Lightweight Intrusion Detection System for IoT Applications Using Deep Learning Approach

Jiangnan Li
Department of Electrical Engineering and Computer Science
University of Tennessee, Knoxville
Knoxville, USA
jli103@vols.utk.edu

***Abstract*—The security of the IoT system has draw considerable attention. One inherent property of IoT is its open working environment, which makes it vulnerable to cyber various kinds of cyber intrusions and attacks. Intrusion detection system is an effective tool to protect the local system from outside attack. However, different from traditional networks, IoT applications were thought to be constituted by large number of sensors that have limited power, computation capability and storage. Therefore, a lightweight security solution is needed. This project builds an lightweight IDS with high accuracy for IoT applications by using deep learning approaches. The evaluation on KDDCup99 dataset shows that the IDS system can achieve competitive performance with state-in-the-art methods while keeping a simpler structure.**

***Index Terms*—intrusion detection, kddcup99, autoencoder, nerual network**

I. Introduction

The Internet of Things (IoT) has draw more and more attention from both researchers and industries due to its enormous potential applications, and will still be a fast-developing innovation that will change humans live style to a great extent. With the rapid development of IoT, a variety of IoT applications have already been implemented and contribute to our every life. Typical examples are Smart Home, wearable devices, smart phones with sensors, Smart Grid and so on [1].

While the community and relatively technologies of IoT are developing fast, security has always been an important factor and inevitable problem when designing an IoT application. Indicating by the fact that IoT will play a more indispensable role in future society, security problem of IoT is getting more critical. If security problem cannot been appropriately solved, it will definitely restrict the development of IoT.

Technically speaking, IoT is a technology born out of a network, and currently IoT applications are still mainly built on the basis of the Internet. Therefore, IoT also faces the security problems that arise in the Internet in same or different ways, such as bad data intrusion, DoS and so on[2]. Moreover, the lower layer of IoT system was thought to be constructed with enormous cheap sensors that has limited memory, computation capacity and power. Therefore, lightweight security solution is required for IoT applications.

To enhance the security level of IoT and protect it from outside attack, different approaches have been investigated[5][6][7][8]. An intrusion detection system (IDS) is an effective solution to isolate IoT application from outside network and provide the inside system protection. Technical speaking, IDS is deemed as an improved version of firewall which can provide far more powerful protection to the local network systems. In traditional networks, different kinds of IDS was investigated. In recently years, as the deep learning technologies developing fast, technologies such as Artificial Neural Networks, Recurrent Neural Networks are used to build IDS for traditional networks. However, these methods are either with high complexity or relatively low accuracy which makes them inappropriate to IoT applications.

In this project, a lightweight IDS was designed and implemented using deep learning approaches. To reduce the computation capacity and memory requirements, a shallow neural network was designed to act as classifier to determine whether a connection is benign. Moreover, an autoencoder was designed to reduce the number of dimention of input data. The evaluation on the benchmark dataset KDDCup99 demonstrates that the lightweight IDS can provide competitive performance compared with state-in-the-art intrusion detection methods.

The rest of this report is organized as follow: In section II, related work on IDS design will be presented. Then, section III will describe the detailed design of the lightweight IDS for IoT applications. Section IV will introduce the implementation of the IDS. In section V, performance evaluation was presented. Then, future work based on this project will be presented in Section VI. Finally, section VII concludes the project.

II. Related Work

Currently, intrusion detection systems can be divided into two categories: anomaly based and misuse based [3]. Anomaly based IDS statistically models the normal work state of the network, and flags connection which was anomaly to the system. On the other hand, misuse based IDS detects attacks based on previously defined knowledges. It is usually considered that anomaly detection systems can detect the effect of bad behavior, while misuse detection system recognize known bad behavior based preset rules [4].

Although the amount of specific types of cyber attack is large, most of them will fall into several categories with specific characteristics, and this naturally leads to solutions based on deep learning tools. Some previous research using

traditional machine learning algorithms, such as K-Means Clustering, Naive Bayes, and Supportive Vector Machine.

