Long Short Term Memory Recurrent Neural Network Classifier for Intrusion Detection

Conference Paper · February 2016

DOI: 10.1109/PlatCon.2016.7456805

Jihyun Kim
Pusan National University
14 PUBLICATIONS **78** CITATIONS
SEE PROFILE

Jaehyun Kim
Pusan National University
2 PUBLICATIONS **24** CITATIONS
SEE PROFILE

Thi-Thu-Huong Le
Pusan National University
10 PUBLICATIONS **61** CITATIONS
SEE PROFILE

Long Short Term Memory Recurrent Neural Network Classifier for Intrusion Detection

Jihyun Kim, Jaehyun Kim, Huong Le Thi Thu, and Howon Kim

Abstract—Due to the advance of information and communication techniques, sharing information through online has been increased. And this leads to creating the new added value. As a result, various online services were created. However, as increasing connection points to the internet, the threats of cyber security have also been increasing. Intrusion detection system(IDS) is one of the important security issues today. In this paper, we construct an IDS model with deep learning approach. We apply Long Short Term Memory(LSTM) architecture to a Recurrent Neural Network(RNN) and train the IDS model using KDD Cup 1999 dataset. Through the performance test, we confirm that the deep learning approach is effective for IDS.

Index Terms—Intrusion detection system, Long short term memory, Recurrent neural network

1 Introduction

With increasing the importance of cyber security, researches about Intrusion Detection System (IDS) have been actively studying. IDS protect a network system from malicious software attacks. There are two types IDS according to an object of observation [1]. The first one, Host-based IDS (HIDS), watches the host system operation or states. It detects system events such as unauthorized installation or access. Also, it checks the state of ram or file system whether there is an expected data or not. Because the detection of HIDS is based on the system event log, a false alarm ratio is low. But it cannot analyze behaviors related to the network. The second one, Network-based IDS (NIDS) is placed on DMZ or choke point of the network edge. It observes a real-time network traffic and analyzes it for detecting unauthorized intrusions or the malicious attacks. The detection techniques are two types [2]. The first technique is a behavior-based intrusion detection called anomaly detection. It catches attacks by comparing an abnormal behavior to a normal behavior. The second technique is a knowledge-based intrusion detection called misuse detection. This one detects the attacks based on the known knowledge.

Most researchers have studied about the detection techniques of IDS. In recent years, they try to apply machine learning to IDS. Yihua Liao et al. applied k-Nearest Neighbor (k-NN) to IDS [3]. They used k-NN classifier for the anomaly detection. Andrew H. Sung et al. took an IDS performance test with Support Vector Machine (SVM) and Artificial Neural Networks (ANN) [4]. They reduced features according to the influence toward classification of attacks, but the performance was similar to the result which used the original features. Maheshkumar Sabhnani et al. used several machine learning algorithms such as multi-layer perceptron (MLP), k-Means, decision tree [5]. Also they designed a multi-classifier model. They confirmed that the multi-classifier model had a better performance than comparison models.

We extend this research flow to deep learning approach. Deep learning achieves high level abstractions in data by using a complex architecture or composition of non-linear transformations. Therefore, we can acquire a high detection rate. In this paper, we apply Long Short Term Memory (LSTM) to Recurrent Neural Network (RNN) and use it for an IDS model [6] [7]. We train the model by using KDD Cup 1999 dataset and measure the performance. Through the experiments, we find an optimal hyper-parameter for LSTM-RNN and confirm the detection rate and false alarm rate.

The rest of the paper is organized as follows: we introduce the related works in section 2 and give a brief description of LSTM in section 3. In section 4, we explain about a dataset and take the experiments in section 5. We make a conclusion in the last section.

2 Related Works

In the real world, the Intrusion Detection System (IDS) has a vital role in detecting the intrusion. There are two kinds of detection, including anomaly- based and misuse-based [8] [9]. To detect activities that differ from established patterns for users, we use to anomaly-based. On the other hand, misuse-based compares users? activities with the known behaviors of attackers.

