Computers, Materials & Continua CMC, vol.65, no.1, pp.309-320, 2020

Applying Stack Bidirectional LSTM Model to Intrusion Detection

Ziyong Ran[1], Desheng Zheng[1,*], Yanling Lai[1] and Lulu Tian[2]

Abstract: Nowadays, Internet has become an indispensable part of daily life and is used in many fields. Due to the large amount of Internet traffic, computers are subject to various security threats, which may cause serious economic losses and even endanger national security. It is hoped that an effective security method can systematically classify intrusion data in order to avoid leakage of important data or misuse of data. As machine learning technology matures, deep learning is widely used in various industries. Combining deep learning with network security and intrusion detection is the current trend. In this paper, the problem of data classification in intrusion detection system is studied. We propose an intrusion detection model based on stack bidirectional long short-term memory (LSTM), introduce stack bidirectional LSTM into the field of intrusion detection and apply it to the intrusion detection. In order to determine the appropriate parameters and structure of stack bidirectional LSTM network, we have carried out experiments on various network structures and parameters and analyzed the experimental results. The classic KDD Cup'1999 dataset was selected for experiments so that we can obtain convincing and comparable results. Experimental results derived from the KDD Cup'1999 dataset show that the network with three hidden layers containing 80 LSTM cells is superior to other algorithms in computational cost and detection performance due to stack bidirectional LSTM model's ability to review time and correlate with connected records continuously. The experiment shows the effectiveness of stack bidirectional LSTM network in intrusion detection.

Keywords: Stack bidirectional LSTM, KDD Cup'1999, intrusion detection systems, machine learning, recurrent neural network.

1 Introduction

In recent years, with the rapid development of cyberspace security, intrusion detection has received more attentions than ever. Intrusion detection is an active security protection technology, which can effectively detect the threat of possible invasive attacks, and take corresponding measures to protect cyberspace security.

[1] School of Computer Science, Research Center for Cyber Security, Southwest Petroleum University, Chengdu, 610500, China.

[2] Brunel University London, Uxbridge, Middlesex, UB8 3PH, UK.

[*] Corresponding Author: Desheng Zheng. Email: zheng_de_sheng@163.com.

Received: 11 February 2020; Accepted: 01 May 2020.

Nowadays, the main research direction of intrusion detection classification is network-based intrusion detection, upon which many intrusion detection technologies are used. Over time, more and more intrusion detection technologies will appear and become available. Among these technologies, machine learning is a typical representative.

Machine learning technology is widely used in modern society. Among them, it can be divided into two categories: "supervised learning" and "unsupervised learning" [Schrider and Kern (2018)]. The classification of network intrusion detection can be conducted through "supervised learning" to collect network traffic and build reasonable models for training. In terms of model selection, traditional machine learning methods can be selected, such as using models such as support vector machine (SVM), decision tree classification algorithm and Bayesian network for intrusion detection. It can also be classified based on current popular deep learning methods, such as constructing neural networks using convolutional neural networks (CNN), recursive neural networks (RNN), deep belief networks (DBN), and restricted Boltzmann machines (RBM). There are more applications of deep learning in both. Because the complex neural network structure has better results, the reliable support of CPU and GPU hardware devices brings speed guarantee to the training model. Today we live in the era of big data, deep learning methods are more suitable for big data than traditional machine learning methods. The complex neural network structure in deep learning can fully mine more valuable information from massive data [Sohangir, Wang, Pomeranets et al. (2018)].

In this paper, the application of the improved stack bidirectional LSTM neural network model in network intrusion detection classification is studied. LSTM is applied to the KDD Cup'1999 dataset (referred to as KDD99) which is commonly used in network intrusion detection. When studying network intrusion classification, we focus on discovering whether there are new models which is rarely used in previous studies to classify the traffics of KDD99 dataset, and this is getting a good result. The structure of this paper is as follows. The second chapter summarizes the application of deep learning in intrusion detection. The third chapter introduces the stack bidirectional LSTM. The fourth chapter elaborates the experimental process, and the fifth chapter draws the conclusion.

2 Related works

Machine learning techniques has long been combined with network intrusion detection systems, but due to the scarcity of expensive training data, one of the most comprehensive datasets currently in widespread use is still the DARPA competition dataset. In 1999, DARPA held the KDD cup at the fifth international conference on knowledge discovery and data mining. The task of this contest is to divide preprocessed connection records into normal traffic or one of four attack categories given, namely "DOS", "probe", "r2l" and "u2r". KDD99 provides a complete set of training and testing, as well as a "10%" modified subset version of the training set which is used by most of the published articles.

