 tunneling		Ipv4gretunel,ipv4_ipv6_tunel
Http		alphflHttp,ptmpHttp,mptpHttp,ptmplaHttp,mptplaHttp
Other		ttl_error,hostout,netout,icmp_error

neural network structure. In [11], A semi-supervised learning intrusion detection system based on RBM is proposed.

Statistic-based methods.[13][14][15] are based on statistical methods to learn the feature distribution in data traffic. In [14], the entropy is introduced as evaluation indicator for results. In [15], this paper used a mathematical model called ASTUTE (A Short-Timescale Uncorrelated Traffic Equilibrium).

2.2 Intrusion Detection Dataset

In this section, three public datasets for machine learning based intrusion detection are introduced.

KDD CUP 1999. KDD CUP 1999 is divided into training set and test set, which are obtained through network traffic collection of 7 weeks and 2 weeks respectively. Each data item in the dataset is a one-dimensional vector, which has 41 fields obtained by feature extraction of fully connected traffic stream. The

format is shown in Figure 2-1. These 41 domains can be specifically divided into four characteristics. The training set contains 24 attacks, and the test set contains 14 attacks shown in Table 2-1. However, this data set has two shortcomings. First, it was produced in 1999. It is too old to be time-sensitive and representative. It contains many outdated attacks and doesn`t cover new emerging attacks. Secondly, each item in the dataset consists of extracted feature, which has little value for intrusion detection in the vehicular network.

```
0,tcp,http,SF,215,45076,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,1,1,0.00,0.00,0.00,
0.00,1.00,0.00,0.00,0,0,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,normal.
```

Figure 2-1 KDD CUP 99 dataset`s data item format

CICIDS2017. CICIDS2017 dataset is composed of traffic packet, which is stored in the format of a .pcap file. The structure of the file is shown in Table 2-4. This dataset contains 9 different types of attack behaviors as is shown in Table 2-2. The dataset is obtained by injecting artificially simulated attack packets into normal network traffic. The dataset collected traffic of 5 days. The disadvantage of it is that the attack traffic is not efficient and may result in an under-fitting problem.

Table 2-4 pcap file format

Global Header	Packet Header	Packet Data	Packet Header	Packet Data	Packet Header	Packet Data	……

MAWILab. MAWILab is also a dataset in packet format, and its attack behavior types are shown in Table 2-3. The dataset has four categories: anomalous, suspicious, notice, and benign. In this paper, only the data traffic that has been confirmed as anomaly is considered. The amount of this dataset is very large, and it is updated almost every day, but its downside is that the attack type covered is insufficient.2.3 Intrusion Detection in Vehicular network.

With the rapid development of the Internet of Vehicles, the car is no longer an independent embedded system, but becomes more like a software platform connecting into the Internet, which not only realizes unmanned technology, such as vehicle to vehicle (V2V), but also obtains rich multimedia content and online updating of vehicle software from cloud server such as Telematics Service Provider (TSP). However, while bringing convenience to customers and depots, these technologies also introduces security issues in the general network into the vehicular network. The CAN bus protocol running in the vehicle system is a low-level protocol, which does not support any inherent security functions, and there is no encryption in the standard CAN implementation, making the CAN network unable to prevent the interception of the middleman data packet. These drawbacks result in the vehicle`s lack of effective defense measures. Therefore, to ensure the safety of vehicles and prevent malicious attacks, building an intrusion detection system in the vehicular network has become a top priority, whose job is to analyze and detect the data stream from the outside and ensure the safety of vehicle

3 TS-VNIDD: the Proposed Dataset

In this section, we will introduce a dataset named TS-VNIDD for machine learning based intrusion detection in a vehicular network environment. Each data item in this dataset is stored in the form of packet. The abnormal traffic and normal traffic are efficient, and there are more than 40 kinds of attack behaviors in abnormal traffic.

3.1 Design Goals

Size. First of all, we hope that the dataset we design can be sufficient in quantity to avoid over-fitting problems caused by the lack of data. Referring to CICIDS2017 and MAWILAB traffic scale, our data volume could not be as small as the former, whose training samples are insufficient, and could not be as large as the latter bloated. Our dataset size will guarantee about 100GB at the byte level, and reaches the number of 500 million at packets level, of which normal traffic accounts for about 60%, abnormal traffic accounts for about 40%. The dataset we provide is not divided to a training set, a validation set, and a test set separately.

Attack behaviors. There are many types of attacks in the vehicular network. According to the purpose of the attack, it can be simply divided into four categories which are message replaying, message spoofing, stealing information, and denial of service. As a dataset for the vehicular network, it should cover attack behaviors on the vehicle platform as much as possible. Starting from the main protocol which the vehicle terminal named the T-box relies on and the type of terminal`s host system, four types of attack behaviors are simulated, namely dos, scan, protocol exploit, and system exploit. The specific classification is described in the following sections.

Scenes. The dataset we would like to design is for the vehicular network environment. As is shown in Figure 3-1, traffic from TSP, cloud server, users and malicious attackers, through the Internet and then is sent to the vehicle. Before entering the protected parts of the vehicle, it will pass through the vehicle terminal which will perform data processing according to the relevant protocols of the vehicular network like TSP&OTA. Then the data stream will enter the interior of the vehicle. We perform vehicular network traffic simulation based on these protocols and get the dataset named TS-VNIDD.

