An Empirical Evaluation of Deep Learning for Network Anomaly Detection

Conference Paper · March 2018

DOI: 10.1109/ICCNC.2018.8390278

Ritesh Malaiya
University of Texas at Dallas
2 PUBLICATIONS **2** CITATIONS

SEE PROFILE

Donghwoon Kwon
Texas A&M University-Commerce
15 PUBLICATIONS **62** CITATIONS

SEE PROFILE

Jinoh Kim
Texas A&M University-Commerce
50 PUBLICATIONS **268** CITATIONS

SEE PROFILE

Ikkyun Kim
University of Science and Technology, Korea
49 PUBLICATIONS **139** CITATIONS

SEE PROFILE

An Empirical Evaluation of Deep Learning for Network Anomaly Detection

Ritesh K. Malaiya* Donghwoon Kwon* Jinoh Kim*, Sang C. Suh*, Hyunjoo Kim†, Ikkyun Kim†
Texas A&M University, Commerce, TX 75428, USA *
ETRI, Daejeon, 34129, Korea †
Email: rmalaiya@leomail.tamuc.edu, {donghwoon.kwon,jinoh.kim,sang.suh}@tamuc.edu, {hjookim,ikkim21}@etri.re.kr

Abstract—**Deep learning has been given a great deal of attention with its success story in many areas such as image analysis and speech recognition. In particular, deep learning is good at dealing with high-dimensional data exhibiting non-linearity. Our preliminary study reveals a very high degree of non-linearity from network traffic data, which explains why it is hard to improve the detection accuracy by using conventional machine learning techniques (e.g., SVM, Random Forest, Adaboosting). In this study, we empirically evaluate deep learning to see its feasibility for network anomaly detection. We examine a set of deep learning models constructed based on the Fully Connected Network (FCN), Variational AutoEncoder (VAE), and Long Short-Term Memory with Sequence to Sequence (LSTM Seq2Seq) structures, with two public traffic data sets that have distinctive properties with respect to the distribution of normal and attack populations. Our experimental results confirm the potential of deep learning models for network anomaly detection, and the model based on the LSTM Seq2Seq structure shows a highly promising performance, yielding 99% of binary classification accuracy on the public data sets.**

1. Introduction

While the importance of network anomaly detection has been emphasized over decades, it is still highly challenging to identify malicious events from the network due to the evolution of cyber-attacks. For instance, a ransomware attack (known as “WannaCry”) hit the global world on May 12th, 2017, affecting over 10,000 organizations in over 150 countries [1]. The damage of WannaCry is severe locking of user files that compromises the primary security goals of availability and integrity. Prior to WannaCry, serious distributed denial of service (DDoS) attacks overwhelmed several datacenters on October 21st, 2016, which caused Twitter, Spotify, and other major sites to closed down [2]. Emerging DoS attacks can utilize a botnet comprising hundreds of thousands of IoT devices [3], and the impact of such attacks will be critical with the increasing use of mobile and IoT devices.

Machine learning has been extensively studied for network anomaly detection [4], [5], [6], [7]. However, using conventional machine learning techniques has been largely limited without significant improvement of detection accuracy. For example, we observed less than 83% of detection accuracy with Support Vector Machine (SVM), Random Forest and Adaboosting, from the evaluation with the NSL-KDD data set [8]. Given that emerging attacks will be much more sophisticated than ever, relying only on such traditional techniques may lead to the failure of detection with an unacceptable performance.