[5] implemented intrusion detection system for wireless sensor network based on KNN classification algorithm. In 2008, [6] presented an intrusion detection system using random forest algorithms. In recently years, methods of implementing typical machine learning algorithms keep improving. In 2015, [7] implemented IDS based on K-Means Clustering, and its overall accuracy evaluated based on KDDCup99 dataset reaches 97.26%. In the same year, IDS based on Naive-Bayes Classification was also presented and reached an accuracy of 97%[8].

[9] used support vector machine to separate illegal network packets from the traffic flow and obtains 95.26% overall accuracy based on the evaluation on KDDCup99 dataset. [10] evaluates different machine learning algorithms and compares their capacities using KDD99 datasets.

As the community and technologies on deep learning developed? IDS based on neural networks was also presented. [11] used Recurrent Neural Network to build IDS and reached an overall accuracy of 97.54% in 2016. One of the most recent Neural Network based IDSes was [12] presented by Roy et al, they built an deep neural network to classify a network connection as benign or malicious, and reach an overall accuracy of 97.76%. There are also hybrid methods that combines typical machine learning algorithm and deep learning approaches together. Putchala et al. firstly used random forest algorithm to determine the most influential feature of the input data, and then designed an Recurrent Neural Network to complete the classification[13]. The evaluation on KDDCup99 shows their method achieves an overall accuracy of 98.91%.

A. Preliminary

1) KDDCup99 Dataset: KDDCup99 dataset was used as the standard dataset for The Third International Knowledge Discovery and Data Mining Tools Competition for intrusion detection. The original dataset was built following the 1998 DARPA Intrusion Detection Evaluation Program which was prepared and managed by MIT Lincoln Labs. The KDDCup99 was a standard version of this dataset. The connection data in the dataset was derived from nine weeks of raw TCP dump data for a local-area network (LAN) simulating a typical U.S. Air Force LAN. Although KDDCup99 have published for nearly twenty years, it is still regarded as the benchmark dataset for intrusion detection research. Currently, KDDCup99 dataset is available on the UCI KDD Archive. In this research, ten percent of the original dataset was used.

KDDcup99 contains connection data and corresponding labels. Each connection data is consist of 41 features, such as duration, protocol_type, service, src_bytes and so on. Some of the features are continuous and numerical, such as duration, hot, num_root. However, there are three features that are symbolic, namely protocol_type, service, and flag. Each connection is labeled as normal or as an attack, with one specific attack type. All attacks fall into four categories. As shown in Table I.

TABLE I
KDDCUP99 ATTACKS AND CATEGORIES

Attacks	Categories
back, land, neptune, pod, smurf, teardrop	DOS
buffer_overflow, loadmodule, perl, rootkit	U2R
ftp_write, guess_passwd, map, multihop pfh, spy, warezclient, waremaster	R2L
ipsweep, nmap, portsweep, satan	PROBE

2) Autoencoder: Autoencoder is an unsupervised learning models of neural network. The simplest structure of an autoencoder can be a feedforward, non-recurrent neural network that similar to multilayer perceptron. An autoencoder is always consist of two parts, namely the encoder and the decoder. The numbers of dimensions of the input layer and output layer of autoencoder are the same. There is always hidden layers connecting the input and the output. Usually we expect the number of dimension of hidden layer to be smaller than input layer. Autoencoder is believed to be an training model that can learn a high level representation of a set of data. Therefore, an autoencoder is the powerful tool to reduce the dimension of input data, and further reduce the computation requirement.

B. Our Contributions

Our contribution can be summarized as follow.

- First, we design and implement an autoencoder followed by a feedforward neural network that can be used as the classifier for intrusion detection.
- Then, we make a table to show how the dimension of hidden layer influences the overall accuracy of the system by trying different network structure, such that readers can select the simplest structure while keeping the overall accuracy at a relatively high level.
- Finally, we compare our work with the state-in-the-art methods.

III. INTRUSION DETECTION SYSTEM DESIGN

A. System Overview

The project aims to build effective Intrusion Detection System for IoT applications. However, since the author does not found any valid public IoT application dataset over the Internet, this project uses KDDCup99 the network connections dataset. Although this project uses general network dataset KDDCup99 for training and evaluation, the author claims that the basic structure of the Intrusion Detection System described in this project can be adapted to most IoT applications.