3.2 Design Methods

There are two popular methods that are used to build a dataset. One is packet capturing and the other is simulation. Package capturing sniffs and intercepts the data traffic transmitted over the real network to achieve the purpose of obtaining packets by using some

equipment or software, like sniffer, wireshark. Simulation is to artificially build a local area network, in which several machines are arranged, and these machines can generate a variety of traffic depending on the needs of the task.

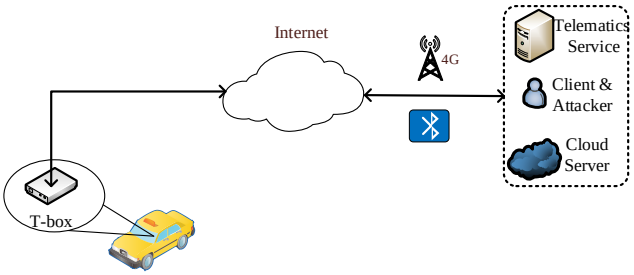

Figure 3-1 Scenes of vehicular network

The scene we are facing is the vehicular network where data is very related to privacy and interests. Because these data are from the communication inside the vehicle, the vehicle factory is paying much attention to the confidentiality of these data, and is not willing to publish these data. We are cooperating with SAIC Motor, and they are always maintaining a cautious attitude towards the provision of data. Therefore, it is quite unrealistic to build a dataset by capturing packets in real vehicular network. We use simulation as a means of data traffic generation. Based on the advantages of our cooperation with SAIC Motor, we can simulate vehicle behavior and various attack behaviors based on corresponding TSP&OTA protocol of vehicular network to generate perfect traffic.

3.3 Design Process

In this section we will introduce the process of dataset building and several attacks that are generated when simulating vehicle behavior. The equipment we used are the vehicle terminal T-box provided by SAIC Motor, several computers, as well as sniffer.

Introduction to the protocol. TSP&OTA protocol is a vehicle terminal protocol, TSP is Telematics Service Provider which can provide navigation information, and multimedia content services. OTA is Over The Air technology, which can be used to update software in the vehicle online. This protocol is responsible for the communication between vehicles and the cloud platform to build the vehicular network. The protocol hierarchy of the vehicular network is based on the ISO/OSI model, but the implementation is different. Its header also has fields like packet length, version number, checksum, and etc. The difference is that it adds information such as device ID that is closely related to the vehicle. Based on TSP&OTA protocol, information about the vehicle, such as GPS, fault data information, anti-theft reminder, various device usage information, are sent to the server for analysis, while T-box remotely receives some resources sent from the server, like multimedia resources, upgrade files. It became one of the most significant protocol in vehicular network.

Introduction to attack behaviors. In the process of building a dataset, we simulated several types of attack behaviors, as is shown in Table 3-1. There are 7 kinds of attacks in the table, each of which contains several specific types. Brute force is a violent cracking of keys and passwords through constant error trialing. According to the system on T-box, we added attack traffic based on the SSH protocol. Heartbleed is a famous attack that achieves intrusion purposes through exploits of the TLS protocol. TTL error is a vulnerability exploiting the ipv4 protocol. By modifying the value of the TTL field of the packet at the intermediate node, it has a certain impact on the destination host. Alpha flow is a data stream which is a point-to-point traffic with more than 1000 packets, and its purpose is to consume the bandwidth of the host, letting the host's resources run out. Scan can be divided into two types of port scan and network scan. The former is more characteristic, and the latter is general. Port scan is probing all ports of the destination host, if active port is found, it will send a report to the attacker and then take the next step. Network scan is more inclined to describing the scanning of hosts in the network. In this process, it attempts to get information of the desired IP address, and then confirm whether the corresponding host is connected to the Internet to work. The last one is Dos attack, which is designed to take up host bandwidth or resources. Dos attack in general is based on the TCP or UDP protocol. The Dos attack based on the TCP protocol utilizes the three interactions mechanism of protocol to establish a connection with the target host by forging a large number of source hosts, accounting for consuming the resources and memory of the destination host. The Dos attack based on the UDP protocol utilizes the loss-tolerating connections feature of the protocol and sends a large number of UDP packets to the host providing the UDP service to impact the target.

Traffic simulation. As shown in Figure 3-2, we use the T-box as the source of vehicle traffic generation, allowing it to interact and communicate with external servers. Combined with the TSP&OTA protocol, we simulated several kinds of attack traffic mentioned in previous section, and then use the sniffing tool to capture the generated traffic stream.

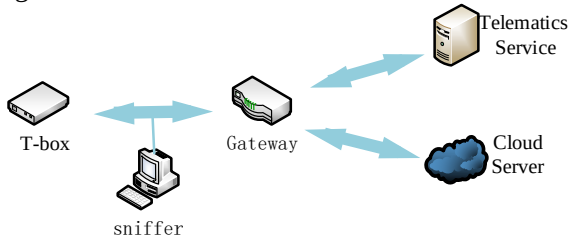

Figure 3-2 Process of traffic simulation

Table 3-1 Classification of attack behavior in TS-VNIDD

Attack Behavior	Specific Type
Brute Force	SSH-Patar
Heartbleed	-
TTL Error	-
Alpha Flow	alphfl,malphfl,salphfl,point_to_point,heavy_hitter
Port Scan	posca,ptpposca,poscaFIN,poscaRST,poscaSYN,poscaUDP
Network Scan	ntscUDP,ptpposcaUDP,ntscSYN,sntscSYN,ntscTCP,ntscFIN,NtscACK, ntscRST,dntscSYN,ntscSYN_ACKresponse,ntscTCP_RSTresponse, ntscUDP_response
Dos	ptpDosSYN,sptpDosSYN,DDosSYN,DDoS_SYN_ACK,DDoSUDP