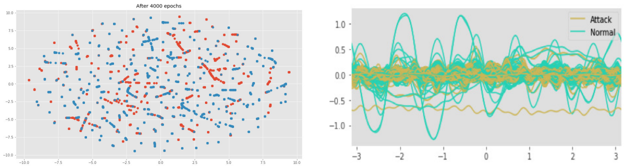
In our preliminary study, we observed a very high degree of non-linearity from the public network traffic data sets including NSL-KDD [8] and Kyoto University Honeypot data (“Kyoto-Honeypot” in short) [9]. We assume that the non-linear property is the root reason why the traditional machine learning techniques do not work well for network anomaly detection. This motivates us to explore the potential of deep learning, which is good for representational learning and powerful to deal with the high-dimensional data exhibiting non-linearity [10], [11]. In fact, deep learning has widely been considered for a diverse range of applications including image processing, natural language processing, computer vision, and so forth [12], [13]. Comparatively, it has not been thoroughly investigated for network anomaly detection, and only a few studies employed deep learning techniques without great details [14], [15], [16], [17], [18]. In this study, we conduct an empirical evaluation of deep learning in depth to validate its feasibility for network anomaly detection.

The key contribution of this paper is that we thoroughly examine a set of deep learning models established based on Fully Connected Network (FCN) [19], Variational AutoEncoder (VAE) [20], and Long Short-Term Memory (LSTM) [21] structures. The traffic data sets employed for our evaluation have the distinctive characteristics: the population of Kyoto-Honeypot is highly skewed and the vast majority of the data is for attack records, whereas NSL-KDD contains a relatively balanced number of normal and attack records without significant biases, and thus, it would be helpful for evaluating the constructed models in a thorough way. Our experimental results confirm the potential of the evaluated deep learning models for network anomaly detection, with significantly improved performance compared to conventional machine learning techniques. In particular, the model based on the LSTM Seq2Seq structure shows the highly promising performance, yielding 99% of binary classification accuracy for both traffic data sets.

This paper is organized as follows. In the next section, we discuss the motivation of this work with a brief description of the data sets, and present a summary of the closely

TABLE 1. ACCURACY OF CONVENTIONAL ML TECHNIQUES

Training	Testing	Adaboosting	SVM	Random Forest
Train-	Test+	82.5%	79.6%	78.3%
Train-	Test-	65.5%	56.5%	53.4%
Train+	Test+	80.5%	79.1%	76.1%
Train+	Test-	58.7%	56.4%	50.3%



(a) t-SNE plot

(b) Andrews curve

Figure 1. Distribution of NSL-KDD data points using (a) t-SNE plot and (b) Andrews curve. The results show a high degree of non-linearity.

related studies. In Section 3, we illustrate the deep learning models for network anomaly detection, based on FCN, VAE, and LSTM Seq2Seq structures. We evaluate the presented deep learning models in Section 4 with the two traffic data sets. Finally, we conclude our presentation in Section 5.

2. Background

2.1. Why Deep Learning for Network Anomaly Detection?

In our preliminary study, we examined a set of traditional machine learning techniques with the NSL-KDD data set. NSL-KDD [8] is a modified version of the KDDCup 1999 connection data set [22] that substantially reduced redundant records to minimize the classification bias. The data set contains two files for training ("Train+" and "Train-") and two other files for testing ("Test+" and "Test-"). Note that Train- and Test- are subsets of Train+ and Test+, respectively, to give different degrees of difficulty in classification. Each record in the data set consists of 41 features with the associated label indicating whether the data record is normal or anomalous. The distributions between normal and anomalous connections are fairly well "balanced" between 46% – 82% in the data set. Table 1 summarizes the classification accuracy using the machine learning techniques. As seen from the table, the resulted accuracy is limited to less than 83%.

To see why, we examined the distribution of the data points in NSL-KDD, using t-Distributed Stochastic Neighbor Embedding (t-SNE) and Andrews Plot, which are embedding tools to visualize high-dimensional data. Figure 1 illustrates the result conducted with 10% of the data points randomly chosen from Train+. The t-SNE plot in the figure shows that normal and attack data points share the same feature space, which makes it very hard to classify them into either normal or attack. The Andrews curve also shows that normal and attack points are severely overlapped. This indicates that discriminating attack from normal data points would be quite challenging.

