Southeast-Asian J. of Sciences: Vol 5, No. 2 (2017) pp. 111-125

INTRUSION DETECTION USING DEEP NEURAL NETWORK

Pham Hoang Duy and **Nguyen Ngoc Diep**

Department of Information Technology
Posts and Telecommunications Institute of Technology (PTIT)
Hanoi, Vietnam
e-mail: diepnguyenngoc@ptit.edu.vn

Abstract

Intrusion detection is very attractive topic for both system administrators and security researchers. The problem of intrusion detection can be tackled by machine learning models, based on statistical algorithms or artificial neural networks, to identify abnormal behaviours from those of users accessing systems. The recent development of machine learning techniques and increasing computational power of graphical processing units contribute significantly to the wide spread of the deep learning technique. This report investigates the application of deep neural network to the problem of intrusion detection and compares with typical machine learning techniques based on NSL-KDD dataset.

1 Introduction

The development of computational devices and the wide spread of network applications such as e-commerce, social networks, and cloud computing make solutions of information security not only complicated but also pressing and necessary. Intrusion to a system can be defined as attempts to violate security properties of the system including confidentiality, integrity, and availability or bypasses security protection mechanisms of computational systems or networks. In other words, attackers try every possible activity to gain access to their targets; hence, their activities infringe security policies of the systems. In order

Key words: Intrusion detection, NSL-KDD dataset, deep learning.

to effectively prevent unauthorised access, the systems should be equipped with intrusion detection mechanisms and promptly informed about activities causing damage to these systems' information security.

Intrusion detection can be defined as the process identifying and responding to malicious activities to the systems. This can be done by monitoring events during the use of computational systems or networks and investigating these events for any possible signatures of intruders. Therefore, it plays a vital part in intrusion detection systems (IDS). IDS can be either hardware or software system that enables an automatic detection of malicious activities among users' access. Two popular approaches to IDS include signature detection and anomaly detection.

Identifying signatures of intrusions is the traditional and essential method for IDS [7]. These signatures can be either models of activities or sequences of characters corresponding to well-known attacks or threats. To discover an attack, IDS compares the models against events recorded during users' access to the systems. This technique is also referred as knowledge-base technique since it makes use of knowledge repository about well-known intrusions. Intuitively, this technique inefficiently tackles with intrusions which have not been recorded in the knowledge repository despite its accuracy and reliability.

The method based on anomaly detection [7] play a vital role in intrusion detection system. The anomaly activities are distinguished from normal ones by investigating users' profiles based on monitoring users' access to the systems, such as routine access, network connections, and hosts, over a certain period of time. Also, IDS can compare these profiles against recorded events in order to identify critical attacks. The anomaly detection method provides effective and efficient tools for system administrators as well as powerful users in order to successfully tackle with unknown attacks or novel malicious activities.

The problem of distinguishing anomaly among behaviours or access to systems' resources is a typical focus of machine learning techniques [15]. Essentially, machine learning techniques provide mathematical models enabling the automatic classification of users' behaviours (activities) based on features (attributes) of those behaviours. The behaviours' classification can be varied between binary and multiple classifications depending on the model design. Some of popular machine learning techniques consist of decision tree C4.5 [10], support vector machine [11], and artificial neural networks [6].

Recently, the invention of deep learning significantly influences on the development of machine learning, especially in the domain of speech recognition, image processing and natural language processing [4]. The distinctive features of deep learning account for using a huge volume of data and a large set of attributes comparing with traditional techniques. Therefore, deep learning techniques significantly improve the performance of making sense of the data. At the present, there are several research works on the intrusion detection using deep learning techniques against the dataset of KDD 99 [5] or NSL-KDD

[1, 6, 8, 12]. However, these investigations have not analysed comprehensively the performance of the proposed models with respected to output classes and the problem of imbalance of output classes in these datasets.

This paper proposes a deep neural network model for the intrusion detection and investigates the performance of the model against NSL-KDD dataset [14] using more reasonable validation method. Also the paper compares and evaluates the performance of proposed model with several well-known classification models. The proposed model is extended with a penalize algorithm for those output classes encountering the unbalanced data in order to improve the quality of classification. Finally, the paper applies the feature reduction for the dataset based on feature ranking technique so that the quality of the model can be improved, especially for the systems with limited computational capacity.