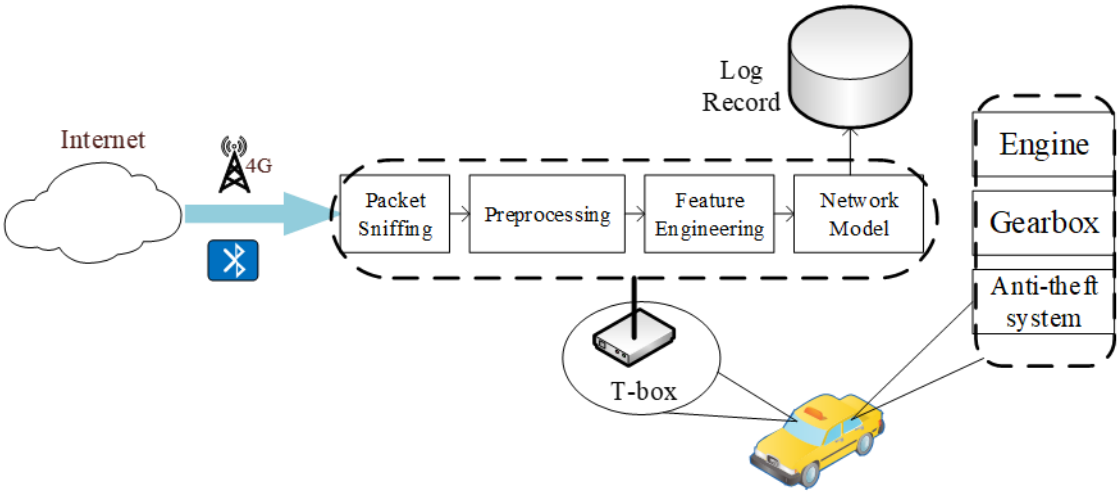

Figure 4-1 vehicular network intrusion detection system

3.4 the Dataset Characteristics

Compared to the previous dataset, the dataset TS-VNIDD has the following characteristics.

1) The type of attack behavior is comprehensive. TS-VNIDD is established based on the attack behavior of CICIDS2017 and MAWILab. there are more than 40 kinds of attack behaviors, covering various common popular attack types.

2) The amount of traffic is sufficient. The data traffic of TS-VNIDD consists of two parts, normal traffic and abnormal traffic, accounting for 60 GB and 40 GB respectively, and the packet amount is of the rank of 100 million. Adequate data traffic allows the model to avoid overfitting problem during training and can also check the robustness of the intrusion detection during the testing phase.

3) Fully reflecting the characteristics of vehicular network traffic. We generated the dataset based on the TSP protocol in the vehicle-server network. On the one hand, TS-VNIDD can be completely distinguished from traffic in a general network such as CICIDS2017 and MAWILab. on the other hand, the trained model will be more sufficient in intrusion detection of the vehicular network.

4 Vehicular network Intrusion Detection System Based on Deep Learning

In order to evaluate our dataset, this paper designed a vehicular network intrusion detection system based on deep learning, which consists of six parts and is shown in Figure 4-1.

Packet sniffing. When the external server like TSP sends data packet to the vehicle through 4G or Bluetooth, the data packet can be collected by the network capturing mechanism of the Linux system in the vehicle terminal named T-box. In the offline model training process, this part is replaced by the TS-VNIDD.

Data analysis and preprocessing. The data packets in TS-VNIDD are stored in hex. The value of each field ranges in [0, 15], so the packet can be normalized making the value of each field distributed between [0, 1], as is in formula (4-1) (4-2) (4-3). Such a process can make the convergence speed of the model training faster, and the final model accuracy higher. The labels in the dataset are discrete values, we use one-hot encoding method to map labels to continuous values in European space.

$$X_{max} = \max\{X_j\} \qquad (4-1)$$

$$X_{min} = \min\{X_j\} \qquad (4-2)$$

$$X_{ij}^{'} = \frac{X_{ij} - X_{min}}{X_{max} - X_{min}} \qquad (4-3)$$

Feature engineering. The features we extracted is packet header which is shown in Figure 4-2. The reason for extracting the packet header is as follows. First, we are building an intrusion detection system in vehicular network, which must ensure real-time and high efficiency. Secondly, the payload portion of the packet is generally encrypted in the Internet, which results in the extracted feature with a poor performance in detection. Finally, through our analysis, most of the network attacks are included in the packet header field. For the above considerations, we chose the data packet header as part of our feature.

Packet Info (16 Byte)
Ethernet Header (14 Byte)
IP Header (20 Byte)
TCP Header (20 Byte)

Figure 4-2 Extracting part of the packet header

Network model. The model we used in the system is a deep neural network. Compared with the general rule-based detection system, deep learning can predict new unknown attacks through existing attack behaviors, and have better detection effects. Moreover, the establishment of the rule set is avoided, and the cost of manual rule constructing is reduced. By constructing a deep neural network, the traffic distribution of the vehicular network can be more fully learned, thereby effectively detect malicious data packets.