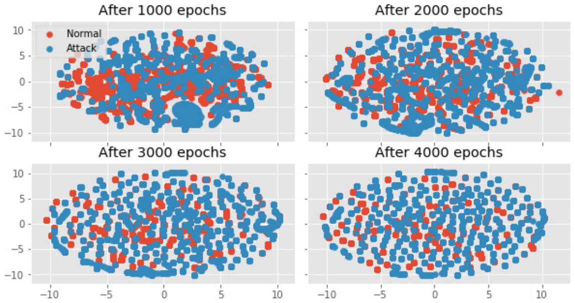


Figure 2. t-SNE plot of Kyoto data points based on each 1,000 epoch. The results show high non-linearity like the NSL-KDD distribution.

The Kyoto-Honeypot data set contains a large set of data points captured and analyzed through honeypots since 2009. The data is highly "unbalanced" with respect to the fractions of the normal and attack data points, and 97% of them are for attacks. This is because the data is collected from the honeypot system. The number of features in this data set is 24 in total (14 basic and 10 extended features), and we excluded six minor features related to the host and port information in our experiments. We conducted t-SNE against a daily data set with different epoch values (from 1,000 to 4,000 epochs). Figure 2 demonstrates the results of t-SNE based on each 1,000 epoch. The resulted plots also suggest a high degree of non-linearity for the Kyoto-Honeypot data.

It is known that deep learning is good at dealing with high dimensional data with non-linearity [11], which motivates us to examine deep learning methods for network anomaly detection. We next briefly discuss the past studies that employed deep learning techniques for network anomaly detection, and introduce a set of deep learning models we designed for evaluating in the following section.

2.2. Related Work

This section summarizes three closely related studies to our work. The work in [14] utilized Deep Neural Network (DNN) for anomaly detection in a software-defined networking environment. In the proposed model, the network statistics are transferred to the DNN model to detect intrusions. A set of different learning rates (0.1, 0.01, 0.001, and 0.0001) were used to measure the performance. The authors reported 75% of F-measure when the learning rate is 0.001 as the best performance against the NSL-KDD data.

In [15], an energy-based model in conjunction with three different types of deep learning models (fully connected, recurrent, and convolutional) was proposed for anomaly detection. In the proposed model, a score matching algorithm is used for training, and two decision criteria were suggested based on the energy score and reconstruction error to make a proper decision of model usage. The reported F-measures for the two criteria are 73.9% and 73.2%, which implies further optimization would be necessary.

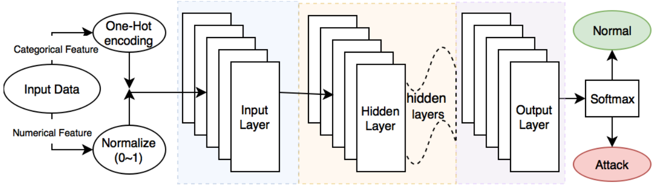


Figure 3. Overview of the FCN model. The number of hidden units and layers are configurable, and the loss function used in this model is cross entropy.

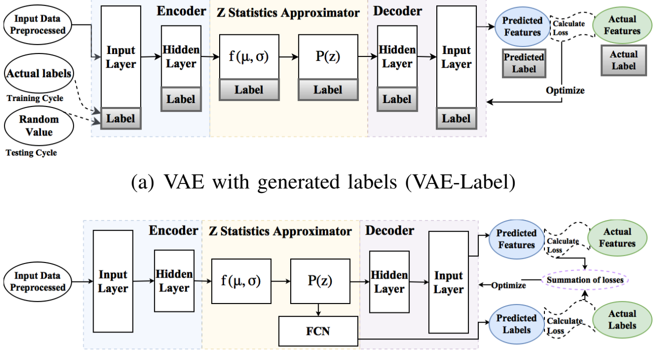
The authors in [16] proposed an anomaly detection model based on Deep Neural Network (DNN). This DNN model is composed of one input layer, three hidden layers, and one output layer. In the first fine tuning step, the first two hidden layers train the model in an unsupervised manner with AutoEncoder, and the classification takes place in the last hidden layer using Softmax. The second fine tuning step performs back-propagation on all hidden layers. The authors reported 97.5% accuracy for binary classification (normal and attack) with NSL-KDD. As will be presented shortly, our designed model based on the LSTM Seq2Seq structure yields 99% of classification accuracy in various settings.