2 Overview of Intrusion Detection using Deep Neural Network

A typical traditional network of perceptron for classification model constitutes of three layers including input, hidden and output layer. This architecture is applied in the process of making sense of data by training perceptron in the hidden and output layers according to the training data. Hence, this network can learn the representation of the given dataset. This architecture can be deepened by providing additional hidden layers so that features of the given dataset are transformed at every hidden layer. Each transformation can be considered is a deduction step in the representation of the dataset.

Network of perceptron with multiple hidden layers, convolution deep neural networks, and recurrent deep neural networks are popular approaches in deep learning models. The main motivation of applying deep learning models is the efficiency compared with the traditional approaches. Furthermore, deep learning models provide advanced and novel techniques in learning functions. The success of deep learning models also accounts for the widespread of high performance computing based on graphical processing units. When data is represented as matrix of vectors, the computation is accelerated by specialised hardware and optimised graphical libraries. Therefore, the training and validating processes of the learning model can be conducted effectively and efficiently.

Deep learning models have been applied for classification and detection of behaviours of unauthorised access among normal behaviours. The authors of [1] use recurrent neural network to automatically classifying data access, such as http request, by the real-time recurrent learning technique. In the next step, the classification model applies support vector machine (SVM). The real-time learning technique allows the model can be scalable and suitable for real-time monitoring systems.

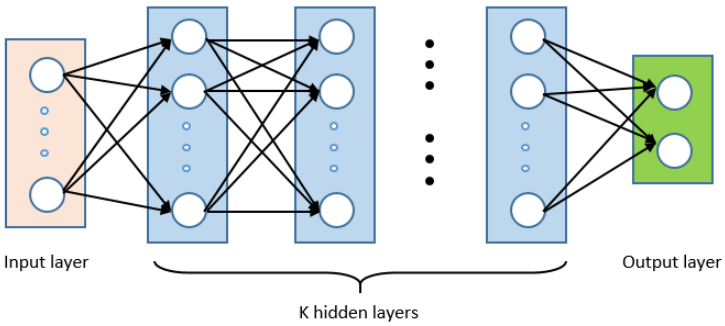

Figure 1: Deep feed-forward neural network with K hidden layers

The works of [6, 12] utilise the long-short term memory (LSTM) architecture for the recurrent neural network in order to develop detection model against KDD 99 dataset [5]. In [12], LSTM architecture is extended by assigning adaptive weights to elements in the network so that these elements can reject unexpected statuses from the inputs. The results are reasonable with the detection rate above 90%. The result from [6] is also encouraging but the proposed model just uses a partial of training data of KDD 99.

The paper [8] applies self-taught learning of deep learning techniques in order to classify intrusion behaviour and conducts experience on the NSL-KDD [14]. Essentially, the classification process is composed of two steps. In the first step, features of the dataset are obtained by the technique of sparse auto-encoder. The second step uses softmax regression technique over these features in order to fulfil the classification process.

3 Intrusion Detection Model

3.1 Deep neural network for intrusion detection

The proposed model of deep neural network relies on feed-forward neural network with 1 input, 1 output, and K hidden layers (Fig. 1). Every node in each layer fully connects to other nodes in the adjacent layer. Similar to other neural networks, deep neural network can formulate non-linear and complex relationships. The deep hidden layers can improve the feature learning capability especially for those features obtained from the previous layer. Therefore, deep learning approaches can represent complex data with smaller number of neural nodes in comparison with other neural networks.

The architecture of deep neural network based on the feed forward network does not simply extend the number of hidden layers as well as the number of

nodes in each layer but also develops advanced and novel activation functions. In the traditional neural networks, sigmoid function is the popular activation function in hidden layers. With respect to deep neural networks, the sigmoid function is not preferred because this function limits the propagation of nodes given the small or large gradients. Recently, *ReLU* functions, formulated as $f(x)=max(0,\ x)$, is more frequently applied in the hidden layers instead of the sigmoid. Thanks to its simplicity, this function enables efficient and fast training for neural nodes, especially useful for big neural networks [4].