In recent years, almost commercial Intrusion Detection System use to attack existing pattern to recognize. Soft computing is one of techniques, which help to reduce cost of detection. There are some soft computing techniques in IDS for example Artificial Neural Network, genetic algorithm, decision tree, fuzzy logic etc. They are applied to create utilities in the intrusion detection field due to learning and flexibility capability. Among the soft computing techniques, neural network approach is widely used in recent

- *J. Kim is PhD student at Department of Electrical and Computer Engineering, Pusan National University, Busan, Korea.*
E-mail: jihyunkim@pusan.ac.kr
- *Jaehyun Kim and T.T.H Le are master and PhD student at Department of Electrical and Computer Engineering, Pusan National University, Busan, Korea.*
- *H. Kim is Professor at Department of Electrical and Computer Engineering, Pusan National University, Busan, Korea.*

Manuscript received December 06, 2015; revised Deccember 31, 2015.

researches. By using SOM, Heywood et al. proposed a hierarchical neural network for intrusion detection [10]. Feedforward neural network is applied to create an IDS using Back Propagation algorithm to train, is proposed by J.Shum et al. [11]. Mukkamala et al. published other approach by combination between neural network and SVM [12]. Other approach which modified Jordan recurrent neural network is published by Xue et al. [13]. Besides Skaruz et al. also applied successfully Jordan recurrent neural network to detect SQL-based attacks [14]. Even though neural network is quite good for applying to this field, deep learning is another approach can obtain accuracy of detection better than previous approaches. In 2015, our research applied Recurrent Neural Network with Hessian-Free Optimization to train DARPA data set [15]. We obtained the detection rate 95.37%. We continue to use the other method in deep learning to detect modern attacks also malwares. By this work, we apply Long Short Term Memory Recurrent Neural Network in IDS.

3 Long Short Term Memory

Recurrent Neural Network is the most famous model to training the sequence data. The conventional RNN has trouble when it is used to train with a long step size. In this section, we briefly discuss the formal of RNN and the vanishing problem. Then, we describe Long Short Term Memory to address this problem.

3.1 Recurrent Neural Network

Recurrent Neural Network (RNN) is extension of a convention feed-forward neural network. Unlike feedforward neural networks, RNN have cyclic connections making them powerful for modeling sequences. We assume that an input sequence, the hidden vector sequence, and output vector sequence denoted by X, H and Y respectively. Input sequence is given by $X = (x_1, x_2, ..., x_T)$. A traditional RNN calculates the hidden vector sequence($H = (h_1, h_2, ..., h_T)$) and output vector sequence($Y = (y_1, y_2, ..., y_T)$) with $t = 1$ to T as follows:

$$h_t = \sigma(W_{xh}x_t + W_{hh}h_{t-1} + b_h) \tag{1}$$

$$y_t = W_{hy}h_t + b_y \tag{2}$$

where function σ is a nonlinearity function, W is a weight matrix, b is a bias term.

The convention RNN used to Back Propagation Training Time (BPTT) to handle a variable-length sequence input [16]. In BPTT, the model is first trained with the training data. Then the output error gradient is saved for each time step. The RNN is hard to train, however, it makes the gradient is exploding or vanishing while training with BPTT algorithm. Bengio et al. mentioned and addressed this problem in [17].

3.2 Long Short Term Memory

Long Short-Term Memory (LSTM) is an architecture which is proposed by Hochreiter and Schmidhuber [6]. Figure 1 shows a single LSTM cell. And we describe the equations to compute the values of three gates and cell state.

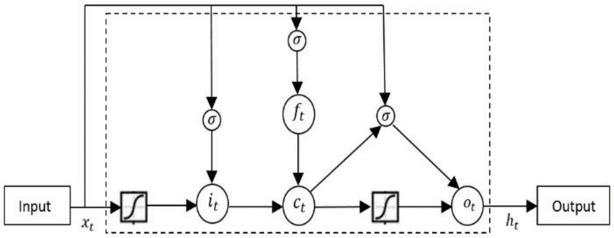


Fig. 1. Long Short Term Memory Cell

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i) \tag{3}$$

$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f) \tag{4}$$

$$c_t = f_tc_{t-1} + i_ctanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \tag{5}$$

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o) \tag{6}$$

$$h_t = o_ttanh(c_t) \tag{7}$$

σ is the logistic sigmoid function, and i, f, o and c are respectively the input gate, forget gate, output gate and cell state. W_{ci}, W_{cf} and W_{co} are denoted weight matrices for peephole connections. In LSTM, three gates(i, f, o) control the information flow. The input gate decides the ratio of input. When calculating the cell state, this ratio has effect on the equation (5). The forget gate passes the previous memory h_{t-1} or not. The ratio of the previous memory is calculated in the equation (4) and used for the equation (5). The output gate determines whether passing the output of memory cell or not. The equation (7) shows this process. By using LSTM, we can solve the vanishing and exploding gradient problems due to the three gates. In LSTM-RNN architecture, the recurrent hidden layer is replaced by LSTM cell.