After the challenge, some new articles about training the KDD99 dataset using different algorithms were published. Hu et al. [Hu and Hu (2005)] constructed a network-based intrusion detection system using the classic Adaboost algorithm. Song et al. [Song, Heywood and Zincir-Heywood (2005)] and Jyothsna et al. [Jyothsna, Prasad and Prasad

(2011)] used genetic programming method to discuss and implement the parameters and evolution process of genetic algorithm in detail. Luo et al. [Luo, Wang, Cai et al. (2019)] concentrated on the abnormal data and generate artificial abnormal samples by machine learning method. A network intrusion detection method based on self-organizing feature mapping was studied [Kayacik, Zincir-Heywood and Heywood (2007)]. Horng et al. [Horng, Su, Chen et al. (2011)] proposed an intrusion detection system based on SVM, which combined hierarchical clustering algorithm, simple feature selection process and support vector machine technology to enable the obtained support vector machine model to classify network traffic data more accurately. By adopting the approximation ability of neural network, Wang et al. [Wang, Zhang, Zhou et al. (2019)] proposed a novel adaptive neural control strategy, which removes the linear growth condition assumption for nonlinearities in existing finite-time studies.

On the other hand, the research of intrusion detection system is making progress. Liao et al. [Liao, Lin, Lin et al. (2013)] attempted to give a more detailed image for a comprehensive overview. Through extensive investigation and complex organization, a classification overview of modern IDS is presented. Otoum et al. [Otoum, Kantarci and Mouftah (2019)] presented a comprehensive analysis of the use of machine and deep learning (DL) solutions for IDS systems in wireless sensor networks (WSNs) and introduce restricted Boltzmann machine-based clustered IDS. Zhao et al. [Zhao, Zhang, Shi et al. (2019)] first used the analytic hierarchy process (AHP) and natural breakpoint method (NBM) to implement an AHP-NBM comprehensive evaluation model to assess the national vulnerability. Modi et al. [Modi, Patel, Borisaniya et al. (2013)] investigated different intrusions that affect the availability, confidentiality, and integrity of cloud resources and services. And they investigate recommendations about adding intrusion detection systems (IDS) and intrusion prevention systems (IPS) to the cloud. The main point of Sommer et al. [Sommer and Paxson (2010)] is that the task of discovering attacks is fundamentally different from these other applications, making it more difficult for the intrusion detection community to use machine learning effectively. Peddabachigari et al. [Peddabachigari, Abraham, Grosan et al. (2007)] proposed an approach that combines Decision trees (DT) and support vector machines (SVM) as a hierarchical hybrid intelligent system model (DT-SVM) and an ensemble approach combining the base classifiers. The cloud computing environment experiences several security issues such as Dos attack, replay attack, flooding attack [Kalaivani, Vikram and Gopinath (2019)]. Kim et al. [Kim, Lee and Kim (2014)] proposed a hybrid intrusion detection method that layered misuse detection model and anomaly detection model into the decomposition structure. Firstly, a misuse detection model is established based on C4.5 decision tree algorithm, and then the normal training data is decomposed into smaller subsets. It is expected that in the adjacent times, portable malware will contain serious risks. Consequently, the researchers are regularly looking for explanations to handle these afresh-familiarized threats. Therefore, a necessity for a smart and useful defence panels, such as Intrusion Detection and Anticipation Systems (IDAS) is a compulsory consideration [Tariq (2019)]. Javaid et al. [Javaid, Niyaz, Sun et al. (2016)] proposed a deep learning-based approach for developing an efficient and flexible NIDS: self-taught learning. Zhou et al. [Zhou, Leckie and Karunasekera (2010)] summarized the research progress on the use of collaborative intrusion detection system (CIDSs) to detect

such attacks. Chaabouni et al. [Chaabouni, Mosbah, Zemmari et al. (2019)] reviewed existing NIDS implementation tools and datasets as well as free and open-source network sniffing software. Raza et al. [Raza, Wallgren and Voigt (2013)] evaluated a new Internet of things intrusion detection system, SVELTE, against routing attacks such as spoofing or changing information and selective forwarding implementations. Patel et al. [Patel, Taghavi, Bakhtiyari et al. (2013)] investigated, explored, and informed researchers on the latest development of IDPSs and alarm management technologies, providing a comprehensive taxonomy, and investigated possible solutions for detecting and preventing intrusion in cloud computing systems.