For classification problem using neural network, these two functions are not adequate and appropriate in particular for those problems requiring more than two classes. Therefore, *softmax* function is preferred. This function not only compresses outputs of neural nodes into the range of [0,1] like sigmoid function but also divides these outputs such that their total equals to 1. For the problem differing from the binary classification, the effect of softmax function is similar to the distribution of probabilities of individual classes required the classification. Softmax function is represented as:

$$softmax(\mathbf{z})_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}} \tag{1}$$

where $\mathbf{z}$ is the input vector adjacent to the output layer, K is the total neural nodes of the output layer.

Providing the improvement in activation functions and learning methods, deep feed-forward neural networks enable considerably progress of the performance of classification. This paper proposes a neural network architecture with 4 hidden layers and each layer contains 60 neural nodes in order to identify the users' behaviours including *normal* and *attack* activities. *ReLU* serves as the activation function for the hidden layers meanwhile *softmax* is employed at the output layer. For regularization, a dropout layer is used after the last fully-connected layer. The dropout layer is applied to control over-fitting by removing an individual unit with an arbitrary probability while training the network [17]. This method can decrease over-fitting by avoiding training nodes on all training data, and lead the network to learn more robust features. The neural network is trained using mini-batches with size of 32 and the data is grouping in proportion to the distribution of individual classes in the training set. The precision of the network is optimised using the popular method titled Adam as in Keras library [2] with the learning rate of 0.001.

3.2 Data pre-processing

The deep neural network only accepts the features or attributes of users' behaviours represented as numeric values (specifically in real format) at the input layer. In fact, the actual features can take either numeric or nominal values.

For example, the transportation type can be either "*tcp*" or "*udp*". It is required to transform these values into real ones by using one-hot vector similar to that in natural language processing. The one-hot vector actually constitutes a matrix with size of 1xN so that it is possible to distinguish a word in the dataset with other words. The vector is filled with the value of 0 except the position corresponding to the word and the value at this position is set to 1.

4 Experiment Results and Evaluation

This part presents the dataset used for experimenting intrusion detection based on the classification of users' behaviours. The classification model employed the proposed deep neural network is experimented using the selected dataset . Furthermore, the result from the model is compared against the models using support vector machine (implemented by SVC), decision tree (by CART algorithm), random forest, and stochastic gradient decrease SGD. This part also shows the configuration parameters applied in the experiences of individual models.

4.1 Dataset

NSL-KDD dataset [14] is used to investigate the performance of the deep neural network. This dataset is refined from KDD 99 [5], where duplicated records are removed and the numbers of records are reasonable big for both training and testing sets. Each record constitutes 41 attributes representing different features of information flows as well as users' activities. These records are labelled as normal or anomaly. These attributes can be divided into groups relating to network connections and host traffic.

Typical attributes related to network connections are as follows:

- *duration*: length of connection in real value,
- *protocol_type*: such as tcp,
- *service*: service being used such as ftp,
- *flag*: status of connection can be normal or failure such as SF.

Typical attributes corresponding to host traffic are including, just named a few:

- *dst_host_count*: number of connections to the destination,
- *dst_host_srv_count*: number of connections to the destination with the same port,

Table 1: Classification of attack activities

Class	**Attack activities**
dos	back, land, neptune, pod, smurf, teardrop, mailbomb, proctable, udpstorm, apache2, worm
probe	satan, ipsweep, nmap, portsweep, mscan, saint
r2l	guess_pw, guess_pw, ftp_write, imap, phf, multihop, warezmast, warezcli, xlock, spy, xsnoop, snmpguess, snmpgetatt, httptunnel, sendmail, named
u2r	buf_overflow, loadmodule, rootkit, perl, sqlattack, xterm, ps

- *dst_host_same_srv_rate*: rate of the same services among connections to the host.

Beside the attributes directly representing low level information on network connections and host traffic, the dataset also provides attributes giving abstract and high level information on users' activities such as the number of login failures (*num_failed_logins*), request to host login (*is_host_login*), or attempts to elevate privilege (*su_attempted*).