4 Dataset

KDD Cup 1999 dataset has been used to measure a performance of IDS in many researches. Although the dataset is old, it is good to compare the IDS models. Because there are lots of performance measurement results with the same dataset. That is the main reason why we chooesed KDD Cup 1999 dataset.

There are 4,898,431 network traffics in the dataset and each traffic has 41 features. And 22 attacks are categorized according to their characteristic. Table 1 shows the category of the attacks. DoS attack depletes resources of the target servers and makes them precluding any services. R2L attack enables an unauthorized remote access . U2R attack tries to acquire the superuser authority. Probe attack uses for finding vulnerabilities of the target server.

Because there are too many data records in the original dataset, we use KDD Cup 1999 10 percent data for training and testing. However, the data ratio of the attacks is weighted towards DoS attack. And the others ratio is only 1 percent. It is shown in Figure 2. So the IDS model will be trained unfairly. As a result, DoS attack and normal traffic can easily be detected but the other attacks cannot be caught.

TABLE 1
Category of the attacks

Category	Attacks
DoS	back, land, neptune, pod, smurf, teardrop
R2L	ftp-write, guess-passwd, imap, multihop, phf, spy, warezclient, warezmaster
U2R	buffer-overflow, loadmodule, perl, rootkit
Probe	ipsweep, nmap, portsweep, satan

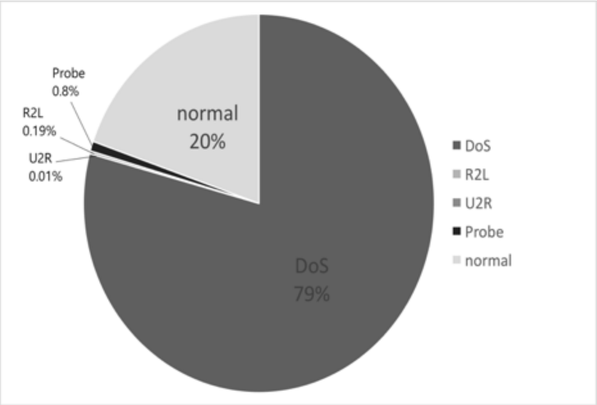


Fig. 2. Data ratio of the attacks

In order to solve this problem, we generate a new training dataset evenly. We extract 300 instances from each attack category except U2R attack. Due to the lack of instances, we only extract 30 data from U2R category. Also, we use 1,000 normal instances.

5 Experiment

In this section, we take two experiments. The first experiment is about finding hyper-parameter values to get the best performance of IDS model. The second experiment is about measuring the performance with the hyper-parameter values acquired from the first experiment. Our experiment environment is as follow:

- CPU : Intel(R) Core(TM) i7-4790 3.60GHz
- GPU : GTX Titan X
- RAM : 8GB
- OS : Ubuntu 14.04

5.1 Evaluation Metric

Generally, Detection Rate (DR) and False Alarm Rate (FAR) are used as the metrics of IDS evaluation. The DR signifies a ratio of intrusion instances detected by IDS model. And the FAR is a ratio of misclassified normal instances. Based on a confusion matrix, equations of the metrics are as follow (TP: true positive, TN: true negative, FP: false positive, FN: false negative):

$$DR = TP/(TP + FN) \tag{8}$$

$$FAR = FP/(TN + FP) \tag{9}$$

As the DR increases and FAR decreases, the performance grows better. Therefore, we use one more metric, efficiency. By using this metric, we can evaluate the IDS model easily.

$$Efficiency = DRFAR \tag{10}$$

5.2 The IDS model set up

Before using the training dataset, we normalized all instances from 0 to 1. The input vector is 41 features and the output vector is composed of 4 attacks and 1 non-attack. Therefore, the input dimension is 41 and the output dimension is 5. And we apply LSTM architecture to the hidden layer. The time step size, batch size and epoch are 100, 50, 500 respectively. We use softmax for the output layer and stochastic gradient decent (SGD) for an optimizer. And the loss function is mean squared error (MSE). The learning rate and hidden layer size will be decided in the next two experiment.

5.3 Experiment 1: Finding hyper-parameter value

Hyper-parameters are parameters for model initiation. Depending on the value of hyper-parameter, the performance is changed. Kalus Greff et al. analyzed the impact of hyper-parameters [18]. Among them, they found that the learning rate and hidden layer size have great effect on the performance. Also, they discovered that the hyper-parameters are independent. Therefore, we take the experiments with changing the values of learning rate and hidden layer size. First, we change the learning rate from 0.0001 to 0.1. In figure 3, we can notice that the DR and the FAR are a growing trend as the learning rate is increased.

