2019 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom)

An Internet of Vehicles intrusion detection system based on a convolutional neural network

Ruxiang Peng[1], Weishi Li[2], Tao Yang[1], Huafeng Kong[3*]
1,The Third Research Institute of the Ministry of Public Security
2,Beijing University of Posts and Telecommunications
3,Wuhan Business Universityline
Shanghai, China
e-mail: robin_kong@qq.com

***Abstract*—With the continuous development of the Internet of Vehicles, vehicles are no longer isolated nodes, but become a node in the car network. The open Internet will introduce traditional security issues into the Internet of Things. In order to ensure the safety of the networked cars, we hope to set up an intrusion detection system (IDS) on the vehicle terminal to detect and intercept network attacks. In our work, we designed an intrusion detection system for the Internet of Vehicles based on a convolutional neural network, which can run in a low-powered embedded vehicle terminal to monitor the data in the car network in real time. Moreover, for the case of packet encryption in some car networks, we have also designed a separate version for intrusion detection by analyzing the packet header. Experiments have shown that our system can guarantee high accuracy detection at low latency for attack traffic.**

Keywords- Information security; Internet of Vehicle; Convolutional neural network (CNN); Intrusion detection system (IDS)

I. INTRODUCTION

With the continuous development of intelligent transportation system, the Internet of Vehicles system has become increasingly mature. In the Internet of Vehicles, vehicles build dynamic ad-hoc networks with other vehicles in the network, road infrastructure, and even pedestrians and passengers in traffic by their built-in on-board computers and on-board terminals to providing real-time traffic to management, thereby making it convenient for the management department to coordinate management.

There is no doubt that the development of the Internet of Vehicles will bring endless convenience to our lives. However, at the same time, the safety of the Internet of Vehicles cannot be neglected.

Since the Internet of Vehicles is mostly designed to networks suitable for vehicle-to-vehicle communication (V2V) and vehicle-to-road communication (V2R), there is usually no reliable infrastructure support. The mode lacking in central control and supervision and the nature of wireless ad-hoc network propagation sharing lead to the existence of security problems not only in the traditional Internet and but also in wireless sensor networks.

Typical attacks in the current Internet of Vehicles include Distributed Denial of Service (DDoS) attack, fake information attack[16], Sybil Attack[17], wormhole attack[18], black hole attacks[19], information stealing and so on.

The above attacks also exist in the traditional Internet or wireless sensor networks, but the damage may be much smaller than the Internet of Vehicles. For example, since most traditional wireless sensor networks aim at information collection, when they are attacked or controlled by hackers, the most serious consequence is probably information loss or information error which will not cause large-scale losses. However, the Internet of Vehicles is different. For example, if the Internet of Vehicles encounters Sybil Attack, the hacker can simulate many virtual vehicle nodes at a certain location. At this time, the vehicle management center and a single vehicle node on the road will receive an alarm similar to traffic congestion. Then the vehicle management center may adjust the time of the traffic lights and the vehicle will automatically stop to avoid the virtual vehicles that does not exist in the front, which is very dangerous.

Most of the attacks in the Internet of Vehicles are remote attacks, that is, sending malicious information to the vehicle nodes through malicious traffic. In order to detect and prevent such attacks, the researchers hope to install an intrusion detection system (IDS) on the on-board vehicle terminal (the gateway device of the vehicle node) to intercept the attack traffic outside the vehicle node.

As the gateway system of the vehicle node, the on-board vehicle terminal generally only undertakes the transmitting function, so the design and the equipment are relatively simple. The general on-board vehicle terminal uses the Linux embedded system armv7l or the Raspberry Pi system, of which storage capacity are usually only a few hundred MB, leading that the large IDS system is difficult to operate.

Therefore, we design and develop an IDS on vehicle terminal, an intrusion detection system of the Internet of Vehicles based on convolutional neural network which implement intrusion detection of traffic flow on the on-board vehicle terminal to ensure vehicles safety.