3. Deep Learning Models

In this work, we design a set of deep learning models from the FCN, VAE, and LSTM-Seq2Seq structures for the purpose of network anomaly detection. FCN is the first generation deep learning model, in which every neuron in fully connected layers is connected to each other [19]. While simple, the performance could be degraded due to the problem of vanishing gradient during back-propagation. AutoEncoder is widely considered for high-dimensional data, and VAE is a variant of AutoEncoder. Since VAE is based on a probabilistic model, VAE works well if we can expect that the final distribution is close to the original one by maximizing the log likelihood. The vanishing gradient problem is no more a critical concern in VAE without the use of back-propagation. The Sequence to Sequence (Seq2Seq) structure is based on Recurrent Neural Network (RNN), and the LSTM cell is a type of RNN cells. Thus, LSTM-Seq2Seq has been considered for time-series data, but we examine this model for network anomaly detection without a time-series concept. We next present our network anomaly detection models constructed based on these deep learning methods.

3.1. FCN model

The first deep learning model we design is based on the simple FCN structure. Figure 3 shows the overview of the FCN model we designed. The first step in this model is the data preprocessing for normalization and transformation. Any numerical feature is normalized using the z-score, while a categorical feature is encoded as a set of dummy numerical values using one-hot encoding. The pre-processed input data is then passed to the fully connected network for training. The training is performed by tuning the parameter configuration:



(a) VAE with generated labels (VAE-Label)

(b) VAE with Softmax (VAE-Softmax)

Figure 4. Overview of two VAE models. VAE-Label is a model based on label inclusion and the labels are treated as an independent feature. VAE-Softmax does not utilize the label information as a feature. The loss is calculated by combining Binary Cross Entropy (BCE) and KL divergence (KLD) in this model.

- Number of hidden units = {1 unit, 10% of the entire features, 20% of the entire features, 40% of the entire features, and 100% entire features}
- Number of hidden layers = {1, 3}
- Epochs = {40}
- Learning rate = {1e-4, 1e-5}

To overcome the vanishing gradient problem which can be caused by Sigmoid, we considered Rectified Linear Unit (ReLU) as the activation function. The Softmax layer with a cross entropy cost function at the end produces the final output, i.e., either normal or attack, as shown in Figure 3.

3.2. VAE models (with/without label inclusion)

VAE is stochastic-based and composed of two networks: encoder and decoder [20]. When the large scale input data x is given to this model, the encoder compresses x into a small scale latent variable z (also known as a hidden representation), which is denoted as $q_\Theta(z|x)$. An essential assumption with respect to the latent variable z is that the underlying distribution is a standard normal distribution ($\mu = 0$ and σ=1). The latent variable is then passed to the decoder to reconstruct the data, denoted as $p_\Phi(x|z)$. Note that Θ and Φ are tunable parameters. Thus, it is important how well the decoder reconstructs input data x from the latent variable z in the VAE method.

We establish two VAE models: VAE-Label and VAE-Softmax as shown in Figure 4. For VAE-Label in Figure 4(a), the labels are treated as an independent feature. The actual labels are included as a feature in the training phase, while the value of the label is randomly chosen from the standard normal distribution in the testing phase. This VAE model performs learning to regenerate the labels out of the provided random values. For the output, labels are

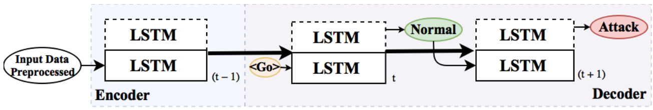


Figure 5. Overview of the LSTM-Seq2Seq model composed of two RNNs for encoding and decoding. We consider two different types of LSTM-Seq2Seq: *shallow* with one LSTM layer and *deep* with three LSTM layers (the figure shows two LSTM layers as an example). The loss function used in this model is Mean Squared Error.

provided in a one-hot encoded format and the regenerated label values are treated as a probability of the given label. We employ the same parameter configuration set as the one used for the FCN model.