5 Benchmark Algorithms and Evaluation Indicators

This paper focuses on machine learning based intrusion detection methods, so the comparison is limited to machine learning methods and does not involve other types of algorithms. We will first describe some mainstream machine learning methods that will be adopted in the benchmarking experiment, and then introduce the evaluation indicators used in this paper.

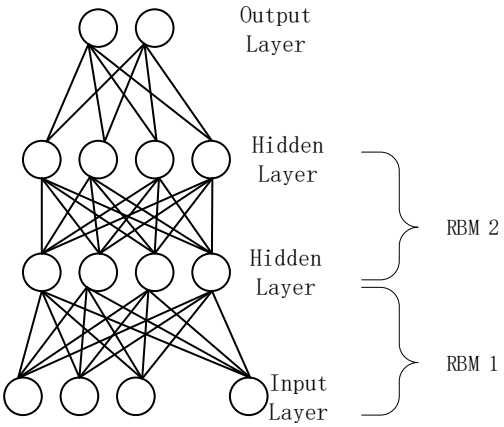

Figure 5-1 DBN network model

5.1 Popular Algorithms

The machine learning methods mentioned below are what we compare in the paper.

DBN. Deep Belief Networks (DBN) [19] is a probability generation model consisting of multiple restricted Boltzmann Machine (RBM). The reason for choosing DBN is that the distribution of input traffic is unknown. In order to be able to make the model fit the input traffic as much as possible, we can use the energy model to solve this problem. The energy model provides objective function for unsupervised learning, making the distribution of input data feasible. all probability distribution can be transformed into an energy-based model. The Boltzmann network is a probability generation model. According to the energy model, the joint distribution between the visual node and the hidden node can be established to obtain the objective function and the target solution. The model is shown in Figure 5-1.

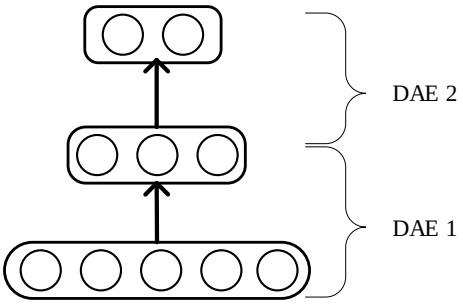

Figure 5-3 SdA network model [30]

CNN. Convolutional neural network [20] is shown in Figure 5-2. CNN utilizes spatial local correlation by implementing a local connection pattern between neurons in adjacent layers. It has two important operations, one is convolution and the other is pooling. Through the convolution operation, more local information can be discovered. The dimension of the Feature Map can be effectively reduced by the pooling. The pooling can also enhance the robustness of the network. When each unit of input data in the adjacent domain is slightly displaced, the output of the pooling layer is unchanged.

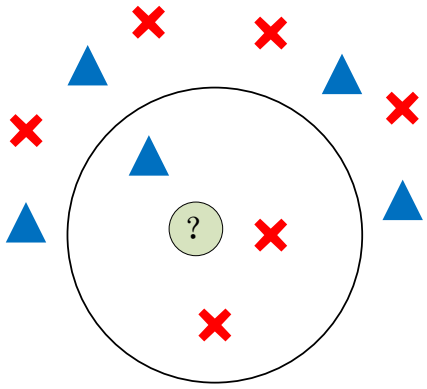

Figure 5-4 KNN method`s model

SdA. Stacked Denoising Autoencoder [21] is similar

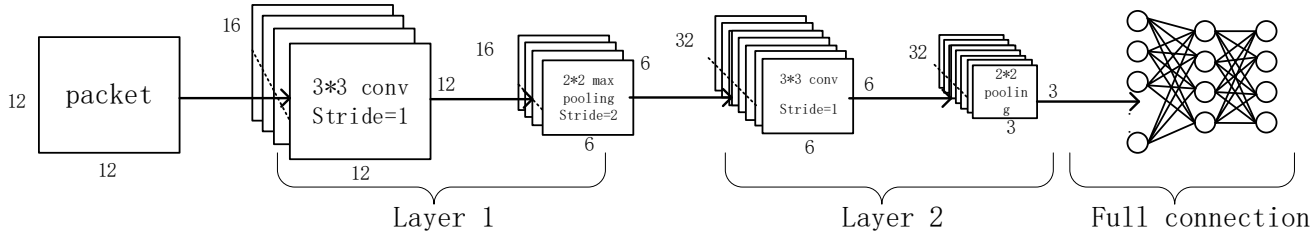

Figure 5-2 CNN network model

in structure to DBN except that each layer of RBM is replaced by Denoising Autoencoder (DAE), as is shown in Figure 5-3. It achieves the goal of learning robust features by introducing random noise into the visible layer of the network.

KNN. K-Nearest Neighbor [22] is one of the simplest machine learning methods. It does not require a training process. As is shown in Figure 5-4, it only selects k samples from the training sample set closest to the test sample in distance. The category of highest frequency occurs in the k samples will be selected as the label of test sample.

Among the several algorithms introduced above, DBN, CNN and SdA are three kinds of deep neural networks. KNN is a general machine learning method.

5.2 Specific Implementation Details

In this section, we will describe the implementation details and settings of the above algorithms.

1) DBN, SdA. We use the C++ interface provided by code[3]. The number of hidden layers is 2, the number of neural units in hidden layers are {80, 40}, and the training data in each batch is 400 items. We totally trained the model 10 epochs.

