A SELF-ATTENTIONAL AUTO ENCODER BASED INTRUSION DETECTION SYSTEM

Bingzhang Hu and Yu Guan

School of Computing, Newcastle University, Newcastle upon Tyne, UK

ABSTRACT

Intrusion detection systems (IDSs) have received increasing attention in recent years due to the rapid development of Internet applications and Internet of Things. Anomaly based IDSs are preferred in many situations due to their capabilities of detecting novel unseen attacks. However, existing works have neither considered the intrinsic relationships within the network traffic data nor the correlations shared among the sub features (i.e. content feature, host-based feature, etc.). In this paper, we propose a self-attentional auto-encoder based intrusion detection system, namely the STAR-IDS, to effectively explore the intrinsic structures of network traffic data and evaluated it on the NSL-KDD dataset. The experimental results show that the proposed STAR-IDS has achieved state-of-the-art performances.

KEYWORDS

Intrusion Detection System, Auto Encoder, Anomaly Detection, Self-attentional

1. INTRODUCTION

Intrusion detection has been a popular topic since the emerging of Internet techniques and has received increasing attention due to the monumental growth of the Internet applications and Internet of Things (IoTs) in the past few decades. Many intrusion detection systems (IDSs) have been proposed to assist the network administrators to detect those abnormal network traffic data which may threaten the computers or network security.

Among the existing IDSs, the very early of them are mostly based on expert systems, in which a list of expert-defined signatures and patterns are matched with the input network traffic data to detect attacks. However, it is never easy to keep the libraries of such signatures and patterns up to date, therefore the rule-based expert systems suffer from unknown attacks when the incoming network traffic data contain system-agnostic attack-related signatures and patterns. Taking advantage of machine learning techniques, anomaly detection based IDSs have been proposed recently to tackle the unknown attacks, where the intrusion detection is usually treated as a classification problem. In anomaly detection based IDSs, network traffics are identified as normal traffics or abnormal ones. For example, [1] proposed a recurrent neural networks based method to identify whether the network traffic is normal or anomalous and further classify abnormal traffics into four attack types: Denial of Service (DOS), User to Root (U2R), Probe (Probing) and Root to Local (R2L), in a supervised fashion. However, supervised methods rely on sufficient training data with annotated labels, which is labour and time consuming. Without requiring labelled data, [2] employed an auto encoder network to distinguish normal and anomalous. The auto encoder is trained only on normal network traffics to minimise the reconstruction error between input and output. Hence in the deployment, the reconstruction error

David C. Wyld et al. (Eds): ISPR, ARIN, MLCL, NBIoT - 2020
pp. 53-61, 2020. CS & IT - CSCP 2020 DOI: 10.5121/csit.2020.100704

can be compared with a threshold as the reconstruction errors of abnormal data are supposed to be higher than those of normal ones.

Although many efforts have been put on learning good models [3] [4] [1] and selecting more efficient features [5], there are very few works looking at the intrinsic relationships of network traffic data. Network traffic data contain prolific information covering the intrinsic, content, host-based and time-based features of network packages [6], which describe different characteristics of network traffic. The anomaly network traffic data can be different with normal ones either at a holistic level or at a sub level, for example, the attack `port scan' is mostly different with normal traffic on the number of TCP connection requests of different ports within a short time [7], or the combination of some sub levels. To effectively and sufficiently leverage such intrinsic relationships within the network traffic data to improve intrusion detection, in this paper we propose a **S**elf-a**T**tentional **A**uto encode**R** based anomaly detection framework, namely STAR-IDS. Different with existing auto encoder based methods that directly measure the reconstruction errors, in the proposed STAR-IDS firstly split the input network traffic data into four sub features, each of which will pass through an independent auto encoder to obtain their corresponding reconstructions. Simultaneously, the original network traffic data in their entireties are fed into a self-attentional module to compute a set of attention weights that are multiplied with the reconstructions to obtain the adjusted reconstructions. Finally, the mean square error (MSE) between adjusted reconstructions and original network traffic data is compared with a threshold, and the traffic data with reconstruction errors higher than the threshold are considered as anomalies.