VAE-Softmax in Figure 4(b) is almost identical to VAE-Label, but it does not consider the label as a feature. In addition, a separate FCN with one hidden layer (ReLU activation) and one Softmax layer is attached to this VAE model. The FCN in this model receives the input from the probability distribution of z ($P(z)$) and produces the label probabilities as output. This model performs learning by adding two losses: Binary Cross Entropy (BCE) and KL divergence (KLD) loss as in [23].

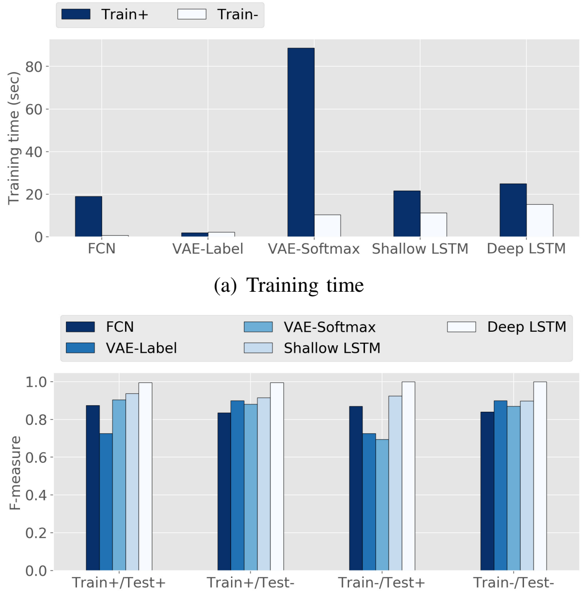
3.3. LSTM-Seq2Seq model

Another model we establish is based on the Sequence to Sequence (Seq2Seq) structure, which is based on Recurrent Neural Network (RNN) [21]. The goal of Seq2Seq is to yield a target sequence and conditional probability $P(y|x)$ through an encoder and decoder with two non-linear activation functions by updating the hidden state (within the encoder and decoder), where the given conditions are:

- x is an input sequence, $x = \{x_1, x_2, ...x_T\}$ in each time step (t)
- y is a representation of a fixed-size vector, $y = \{y_1, y_2, ...y_{T'}\}$, also known as a target sequence ([24], [25]).

There are several important concepts to note here: 1) only the fixed-length context vector generated by the encoder is passed to the decoder; 2) the non-linear activation functions to update the hidden state in the encoder and decoder could be simple (e.g., Softmax) or complex (e.g., LSTM); and 3) the length of x and y can be different. We design our LSTM-Seq2Seq model shown in Figure 5 based on these concepts, and LSTM is chosen for the activation function. Depending on the number of LSTM layers, we can build different types of LSTM-Seq2Seq, and we consider two LSTM-Seq2Seq models: *shallow* with one LSTM layer in the encoder and decoder, and *deep* with three LSTM layers.

In Figure 5, the <Go> signal is a tensor which indicates the input in the decoder (provided in the Tensorflow API). Once the first Seq2Seq in the decoder is performed, the expected outcome is a conditional probability $P(y|x)$ where y indicates *normal*, and x indicates the input sequence of the data set (e.g., NSL-KDD or Kyoto-Honeypot). This outcome is then passed to the second Seq2Seq in the decoder to generate a conditional probability $P(y|x)$ where y indicates *attack*, and x indicates the context vector (i.e., the output of the encoder). The parameter configuration required in this model is the learning rate (1e-2 or 1e-3). The loss function used in this model is Mean Squared Error.



(a) Training time

(b) Performance (F-measure)

Figure 6. Experimental results for the NSL-KDD data set: (a) training time in seconds for Train+ and Train-, respectively, and (b) F-measure for the four combinations of the training and testing files. Overall, LSTM-Seq2Seq shows highly promising performance with the manageable training cost.