Regards to abnormal behaviours, the intrusion activities can be grouped into 4 classes as follows:

- *dos*: attacks on availability of services,
- *probe*: monitoring or probing in order to obtain host information such as port scanning,
- *u2r*: unauthorised access to privileged user's account,
- *r2l*: unauthorised remote access, attackers using a remote computer to gain access to user's computer.

Table 1 describes grouping abnormal behaviours based on attack activities in detail.

As shown in Fig. 2, the instances of *normal* and *anomaly* (including dos, probe, r2l and u2r) are relatively balanced in the whole dataset (both training and testing sets). However, the detail distribution of abnormal behaviours is very different. The *dos* behaviour accounts for the highest at more than 50000 observed instances whereas *u2r* is merely 119 ones.

Overall, NSL-KDD dataset contains about 126.000 samples for 22 attack/intrusion activities. Meanwhile, the testing dataset constitutes above 22.500 samples for 37 attack/intrusion activities. The difference between training and testing sets presents the challenge to classification models. Furthermore, the imbalance among abnormal behaviours results in difficulty in training these models.

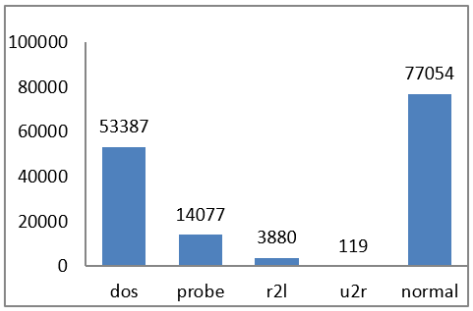

Figure 2: Distribution of abnormal and normal behaviours in the whole dataset of NSL-KDD

4.2 Experimental Settings

4.2.1 Evaluation metrics

Overall accuracy is the simple measurement, which is often used in evaluating classification performance. This measurement accounts for the proportion between the correct classified elements and the total elements being classified. However, due to the data imbalance among the classes required to be classified, this measurement is not really efficient and appropriate [13, 16]. In this situation, the measurement metrics are composed of precision, recall and F1 (harmonic mean). The precision presents the rate of correct classification/prediction of a class, while the recall illustrates the rate of actual correct classification/prediction. F1 presents the average of both precision and recall, therefore, it facilitates the performance comparison among classification models.

These measurements are defined as following formulae:

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{4}$$

where TP (true positives): correct classifications of positive cases; FP (false positives): incorrect classifications of positive cases into negative class; FN (false negatives): incorrect classifications of negative cases into positive class.

In order to evaluate the performance of classification models, 10 folds cross-validation is used in the experiment. This validation technique divides the dataset into 10 parts such that 9 parts is employed for training and the last one is for testing. This technique is repeatedly applied until all of 10 parts are passed through the model. The measurements are computed by applying average functions.

4.2.2 Parameters of classification models

To investigate the performance of the deep neural network, the paper deploys different classification models including: traditional feed-forward network (*perceptron*), support vector machine (SVM), decision tree, random forest, and stochastic gradient descent (SGD). Techniques based on decision tree or random forest algorithms are fundamentals of machine learning techniques. However, the performance of novel technique as SVM attracts researchers and developers to deploy as a reference model for evaluating their classification model. With respect to the representation, decision tree and random forest techniques provide readable and understandable results to human, especially non-expert users.

The goal of intrusion detection model is to classify users' activities into different expected classes as normal and abnormal classes given NSL-KDD dataset. Activities in the abnormal class are grouped into 4 attack/intrusion behaviours namely *dos*, *probe*, *u2r* v *r2l* (according to [14]). In other words, the detection model is required to identify the actual anomaly rather than a simple notification of the abnormal activities.

As presented in Section 4.1, NSL-KDD dataset has quite balance in the number of records of normal and abnormal activities in both training and testing sets. However, the details of grouped abnormal behaviours (*dos*, *probe*, *r2l*, *u2r*) show significantly unbalanced data in individual attack/intrusion activities. This imbalance accounts for both occurrences and types of activities related to the abnormal behaviours.