2) CNN. We use a GPU-based python interface with hidden layers of 2, convolution kernel size of 3*3, and pooling window size of 2*2. Training data in each batch is 200 items. We totally trained the model 10 epochs.

3) KNN. We use the python interface provided by code[4], and the value of K is 5.

5.3 Dataset Split

In this paper, we benchmarked several algorithms described above on the CICIDS2017, MAWILab and TS-VNIDD. Each dataset consists of a training set, a verification set and a testing set, and the percentage of normal data and anomaly data in set are 50% and 50% respectively. The following is a description of the division of the dataset.

1) CICIDS2017. We sampled 15GB of this dataset. We take 10GB as the training set, and the rest of 5GB is divided into two parts, namely the verification set and the testing set.

2) MAWILab. We take the latest part from the dataset about 15GB, and take 10GB as the training set, then the rest are divided into verification set and testing set.

3) TS-VNIDD. The dataset size is about 100GB, 10GB is taken as the training set, and 5GB is divided into the verification set and the testing set.

5.4 Evaluation Indicators

In this section, we have introduced some indicators for evaluation of methods` performance. Figure 5-3 shows the confusion matrix [28]. In this matrix, there are 4 indicators.

TP. Detects abnormal behavior as abnormal behavior, it`s correct detection.

FN: detects abnormal behavior as normal behavior, it`s type-2 error.

FP: detects normal behavior as abnormal behavior, it`s type-1 error.

TN: detects normal behavior as normal behavior, it`s correct detection.

Based on this information, we can get the following evaluation indicators.

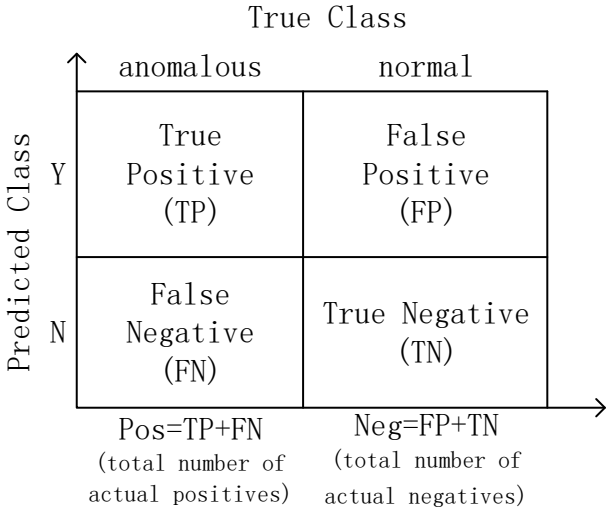

Figure 5-3 Confusion matrix

$$\mathrm{TPR} = \frac{TP}{Pos} = \frac{TP}{TP + FN} \quad (5-1)$$

$$\mathrm{FPR} = \frac{FP}{Neg} = \frac{FP}{FP + TN} \quad (5-2)$$

$$\mathrm{TNR} = \frac{TN}{Neg} = \frac{TN}{FP + TN} = 1 - FPR \quad (5-3)$$

$$\mathrm{FNR} = \frac{FN}{Pos} = \frac{FN}{TP + FN} = 1 - TPR \quad (5-4)$$

3. https://github.com/yusugomori/DeepLearning

4. https://github.com/wepe/MachineLearning

Table 6-1 Evaluation of various methods on three datasets

	CICIDS2017				MAWILab				TS-VNIDD			
	DBN	SdA	CNN	KNN	DBN	SdA	CNN	KNN	DBN	SdA	CNN	KNN
precision	0.9061	0.9372	**0.9647**	0.9646	0.7511	0.8011	**0.9674**	0.9464	0.7860	0.7761	**0.9547**	0.7720
recall	**0.9993**	0.9641	0.9807	0.8772	0.8079	0.8032	**0.8704**	0.7935	0.9089	0.9207	**0.9657**	0.7607
F-score	0.9505	0.9505	**0.9726**	0.9189	0.7785	0.8021	**0.9164**	0.8632	0.8430	0.8423	**0.9602**	0.7663
accuracy	0.9479	0.9497	**0.9724**	0.9225	0.7701	0.8019	**0.9206**	0.8743	0.8307	0.8276	**0.9599**	0.7680

$$\text{Precision} = \frac{TP}{TP+FP} \quad (5-5)$$

$$\text{TPR} = \text{Recall} = \frac{TP}{Pos} = \frac{TP}{TP+FN} \quad (5-6)$$

$$\text{F} - \text{measure} = \frac{2}{\frac{1}{Precision} + \frac{1}{Recall}} \quad (5-7)$$

$$\text{Accuracy} = \frac{TP+TN}{Pos+Neg} \quad (5-8)$$

In addition, we introduced Receiver Operating Characteristics (ROC) for result evaluation. The higher the curve locates, the better the detection ability is.

6 Experimental Results

In this section, we analyzed the experimental results and discussed the effects of various factors on the detection performance, such as datasets, training set sizes, and different machine learning detection methods.

6.1 Comparison of Detection Methods on Various Datasets

We benchmarked the performance of several methods such as DBN, SdA, CNN, KNN, on the three different datasets of CICIDS2017, MAWILab and TS-VNIDD. In particular, KNN has no training process due to its theory. The evaluation results are shown in Table 6-1. Figure 6-1 shows the ROC curve of these method on various datasets. By analyzing the Table and the ROC curve, we can get the following conclusions.