The remainder of this paper is organized as follows. In section 2, we review the related works of IDSs, especially those based on the anomaly detection; In section 3, we introduce the proposed Self-aTtentional Auto encodeR Intrusion Detection System (STAR-IDS). In section 4, we evaluate the proposed STAR-IDS on a benchmark dataset, and we give a brief conclusion and outlook in section 5.

2. RELATED WORKS

Intrusion detection systems can be generally divided into three categories: Signature-based (knowledge-based) Detection (SD), Anomaly-based (behaviour-based) Detection (AD) andStateful Protocol Analysis (specification-based) (SPA) [8]. In signature-based detection systems, predefined unique patterns (*e.g.* a sequence of code, a pattern or string that corresponds to a known attack, the hash code of a known bad file, *etc.*) are compared with incoming network activities to detect the intrusions. These patterns are either defined heuristically or by domain experts. In anomaly-based detection methods, the anomalies, which are defined as those network activities which differ from others enough to raise suspicion, are detected as intrusions. For the stateful protocol analysis, the protocol states are known to the system thus the vendor-developed generic profiles to specific protocols can be utilised to distinguish intrusions. Despite the effectiveness and simpleness of SDs in detecting known attacks and the ability of identifying unexpected sequences of commands in SPAs, ADs are standing out because of their abilities in detecting unknown attacks. In this section, we mainly review the existing works that belong to anomaly-based intrusion detection.

Anomaly-based intrusion detection systems typically employ supervised machine learning techniques as their backends. For example, [3] proposed a model that combines the Random Tree and Naive-Bayes Tree to classify the incoming network traffic data into two classes: normal and abnormal traffic. To explore the effectivities of features, [9] performed a feature selection and then employed the support vector machine (SVM) to detect abnormal traffics. Inspired by natural language processing, network traffic data are treated as documents in [10] and classified K-

Nearest Neighbour (K-NN). Recently, deep neural networks, as an emerging powerful machine learning technique, have also attracted much attention from the intrusion detection community. [1] proposed a recurrent neural networks (RNN) based method for intrusion detection, where each network traffic data can be identified as normal or abnormal and then further classified into a specific category including DOS, U2R, Probing and R2L. Similarly, [4] employed long-short-term-memory (LSTM) for intrusion detection and discussed the impacts of a variety of neural network architectures.

However, the aforementioned supervised learning methods, especially the deep neural networks, rely on a large volume of annotated data, which is time and labour consuming to be obtained. To tackle this problem, unsupervised approaches have been utilised. For example, [11] proposed a framework combining the self-taught learning and MAPE-K framework to deliver a scalable, self-adaptive and autonomous intrusion detection system. [2] proposed an auto-encoder based method, in which the reconstruction error between decoded traffic data and original incoming traffic data is used to validate if incoming traffic is abnormal.