Many achievements have been made in the application of different algorithms and models in intrusion detection system. As one of the main development directions in the information field, big data technology can be applied for data mining, data analysis and data sharing in the massive data, and it created huge economic benefits by using the potential value of data [Wang, Yang, Wang et al. (2020)]. Greff et al. [Greff, Srivastava, Koutník et al. (2016)] demonstrated the forget gate and the output activation function to be the most critical components. Wang et al. [Wang, Hao, Ma et al. (2010)] proposed a fuzzy neural network (fc-ANN) based on artificial neural network and fuzzy clustering to solve problem. Li et al. [Li, Zhu, Huang et al. (2019)] demonstrated that machine learning methods are useful for studying the differences and commonalities of different quantum related quantization methods. To achieve high bandwidth utilization and low latency, Yu et al. [Yu, Qi, Li et al. (2018)] presented a dynamic flow scheduling mechanism based on OpenFlow protocol which enables monitoring the global network information by a centralized controller. Skaruz et al. [Skaruz and Seredynski (2007)] proposed an SQL attack detection method based on neural network. Mukherjee et al. [Mukherjee and Sharma (2012)] applied an effective classifier naive bayes to the reduction dataset for intrusion detection. Bivens et al. [Bivens, Palagiri, Smith et al. (2002)] demonstrated that neural network can be effectively applied to supervised and unsupervised learning methods of network data. Laskov et al. [Laskov, Düssel, Schäfer et al. (2005)] developed an experimental framework and proved that supervised learning technology applied to KDD Cup'1999 training data is superior to unsupervised learning technology.

3 Overview of approach

3.1 Data preprocessing

Each record in the KDD99 dataset contains 41 input features which can be divided into basic features and advanced features, and these input features include 9 discrete variables and 32 continuous variables. All traffic is either classified as "normal" or as one of the four attack types: denial of service attack (dos), network probe attack, remote-to-local attack (r2l), and user-to-administrator attack (u2r).

Since the dataset contains non-numeric information, we converted the non-numeric information into numeric values before the experiment in order to use data standardization processing. Each record in the KDD99 dataset is marked as a normal or a specific attack and is described as a vector with 41 attributes. These attributes include 38 continuous or discrete numerical attributes and 3 classification attributes. We convert symbols to integers and maintain the continuity of the data.

3.2 Stack bidirectional LSTM

LSTM is a kind of network proposed on the basis of RNN, which can learn long-term dependence and solve the gradient disappearance problem of RNN.

LSTM has a chain structure, and the repeating modules have different structures. Instead of just one layer of neural networks, there are four layers that interact in special ways, as shown in Fig. 1 [Colah (2015)]:

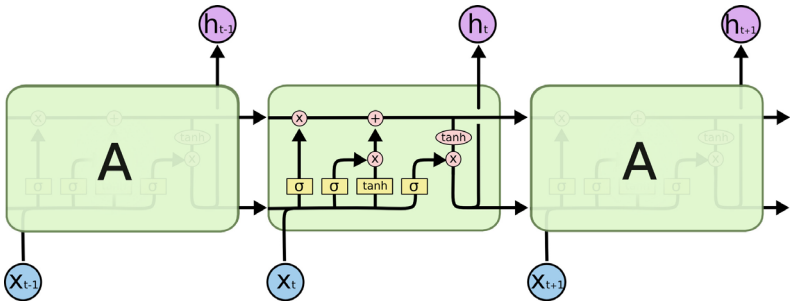

Figure 1: Interaction layer with four special ways

LSTM controls the inflow of information through the "gate". Information can be either forgotten or remembered considering whether it flows into the internal layer structure of LSTM. The gate is composed of sigmoid function and point multiplication operation. The output value of the sigmoid function is in the interval [0,1], 0 means completely discarded, and 1 means completely passed. The LSTM unit has three such gates: forget gate, input gate and output gate.

