An Approach for Host-Based Intrusion Detection System Design Using Convolutional Neural Network

Nam Nhat Tran(✉), Ruhul Sarker, and Jiankun Hu

University of New South Wales Canberra at the Australian Defence Force Academy, Canberra, Australia
nam.tran@student.adfa.edu.au, {r.sarker,j.hu}@adfa.edu.au

Abstract. Along with the drastic growth of telecommunication and networking, the cyber-threats are getting more and more sophisticated and certainly leading to severe consequences. With the fact that various segments of industrial systems are deployed with Information and Computer Technology, the damage of cyber-attacks is now expanding to physical infrastructure. In order to mitigate the damage as well as reduce the False Alarm Rate, an advanced yet well-design Intrusion Detection System (IDS) must be deployed. This paper focuses on system call traces as an object for designing a Host-based anomaly IDS. Sharing several similarities with research objects in Natural Language Processing and Image Recognition, a Host-based IDS design procedure based on Convolutional Neural Network (CNN) for system call traces is implemented. The decent preliminary results harvested from modern benchmarking datasets NGIDS-DS and ADFA-LD demonstrated this approachs feasibility.

Keywords: Intrusion Detection System · Host-Based Convolutional Neural Network

1 Introduction

The issues of cyber security have increasingly attracted the social concerns in recent decades. The catastrophic consequences of cyber-crimes are the main factor contributing to the development of security systems. Almost every regime of the contemporary digital society, such as industry, military, business, medicine, and so on, are involved with the cyber-infrastructure, it is now critical to protect the integrity of information. An Intrusion Detection System (IDS) is deployed with the purpose to provide a tool to monitor system and network operations against malicious activities and policy violations [20]. The IDS plays role as a guardian to classify the activities and then to trigger defense mechanisms, if necessary. The ultimate target of an IDS is to prevent cyber-attacks, or at least mitigate an ongoing attack. Thus, an IDS must theoretically possess sensible reaction, precise detection mechanism, and secure defense techniques against

J. Hu et al. (Eds.): MONAMI 2017, LNICST 235, pp. 116–126, 2018.
https://doi.org/10.1007/978-3-319-90775-8_10

malevolent attackers. From these points of view, an IDS is a crucial component that contributes to the procedure of forensic analysis in order to identify security breaches and vulnerabilities. Moreover, the potential of an IDS is not limited to detect cyber-attacks but it also expands to noticing abnormal system behavior to detect accidents or undesired conditions [39].

The quick development of the cyberspace has led to the booming in cyber-threats, which result in the increasing of various types of attacks. Therefore, an attack is either a defined one whose signature or pattern has already been discovered, or a brand new case with unknown signature. Based on the detection methods, IDSs are categorized into misuse detection and anomaly detection. Misuse or signature-based detection operates based on the principle of comparing the collected data with a database of known attack signatures in order to determine whether a pattern is matched. It is capable of detecting predefined threats but has no ability to cope with novel threats. More importantly, with the advance of technology, attackers are able to create polymorphic threats. This is a serious loophole of the signature-based detection techniques. Anomaly detection, however, is able to resolve the problem with unknown patterns by using machine learning algorithms to build a model for normal, trustworthy activities. Any excessive deviation from the normal model would be considered as a malicious threat, thus results in alerts for protection system. The main drawback of an anomaly IDS is its suffering from false positives such that a previously unknown legitimate operation would be classified as malicious. IDSs are also sorted out by where the detection takes place, forming the Network based IDS (NIDS) and the Host based IDS (HIDS). A network based IDS deploys one or several points of observation to collect and monitor inbound and outbound network traffics, then analyses such data to make decision. A host based IDS, which is considered as the final layer of defenses, works primarily on individual hosts or devices on a network. It monitors the critical components including but not limited to configurations, system logs, system processes, files, or network interfaces of the host, compare the current states with the previous stable states. Any modification that significantly deviates the system from the normal state brings up an alert for further action. Among those various types of systems raw data, system call traces have been extensively used as a specific measure for security evaluation since [13,14]. As a low-level direct interaction with the Unix-based systems kernel, system call traces could provide rich source of activity meanings, thus it is a top list priority object for security monitoring purpose [16,17].