First, the CNN methods achieved the best performance among the baselines on TS-VNIDD which results from its powerful local feature extraction. Compared to other methods, CNN can obtain the subtle change in traffic by convolution and max pooling, while other methods, such as DBN and SdA, implement pre-train process directly; KNN even has no the model-training stage, only depending on the distance among traffic.

Second, among these three datasets, the four methods have the best evaluation results on CICIDS2017, while MAWILAB are the worst, and the results on TS-VNIDD are close to MAWILAB. We analyzed the results and got the following reasons. In CICIDS2017, the attack data traffic was simulated by the deployed machine. This simulation behavior is much more deliberate than the real attack behavior. Therefore, distribution of artificial traffic is unnatural compared to that of real network traffic, which makes it easy to identify anomalies in traffic by detection algorithms. In MAWILAB, traffic is collected entirely from the real network, and the traffic is labelled by several detectors during the collection process.

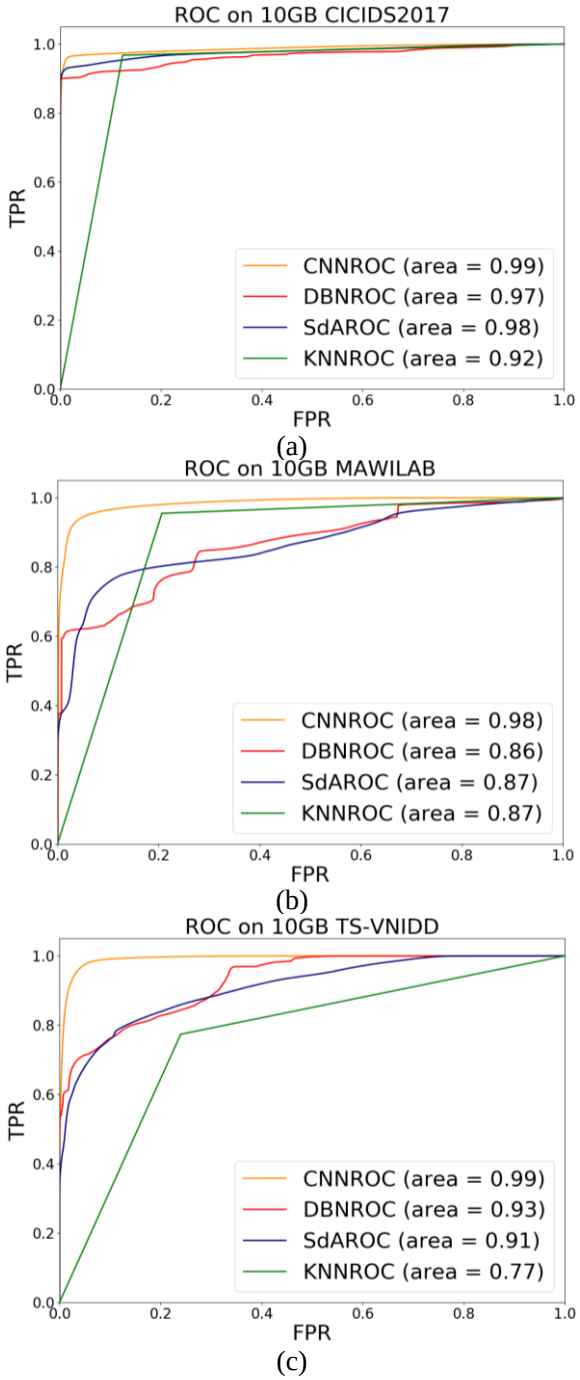

Figure 6-1 (a) the ROC curve of these four methods on various datasets. (a) CICIDS2017;(b) MAWILAB;(c) TS-VNIDD

Table 6-3. Result of TS-VNIDD by CNN

(a) Evaluation of multi classification on TS-VNIDD by CNN

precision	recall	F-score	accuracy
0.9091	0.9110	0.9100	0.9338

(b) Recall of different attack behaviors on TS-VNIDD

ttl_error	Dos	alphfl	portScan	ntscUDP	ntscTCP	sshPatator	heartBleed
0.0	0.6518	0.3648	0.2809	0.7227	0.7339	0.6446	0.0

Because it is real traffic, in which the attack behavior is more concealed, making it difficult to detect attacks. Based on such conditions, a considerable number of data labels in MAWILAB may be incorrect. As a result, the classification effect of models trained with MAWILAB in our experiments was not ideal compared to results on CICIDS2017 by same methods. Compared to the former two datasets, the result on TS-VNIDD is close to MAWILAB, because this dataset has a more comprehensive coverage on attack behaviors, and traffic of it is captured on a real vehicular network on T-box. Therefore, to obtain an excellent performance on our dataset is a tough job. Namely, the TS-VNIDD dataset can be viewed as the most challenging one among these three datasets in vehicular network.