The first step of information entering LSTM will enter the forget gate, which reads h_{t-1} and x_t and outputs a value between 0 and 1 to select the degree of forgetting information of the previous unit. As shown in Fig. 2, while the output of the forget gate f_t is shown in Eq. (1) [Colah (2015)]:

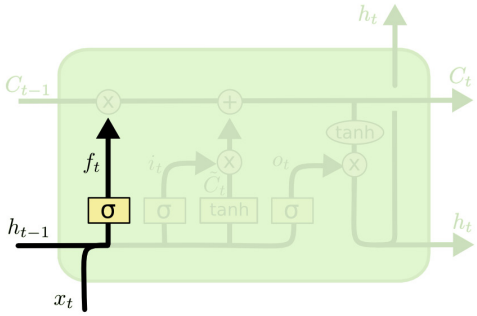

Figure 2: Forget gate

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right) \tag{1}$$

The information will then enter the input gate, in which the newly entered information will be filtered, and the selected new information will enter the next structure. The output i_t of the input gate is calculated by h_{t-1} and x_t as Eq. (2), and the input gate and tanh

function combine with the mathematical operation. The generated c_t represents the cell state after the update of the input gate. The calculation process is as Eqs. (3) and (4). As shown in Fig. 3 [Colah (2015)]:

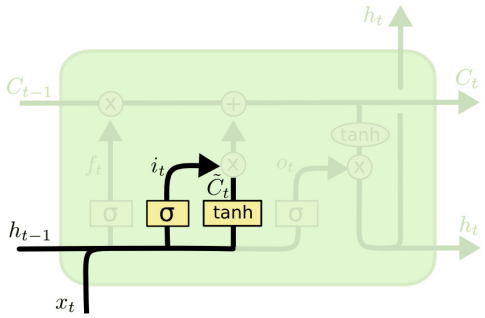

Figure 3: Input gate

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{2}$$

$$\widetilde{C_t} = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \tag{3}$$

$$C_t = f_t * C_{t-1} + i_t * \widetilde{C_t} \tag{4}$$

Finally, the information goes to the output gate, which determines how much cell state to discard. As shown in Fig. 4, output h_t of the output gate as Eqs. (5) and (6) [Colah (2015)]:

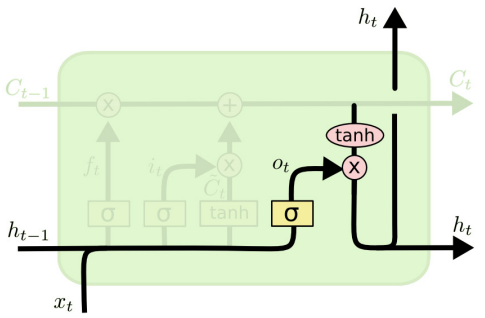

Figure 4: Output gate

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \tag{5}$$

$$h_t = o_t * \tanh(C_t) \tag{6}$$

The stack bidirectional LSTM model used in this study improves the standard LSTM model, and the output results can act on the input in the opposite direction. The continuous use of three-layer LSTM network structure is called stack LSTM. In this paper, the forward-reverse-forward three-layer network structure is called a stack bidirectional module.

3.3 Model parameters

We start with a network with 80 hidden layers, and we use sequence input neurons to input into the hidden layer, and then we add a peephole connection, finally we output layer with 5 output neurons.

Learning rate and epochs are important parameters that affect the output result. We start with 0.001 and gradually increase it to 0.1, and set the initial epochs to 200.

As the number of stack-type bidirectional LSTM modules increases, the model will become complex and easy to lead to overfitting. Therefore, the experiment starts with the number of 1 and increases by 1 each time.

The advantage of choosing cross entropy as the loss function is that the algorithm is integrated in the interface and has high comparability with other algorithms.

3.4 Optimization function

In this paper, the importance of each parameter is different, so different parameters should be adjusted dynamically and different learning rate should be adopted to make the objective function converge faster. So Adagrad is the best choice as an optimization function. The formula is as Eqs. (7) and (8): the gradient is obtained by solving the partial derivative, and then the adaptive to the independent variable w is realized by the accumulation of the gradient g^t of each time step t and the control of learning rate η.

$$w^{t+1} = w^t - \frac{\eta}{\sqrt{\sum_{i=0}^{t}(g^i)^2}} g^t \tag{7}$$

$$g^t = \frac{\partial l(\theta^t)}{\partial w} \tag{8}$$

4 Experiment

4.1 KDD99

The dataset we use is called the KDD Cup'1999 dataset [Lippmann, Haines, Fried et al. (2000)], and they were sponsored by the defense advanced research projects agency (DARPA ITO) and the air force research laboratory (AFRL/SNHS). The 1998 DARPA intrusion detection evaluation network simulated an air force base LAN used to collect seven weeks of training data and two weeks of test data. The data collected contained more than 200 instances, 39 of which were primarily network-based and embedded in backend traffic similar to the air force base LAN.