It is the broad yet deep prevalence of the Internet and Computer Technology makes the traditional signature-based security resolution obsolete in dealing with such polymorphism. Therefore, anomaly-based IDS is an indisputable choice as the IDS step-by-step becomes an indispensable part of an ICT infrastructure. Various detection paradigms have been proposed, that were built based on classification techniques, statistical theories, information theories, and so on [3]. Some modern concepts also contribute to the increased complexity of the problem. The brilliant emerging of the next generation cellular networks and the introduction of a new class of client, Internet of Things (IoTs) are amongst

the evidences to demonstrate that an attack could be established from everywhere. In addition, the network of sensors also poses a new challenge on the traditional security and intelligence system with a new burden in terms of data, so called Big Data [21]. The enormous amount of data from fragmented sources is inherently a huge obstacle to the effort in searching for a comprehensive yet efficient security solution. In summary, the task for constructing comprehensive IDS framework solution is now required to incorporate many more techniques as well as to consider many other aspects of the problem. Machine learning classification techniques such as Neural Network is always considered first thanks to its potential in dealing with the constantly dynamic and evolving network threats. Especially, one of Neural Network branch, CNN is very effective when dealing with classification given enormous amount of data [26]. The decent results achieved recently with CNN through applications in NLP and Image Recognition has inspired this research. This paper is organized as follow: the first part presents introduction while the second part is dedicated to reviewing related works in the literatures; the third part discusses the research methodology, and the fourth one demonstrations works preliminary results; finally, conclusion and future research is drawn in the fifth part.

2 Related Work

The ability of malicious cyber threat to transform and protect itself from the signature-based IDS has led to the emerging of anomaly-based tools. Despite the fact that those machine learning based techniques could possibly introduce some false alarms, their provision of unknown detection is extremely useful to deal against polymorphic mechanism [4]. Several machine learning techniques are used extensively, includes but not limited to Support Vector Machine [18,25,30], Neural Network [4,29], k-means Clustering (kMC) [37] and k-nearest neighbor (kNN) [27]. Among these, SVM which is based on the empirical risk minimization principle, has very high reputation as a popular and handy classification while Neural Network is gradually becoming a favor choice in IDS design thanks to its flexibility and effective classifying. The work of authors from [37] which is based on transforming system call traces into frequency vectors before reducing the dimension using Principle Component Analysis is conducted on the same ADFA-LD dataset as in this research. Therefore, these work that had conducted with kMC algorithm is correspondingly suitable to compare with our proposed work here, which fundamentally is also based on system call traces from ADFA-LD dataset.

A Convolutional Neural Network is fundamentally a Neural Network whose raw input data is divided into sub-regions and then using a specified number of filters to perform convolution before feeding through an activation function to build feature maps. The filters in convolutional layer has fixed size and will be slide horizontally then vertically (or vice versa) on "the surface" of input data until every sub-regions are covered. A non-linear activation function that is applied in the next step is typically a ReLU (*Rectified Linear Unit*) $f(x) = max(0, x)$ in

order to improve training performance on large and complex datasets [33]. On the next step, a pooling layer is deployed for down-sampling purpose (either max- or average-pooling) that could help to increase performance efficiency. The combination of three stacked layers convolutional, ReLU and pooling may be repeated in case of extracting features from high dimensional input data. After that, a fully connected layer, whose every node is connected to all the nodes from previous layer, is deployed to perform classification task. A softmax layer is also used to convert classification results into probabilities i.e., how likely an object is classified into one class.

The brilliant success of Convolutional Neural Network in classifying tasks from various technology giants like Google [23] or Facebook [2,19] has inspired the idea of using CNN for designing a multiple classification IDS. Without mentioning the achievements from industrial firms, CNN has its own reputation for applications in image processing [6,7,11,26] as well as natural language processing [8,15]. However, it must be noted that the approaching method for image processing CNN application is differed from the way a CNN-based tool can handle an NLP task. This is due to the nature of input data for each application as the raw input data for an image processing application has at least two dimensions, thus convolutional filters (kernels) would be able to slide either vertically or horizontally to extract features from sub-regions of the input data; while the representation data for NLP applications have meaning along only one specified dimension, therefore the movement of the filters are strictly restricted. The case of input data for an IDS application is very similar to NLP applications, therefore, in order to exploit CNN for designing an IDS, input data sources must be "crafted and polished" into a suitable form. According to Ronan and Jason [8], a sentence can be transformed into a matrix by considering each word as a column vector that corresponding to an index from a finite dictionary. This rudimentary representation is then used as the input for a deep neural network. This method establishes a new way to approach CNN as any piece of information can possibly be digitalizing as a string of numbers. Hence, an observation or a group of observations from a training data set can be converted into a matrix during input design procedure. Sharing the same approach method, Zhang and Wallace in [38] also transformed a sentence of words into a matrix whose each row represents a word and the number of column is the dimension of set of word. One novel characteristic from this work is the employment of various-size filters in the convolutional layers before applying 1-max pooling layer to synchronize the result from previous layer. This strategy is a noticeable suggestion when it comes to apply for CNN-based HIDS design process.