Since the goal of this project is to design a lightweight Intrusion Detection System with high accuracy, one straightforward methods is to reduce the original data dimensions. Meanwhile, the structure or layer of the neural networks should be as simple as possible. To achieve this, the basic idea of the project is using an antoencoder to reduce the dimension of the input data followed by an full connected neural network to act as a classifier. The autoencoder should be firstly trained and stored. After that, the full connected neural network was trained with

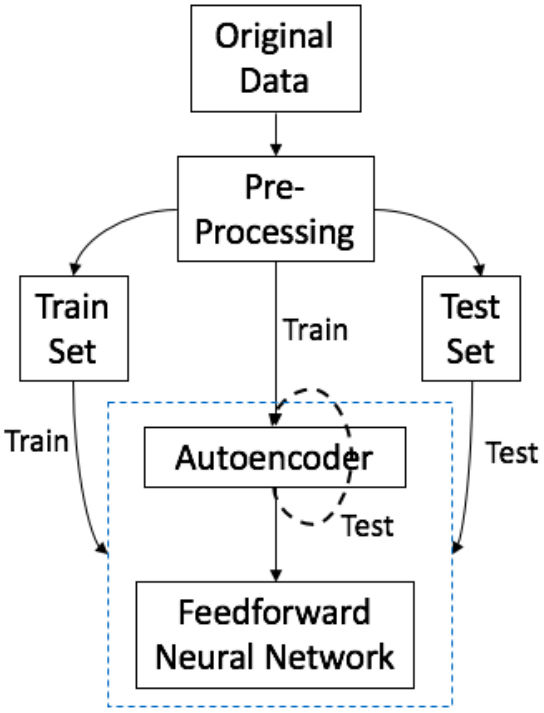

Fig. 1. System Design Overview

the input as the output of the trained autoencoder. The data flow of this project was shown in Figure 1.

B. Data Preprocessing

As previously described, the original KDDCup99 dataset contains several symbolic feature that cannot act as the input for the autoencoder directly, such as protocol_type, and service. Meanwhile, the label for each connection is also symbolic, namely various kinds of attacks. Therefore, the first step is to convert these symbolic features and labels to numeric data. To achieve this, we simply use integers in the range of the total number of different categories of the feature. For example, feature protocol_type contains three different categories, there are udp, icmp and tcp. To numeric these data, we convert all udp to 0, all icmp to 1 and tcp to 2. Beside that, we apply one-hot encoding to transfer attack types to numeric vectors.

After that, to reduce the effect of one feature with large range influence the overall performance of the system, an normalization process of the input data was made. There are different normalization methods such as coefficient of variation and standardized moment. In this work, we select standard score normalization algorithm to normalize the original data.

C. Training of Autoencoder

Autoencoder is a useful application of unsupervised learning technology of neural network. After pro-processing procedure, the dataset was ready for training. There are totally three layers of the autoencoder in this research, namely the input layer, followed by one hidden layer and one output layer. Apparently, the number of dimensions of the input and output layer is set to 41. We use relu as the activation function of the hidden layer and sigmoid of the output layer. The cost function of the autoencoder was ?Mean Square Error?. Figure 2 gives the overview of how we training the autoencoder.

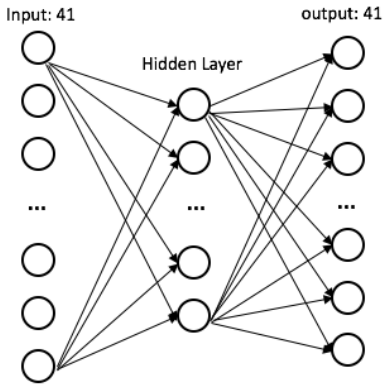

Fig. 2. Autoencoder Training

D. Training of Neural Network Classifier

Neural Network was an powerful tool to do pattern recognition and classification. This project uses and full connected neural network to act as an classifier to label an connection as normal or what kinds of attack it is. There are total three layers of the feedforward network in this project, there are input layer, one hidden layer, and output layer. The dimension of input layer of the classifier was set to the dimension of the autoencoder?s hidden layer, and the dimension of the output layer was set to five due to the one hot encoding of five kinds of labels. The input of the feedforward neural network should be the output of the trained autoencoder. The original data was firstly compressed from 41 dimension to relatively fewer dimensions after the trained encoder, then the highly extracted data was sent to the classifier.