The proposed deep neural network is built and trained by Keras library [2] and the model parameters are described in Section 3. Meanwhile, parameters of other classification models are constructed and trained by using default values of Scikit-learn library [9]. In addition, due to the imbalance in the dataset the paper applies weighted parameters to interested classes. That is assigning a weight value to classes having substantially more instances than the other classes in order to improve the classification performance. This approach is facilitated by *class_weight* parameter in Keras library [2].

4.2.3 Input reduction for neural network using feature ranking

In order to accelerate the processing speed of the deep neural network as well as other models, the selection of appropriate features/attributes at the input

Table 2: F1 of deep neural network and other models

Classification model	**Mean of F1 score** (%)
DNN	**96.2**
Perceptron	71.2
Random Forest	95.6
SVM	86.6
SGD	81.2
CART	91.8

is very important. In fact, some features have minor or no impact at all on the classification performance of interested models. In this paper, the dataset is composed of 41 features but the number of features increases 3 times (to about 120) after pre-processing input data. Therefore, the feature reduction is required to improve the overall speed and performance. The algorithm for ranking features, *Input Perturbation* or *Perturb* [3], has been well-known and suitable for neural networks, therefore, it is selected for the proposed model. This algorithm justifies the change of mean squared error (MSE) of neural networks by adding a white-noise error in every input node while keeping those of other input nodes unchanged. The change in MSE for every *Input Perturbation* reflects the important level of predicted variables. The experiment result is presented in Sec. 4.4.

4.3 Evaluation of classification models

The order of results presented in the below tables and figures respectively is the deep neural network model, *DNN*; traditional neural network, *perceptron*; decision tree, *CART*; random forest; support vector machine, *SVM*; and stochastic gradient descent, *SGD*.

Table 2 enables a direct assessment of interested models by using F1-score, which combines both precision and recall. As can be seen from the table, the models including DNN, random forest, and CART lead the group with F1 value above 91%. Precisely, the proposed deep neural network model ranks the top by having F1-score at 96.2%; closely followed by the models of random forest and CART at 95.6% and 91.8% respectively. The remained models lag far behind by a magnitude of around 10%. The traditional neural network has the poorest performance with F1-score at 71.2%. This result shows that DNN model can learn more useful features to better identify intrusion behaviours on account of a reasonable amount of hidden layers.

Besides the overall performance presented in Table 2, the below part also illustrates the detailed capability of these models by examining factors of precision, recall, and F1-score for individual classified classes (normal, dos, probe,

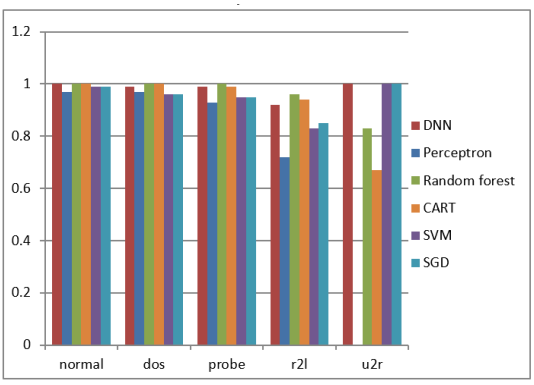

Figure 3: Precision of classification models

r2l, and u2r). These classes are corresponding to users' behaviours and should be identified by these models.

As shown in Fig. 3, all models can accurately identify the normal behaviours of users with precision factors fluctuated from 98% to 100%. In fact, traditional classification techniques, namely CART and random forest, shows a very good performance against the other techniques except for DNN model. The traditional neural network, perceptron, has the lowest performance. The result partially accounts for the big proportion of the normal behaviour in the examined dataset.

The problem in performance arises when investigating the classification results for abnormal behaviours from users' activities. The traditional techniques provide the best accuracy for attack behaviours, *dos* and *probe*, at almost 100%. The performance of DNN model (at 99%) is equally good to that of CART model and catches up with that of the random forest model. The remained models show the precision ranging from 93% to 95%.

The remarkable point is precision of classifying *u2r* attack behaviour. As shown in the previous section, this behaviour accounts for a small portion in the dataset (around 0.1% over all dataset). DNN model can detect all intrusion activities (100%). This performance is similar to that of SVM and SGD models, but the traditional neural network does not identify any case of this attack behaviour. The random forest and decision tree models have been shown relatively poor performance in this situation with the precision at around 83% and 67% respectively.