4. Evaluation

In this section, we report the experimental results performed with the two data sets: NSL-KDD ("balanced") and Kyoto-Honeypot ("unbalanced"). The experiments were conducted on a dedicated machine in the Google cloud.

4.1. Experimental results with NSL-KDD

As discussed in Section 2.1, the NSL-KDD data set is fairly well balanced with respect to the population of normal and attack data points. For the evaluation purpose, we employ *training time* for training complexity and *F-measure* for performance. F-measure (also known as *F1-score*) combines precision and recall and is widely used for measuring classification performance.

Figure 6 shows the experimental results for the NSL-KDD data set. We begin by reporting training time shown in Figure 6(a). The figure shows the time taken (in seconds) to train each model using the two training files. We can see that VAE-Label is the cheapest and the training cost is almost negligible for both training files. FCN is the next in terms

TABLE 2. TRAINING AND TESTING DATA SETS SELECTED FROM KYOTO-HONEYPOT

Data set	Dates	# records/day
Training	January 1–7, 2014	268K
Testing	December 1–31, 2015	236K

TABLE 3. EXPERIMENTAL RESULT WITH A DAILY DATA ON DECEMBER 1ST, 2016 IN KYOTO-HONEYPOT

Model	Precision	Recall	F-measure	MCC
FCN	99.7%	87.4%	93.1%	0.37
VAE-Label	97.5%	75.3%	85.0%	0.05
VAE-Softmax	98.1%	90.1%	93.9%	0.19
Shallow LSTM	98.3%	99.6%	99.0%	0.60
Deep LSTM	100.0%	100.0%	100.0%	1.0

of the training complexity, and the time taken for Train- is negligible. The two LSTM-Seq2Seq models (shallow and deep) show greater overheads than FCN and VAE-Label, but the training cost is still manageable (i.e., less than 25 seconds for Train+ including over 125K data points) and no significant difference between shallow and deep (22% gap at max). In contrast, VAE-Softmax running without the label information seems not scalable showing a high degree of training overhead for Train+.

Figure 6(b) shows the performance for binary classification. From the figure, the deep LSTM-Seq2Seq model shows a very promising performance (>99% of F-measure), regardless of the combinations of training and testing data sets. Even shallow LSTM-Seq2Seq consistently outperforms the other models based on FCN and VAE. Although FCN and VAE models perform worse than LSTM-Seq2Seq showing around 70%–90% of F-measure, these are still better than what we observed with the conventional machine learning techniques (see Table 1).

4.2. Experimental results with Kyoto-Honeypot

As described in Section 2.1, Kyoto-Honeypot consists of a large set of daily connection data collected from honeypots since 2009. We selected a subset of data for our experiments, as shown in Table 2. We intended to have a two-year gap for the training and testing data sets; other than that, there was no preference in the data selection. The data is highly skewed and 97% of the records are for attacks.

In this experiment, we do not rely only on F-measure to estimate performance because it could lead to a critical bias due to the high degree of skewness. For example, if a model being evaluated simply classifies the entire data points into attack, the resulted F-measure value would still be very good as the vast majority of the data points are for attacks. For this reason, we employ a measure of Mattew Correlation Coefficient (MCC) to estimate the quality of binary classification [26]. The interpretation of the MCC value would be -1.0 (poor), 0.0 (random), and 1.0 (good).

The training time taken for each model to learn from one-day data is: FCN (3.93 sec), VAE-Label (3.87 sec), VAE-Softmax (6.54 sec), and LSTM-Seq2Seq (9.20 sec). We can see that LSTM-Seq2Seq takes longer than the others, while FCN consumes the smallest time for training. However, we observed a very poor classification accuracy only with a one-day data for training the FCN and VAE models. In contrast, LSTM-Seq2Seq works very well even with a one-day training data. For this reason, we trained LSTM-Seq2Seq using a one-day data (January 1st, 2014), while the other three models were trained using a seven-day data (January 1–7, 2014).