978-1-7281-4328-6/19/$31.00 ©2019 IEEE
DOI 10.1109/ISPA-BDCloud-SustainCom-SocialCom48970.2019.00234

II. RELATED WORK

Wherever Times is specified, Times Roman or Times New Roman may be used. If neither is available on your word processor, please use the font closest in appearance to Times. Avoid using bit-mapped fonts if possible. True-Type 1 or Open Type fonts are preferred. Please embed symbol fonts, as well, for math, etc.

In this section, we investigate proposed intrusion detection technologies, including traditional intrusion detection technology, machine learning-based intrusion detection technology, and intrusion detection technology in vehicle networking, and analyze their advantages and disadvantages.

Intrusion Detection Technology Based on Machine Learning

With the continuous development of machine learning, the advantages of machine learning for classification and aggregation problems are discovered by security personnel and introduced into the analysis of abnormal traffic.

In some early studies, the researchers tried some simple machine learning algorithms that performed well in classification clustering problems in other fields, such as K-nearest neighbor (k-NN) [6], support vector machine (SVM) [7], Self-organizing maps (SOM) [9], etc., have achieved good results on data sets such as KDD99, NSL-KDD, and DARPA. However, these data sets are very old, and both normal data and attack data are very simple, making it difficult to simulate today's complex network environment. Our experiments also demonstrate that using these algorithms to analyze malicious traffic in newer data sets makes it difficult to achieve the desired results.

Among various machine learning models, neural networks have become an increasingly popular solution for network intrusion detection systems. Their ability to learn complex patterns and behaviors makes them a suitable solution for distinguishing between normal traffic and network attacks. The Convolutional Neural Network (CNN) uses the original data directly as input to the network, avoiding feature extraction and image reconstruction, reducing the number of parameters, and effectively reducing the amount of data processing in the process of extracting features more accurately and abstractly. CNN has shown great effectiveness in the field of image recognition. In [10], Zhanyi Wang pointed out that CNN has a significant effect on the identification of traffic images. For some protocols, the CNN network can get good results through rapid training. In [8], JIA Fan et al. used a multi-layer "convolution-downsampling layer" to extract more accurate features and classified them by a multi-layer perceptron, which showed in the experiment of detecting the KDD99 data set. The superiority of classical detection algorithms compared to SVM.

As a branch of wireless sensor network, the Internet of Vehicles contains many characteristics of wireless sensor networks. For example, it does not have a fixed network topology, and it builds a dynamic network with the running of the vehicle. Similar to wireless sensors, they are lightweight nodes that are difficult to carry intrusion detection systems with heavy loads. Therefore, our research on vehicle network intrusion detection technology begins with the research on intrusion detection systems in wireless sensor networks.

Researchers have matured on intrusion detection systems in wireless sensor networks. In [11], the authors propose a new Bayesian hybrid detection-based defense strategy that uses a weight monitoring system to estimate the opponent's state of motion and uses a weight monitoring system as a last resort.

For solving the problem of low computing power in wireless sensor networks, researchers generally use the method of network sharing computing power. For example, in [12], the author places the monitoring module locally on the wireless sensor, and the detection and processing module consists of several in the network. The nodes are completed together. Thus, when a node discovers attack data, the entire network can be linked to handle the attack. This kind of thinking can solve the problem of low computing power to a certain extent, but the monitoring module still needs a single node to bear itself.

With the development of the Internet of Vehicles, researchers have applied the idea of wireless sensor networks to the Internet of Vehicles, and introduced the idea of machine learning to improve.