The seven-week network traffic collected from DARPA's training data was preprocessed into 5 million tagged and classified connection records, each about 100 bytes long. Two weeks of training data were processed into two million unmarked connections. Basic features extract or derive the header information of IP packets and TCP/UDP segments directly from the tcpdump file for each session. Each connection record is generated when the connection terminates or the Bro closes. Tcpdump's list file comes from training data that DARPA uses to mark connection records. It is important to note that sample distributions for probe attacks, r2l attacks, and u2r attacks vary significantly between the training and test sets. Therefore, the huge difference between KDD99's training set and test set, testing the model's ability to recognize attacks that have never been seen, makes the test set results more convincing, and it remains one of the most comprehensive datasets.

4.2 Experimental results

In this paper, a total of three sets of experiments were conducted, each of which included the Bi-LSTM parameter experiment and the control experiment of three common network models. The performance of the model was evaluated by using the KDD99 dataset, and the charts were drawn.

4.2.1 Experimental environment

The operating system used in the experiment is Ubuntu 18.04, CPU is Intel(R) Xeon(R) CPU E5-2609 v4 @ 1.70 GHz, GPU is Nvidia 1080ti, and matched with 8 GB of memory.

4.2.2 Experimental results

Under the model of three-layer stacked bi-directional LSTM module neural network with 80 hidden layers, we obtained the optimal result when the learning rate was 0.1.

We used the 10 percent modified version of KDD99 as the training set and corrected as the test set, which obtained 91.6% accuracy and 0.9 cost.

As shown in the Fig. 5, four different colors represent four different network models and we have compared three different network models. It can be seen that the accuracy obtained by using CNN model does not get a stable result. The MLP model is relatively simple, reaching convergence after 10 epochs, while the LSTM and Bi-LSTM models are relatively complex, reaching convergence after 150 and 30 epochs, respectively. Among the three convergent models, Bi-LSTM achieved the best results in terms of accuracy. Correspondingly, the cost of CNN still fluctuates in 200 epochs. Meanwhile, the cost of MLP keeps in a good level, and the curve of LSTM and Bi-LSTM reaches a stable level after 160 and 60 epochs, respectively. As can be seen from the Fig. 5, Bi-LSTM model provides a good compromise between accuracy and cost. The training process of stack bi-directional LSTM model was stable after 50 epochs, and stable after 75 epochs, and the performance was slowly optimized with the increase of epochs.

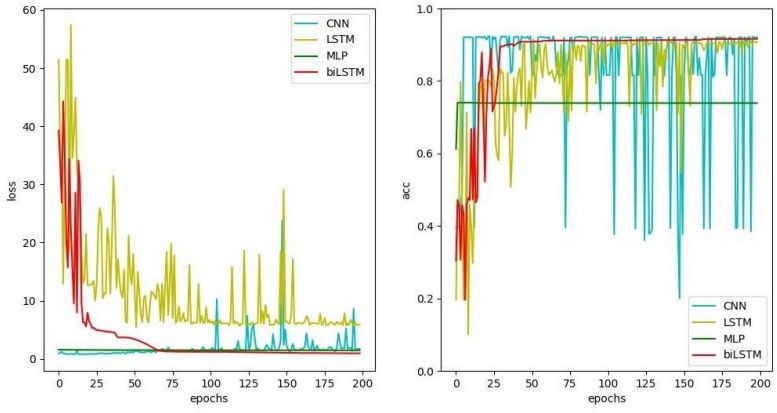

Figure 5: Performance of four network models

Tab. 1 shows the performance of LSTM and stack bidirectional LSTM networks with different feature sets on KDD99 data. There is no data because the CNN model cannot

obtain stable results. It can be seen that Bi-LSTM model is superior to other models in various indicators.

Table 1: Data from four models

Model	Acc (%)	Cost
MLP	73.90	1.46
CNN	/	1.68
LSTM	90.78	5.83
Bi-LSTM	91.63	0.94
Portnoy [Portnoy (2000)]	65.7	/

It can be seen that in the case of stable accuracy, the cost is greatly reduced, making it insensitive to the output deviating from the real value. Therefore, it is beneficial to keep the model stable when outliers exist in the observation.