TABLE 4. QUALITY SCORE (MCC) BETWEEN DECEMBER 1–31, 2015 IN KYOTO-HONEYPOT

Model	Best	Worst	Average
FCN	0.54	0.02	0.19
VAE-Label	0.06	0.00	0.01
VAE-Softmax	0.38	0.00	0.09
Shallow LSTM	0.84	0.10	0.44
Deep LSTM	1.00	1.00	1.00

Table 3 compares the models with a set of measures. The testing data used for this experiment is a one-day data collected on December 1st, 2015. From the table, we can see that LSTM-Seq2Seq outperforms the other models. Interestingly, deep LSTM-Seq2Seq works almost perfect even with the highly skewed data. From the F-measure, the other models seem to be working well, but their MCC values are ranged between 0.05–0.37, which are far from 1.0. In contrast, deep LSTM-Seq2Seq shows MCC=1.0 and shallow LSTM-Seq2Seq yields MCC=0.6.

We conducted the experiments against the entire data collected in December 2015 on a daily basis; hence, 31 independent results exist. Table 4 compares the constructed models with respect to MCC. In the table, Best is the best result out of 31 days and Worst is the opposite. Average is the aggregated MCC over the 31 days. The deep LSTM-Seq2Seq model works consistently across the entire days. We can also see that shallow LSTM-Seq2Seq outperforms the FCN and VAE models, but it shows a high degree of variances over days. The result shows that the VAE models do not deal with highly skewed data sets well.

From the experiments with the balanced (NSL-KDD) and unbalanced (Kyoto-Honeypot) data sets, we observed that deep LSTM-Seq2Seq works very well, yielding over 99% of accuracy with the manageable learning complexity. The FCN and VAE models showed worse results than the LSTM-Seq2Seq model but outperform the conventional machine learning techniques with much higher accuracy.

5. Conclusions

While deep learning has widely been considered for a diverse range of applications, few studies have been conducted to examine deep learning for network anomaly detection in depth. In this study, we claim why deep learning is essential for network anomaly detection, by showing a degree of non-linearity of the network traffic data. To see the feasibility of deep learning, we designed a set of deep learning models based on FCN, VAE, LSTM-Seq2Seq structures, and examined the constructed models with the public traffic data sets of NSL-KDD and Kyoto-Honeypot. Our experimental results are interesting and the model based on the LSTM Seq2Seq structure shows a highly promising performance yielding 99% of binary classification accuracy on both traffic data sets. We also confirmed that the other models work better than the conventional machine learning techniques with greater accuracy.

As one of the future directions, we plan to examine semi-supervised learning through GANs (Generative Adversarial Networks) with several benefits including the powerful handling of skewed data (like Kyoto-Honeypot). Another interesting direction would be to evaluate and optimize the deep learning models from the perspective of online analysis that requires utilizing a subset of traffic variables readily available at the collection time (rather than accounting the entire features).

Acknowledgment

This work was supported in part by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIP) (No.2016-0-00078, Cloud-based Security Intelligence Technology Development for the Customized Security Service Provisioning).

References

[1] Gartner Provides Three Immediate Actions to Take as WannaCry Ransomware Spreads. http://www.gartner.com/newsroom/id/3715918.

[2] Large DDoS attacks cause outages at Twitter, Spotify, and other sites. http://www.cisco.com/c/dam/en/us/solutions/collateral/service-provider/visual-networking-index-vni/complete-white-paper-c11-481360.pdf.

[3] http://www.eweek.com/security/ddos-attack-snarls-friday-morning-internet-traffic.html.

[4] Dapeng Liu, Youjian Zhao, Haowen Xu, Yongqian Sun, Dan Pei, Jiao Luo, Xiaowei Jing, and Mei Feng. Opprentice: Towards practical and automatic anomaly detection through machine learning. In *ACM IMC*, pages 211–224, 2015.