3. METHODS

3.1. Overview

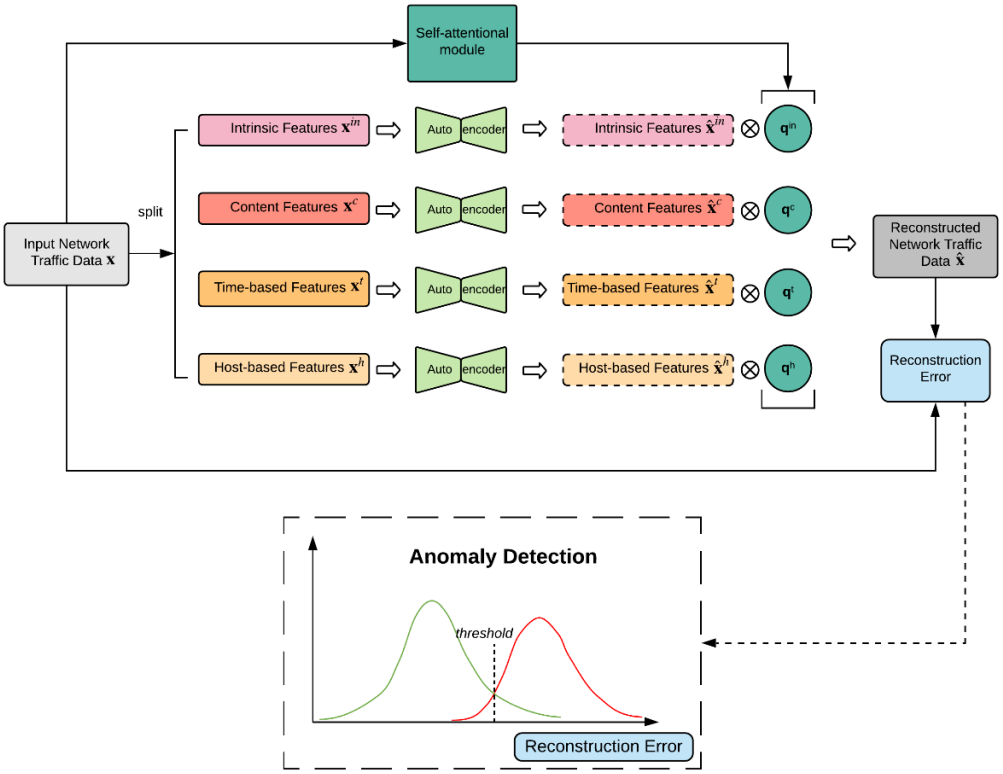

Figure 1. The framework of proposed STAR-IDS. The incoming network traffic data will be firstly split into four sub features, *i.e.* intrinsic features, content features, time-based features and host-based features. Each sub feature is encoded and decoded by an independent auto encoder.Simultaneously, the network traffic data is passed through a self-attentional module to obtain aset of attention weights, which is later multiplied with decoded sub features to get the adjusted reconstructed network traffic data. Finally, the reconstruction error is computed by measuring the l_2 loss between reconstructed and original traffic data, and the error is compared with a threshold to determine if the input traffic data is normal or abnormal.

The framework of proposed STAR-IDS is illustrated in Figure 1, from which we can see the input network traffic data, for example the i^{th} record $\mathbf{x}_i$, is first split into four subcategories: the intrinsic feature $\mathbf{x}_i^{in}$, which contains the basic information about the packet; the content feature $\mathbf{x}_i^c$, which covers the login trials, su attempts, number of roots, *etc.* information; the time-based feature $\mathbf{x}_i^t$, which records the traffic input over a two-second window; and the host-based feature $\mathbf{x}_i^h$, which shows the information over a series of connections. Then the sub features aresubsequently fed into four auto encoders to obtain their reconstructed vectors $\hat{\mathbf{x}}_i^{in}$, $\hat{\mathbf{x}}_i^c$, $\hat{\mathbf{x}}_i^t$and $\hat{\mathbf{x}}_i^h$. Simultaneously, the input network traffic data $\mathbf{x}_i$ is fed into a self-attentional module to compute four attentional weights $\mathbf{q}_i^{in}, \mathbf{q}_i^c, \mathbf{q}_i^t$ and $\mathbf{q}_i^h$ that are later on multiplied by the reconstructed feature vectors to get the adjusted reconstructed feature $\hat{\mathbf{x}}_i$. Finally, the reconstructed error is obtained between adjusted reconstructed feature and original input network traffic data and then compared with a threshold to determine if input data is anomaly. In the following subsections, we in turn introduce each component in the proposed STAR-IDS.