Fig. 4 presents the recall factors of examined models. DNN model is in the top followed by CART and random forest models giving the classification of normal, dos, and probe behaviours. The quality of these models is significantly fluctuated when identifying *r2l* and *u2r*. The results of perceptron and SVM

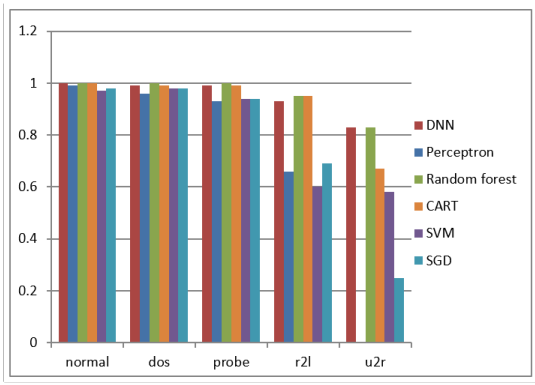

Figure 4: Recall of classification models

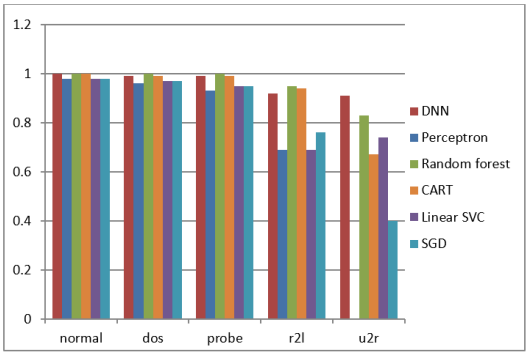

Figure 5: F1-score of classification models

models are just around 60%. In particular, perceptron model cannot detect any *u2r* attack from users' activities.

Fig. 5 outlines general capability of interested models by using F1-score factor. DNN model is equally competent to traditional techniques when identifying users' normal behaviours. The other models stay behind with the difference of about 2%. With respect to detecting abnormal behaviours, DNN model catches up with the random forest model and situates far from the perceptron and SVM models, but outperforms the other models in detecting *u2r* attack.

The result of DNN model which is deployed in this paper cannot be directly compared to researches using recurrent neural networks partially due to differences in measurement metrics and performance factors published in these works. Only the paper [12] provides detailed performance on individual behaviours classified by the research model. Despite of that, the result of this

Table 3: Top 20 important input features for neural network using Perturb feature ranking method

Feature	Error	Importance
dst_host_rerror_rate	1.378967	1
service-hostnames	1.238241	0.897948
dst_host_same_src_port_rate	0.906749	0.657557
dst_host_serror_rate	0.535132	0.388067
diff_srv_rate	0.531129	0.385164
dst_host_same_srv_rate	0.44682	0.324025
dst_host_count	0.444039	0.322008
dst_host_srv_count	0.331666	0.240517
srv_count	0.330741	0.239847
count	0.317892	0.230529
service-domain	0.284899	0.206603
wrong_fragment	0.276895	0.200799
hot	0.226532	0.164276
dst_host_diff_srv_rate	0.210075	0.152342
duration	0.202622	0.146938
service-shell	0.191343	0.138758
rerror_rate	0.18309	0.132773
service-echo	0.179432	0.13012
dst_host_srv_rerror_rate	0.174967	0.126883
same_srv_rate	0.156907	0.113786

paper illustrates another aspect on the performance of deep neural networks for the problem of user behaviours classification or intrusion detection.

4.4 Experimental result for feature reduction

The ranking of the top 20 input features using Perturb feature ranking method [3], which are useful and important for neural network given NSL-KDD dataset, is presented in Table 3. Using this compact feature set as input data for the proposed deep neural network model to recognize intrusion behaviours, we still achieve a promising result with F1 value of 93.6%, less than only 2.6% compared to the proposed model with full input features (with F1 value of 96.2%). In this modified model, we only use 16.5% number of input features (20 of 121 features) and that results in significantly increasing the performance of the model. This modified model is suitable for the systems with limited computational capacity as mobile devices.