In this project, the activation function of the hidden layer was set to relu and output layer was softmax. The cost function of the feedforward network was ?cross-entropy?. The structure of training the classifier was shown in Figure 3.

IV. Implementation

A. Environment

Dataset: Ten percent KDDCup99 dataset
Computer: hydra4@eecs.utk.edu
Operating system: Red Hat Enterprise Linux 7
Processor: Intel Core i7-6700 @ 3.40GHz (Quad Core)
Memory: 16GB (2x8G) DDR4 SDRAM @ 2133MHz
GPU: GeForce GT 745, 4GB, HDMI
Programming Language: Python 2.7
Library Used: Keras, Tensorflow, Pandas, Scikit-Learn

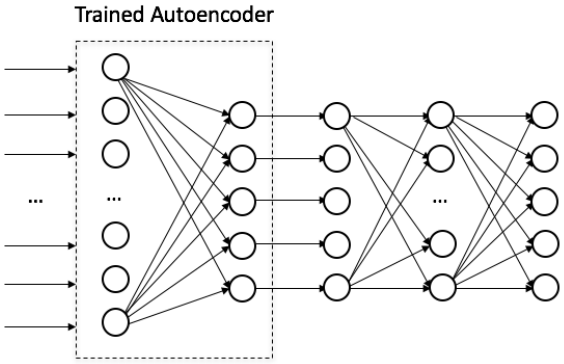

Fig. 3. Autoencoder Training

B. Implementation

In order to find the appropriate structure for intrusion detection system, dimensions, we tried different hidden layers for both autoencoder and feedforward classifier.

Before training process, we used StandardScaler() function provided by sklearn library for pre-processing. During each round(different dimension of hidden layers), to train the autoencoder, we used ?adadelta? optimizer provided by Keras library. The first twenty thousand connection data are used for validation at each training step. In this project, an autoencoder was set to be trained for 10 epochs with the batch size set to 256. The trained autoencoder was then used to compress the input dimension of the original data for the feedforward network.

To train the classifier, the processed data was randomly split to training data and test data with proportion set to 70% and 30% respectively. The first twenty thousand connection datas are used for validation at each training step. Each feedforward network was set to be trained for 10 epochs with the batch size set to 512. The optimizer of classifier was ?rmsprop? providing by the Keras library with the learning rate set to 0.0001.

V. Performance Evaluation

A. Metrics

This project aims to build an effective lightweight intrusion detection system, which means that the structure of the system should be as simple as possible while keeping a relatively high detection accuracy. However, how to determine a structure is simple or complex is not straightforward. Firstly, most related work on intrusion detection system using neural network does not provide fully description of their network structure. Furthermore, due to the difference in dataset, training and testing environment, implementation methods, hyper parameters selection, it is different to compare two classifier to determine which one is simpler or more complex. Therefore, we only focus on our system and selecting the simplest structure.

For lightweight consideration, we will select number of nodes, number of wights, training time, testing time as our metrics. Then, the overall accuracy of the multi-class classification training was defined as the rate that predictive label is the same as real label. Finally, for detection consideration of each attacks, we will consider false positive rate, false negative rate, true positive rate, true negative rate, and accuracy for each kind of attacks, as shown in Table II.

TABLE II
Evaluation Metrics For Each Attack

Real versus Predict	Positive	Negative
Positive	True Positive (TP)	False Negative (FN)
Negative	False Positive (FP)	True Negative (TN)

The accuracy of detecting each attack was defined as proportion of the sum of True Positive and True Negative in all detecting tasks, as shown in Equation 1:

$$Accuracy = \frac{TP + TN}{TP + TN + FN + FP} \tag{1}$$

B. Selecting Lightweight Structure

As described in Section IV, we set different dimensions of the hidden layer of both autoencoder and feedforward network to find out the simplest structure while keeping a relatively high detection accuracy. In this project, we set the autoencoder hidden layer dimension in the range from one to fifteen and feedforward network from one to twenty-five. Figure 4 shows how different hidden layer dimensions influence the overall accuracy of the intrusion detection system. Futuremore, we also make a table (see attached page 1) to demonstrate the detailed evaluation result for each dimensions combination, which contains the accuracy, autoencdoer training time, feedforward network training time and test time. By doing that, readers can make a compromise between accuracy and structure, which makes the detection system adaptive to different actual applications. For example, in the situation where 95% accuracy is good enough, a simpler structure will reduce the requirements for memory and computational power. Table III picks several typical results to the show the overall accuracy of our detection system with different dimension setting.