3 Proposed Method

The main element to consider when analyzing host log files of a Unix-based system is the system calls. System call is defined as a request to the kernel that is generated by an ongoing process through interrupt mechanism [35]. This request is sent to the kernel because the active process needs to access some resources.

Therefore, a system call has a great impact on system state and is exactly an object under examining when it comes to system monitoring. System calls, however, are investigated as chain rather than a standalone one [5]. This is because a malicious activity normally contains a series of tasks, each of which has a specific request to the kernel. Thus, in order to identify an attack pattern, a series of several successive system calls and the related information are collected in a window and then analyzed. These windows are usually shifted chronologically, overlap each other with a predefined step for feature extraction purpose. This method was used extensively in several literatures, including [12,22,34].

Taking the main element under examining for a Host-based IDS as sequences of system calls is investigated through [12,22,34]. However, there is no general guideline or at least, a rule-of-thumb for the task of selecting adequate length for system call sequence. As this research is only at starting phase, taking on this problem heuristically is possibly a decent solution. Raw input data will be chunked into parts with length of a sliding window (which we is trying to find an optimized value heuristically) to form the input for training data as well as testing data for the CNN-based IDS model.

3.1 Data Sets

The datasets for benchmarking purpose here are Next Generation Intrusion Detection Systems Data Set (NGIDS-DS) and ADFA Linux Dataset (ADFA-LD), the two modern datasets were generated under the next generation cyber range infrastructure of the Australian Centre for Cyber Security at the Australian Defence Force Academy [31,36]. Although there are several famous benchmarking data sets such as MIT Lincoln Laboratory's DARPA [10], KDD Cup 1999 Data [24], and NSL-KDD Dataset (an improved version of KDD'99) [32] etc., a few to names, ADFA-LD and NGIDS-DS were chosen for several reasons. First, these datasets were generated based on the modern computing infrastructures, contains up-to-date knowledge, thus it is able to reflect the latest characteristics and realistic performance of recent attacks. Second, the KDD'99 or DARPA are not only obsolete but also contains redundant information that might degrade the performance of the modern training system [28]. In addition, the NGIDS contains both host log files and .pcap (packet capture file), so it is convenient to analyze either HIDS or NIDS, or even perform a combined analysis. This is an important step toward building a comprehensive IDS, which is an indispensable part of the intelligence system for improving Critical Infrastructure Situational Awareness in future.

NGIDS-DS contains 99 host log files with csv format. Each record fully describes the relevant information about happened event, for both normal and malicious activities, including (Fig. 1) timestamp (date and time), event ID, path, process ID, system calls, attack category, attack subcategory, and label ("1" is marked for an attack and "0" is for a normal activity). A sliding window with fixed size will slide chronologically, extracts system calls and corresponding labels to form raw input data for CNN. The sliding windows may or may not overlap each other, which is also a parameter to determine. The principle of

making label for a sliding window is based on the fact that if a window contains any event marked "1", the label for such window should be "1" also. Only when every activity inside a window are marked "0", the label for that window is "0". Figure 1 illustrates the sliding window method with size of 5 and no overlap. ADFA-LD is already divided into Attack, Training and Validation datasets. Each dataset contains various files of system call traces. Deploying the same sliding window approach as with NGIDS-DS, we also extracted raw input data for both training and testing phases successfully.

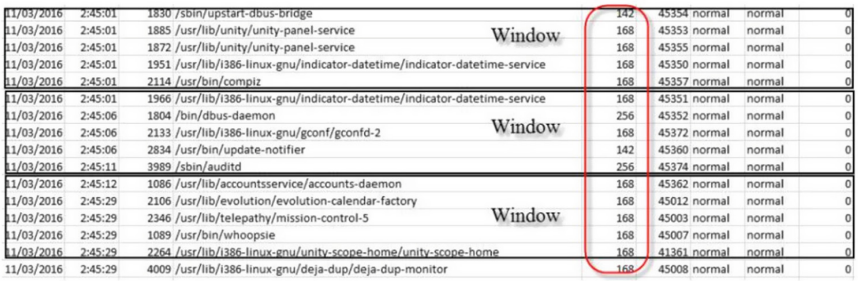

Fig. 1. NGIDS-DS with sliding window for extracting data for CNN-based IDS training and testing phases