3.2. Auto Encoder

Auto encoder is a type of artificial neural network that is first proposed in [12] to learn internal representations by error-propagation. It is later employed in many machine learning frameworks to learn efficient representations of the input data in an unsupervised manner. The variations of auto encoder, including Sparse Auto Encoders [13], which impose the sparsity of the learnt embeddings; Denoising Auto Encoders [14], which aim to recover corrupted data by manually constructing corrupted-clean data pairs for training; Convolutional Auto Encoders [15], which introduce the convolution operations instead of linear mappings to learn the semantics; and *etc.*, have been widely used in anomaly based intrusion detection. However, none of the existing works have considered the intrinsic relations among the network traffic data and treat the sub features independently. Quoting the example in [7], if a large number of TCP connection requests to a very large number of different ports are observed within a short time, one could assume that someone is committing a `port scan'. Such kind of anomaly pattern can be found in host-based features. However, treating the network traffic data as an entirety may break such intrinsic structure. Therefore, to explicitly learn the semantic latent representations on each sub feature, we introduce four independent auto encoders to extract and learn the latent representations of each sub feature respectively. As shown in Figure 1, the sub features $\mathbf{x}^{in}$, $\mathbf{x}^c$, $\mathbf{x}^t$and $\mathbf{x}^h$ are fed into the corresponding auto encoders to obtain their reconstructed vectors as:

$$\hat{\mathbf{x}}^s = AE^s(\mathbf{x}^s), \tag{1}$$

where $s \in S = \{in, c, t, h\}$ denoting different sub features. The auto encoders in STAR-IDS

adopted the simplest fully connected architecture with a pre-defined number of hidden units, hence the mapping function AE can be written as:

$$AE^s(\mathbf{x}^s) = \sigma\left(W_{l^s}^s h_{l^s}^s + b_{l^s}^s\right),$$

$$h_l^s = \sigma(W_{l-1}^s h_{l-1}^s + b_{l-1}^s),$$

$$\cdots$$

$$h_1^s = \sigma(W_0^s \mathbf{x}^s + b_0^s), \tag{2}$$

Where σ denotes the activation function, $W\ and\ b$ are trainable weights and h are outputs of each layer. The details of the hyper-parameters and implementations of STAR-IDS can be found at https://github.com/u112358/STAR-IDS.

3.3. Self-attentional Adjusted Reconstruction

Comparing the summation of reconstruction errors between $\mathbf{x}^s$and$\hat{\mathbf{x}}^s$with a threshold, one can distinguish the anomaly from normal records. However, there exist some normal records with high reconstruction errors because they have never been seen by the auto encoder. We assume the intrinsic information in each sub features and the correlations between them in normal records are different from those of anomaly, and such kind of information can be utilised to adjust the reconstruction error. To this end, we proposed a self-attentional module, which is shown in Figure 1. Inspired by Spatial Transformer Networks (STN) [16], where a localisation net is employed to obtain a set of transformation coefficients that can recover the spatial manipulation of the input image, the self-attentional module adopts a regression network design whereby a set of attention weights is obtained from the input network traffic data and used to adjust the reconstructed sub features. Denoting the self-attentional module as f, the attentional weights can be written as $\mathbf{q} = f(\mathbf{x})$. Therefore, the adjusted reconstruction vectors can be obtained by:

$$\hat{\mathbf{x}} = [AE^s(\mathbf{x}^s) \otimes \mathbf{q}^s],$$

where $[\cdot]$ denotes the concatenation of each adjusted reconstructed sub features and $\otimes$ denotes the element-wise multiplication.

3.4. Anomaly Detection

After obtaining the adjusted reconstructed vector, the anomaly detection can be conducted by measuring the mean square error between adjusted reconstructed vector $\hat{\mathbf{x}}$ and the original incoming traffic data $\mathbf{x}$ and compare it with a pre-defined threshold δ. If the mean square error $e = \|\hat{\mathbf{x}} - \mathbf{x}\|_2^2$ is higher than the δ, then the incoming traffic data will be considered as an anomaly record and vice versa.