III. FRAMEWORK

A. OverView

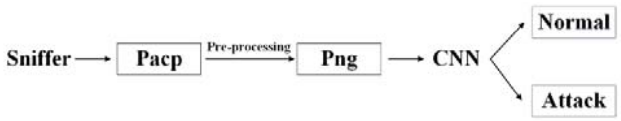

Fig. 1 The Framework of IDS on the vehicle terminal

IDS on the vehicle terminal is mainly composed of three modules: data collection module, data preprocessing module and detection module. The data collection module is divided into two parts, static data read from data set and dynamic data sniffed by Sniffer based on libcap in bypass according to different processes of training and testing. In the pre-processing module, we convert the data packets from data collection module into a traffic matrix byte by byte and save it as a png format as the input of convolutional neural network (CNN). Finally, CNN model determines whether traffic is malicious traffic or not based on its previously trained model. If it is malicious traffic, system will alarm immediately and process in conjunction with the firewall. Otherwise, drop the data drop to save storage space.

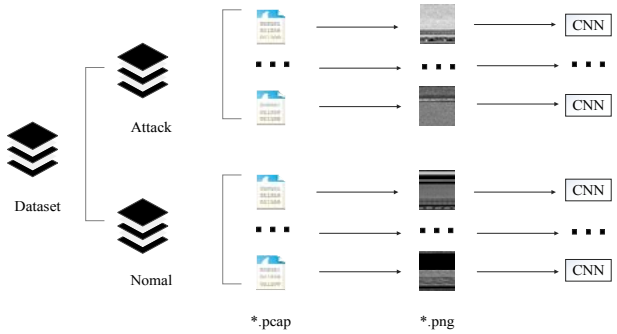

Fig. 2 The Process of Data Preprocessing

The processing of the entire data set preprocessing is shown in Fig.2. In general, we first divide the pcap file in the normal and attack data sets into single packages, then use the information of each byte in pcap file as a pixel in the grayscale image in png format and add the corresponding format head of it. Since the data in the network traffic packet is composed of bytes of which the value ranges from 0 to 255, which is the same as the value range of the pixels in the picture, therefore data normalization is not needed. Finally, we convert every data into a 28bytes*28bytes image in png format as the input to CNN.

In Fig.3, we can clearly distinguish between normal data and abnormal data in terms of packet length, packet header and payload. But these differences are difficult to summarize in language. However, when using CNN model, it can observe the nuances of the pictures and find out the differences between them which can be the key of determining the nature of the data packets.

B. Convolutional Neural Network (CNN)

The convolutional neural network provides an end-to-end learning model. The parameters in the model can be trained by the traditional gradient descent method. The trained convolutional neural network can learn the features in the image and complete the image features. Extraction and classification. As an important research branch in the field of neural networks, the convolutional neural network is characterized in that the characteristics of each layer are obtained by the local region of the upper layer by the convolution kernel of the shared weight. This feature makes the convolutional neural network phase More suitable for learning and expression of image features than other neural network methods.

C. CNN-Entirety Model Structure

In this section, we present a Convolutional Neural Network (CNN) model for traffic quality judgment. Since the data in the Internet of Vehicles includes both encrypted data and non-encrypted data, we hope that IoV-IDS can achieve high accuracy for both types of data. However, since the dataset only has non-encrypted data, encrypting the entire dataset not only needs a lot of manpower and material resources, also has low efficiency.

The CNN-Entirety model consists of the Input layer, the Conv2D-I layer, the AveragePooling-I layer, the Conv2D-II layer, the AveragePooling-II layer, the Flatten layer and the Dense layer. Each Combination makes a different level of analysis of the input layer characteristics. In Combination I, we use a larger convolution kernel to analyze the relationship between two bits that are farther away, such as information in the traffic payload. In Combination II, we used a convolutional layer with a small convolution kernel to extract local features of the traffic image details, such as IP, Port, and so on. After the convolution operation and after the processing of the Pooling layer, the noise can be eliminated, so that CNN can obtain more obvious features and more stable results. Among them, we combine the Conv2D-I layer and the AveragePooling-I layer into a Combination I, and the Conv2DII layer and the AveragePooling-II layer into a Combination II.