Table 6-2 Evaluation of four methods on 15GB TS-VNIDD

	DBN	SdA	CNN	KNN
precision	0.7784	0.7839	0.9908	0.9722
recall	0.9247	0.9184	0.9728	0.2410
F-score	0.8453	0.8458	0.9817	0.3863
accuracy	0.8307	0.8326	0.9819	0.6171

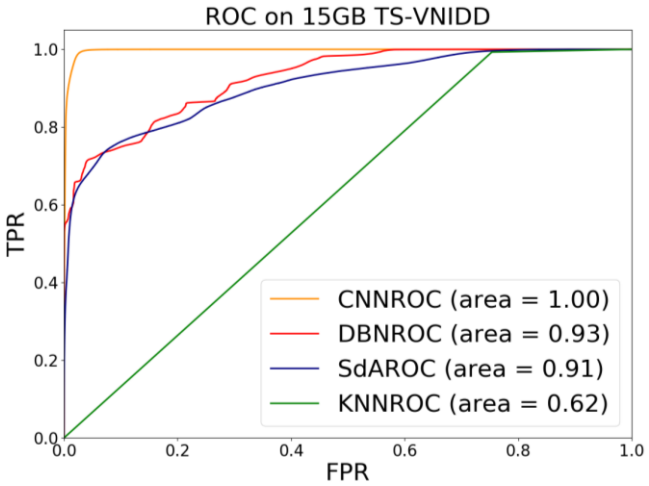

Figure 6-2 the ROC curve of these four methods on TS-VNIDD of 15GB

6.2 the Effect of Different Amount of Training Sets

In this section, we will evaluate whether using a larger training dataset will greatly improve the performance of the detection method. We used 10GB, 15GB of TS-VNIDD for training in the experiment. The experimental results of 10GB are shown in Figure 6-1 and Table 6-1(C), and Table 6-2 as well as Figure 6-2 show the result of 15GB.

As is shown in Figure 6-1(c) and 6-2, the orange lines (CNN) are always in the top, the green lines (KNN) are in the bottom, and the blue lines (SdA) as well as the red lines (DBN) are in the middle. In Table 6-1 and Table 6-2, when the training set increases from 10GB to 15GB, average accuracy of CNN improved from 95.99% to 98.19%, DBN keep constant, and SdA improved from 82.76% to 83.26%.

From the comparison of the above results, we can conclude that sufficient training data can ensure performance of attack detection capability of the intrusion detection system. TS-VNIDD's large scale training data is able to guarantee a reliable model.

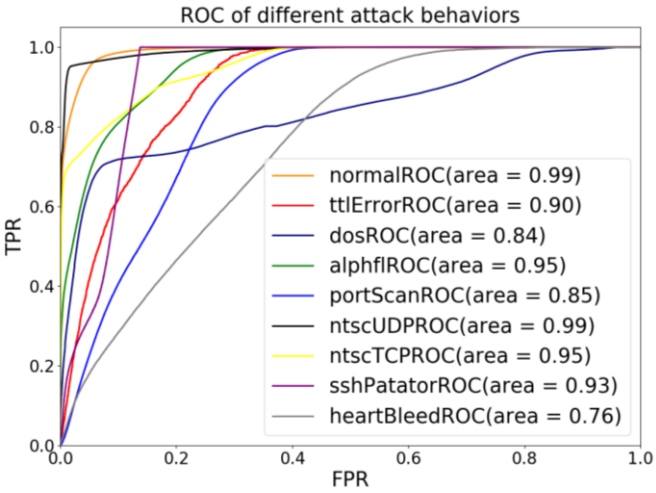

Figure 6-3 ROC curve of different attack behaviors in TS-VNIDD

6.3 Detection Evaluation on TS-VNIDD

In this section, we evaluated the test results on TS-VNIDD. We use the CNN network to extract the data packet header as a feature and conduct multi-classification task on the attack behavior. The experimental results are shown in Table 6-3 and Figure 6-3. We can draw the following conclusions.

As is shown in Table 6-3(b), two attack behaviors, ttl_error and heartBleed, got a recall rate of 0.0. We analyzed the reason and concluded that these two behaviors are of about 5% in our training set which caused an under-fitting problem. Besides these two, the rest also obtained a result with great room for improvement.

Compared to two-classification job, our intrusion detection system obtained an unsatisfactory result in multi classification. In other word, a multi-classification detection on TS-VNIDD is a tough job. The reason are as follows. First, this dataset covers varieties of attack behaviors, making detection of a particular behavior more difficult. Second, its traffic is captured from the vehicle-sever network, which accounts for a more general distribution and the more concealed abnormal traffic.

7 Conclusions

In this paper, we introduce a large-scale machine learning based vehicular network intrusion detection dataset called TS-VNIDD, which provides a more realistic benchmark in the vehicular network scenario compared to other datasets. In the experiment, we applied several machine learning methods for performance evaluation.

According to the experiment result, we still have a long way to go to achieve better detection performance on this dataset. In the future work, we hope to continue to analyze and organize the traffic to expand our dataset, keeping it large-scale and a comprehensive coverage on attack behaviors. Secondly, in the multi-classification detection task, we need to constantly improve the performance of the method on our dataset and propose a vehicular network intrusion detection system based on deep learning. Finally, we expect this paper and the new dataset will motivate more insightful works.

8 Conflict of Interests

We declare that there is no conflict of interests regarding the publication of this paper. We declare that we have no financial and personal relationships with other people or organizations that can inappropriately influence our work.

9 Acknowledgment

The authors would like to thank the anonymous referees for their valuable comments and helpful suggestions. The work is supported by the National Key Research and Development Program of China (No.2016YFB0800402) and the National Natural Science Foundation of China (No.U1836204, No.U1536207).

10 References

[1] Javaid, Ahmad, et al. "A deep learning approach for network intrusion detection system." Proceedings of the 9th EAI International Conference on Bio-inspired Information and Communications Technologies (formerly BIONETICS). ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2016.