5 Conclusion

This paper conducts an investigation on the application of deep neural networks for detecting unauthorised access to computer systems in order to protect information security. Furthermore, the paper examines the performance of deep neural networks against other models using typical and well-known classification techniques including random forest, decision tree, stochastic gradient descent, support vector machine, and traditional neural networks with NSL-KDD dataset. The 10 times cross-validation method is applied in order to evaluate the raw performance of interested models on identifying attack behaviours from users' access. The overall performance of the proposed deep neural network is relatively better than that of the random forest model and situates far from that of the other models.

With respect to detecting actually attacks, the proposed model outperforms the other models. This accounts for the reasonable amount of hidden layers that enable the proposed model to efficiently and effectively learn useful features to detect more accurately intrusion activities. Furthermore, the experimental result of applying reduction technique so that useful and important features can be retained and accelerates the training speed is quite promising. Hence, the deep neural networks can be deployed in systems with limited computational capability.

References

[1] L.O. Anyanwu, J. Keengwe, and G.a. Arome. Scalable Intrusion Detection with Recurrent Neural Networks. *Information Technology: New Generations (ITNG), 2010 Seventh International Conference on*, 6(1):21–28, 2010.

[2] François Chollet. Keras: Deep Learning library for Theano and TensorFlow. *GitHub Repository*, pages 1–21, 2015.

[3] Muriel Gevrey, Ioannis Dimopoulos, and Sovan Lek. Review and comparison of methods to study the contribution of variables in artificial neural network models. In *Ecological Modelling*, volume 160, pages 249–264, 2003.

[4] Aaron Goodfellow, Ian, Bengio, Yoshua, Courville. Deep Learning, 2016.

[5] S Hettich and S D Bay. The UCI KDD Archive [http://kdd.ics.uci.edu]. *University of California, Department of Information and Computer Science*, 1999.

[6] Jaehyun Jihyun Kim, Jaehyun Jihyun Kim, Huong Le Thi Thu, and Howon Kim. Long Short Term Memory Recurrent Neural Network Classifier for Intrusion Detection. *2016 International Conference on Platform Technology and Service (PlatCon)*, pages 1–5, 2016.

[7] Hung-Jen Liao, Chun-Hung Richard Lin, Ying-Chih Lin, and Kuang-Yuan Tung. Intrusion detection system: A comprehensive review. *Journal of Network and Computer Applications*, 36:16–24, 2012.

[8] Quamar Niyaz, Weiqing Sun, Ahmad Y Javaid, and Mansoor Alam. A deep learning approach for network intrusion detection system. In *Proceedings of the 9th EAI International Conference on Bio-inspired Information and Communications Technologies (Formerly BIONETICS), BICT-15*, volume 15, pages 21–26, 2015.

[9] Fabian Pedregosa and G Varoquaux. *Scikit-learn: Machine learning in Python*, volume 12. 2011.

[10] JR Quinlan. *C4. 5: programs for machine learning*, volume 240. 1993.

[11] Bernhard Scholkopf and Alex J. Smola. *Learning With Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. 2002.

[12] Ralf C Staudemeyer. Applying long short-term memory recurrent neural networks to intrusion detection. *Sacj*, (56):136–154, 2015.

[13] Yuchun Tang, Yan Qing Zhang, and Nitesh V. Chawla. SVMs modeling for highly imbalanced classification. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 39(1):281–288, 2009.

[14] Mahbod Tavallaee, Ebrahim Bagheri, Wei Lu, and Ali A Ghorbani. Nsl-kdd dataset. *http://www. iscx. ca/NSL-KDD*, 2012.

[15] Chih-Fong Tsai, Yu-Feng Hsu, Chia-Ying Lin, and Wei-Yang Lin. Intrusion detection by machine learning: A review. *Expert Systems with Applications*, 36(10):11994–12000, 2009.

[16] Konstantinos Veropoulos, Colin Campbell, Nello Cristianini, and Others. Controlling the sensitivity of support vector machines. *Proceedings of the international joint conference on artificial intelligence*, pages 55–60, 1999.

[17] Sida I Wang and Christopher D Manning. Fast dropout training. *Proceedings of the 30th International Conference on Machine Learning*, 28:118–126, 2013.