5 Conclusion

The purpose of this paper is to verify the effectiveness of using stack bidirectional LSTM in intrusion detection. Four different network models were used for training, and experimental results showed that Bi-LSTM achieved a good balance between accuracy and cost. It does not matter that the attack included in the DARPA dataset was not a recent experiment because KDD99 is still one of the most comprehensive datasets and those attacks grouped into traffic classes are still valid. But we do hope that our results will apply.

Funding Statement: This work was supported by Scientific Research Starting Project of SWPU [Zheng, D., No. 0202002131604]; Major Science and Technology Project of Sichuan Province [Zheng, D., No. 8ZDZX0143]; Ministry of Education Collaborative Education Project of China [Zheng, D., No. 952]; Fundamental Research Project [Zheng, D., Nos. 549, 550].

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

Bivens, A.; Palagiri, C.; Smith, R.; Szymanski, B.; Embrechts, M. et al. (2002): Network-based intrusion detection using neural networks. *Intelligent Engineering Systems Through Artificial Neural Networks*, vol. 12, no. 1, pp. 579-584.

Chaabouni, N.; Mosbah, M.; Zemmari, A.; Sauvignac, C.; Faruki, P. (2019): Network intrusion detection for IoT security based on learning techniques. *IEEE Communications Surveys & Tutorials*, vol. 21, no. 3, pp. 2671-2701.

Colah. (2015): *Understanding LSTM Networks*. http://colah.github.io/posts/2015-08-Understanding-LSTMs/.

Greff, K.; Srivastava, R. K.; Koutník, J.; Steunebrink, B. R.; Schmidhuber, J. (2016): LSTM: A search space odyssey. *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 10, pp. 2222-2232.

Horng, S. J.; Su, M. Y.; Chen, Y. H.; Kao, T. W.; Chen, R. J. et al. (2011): A novel intrusion detection system based on hierarchical clustering and support vector machines. *Expert Systems with Applications*, vol. 38, no. 1, pp. 306-313.

Hu, W.; Hu, W. (2005): Network-based intrusion detection using Adaboost algorithm. *Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence*, vol. 1, no. 1, pp. 712-717.

Javaid, A.; Niyaz, Q.; Sun, W.; Alam, M. (2016): A deep learning approach for network intrusion detection system. *Proceedings of the 9th EAI International Conference on Bio-Inspired Information and Communications Technologies*, vol. 1, no. 1, pp. 21-26.

Jyothsna, V.; Prasad, V. R.; Prasad, K. M. (2011): A review of anomaly-based intrusion detection systems. *International Journal of Computer Applications*, vol. 28, no. 7, pp. 26-35.

Kalaivani, S.; Vikram, A.; Gopinath, G. (2019): An effective swarm optimization-based intrusion detection classifier system for cloud computing. *5th International Conference on Advanced Computing & Communication Systems*, vol. 1, no. 1, pp. 185-188.

Kayacik, H. G.; Zincir-Heywood, A. N.; Heywood, M. I. (2007): A hierarchical SOM based intrusion detection system. *Engineering Applications of Artificial Intelligence*, vol. 20, no. 4, pp. 439-451.

Kim, G.; Lee, S.; Kim, S. (2014): A novel hybrid intrusion detection method integrating anomaly detection with misuse detection. *Expert Systems with Applications*, vol. 41, no. 4, pp. 1690-1700.

Laskov, P.; Düssel, P.; Schäfer, C.; Rieck, K. (2005): Learning intrusion detection: supervised or unsupervised? *International Conference on Image Analysis and Processing*, vol. 1, no. 1, pp. 50-57.

Li, X. Y.; Zhu, Q. S.; Zhu, M. Z.; Huang, Y. M.; Wu, H. et al. (2019): Machine learning study of the relationship between the geometric and entropy discord. *EPL Europhysics Letters*, vol. 127, no. 2, pp. 20009.

Liao, H. J.; Lin, C. H. R.; Lin, Y. C.; Tung, K. Y. (2013): Intrusion detection system: A comprehensive review. *Journal of Network and Computer Applications*, vol. 36, no. 1, pp. 16-24.

Lippmann, R.; Haines, J. W.; Fried, D. J.; Korba, J.; Das, K. (2000): Analysis and results of the 1999 DARPA off-line intrusion detection evaluation. *International Symposium on Recent Advances in Intrusion Detection*, vol. 34, no. 4, pp. 162-182.