TABLE III
Evaluation Result

Au. Dim.	Fd. Dim.	Accu.	Au. Tr. T. (s)	Fd. Tr. T(s)
1	1	0.79212	28.26552	13.53285
1	10	0.97335	28.26552	13.5211
1	20	0.97389	28.26552	14.0647
3	1	0.793005	30.06977	15.9175
3	10	0.97924	30.06977	18.6969
3	20	0.97915	30.06977	17.6531
9	1	0.98172	37.54131	23.69682
9	10	0.996808	37.54131	23.93907
9	20	0.997719	37.54131	26.8627
14	1	0.984049	50.05794	30.79484
14	10	0.998036	50.05794	35.7716
14	20	0.998252	50.05794	32.5213

C. Evaluation

In order to evaluate the performance of the intrusion detection system to detecting each kind of attack, we select

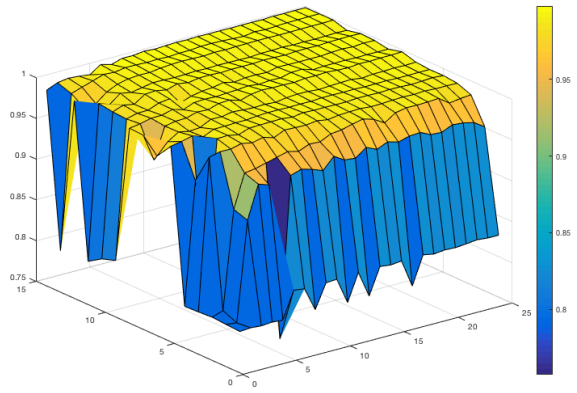

Fig. 4. Intrusion Detecion System Accuracy

one classifier model (Au. Dim. = 9, Fd. Dim = 9) from our previous evaluation process. The evaluation data is 30 percent KDDCup99 data that randomly selected from the origial data, a detailed description of the test data is shown in Table IV.

TABLE IV
EVALUATION DATASET (30% KDDCUP99)

Label	Number
Dos	117469
U2L	16
R2L	324
Probe	1219
Normal	29179
Total	148207

TABLE V
EVALUATION DATASET (30% KDDCUP99)

Real vs. Predict	Dos	U2R	R2L	Probe	Normal
Dos	**116750**	0	0	4	715
U2R	1	**0**	1	1	13
R2L	31	0	**147**	4	142
Probe	28	0	0	**1144**	47
Normal	92	0	8	36	**29043**

Table V lists the evaluation result of the classifier model (Au. Dim. = 9, Fd. Dim = 9). The overall accuracy of this model is 99.24227%. From Table V we can see that both all evaluation metrics of Dos Normal are good. However, for U2L attack, our IDS have very low TP rate. The reason for this phenomenon is that the number of U2R connections in the KDDCup99 dataset is very small. There is only 52 connections in the original dataset for training. In related research, how to increase the detection rate of U2L attack is still a problem. By comparing with related research, we claims that our detection system can provide competitive overall detection accuracy, as shown in Table VI.

TABLE VI
RELATED RESEARCH WORK COMPARISON

Method	Description	Dataset	Accuracy
Yin et al. [16]	RNN, hidden layer = 80	NSL-KDD	99.53%
Jihyun et al. [11]	LSTM, hidden layer = 80	KDD99	96.93%
Shrivas et al. [17]	ANN+Naive Bayes No structure given	KDD99	99.41%
Soni et al. [18]	ANN with feature selection No structure given	KDD99	99.24%
Our Method	Autoencoder+ANN	KDD99	

VI. FUTURE WORK

The intrusion detection system presented in this project aims to build an effective security solution for the Internet of Things applications. However, due to the lack of IoT application packet data, the IDS was trained and evaluated on KDDCUP99 data set. In the future, if related IoT dataset was found, the IDS will be adapted to real IoT data and IoT applications. Meanwhile, as the device finger printing technology was recently thought to be used to smart grid [15], the combination of IDS with device finger printing can also be an new direction.