[5] Evangelos E. Papalexakis, Alex Beutel, and Peter Steenkiste. Network anomaly detection using co-clustering. In *Proceedings of the 2012 International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2012)*, ASONAM '12, pages 403–410, 2012.

[6] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM Comput. Surv.*, 41(3):15:1–15:58, July 2009.

[7] Yang Liu, Linfeng Zhang, and Yong Guan. A distributed data streaming algorithm for network-wide traffic anomaly detection. *SIGMETRICS Perform. Eval. Rev.*, 37(2):81–82, October 2009.

[8] Mahbod Tavallaee, Ebrahim Bagheri, Wei Lu, and Ali A. Ghorbani. A detailed analysis of the kdd cup 99 data set. In *Proceedings of the Second IEEE International Conference on Computational Intelligence for Security and Defense Applications*, CISDA'09, pages 53–58, 2009.

[9] Kyoto. http://www.takakura.com/Kyoto_data/.

[10] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

[11] Quoc V Le et al. A tutorial on deep learning part 2: Autoencoders, convolutional neural networks and recurrent neural networks. *Google Brain*, 2015.

[12] Li Deng and Dong Yu. Deep learning: Methods and applications. *Found. Trends Signal Process.*, 7(3&#8211;4):197–387, June 2014.

[13] Maryam M Najafabadi, Flavio Villanustre, Taghi M Khoshgoftaar, Naeem Seliya, Randall Wald, and Edin Muharemagic. Deep learning applications and challenges in big data analytics. *Journal of Big Data*, 2(1):1, 2015.

[14] Tuan A Tang, Lotfi Mhamdi, Des McLernon, Syed Ali Raza Zaidi, and Mounir Ghogho. Deep learning approach for network intrusion detection in software defined networking. In *Wireless Networks and Mobile Communications (WINCOM), 2016 International Conference on*, pages 258–263. IEEE, 2016.

[15] Shuangfei Zhai, Yu Cheng, Weining Lu, and Zhongfei Zhang. Deep structured energy based models for anomaly detection. In *International Conference on Machine Learning*, pages 1100–1109, 2016.

[16] Sasanka Potluri and Christian Diedrich. Accelerated deep neural networks for enhanced intrusion detection system. In *Emerging Technologies and Factory Automation (ETFA), 2016 IEEE 21st International Conference on*, pages 1–8. IEEE, 2016.

[17] Yuancheng Li, Rong Ma, and Runhai Jiao. A hybrid malicious code detection method based on deep learning. *International Journal of Security and Its Applications*, 9(5), 2015.

[18] Xiaoling Tao, Deyan Kong, Yi Wei, and Yong Wang. A big network traffic data fusion approach based on fisher and deep auto-encoder. *Information*, 7(2):20, 2016.

[19] Wenlin Chen, James Wilson, Stephen Tyree, Kilian Weinberger, and Yixin Chen. Compressing neural networks with the hashing trick. In *Int. Conference on Machine Learning*, pages 2285–2294, 2015.

[20] Jinwon An and Sungzoon Cho. Variational autoencoder based anomaly detection using reconstruction probability. Technical report, Technical Report, 2015.

[21] Pankaj Malhotra, Lovekesh Vig, Gautam Shroff, and Puneet Agarwal. Long short term memory networks for anomaly detection in time series. In *Proceedings*, page 89, 2015.

[22] KDD Cup 1999 Data. http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html.

[23] Manuel Lopez-Martin, Belen Carro, Antonio Sanchez-Esguevillas, and Jaime Lloret. Conditional variational autoencoder for prediction and feature recovery applied to intrusion detection in iot. *Sensors*, 17(9):1967, 2017.

[24] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[25] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014.

[26] Pierre Baldi, Søren Brunak, Yves Chauvin, Claus AF Andersen, and Henrik Nielsen. Assessing the accuracy of prediction algorithms for classification: an overview. *Bioinformatics*, 16(5):412–424, 2000.