For more evaluating precisely, we calculate the efficiency. Although the DR is the lowest value, we get the best efficiency when we set the learning rate 0.01 (Table 2). If the learning rate is too small, the IDS model would be trained too accurately. So the model can easily catch the intrusion instances but it also considered the normal instances as the intrusion ones.

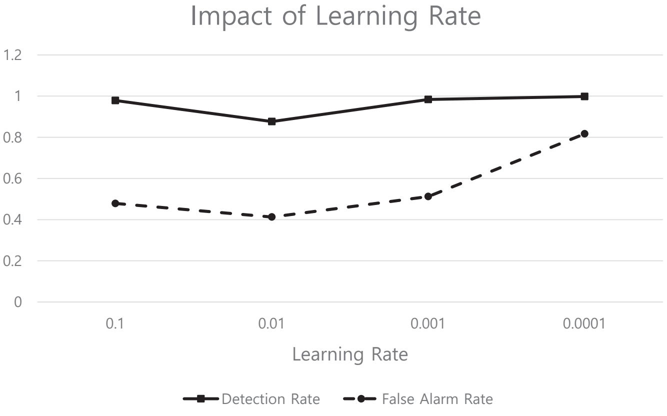


Fig. 3. Performance test with increasing the learning rate

Now, we fix the learning rate(0.01) and take the other experiment with changing the hidden layer size from 10 to 90. Figure 4. shows an impact of the hidden layer size. With

TABLE 2
Efficiency by the learning rate

	0.0001	**0.001**	**0.01**	**0.1**
DR	0.998	0.984	0.877	0.979
FAR	0.817	0.512	0.413	0.479
Efficiency	**1.222**	**1.921**	**2.124**	**2.045**

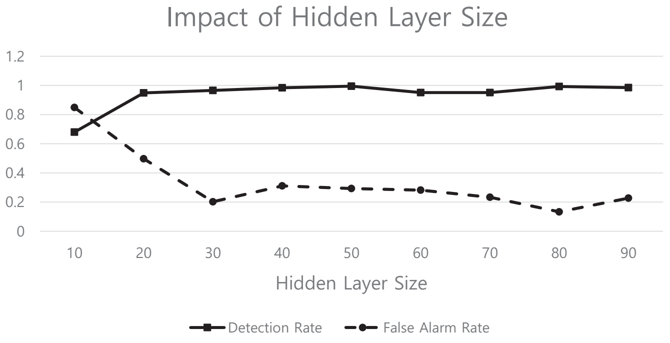


Fig. 4. Performance test with increasing the hidden layer size

growing the size, the DR is an increasing trend and the FAR is a decreasing trend.

In common with the previous test, we calculate the efficiency of the IDS model(Table 3). When we set 80 for the hidden layer size, we could get the highest efficiency. As a result, the best values for the learning rate and hidden layer size are 0.01 and 80 respectively.

TABLE 3
Efficiency by the hidden layer size

Size	DR	FAR	Efficiency
10	0.68	0.849	**0.800942285**
20	0.949	0.497	**1.90945674**
30	0.966	0.202	**4.782178218**
40	0.984	0.311	**3.163987138**
50	0.995	0.293	**3.395904437**
60	0.951	0.282	**3.372340426**
70	0.951	0.233	**4.081545064**
80	0.993	0.133	**7.466165414**
90	0.985	0.227	**4.339207048**

5.4 Experiment 2: Measuring the performance

According to the result of section 5.3, we set the hyper-parameters for training the IDS model. For testing, we generated 10 test datasets which are selected from kddup.data.txt. In each dataset, there are 5,000 selected instances randomly. Figure 5. shows the DR and FAR when we take a test with each test dataset. We summaries the result in Table 4. The average DR is 0.9879003. It means the attack detection percent is 98.8% among the total attack instances. The average FAR is 0.1003805. About 10% normal instances are misclassified.

The average percentages of each attack detection are represented in Figure 6. DoS and Normal instances are well detected but U2R instances are never detected. Because only 30 U2R instances are used for training the model.