[2] Shone, Nathan, et al. "A deep learning approach to network intrusion detection." IEEE Transactions on Emerging Topics in Computational Intelligence 2.1 (2018): 41-50.

[3] Ma, Tao, et al. "A hybrid spectral clustering and deep neural network ensemble algorithm for intrusion detection in sensor networks." Sensors 16.10 (2016): 1701.

[4] Potluri, Sasanka, and Christian Diedrich. "Accelerated deep neural networks for enhanced intrusion detection system." 2016 IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA). IEEE, 2016.

[5] Wu, Chao, Yike Guo, and Yajie Ma. "Adaptive anomalies detection with deep network." Procee ding of the Seventh International Conference on Adaptive and Self-Adaptive Systems and Applications. 2015.

[6] Alom, Md Zahangir, VenkataRamesh Bontupalli, and Tarek M. Taha. "Intrusion detection using deep belief networks." 2015 National Aerospace and Electronics Conference (NAECON). IEEE, 2015.

[7] Lee, Honglak, et al. "Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations." Proceedings of the 26th annual international conference on machine learning. ACM, 2009.

[8] Diro, Abebe Abeshu, and Naveen Chilamkurti. "Distributed attack detection scheme using deep learning approach for Internet of Things." Future Generation Computer Systems 82 (2018): 761-768.

[9] Kang, Min-Joo, and Je-Won Kang. "Intrusion detection system using deep neural network for in-vehicle network security." PloS one 11.6 (2016): e0155781.

[10] Kim, Jin, et al. "Method of intrusion detection using deep neural network." 2017 IEEE International Conference on Big Data and Sm art Computing (BigComp). IEEE, 2017.

[11] Fiore, Ugo, et al. "Network anomaly detection with the restricted Boltzmann machine." Neurocomputing 122 (2013): 13-23.

[12] Alrawashdeh, Khaled, and Carla Purdy. "Toward an online anomaly intrusion detection system based on deep learning." 2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA). IEEE, 2016.

[13] Lakhina, Anukool, Mark Crovella, and Christophe Diot. "Diagnosing network-wide traffic anomalies." ACM SIGCOMM computer communication review. Vol. 34. No. 4. ACM, 2004.

[14] Lakhina, Anukool, Mark Crovella, and Christophe Diot. "Mining anomalies using traffic feature distributions." ACM SIGCOMM computer communication review. Vol. 35. No. 4. ACM, 2005.

[15] Silveira, Fernando, et al. "ASTUTE: Detecting a different class of traffic anomalies." ACM SIGCOMM Computer Communication Review 41.4 (2011): 267-278.

[16] Iman Sharafaldin, Arash Habibi Lashkar i, and Ali A. Ghorbani, “Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization”, 4th International Conference on Information Systems Security and Privacy (ICISSP), Portugal, January 2018

[17] R. Fontugne, P. Borgnat, P. Abry, K. Fukuda. "MAWILab: Combining diverse anomaly detectors for automated anomaly labeling and performance benchmarking". ACM CoNEXT 2010. Philadelphia, PA. December 2010.

[18] Hinton, Geoffrey E. "Deep belief networks." Scholarpedia 4.5 (2009): 5947.

[19] Convolutional Neural Networks (LeNet) - DeepLearning 0.1 documentation. DeepLearning 0.1. LISA Lab. [31 August 2013].

[20] Vincent P, Larochelle H, Lajoie I, et al. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion[J]. Journal of machine learning research, 2010, 11(Dec): 3371-3408.

[21] Han, Eui-Hong Sam, George Karypis, and Vipin Kumar. "Text categorization using weight adjusted k-nearest neighbor classification." Pacific-asia conference on knowledge discovery and data mining . Springer, Berlin,

Heidelberg, 2001.
[22] Bhuyan, Monowar H., Dhruba Kumar Bhattacharyya, and Jugal K. Kalita. "Network anomaly detection: methods, systems and tools." Ieee communications surveys & tutorials 16.1 (2014): 303-336.
[23] Denning, Dorothy E. "An intrusion-detection model." IEEE Transactions on software engineering 2 (1987): 222-232.
[24] Ghorbani, Ali A., Wei Lu, and Mahbod Tavallaee. Network intrusion detection and prevention: concepts and techniques. Vol. 47. Springer Science & Business Media, 2009.
[25] Panda, Mrutyunjaya, and Manas Ranjan Patra. "Network intrusion detection using naive bayes." International journal of computer science and network security 7.12 (2007): 258-263.
[26] Li, Wei. "Using genetic algorithm for network intrusion detection." Proceedings of the United States Department of Energy Cyber Security Group 1 (2004): 1-8.
[27] Wang, Jun, et al. "A real-time intrusion detection system based on PSO-SVM." Proceedings. The 2009 International Workshop on Information Security and Application (IWISA 2009). Academy Publisher, 2009.
[28] Abbes, Tarek, Adel Bouhoula, and Michaël Rusinowitch. "Protocol analysis in intrusion detection using decision tree." International Conference on Information Technology: Coding and Computing, 2004. Proceedings. ITCC 2004.. Vol. 1. IEEE, 2004.
[29] Yin, Chuanlong, et al. "A deep learning approach for intrusion detection using recurrent neural networks." Ieee Access 5 (2017): 21954-21961.
[30] Vincent, Pascal, et al. "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion." Journal of machine learning research 11.Dec (2010): 3371-3408.