4. EXPERIMENTS

4.1. Dataset and Pre-processing

In this section, we evaluated the proposed STAR-IDS on the most popular NSL-KDD dataset, where the redundant records in KDD'99 [6] have been removed manually. Table 1 shows the statistic information of the NSL-KDD dataset, where we can see the proportion of normal and anomaly network traffic records in KDDTrain+ and KDDTest+ are very close. However, the distribution of each attack type (*i.e.* DoS, Probing, R2L and U2R) in KDDTrain+ and KDDTest+ are different. In KDDTrain+, the R2L and U2R attack records are minorities while in KDDTest+ the number of R2L records is close to that of Probing and occupies nearly a quarter of total records.

Table 1. Statistics of NSL-KDD Dataset.

	Abnormal				Normal	Total
	DoS	Probing	R2L	U2R		
KDDTrain+	45927	11656	995	52	67343	125973
KDDTest+	7458	2421	2754	200	9711	22544

The original network traffic records in NSL-KDD dataset are stored as 41-dimensional vectors, containing both numerical values and categorical values. To feed the data into proposed neural network, we converted the categorical values using one-hot encoding while the numerical values are normalised to range $[0,1]$. Hereby the final dimension of the data after pre-processing is 122.

4.2. Evaluations

4.2.1. Evaluation Metrics

Intrusion detection problem can also be regarded as binary classification (normal *vs.* anomaly) problem. In our experiments, four standard evaluation metrics, including accuracy, precision, recall and f-score are used to evaluate the performances of proposed method. The above-mentioned metrics are defined as:

$$Precision = \frac{\mathrm{TP}}{TP + FP},$$

$$Recall = \frac{\mathrm{TP}}{TP + FN},$$

$$F - score = \frac{2 \times Precision \times Recall}{Precision + Recall},$$

$$Accuracy = \frac{\mathrm{TP+TN}}{TP + FP + TN + FN}, \tag{3}$$

where TP (True Positive) denotes the correctly classified positive samples, FP (False Positive) denotes those negative samples classified as positive ones, TN (True Negative) denotes the correctly classified negative samples and FN (False Negative) denotes those positive samples classified as negative ones. It is important and necessary to consider various evaluation metrics at the same time as they can investigate the proposed method comprehensively hence illustrate us a complete picture of STAR-IDS.

4.2.2. Anomaly Detection on KDDTrain+

We first evaluated our proposed STAR-IDS on the KDDTrain+ Dataset. We followed the settings in [2] and conducted the training and testing process only on the KDDTrain+. The KDDTrain+ is randomly split into training, validation and test subsets, where the training subset contains 53,873 normal records only, while validation subset and test subset both contain 6,735 normal records and 6,735 anomaly records. Table 2 shows the detection performance between the proposed STAR-IDS and existing methods. We can find that among unsupervised methods, the proposed STAR-IDS has achieved better performances than [17] while yielded comparable

results with [2]. Regarding supervised methods [1] and [18], since the precision and recall rates are not given by the original papers, we leave them as absences. We can find although there is a gap between STAR-IDS and supervised methods, the results of STAR-IDS are also acceptable considering it using fewer data and no additional annotation information.

Table 2. Anomaly Detection Performances on NSL-KDD Dataset (KDDTrain+ only).

Methods	**Accuracy** (%)	**Precision** (%)	**Recall** (%)	**F-score** (%)
Auto Encoder [17]	93.62	91.39	96.33	93.80
De-noising AE [17]	94.35	94.26	94.43	94.35
AutoIDS [2]	96.45	**95.56**	**97.43**	96.49
STLNIDS [18]	98.30	n/a	n/a	**98.84**
RNN-IDS [1]	**98.81**	n/a	n/a	n/a
STAR-IDS (Ours)	95.59	95.26	95.77	95.51

4.2.3. Generalisation Abilities Analysis on KDDTest+

To further investigate the performance of proposed STAR-IDS, especially the generalisation abilities, we trained STAR-IDS on KDDTrain+ and evaluated it on KDDTest+ because of the various distribution in them. We compared the proposed STAR-IDS with several state-of-the-art methods and the results are shown in Table 3. We can find the proposed STAR-IDS has yielded the best performance among all the methods, even the supervised methods. Considering the results shown in Table 2 and Table 3, we can find the proposed STAR-IDS is stronger in detecting unseen abnormal traffic data, given that some records in KDDTest+ are novel attacks that never appears in KDDTrain+. Also, combining the results in Table 2 and Table 3, we can conclude that the STAR-IDS has better generalisation abilities and supervised methods, as well as AutoIDS, may overfit on the training set.