TABLE 4
Result summary

	DR	FAR	Efficiency
Best	0.989583	0.07781	**12.692786**
Worst	0.984807	0.129093	**7.663612**
Average	0.9879003	0.1003805	**10.005282**

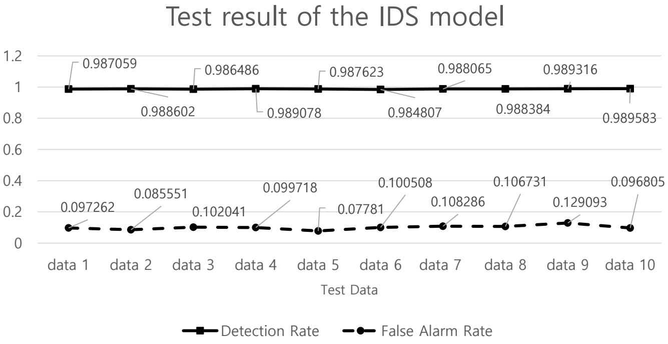


Fig. 5. Performance test result

In order to an objective evaluation, we compare our result to other classifier algorithms. Even though the FAR is a little bit higher than other algorithms, percentages of the DR and Accuracy are the best.

TABLE 5
Comparison with other algorithms

	DR(%)	FAR(%)	Accuracy(%)
GRNN	59.12	12.46	87.54
PNN	96.33	3.34	96.66
RBNN	69.83	6.95	93.05
KNN	45.74	46.49	90.74
SVM	87.65	6.12	90.4
Bayesian	77.6	17.57	88.46
LSTM-RNN	**98.88**	**10.04**	**96.93**

6 Conclusion

In this paper, we implemented the IDS classifier based on LSTM-RNN and evaluated the IDS model. For training phase, we generated a dataset by extracting instances from KDD Cup 1999 dataset. In order to find the proper learning rate and hidden layer size, we took an experiment with changing the values. For testing phase, we made 10 test dataset and measured the performance. By comparing it to other IDS classifier, we found that the attacks are well detected by LSTM-RNN classifier. Because we have the highest DR and Accuracy even though the FAR is slightly above the other ones.

To improve the FAR, we will analyze the relation between the dataset and initialized weight values. The reason is that we found the performance was changed even though all hyper-parameters are the same. The one variable is the initialized weight values. This will be a good subject for our further research.

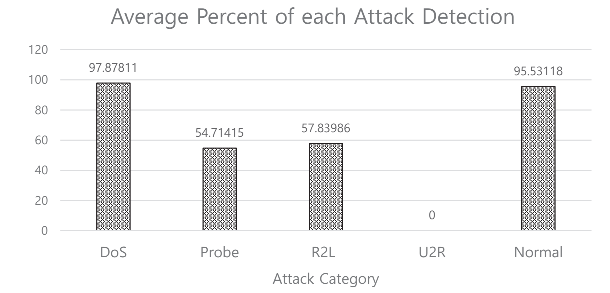


Fig. 6. Average percentage of each attack detection

Acknowledgments

This work was supported by Institute for Information & communications Technology Promotion(IITP) grant funded by the Korea government(MSIP) (No.10043907, Development of high performance IoT device and Open Platform with Intelligent Software)

References

[1] Bai, Yuebin, and Hidetsune Kobayashi, *Intrusion detection systems: technology and development*, AINA 2003. 17th International Conference on. IEEE, 2003

[2] Depren, Ozgur, et al., *An intelligent intrusion detection system (IDS) for anomaly and misuse detection in computer networks*, Expert systems with Applications 29.4, pp.713-722, 2005

[3] Liao, Yihua, and V. Rao Vemuri, *Use of k-nearest neighbor classifier for intrusion detection*, Computers & Security 21.5, pp.439-448, 2002

[4] Sung, Andrew H., and Srinivas Mukkamala, *Identifying important features for intrusion detection using support vector machines and neural networks*, Applications and the Internet, 2003

[5] Sabhnani, Maheshkumar, and Grsel Serpen. , *Application of Machine Learning Algorithms to KDD Intrusion Detection Dataset within Misuse Detection Context*, MLMTA, 2003

[6] Hochreiter, Sepp, and Jrgen Schmidhuber, *Long short-term memory*, Neural computation 9.8, pp.1735-1780, 1997

[7] LukoEvilus, Mantas, and Herbert Jaeger, *Reservoir computing approaches to recurrent neural network training*, Computer Science Review 3.3 pp.127-149, 2009

[8] S. Taeshik, M. Jongsub, *A hybrid machine learning approach to network anomaly detection*, ScienceDirect, Information Sciences 177, pp.37993821, 2007