Luo, M.; Wang, K.; Cai, Z.; Liu, A.; Li, Y. et al. (2019): Using imbalanced triangle synthetic data for machine learning anomaly detection. *Computers, Materials & Continua*, vol. 58, no. 1, pp. 15-26.

Modi, C.; Patel, D.; Borisaniya, B.; Patel, H.; Patel, A. et al. (2013): A survey of intrusion detection techniques in cloud. *Journal of Network and Computer Applications*, vol. 36, no. 1, pp. 42-57.

Mukherjee, S.; Sharma, N. (2012): Intrusion detection using naive Bayes classifier with feature reduction. *Procedia Technology*, vol. 4, no. 1, pp. 119-128.

Otoum, S.; Kantarci, B.; Mouftah, H. T. (2019). On the feasibility of deep learning in sensor network intrusion detection. *IEEE Networking Letters*, vol. 1, no. 2, pp. 68-71.

Patel, A.; Taghavi, M.; Bakhtiyari, K.; JúNior, J. C. (2013): An intrusion detection and prevention system in cloud computing: a systematic review. *Journal of Network and Computer Applications*, vol. 36, no. 1, pp. 25-41.

Peddabachigari, S.; Abraham, A.; Grosan, C.; Thomas, J. (2007): Modeling intrusion detection system using hybrid intelligent systems. *Journal of Network and Computer Applications*, vol. 30, no. 1, pp. 114-132.

Portnoy, L. (2000): *Intrusion Detection with Unlabeled Data Using Clustering (Ph.D. Thesis)*. Columbia University.

Raza, S.; Wallgren, L.; Voigt, T. (2013): Svelte: real-time intrusion detection in the Internet of Things. *Ad Hoc Networks*, vol. 11, no. 8, pp. 2661-2674.

Schrider, D. R.; Kern, A. D. (2018): Supervised machine learning for population genetics: a new paradigm. *Trends in Genetics*, vol. 34, no. 4, pp. 301-312.

Skaruz, J.; Seredynski, F. (2007): Recurrent neural networks towards detection of sql attacks. *IEEE International Parallel and Distributed Processing Symposium*, vol. 1, no. 1, pp. 1-8.

Sohangir, S.; Wang, D.; Pomeranets, A.; Khoshgoftaar, T. M. (2018): Big data: deep learning for financial sentiment analysis. *Journal of Big Data*, vol. 5, no. 1, pp. 3.

Sommer, R.; Paxson, V. (2010): Outside the closed world: on using machine learning for net-work intrusion detection. *IEEE Symposium on Security and Privacy*, vol. 1, no. 1, pp. 305-316.

Song, D.; Heywood, M. I.; Zincir-Heywood, A. N. (2005): Training genetic programming on half a million patterns: an example from anomaly detection. *IEEE Transactions on Evolutionary Computation*, vol. 9, no. 3, pp. 225-239.

Tariq, U. (2019): Intrusion detection and anticipation system (IDAS) for IEEE 802.15.4 devices. *Intelligent Automation and Soft Computing*, vol. 25, no. 2, pp. 231-242.

Wang, F.; Zhang, L. L.; Zhou, S. W.; Huang, Y. Y. (2019)**:** Neural network-based finite-time control of quantized stochastic nonlinear systems. *Neurocomputing*, vol. 362, no. 1, pp. 195-202.

Wang, G.; Hao, J.; Ma, J.; Huang, L. (2010): A new approach to intrusion detection using artificial neural networks and fuzzy clustering. *Expert systems with Applications*, vol. 37, no. 9, pp. 6225-6232.

Wang, J.; Yang, Y. Q.; Wang, T.; Sherratt, R. S.; Zhang, J. Y. (2020): Big data service architecture: a survey. *Journal of Internet Technology*, vol. 21, no. 2, pp. 393-405.

Yu, H. S.; Qi, H.; Li, K. Q.; Zhang, J. H.; Xiao, P. et al. (2018): Openflow based dynamic flow scheduling with multipath for data center networks. *Computer Systems Science and Engineering*, vol. 33, no. 4, pp. 251-258.

Zhao, G.; Zhang, Y.; Shi, Y.; Lan, H.; Yang, Q. (2019): The application of BP neural networks to analysis the national vulnerability. *Computers, Materials, Continua*, vol. 58, no. 2, pp. 421-436.

Zhou, C. V.; Leckie, C.; Karunasekera, S. (2010): A survey of coordinated attacks and collaborative intrusion detection. *Computers & Security*, vol. 29, no. 1, pp. 124-140.