Table 3. Anomaly Detection Performance on NSL-KDD Dataset (KDDTrain+ & KDDTest+).

Methods	Accuracy (%)
Random Tree [3]	88.46
Random Tree and NBTree [3]	89.24
RNN-IDS [1]	83.28
DCNN [4]	85.00
STLNIDS [18]	88.39
LSTM [4]	89.00
Auto Encoder [17]	88.28
AutoIDS [2]	90.17
STAR-IDS (Ours)	**91.31**

5. CONCLUSIONS

In this paper, we proposed a self-attentional auto encoder based intrusion detection system, which takes the intrinsic relationships within the network traffic data as well as the correlations

between sub features of traffic data into account. Systematic experiments have been conducted and shown the proposed STAR-IDS is efficient and robust. However, there are still spaces to be improved in our work in the future. Firstly, the performances on the KDDTrain+ dataset show that the STAR-IDS is moderate, although the reason could be those methods are overfitted. Secondly, since the anomaly detection in STAR-IDS is based on the threshold, it will be interesting to discuss the impacts of the threshold on the performance. Finally, in the experiments, we have observed that some normal network traffic data get high reconstruction errors. It is worth to further explore that if the threshold-based STAR-IDS can be utilised for outlier detection.

ACKNOWLEDGEMENTS

This work is supported by the Engineering and Physical Sciences Research Council (EPSRC) Project CRITiCaL: Combatting cRiminals In the CLoud (EP/M020576/1).

REFERENCES

[1] C. Yin, Y. Zhu, J. Fei and X. He, "A deep learning approach for intrusion detection using recurrent neural networks," Ieee Access, vol. 5, p. 21954–21961, 2017.

[2] M. Gharib, B. Mohammadi, S. H. Dastgerdi and M. Sabokrou, "AutoIDS: Auto-encoder Based Method for Intrusion Detection System," arXiv preprint arXiv:1911.03306, 2019.

[3] J. Kevric, S. Jukic and A. Subasi, "An effective combining classifier approach using tree algorithms for network intrusion detection," Neural Computing and Applications, vol. 28, p. 1051–1058, 2017.

[4] S. Naseer, Y. Saleem, S. Khalid, M. K. Bashir, J. Han, M. M. Iqbal and K. Han, "Enhanced network anomaly detection based on deep neural networks," IEEE Access, vol. 6, p. 48231–48246, 2018.

[5] S. Aljawarneh, M. Aldwairi and M. B. Yassein, "Anomaly-based intrusion detection system through feature selection analysis and building hybrid efficient model," Journal of Computational Science, vol. 25, p. 152–160, 2018.

[6] M. Tavallaee, E. Bagheri, W. Lu and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in 2009 IEEE symposium on computational intelligence for security and defense applications, 2009.

[7] M. H. Bhuyan, D. K. Bhattacharyya and J. K. Kalita, "Network anomaly detection: methods, systems and tools," Ieee communications surveys & tutorials, vol. 16, p. 303–336, 2013.

[8] H.-J. Liao, C.-H. R. Lin, Y.-C. Lin and K.-Y. Tung, "Intrusion detection system: A comprehensive review," Journal of Network and Computer Applications, vol. 36, p. 16–24, 2013.

[9] P. Kushwaha, H. Buckchash and B. Raman, "Anomaly based intrusion detection using filter based feature selection on KDD-CUP 99," in TENCON 2017 - 2017 IEEE Region 10 Conference, 2017.