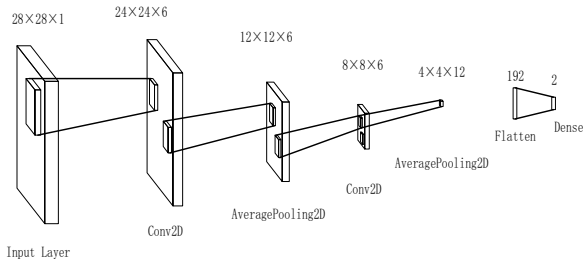

Fig. 3 CNN-Entirety model structure

Each Combination makes a different level of analysis of the input layer characteristics. In Combination I, we use a larger convolution kernel to analyze the relationship between two bits that are farther away, such as information in the traffic payload. In Combination II, we used a convolutional layer with a small convolution kernel to extract local features of the traffic image details, such as IP, Port, and so on. After the convolution operation and after the processing of the Pooling layer, the noise can be eliminated, so that CNN can obtain more obvious features and more stable results.

A convolutional layer consists of Eq.1, where f is the length of the convolution kernel, s is stride, p is padding, b is bias, w is weight, c is channel, l is layer , L_l is the size of Z_l. The specific relationship is shown in the following formula, $(i,j) \in 0,1,\ldots,l_{L+1}$,

$$l_{L+1} = \frac{L_l + 2p - f}{s} + 1$$
$$Z^{l+1}(i,j) = [Z^l \times w^l](i,f) + b$$
$$Z^{l+1}(i,j) = \sum_{k=1}^{c}\sum_{x=1}^{f}\sum_{y=1}^{f}[Z_k^l(s*i+x, s*j+y) * w_k^{l+1}(x,y)] + b \quad (1)$$

The convolutional layer contains the excitation function (Eq.2) to help express complex features. We use the sigmoid excitation function (Eq.3) and the ReLU excitation function (Eq.4) after the two convolutions.

$$A_{i,f,k}^{l} = f\left(Z_{i,f,k}^{l}\right) \quad (2)$$

$$f(x) = \sigma(x) = \frac{1}{1+e^{-x}} \quad (3)$$

$$f(x) = \mathrm{ReLU}(x) = \begin{cases} 0, x < 0 \\ x, x \geq 0 \end{cases} \quad (4)$$

After feature extraction in the convolutional layer, the output image is passed to the pooling layer for feature selection and information filtering. The pooling layer contains a preset pooling function whose function is to replace the result of a single point in the feature map with the feature graph statistic of its neighboring area. The calculation formula for the pooling layer can be simplified to Eq. 5, where p is the pre-specified parameter, and here we use the average pooling, ie p = 0.5.

$$A_k^l(i,j) = \left[\sum_{x=1}^{f}\sum_{y=1}^{f} A_k^l(s*i+x, s*j+y)^p\right]^{\frac{1}{p}} \quad (5)$$

We also use a backpropagation algorithm to adjust the model parameters. For the loss function E we use the categorical cross entropy algorithm, which is calculated as the Eq.6, where p is the two-dimensional tensor of coding_dist representing the feature. q is true_dist, which represents the symbol vector of the integer.

$$E(p,q) = -\sum_{x} p(x)\log(q(x)) \quad (6)$$

In order to reduce training time and improve the accuracy of gradient descent, we used the RMSprop optimization function (Eq.9,10) to adjust the learning rate. The calculation formula for the RMSprop optimization function is as follows.

$$\Delta w(t) = \propto \Delta w(t-1) - \varepsilon\frac{\partial E}{\partial w}(t) \quad (9)$$

$$w_{t+1} = w_t + \Delta w(t) \quad (10)$$

IV. EXPERIMENTS

In order to satisfy the training of convolutional neural network model and the testing in the actual environment, this paper builds a complex software and hardware environment. The training and updating of the model is set on the PC, and the testing phase in the actual environment is set on the embedded system of the vehicle. The following is a brief introduction of the environment used in the experiment process:

Hardware Environment: PC with an Intel Core i5-8400 CPU (2.8 GHz), RAM is 8 GB, GPU is TITAN(X) (Pascal).