VII. CONCLUSION

This project aims to design a lightweight intrusion detection system for the Internet of things applications. To achieve this, an autoencder was used to extract the potential features and reduce the dimensions of the input data. After the autoencdoer there is an shallow full connected neural networks act as the classifier. The evaluation on the benchmark KDDCUP99 dataset demonstrates that the IDS can have competitive performance compared with state-in-the-art methods.

REFERENCES

[1] Shah, Sajjad Hussain, and Ilyas Yaqoob. "A survey: Internet of Things (IOT) technologies, applications and challenges." Smart Energy Grid Engineering (SEGE), 2016 IEEE. IEEE, 2016.

[2] Tan, Jasper, and Simon GM Koo. "A survey of technologies in internet of things." Distributed Computing in Sensor Systems (DCOSS), 2014 IEEE International Conference on. IEEE, 2014.

[3] Butun, Ismail, Salvatore D. Morgera, and Ravi Sankar. "A survey of intrusion detection systems in wireless sensor networks." IEEE communications surveys & tutorials 16.1 (2014): 266282.

[4] Sobh, Tarek S. "Wired and wireless intrusion detection system: Classifications, good characteristics and stateoftheart." Computer Standards & Interfaces 28.6 (2006): 670694.

[5] Li, Wenchao, et al. "A new intrusion detection system based on KNN classification algorithm in wireless sensor network." Journal of Electrical and Computer Engineering 2014 (2014).

[6] Zhang, Jiong, Mohammad Zulkernine, and Anwar Haque. "Random-forests-based network intrusion detection systems." IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) 38.5 (2008): 649-659.

[7] Lin, Xiaodong, and Rongxing Lu. Vehicular ad hoc network security and privacy. John Wiley & Sons, 2015.

[8] Canbay, Yavuz, and Seref Sagiroglu. "A hybrid method for intrusion detection." Machine Learning and Applications (ICMLA), 2015 IEEE 14th International Conference on. IEEE, 2015.

[9] Senthilnayaki, B., K. Venkatalakshmi, and A. Kannan. "Intrusion detection using optimal genetic feature selection and SVM based classifier." Signal Processing, Communication and Networking (ICSCN), 2015 3rd International Conference on. IEEE, 2015.

[10] Almseidin, Mohammad, et al. "Evaluation of machine learning algorithms for intrusion detection system." Intelligent Systems and Informatics (SISY), 2017 IEEE 15th International Symposium on. IEEE, 2017.
[11] Kim, Jihyun, et al. "Long Short Term Memory Recurrent Neural Network Classifier for Intrusion Detection." Platform Technology and Service (PlatCon), 2016 International Conference on. IEEE, 2016.
[12] Roy, Sanjiban Sekhar, et al. "A Deep Learning Based Artificial Neural Network Approach for Intrusion Detection." International Conference on Mathematics and Computing. Springer, Singapore, 2017.
[13] Putchala, Manoj Kumar. Deep Learning Approach for Intrusion Detection System (IDS) in the Internet of Things (IoT) Network using Gated Recurrent Neural Networks (GRU). Diss. Wright State University, 2017.
[14] http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html
[15] Formby, David & Srinivasan, Preethi & Leonard, Andrew & Rogers, Jonathan & Beyah, Raheem. (2016). Who's in Control of Your Control System? Device Fingerprinting for Cyber-Physical Systems 10.14722/ndss.2016.23142.
[16] Yin, Chuanlong, et al. "A Deep Learning Approach for Intrusion Detection Using Recurrent Neural Networks." IEEE Access 5 (2017): 21954-21961.
[17] Shrivas, Akhilesh Kumar, and Amit Kumar Dewangan. "An ensemble model for classification of attacks with feature selection based on KDD99 and NSL-KDD data set." International Journal of Computer Applications 99.15 (2014).
[18] Shrivas, Akhilesh Kumar, and Amit Kumar Dewangan. "An ensemble model for classification of attacks with feature selection based on KDD99 and NSL-KDD data set." International Journal of Computer Applications 99.15 (2014).