[10] Y. Liao and V. R. Vemuri, "Use of k-nearest neighbor classifier for intrusion detection," Computers & security, vol. 21, p. 439–448, 2002.

[11] D. Papamartzivanos, F. G. Mármol and G. Kambourakis, "Introducing deep learning self-adaptive misuse network intrusion detection systems," IEEE Access, vol. 7, p. 13546–13560, 2019.

[12] D. E. Rumelhart, G. E. Hinton and R. J. Williams, "Learning internal representations by error-propagation," in Parallel Distributed Processing: Explorations in the Microstructure of Cognition, MIT Press, Cambridge, MA, 1986, pp. 318-362.

[13] A. Ng and others, "Sparse autoencoder".

[14] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio and P.-A. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," Journal of machine learning research, vol. 11, p. 3371–3408, 2010.

[15] J. Masci, U. Meier, D. Cireşan and J. Schmidhuber, "Stacked convolutional auto-encoders for hierarchical feature extraction," in International conference on artificial neural networks, 2011.

[16] M. Jaderberg, K. Simonyan, A. Zisserman and others, "Spatial transformer networks," in Advances in neural information processing systems, 2015.

[17] Aygun, R. Can and Y. A. Gokhan, "Network anomaly detection with stochastically improved autoencoder based models," in IEEE 4th International Conference on Cyber Security and Cloud Computing (CSCloud, 2017.

[18] A. a. N. Q. a. S. W. a. A. M. Javaid, "A deep learning approach for network intrusion detection system," in Proceedings of the 9th EAI International Conference on Bio-inspired Information and Communications Technologies, 2016.

[19] X. Wu, V. Kumar, J. R. Quinlan, J. Ghosh, Q. Yang, H. Motoda, G. J. McLachlan, A. Ng, B. Liu, S. Y. Philip and others, "Top 10 algorithms in data mining," Knowledge and information systems, vol. 14, p. 1–37, 2008.

[20] N. Sultana, N. Chilamkurti, W. Peng and R. Alhadad, "Survey on SDN based network intrusion detection system using machine learning approaches," Peer-to-Peer Networking and Applications, vol. 12, p. 493–501, 2019.

[21] S. Mukkamala, G. Janoski and A. Sung, "Intrusion detection using neural networks and support vector machines," in Proceedings of the 2002 International Joint Conference on Neural Networks. IJCNN'02 (Cat. No. 02CH37290), 2002.

[22] S. Mukkamala and A. H. Sung, "Detecting denial of service attacks using support vector machines," in The 12th IEEE International Conference on Fuzzy Systems, 2003. FUZZ'03., 2003.

[23] P. Mishra, V. Varadharajan, U. Tupakula and E. S. Pilli, "A detailed investigation and analysis of using machine learning techniques for intrusion detection," IEEE Communications Surveys & Tutorials, vol. 21, p. 686–728, 2018.

[24] E. Hodo, X. Bellekens, A. Hamilton, C. Tachtatzis and R. Atkinson, "Shallow and deep networks intrusion detection system: A taxonomy and survey," arXiv preprint arXiv:1701.02145, 2017.

[25] D. Heckerman, "Bayesian networks for data mining," Data mining and knowledge discovery, vol. 1, p. 79–119, 1997.

[26] N. Chand, P. Mishra, C. R. Krishna, E. S. Pilli and M. C. Govil, "A comparative analysis of SVM and its stacking with other classification algorithm for intrusion detection," in 2016 International Conference on Advances in Computing, Communication, & Automation (ICACCA)(Spring), 2016.

[27] D. H. Ballard, "Modular Learning in Neural Networks.".

[28] D. E. Rumelhart, G. E. Hinton and R. J. Williams, "Learning internal representations by error-propagation," in Parallel Distributed Processing: Explorations in the Microstructure of Cognition., vol. 1, MIT Press, Cambridge, MA, 1986, pp. 318-362.