An algorithm server: CPU is 8*Intel(R) Xeon(R) CPU E5-2637 v4@ 3.50 GHz, memory is 12 GB, GPU is TITAN X (Pascal) 1 block, single-precision floating point computing capacity is about llTFlops, GPU video memory size is 12 G.

The vehicle terminal: The CPU of the vehicle terminal is armv7l, the disk size is 128MB, the operating system is Centos, and the cross-compilation environment is gcc-linaro-4.9.4.

Software Environment: The PC operating system is Windows 10 64bit. The required software includes Pycharm, Python 3.

Algorithms Server: The operating system is Ubuntu 14.04. The required software includes CUDA components, Anaconda3, Tensorflow 1.14 and Keras 2.20.

C. EXPERIMENTAL RESULT

In this section, we introduce our experimental results.

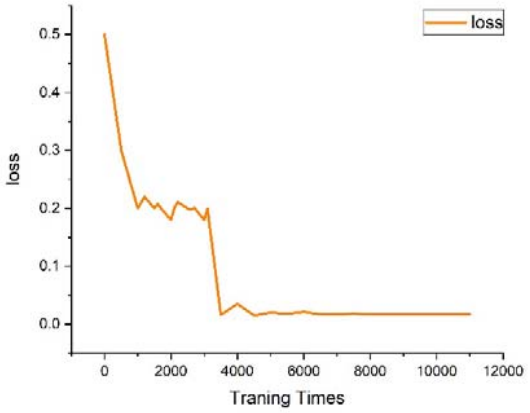

Fig.4 Loss changes during training

Firstly, we train the CNN-Entirety classification model on PC. The Loss changes during the training process are shown in Fig.4. It can be found that in the early stage of training, Loss decreases rapidly, and then achieves the minimum value of Loss at about 0.28. After a period of fluctuation, loss continued to decline and gradually approaches 0. It indicates that we have obtained a more accurate and stable CNN model.

After the training, we tested the model on PC and IDS on the vehicle terminal respectively. The model test on PC is relatively simple. It only needs to use the model to read the static data stored on the hard disk. the test on IDS on the vehicle terminal is Comparatively more complex. the packet sending software on PC side is needed to send data packets to IDS on the vehicle terminal side. IDS on the vehicle terminal uses sniffer to sniff data packets and then carries out packet classification detection.

Table 5: The test results of CNN-Entirety

Total	300,000
Accuracy	99.871%
Normal	149,730
Normal to Attack	35
FRR	0.023%
Attack	150,270
Attack to Normal	3
FAR	0.0019%
Time Effiency on PC	69.356ms/pcap
Time Effiency on vehicle	395ms/pcap

In Table(table) 5, we can see that the accuracy of the CNN-Entirety model is very high, reaching 99.871%, which can meet the needs of intrusion detection in the vehicle network system. However, in terms of usability, on IDS on the vehicle terminal CNN-Entirety needs nearly 0.4 seconds to judge a packet, and it can -not reach the level of real-time detection.

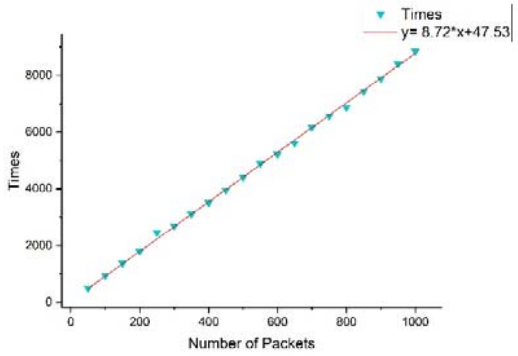

Fig.5 Time efficiency fitting curve of our IDS on PC

In addition, it is difficult to determine the exact time of a single packet because the data in a PC is input in batches.Therefore,we use the linear fitting method (as shown in Fig.5) to get the processing time of a single packet on the PC.