[9] C. Yehui, A. Ajith, Y. Bo, *Hybrid flexible neural-tree-based intrusion detection systems*, International Journal of Intelligent Systems, Volume 22 Issues 4, pp.337-352, April 2007

[10] H. Kayacik, A. Zincir-Heywood, and M. Heywood, *A hierarchical SOM-based intrusion detection system*, In Proc. Elsevier Engineering Application of Artificial Intelligence, pp.439-451, 2007

[11] J. Shum and H.A. Malki, *Network intrusion detection system using neural network*, In Proc. IEEE Fourth Int. Conference on Natural Computation, pp.242-246, 2008

[12] Mukkamala, Srinivas, Andrew H. Sung, and Ajith Abraham, *Intrusion detection using ensemble of soft computing paradigms*, Intelligent Systems Design and Applications. Springer Berlin Heidelberg, pp.239-248, 2003

[13] J.-S. Xue, J.-Z. Sun, and X. Zhang, *Recurrent network in network intrusion detection system*, In Machine Learning and Cybernetics, Proceedings of 2004 International Conference on, vol. 5, pp.26762679, Aug 2004

[14] J. Skaruz and F. Seredynski, *Recurrent neural networks towards detection of sql attacks*, In Parallel and Distributed Processing Symposium, 2007, pp.18, March 2007

[15] K. Jihyun, K. Howon, *Applying Recurrent Neural Network to Intrusion Detection with Hessian Free Optimization*, In Proc. WISA, 2015

[16] Werbos, Paul J., *Backpropagation through time: what it does and how to do it*, Proceedings of the IEEE 78.10, pp.1550-1560, 1990

[17] Bengio, Yoshua, S. Patrice, and F.Paolo, *Learning long-term dependencies with gradient descent is difficult*, Neural Networks, IEEE Transactions on 5.2, pp.157-166, 1994

[18] Greff, Klaus, et al, *LSTM: A Search Space Odyssey*, arXiv preprint arXiv:1503.04069, 2015

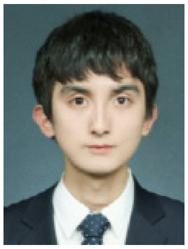
[19] Devaraju, S., and S. Ramakrishnan, *Performance comparison for intrusion detection system using neural network with KDD dataset*, ICTACT Journal on Soft Computing. l 4.3, 2014

[20] Deepika, D., and V. Richhariya, *Intrusion detection with KNN classification and DS-theory*, International Journal of Computer Science and Information Technology and Security 2.2, pp.274-281, 2012

[21] Vaishali Kosamkar, and Sangita S Chaudhari, *Improved Intrusion Detection System using C4.5 Decision Tree and Support Vector Machine*, International Journal of Computer Science and Information Technologies Vol.5(2), pp.1463-1467, 2014

[22] Altwaijry, Hesham, *Bayesian based intrusion detection system*, IAENG Transactions on Engineering Technologies, Springer Netherlands, pp.29-44, 2013

Jihyun Kim was born in XX, on , 1983. He graduated from Pusan National University (PNU) for bachelor in 2010. Then, he continued studying at PNU for master course. His special fields of interest included Machine Learning, Deep Learning, IDS. He received master degree at PNU in 2012. Currently, he is PhD student at Electric Electronic Computer Science Engineering, PNU.



Jaehyun Kim was born in Pusan, on December 1990. He graduated from Pusan National University (PNU) for bachelor. His special fields of interest included Machine Learning, Deep Learning, IDS, face recognition. From 2015 to now, he is master student at Electric Electronic Computer Science Engineering, PNU.

Huong Le Thi Thu was born in Vietnam, on August 1985. She graduated from Hung Yen University of Technical and Education (HYUTE) in Vietnam for bachelor in 2007. In 2008, she worked at HYUTE as lecture. In 2013, she received master degree at Hanoi University of Science and Technology, Vietnam. Her interested area included Machine Learning, Deep Learning, IDS, NILM. Currently, she is PhD student at Electric Electronic Computer Science Engineering, PNU

Howon Kim was born in Busan. He received bachelor degree at Kyungpook National University, Korea in 1993. In 1995, he graduate for master course at Department Electron Electric Engineering, POSTECT. Then, he got doctor degree at POSTECH in 1999. His employment experiment included researcher/team leader at ETRI from 1998 to 2008. From 2008 to now, he is Associate Professor at Pusan National University Computer Science and Engineering. His interested fields are IoTs, Smart Grid Security, RFID/USN Security, PKC Cryptography, VLSI, Embedded System Security.