V. CONCLUSION

In order to detect malicious attack traffic information in the Internet of Vehicles, we design and implement a lightweight intrusion detection system IDS on the vehicle terminal based on convolutional neural network. It builds a graphical model of normal network traffic and abnormal network traffic by learning the characteristics of pictured network traffic data (network traffic data in pictures), and uses this model to detect real-time network data. Experiments show that IDS on the vehicle terminal can achieve high detection accuracy in embedded systems with very low computing power. In the future, we hope to introduce a detection mechanism for encrypted traffic, not only to judge the character of encrypted traffic based on the information of packet head, but also to analyze its payload, time and other information, so as to detect encrypted data of the Internet of Vehicles more accurately and comprehensively.

ACKNOWLEDGMENT

This research was supported by National Key Research and Development Program of China (grant No. 2018YFC0830400, grant No.2018YFC0806903), and the Key Lab of Information Network Security at the Third Research Institute of Ministry of Public Security(grant No.C18613)

REFERENCES

[1] H. Goto, Y. Hasegawa, and M. Tanaka, “Efficient Scheduling Focusing on the Duality of MPL Representatives,” Proc. IEEE Symp. Computational Intelligence in Scheduling (SCIS 07), IEEE Press, Dec. 2007, pp. 57-64, doi:10.1109/SCIS.2007.357670.

[2] Xu H, Mueller F, Acharya M, et al. Machine learning enhancedreal-time intrusion detection using timing information[C]//International Workshop on Trustworthy & Real-time Edge Computing for Cyber-Physical Systems. 2018.

[3] Wang Y, Li Z. SQL Injection Detection via Program Tracing and Machine Learning [J]. Lecture Notes in Computer Science, 2012, 7646:264-274. Yi-liang LIU1,Ya-li SHI1,Hao FENG2,Liang-min WANG. Intrusion detection scheme based on neural network in vehicle network.

[4] Verma A, Ranga V. Statistical analysis of CIDDS-001 datasetfor network intrusion detection systems using distance-based machine learning[J]. Procedia Computer Science, 2018, 125: 709-716

[5] Kang M J, Kang J W. Intrusion detection system using deep neural network for in-vehicle network security[J]. PloS one, 2016, 11(6): e0155781.

[6] Wang Y, Meng W, Li W, et al. A fog-based privacy-preserving approach for distributed signature-based intrusion detection[J]. Journal of Parallel and Distributed Computing, 2018, 122: 26-35.

[7] S. S. Roy, A. Mallik, and R. Gulati, “A Deep Learning Based Artificial Neural Network Approach for Intrusion Detection”, International Conference on Mathematics and Computing. Springer, Singapore, pp. 44-53, 2017.

[8] T. Ishitaki, R. Obukata, and T. Oda, “Application of Deep Recurrent Neural Networks for Prediction of User Behavior in Tor Networks”, International Conference on Advanced Information Networking and Applications Workshops. IEEE, pp. 238-243, 2017.

[9] M. J. Kang and J. W. Kang, “Intrusion Detection System Using Deep Neural Network for In-Vehicle Network Security”, Plos One, vol. 11, no. 6, 2016.

[10] Wang W, Sheng Y, Wang J, et al. HAST-IDS: Learning Hierarchical Spatial-Temporal Features Using Deep Neural Networks to Improve Intrusion Detection[J]. IEEE Access, 2018, 6(99):1792-1806.

[11] Wang, Zhanyi. "The applications of deep learning on traffic identification." BlackHat USA (2015): 21-26.

[12] Liu Y, Comaniciu C, Man H. A Bayesian game approach for intrusion detection in wireless ad hoc networks[C]//Proceeding from the 2006 workshop on Game theory for communications and networks. ACM, 2006: 4.

[13] Zhang Y, Lee W. Intrusion detection in wireless ad-hoc networks[C]//Proceedings of the 6th annual international conference on Mobile computing and networking. ACM, 2000: 275-283.