3.2 Convolutional Neural Network Based IDS

As mentioned above, CNN has been widely used for visual recognition and natural language processing, which are highly classification-oriented application. However, CNN is unprecedentedly applied to an IDS design procedure, even the input design is not straightforward as the others. Despite that challenge, CNN is undoubtedly suitable for this research in terms of availability of data. Since a CNN training phase requires a huge amount of data [26], the NGIDS-DS and ADFA-LD with up to hundreds of million observations are more than enough to work on. The issue with input data incompatibility was solved by using 4-dimensional (4-D) arrays, a popular feature that is available for frameworks like MATLAB [9] or TensorFlow [1]. An observation from raw input data will be transformed into an element of 4-D array, whose the first two dimensions are matrix sizes with the number of row is inherently one, and the number of column is equal to the size of sliding window. The third dimension of 4-D array is one, equivalently to be the channel in RGB image. The fourth dimension is the number of observation in the data set. This technique proved to work seamlessly with Matlab as well as TensorFlow.

For the sake of simplicity, time saving and due to the property of input data, the CNN architecture was merely deployed as follow with one convolutional layer only:

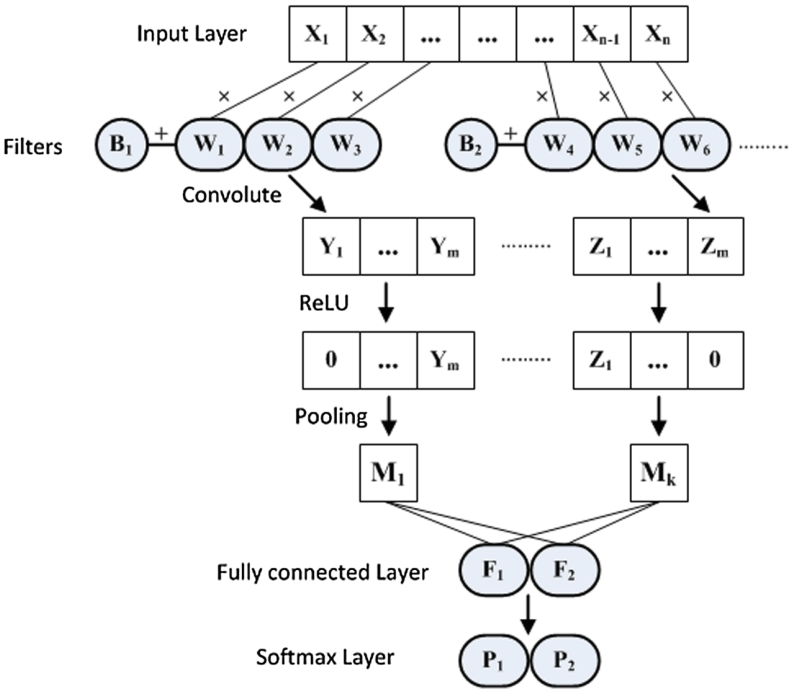

Fig. 2. Convolutional neural network architecture for Host-based IDS

The "Input Layer" has size of either [1 9 1] or [1 5 1], with 9 and 5 is the two best window sizes, which was determined heuristically. With such input data, the size of Filters (kernels) is also [1 x] with x is adjusted accordingly. Applying each filter respectively to the input entity, performs element-wise product between an input value $\mathbf{X}$ with the filter's weight $\mathbf{W}$ then adds a bias $\mathbf{B}$ to the result, an scalar value $\mathbf{Y}$ is yielded. Sliding the filter along the input by a fix stride (step) produces a string of scalar value with the same height as the input data but smaller width. In the next step, those strings of number will be fed through a ReLU layer in order to introduce nonlinearities to the model. A ReLU keeps the same all non-negative values while to replace any input smaller than zero by zero. Then, a pooling layer (either a Max Pooling or Average Pooling depends on whichever provides a better performance) is applied for down-sampling data (a 1-pooling layer, which results in a scalar output, is demonstrated in Fig. 2). A fully connected layer whose each element is connected with all elements of the previous layer, is armed in order to perform classification based on features extracted from these previous layers. At this stage, a dropout layer with changeable drop rate is also deployed for solving over-fitted situations. Finally, softmax layer which plays role as a medium to convert the classification results into probabilities, is presented. The training process was conducted with *Stochastic Gradient Descent with Momentum* (SGDM) algorithm. With SGDM, the size of mini batch is also

an adjustable parameter, which was chosen based on optimized performance at the expense of training time. Two regularization methods L1 and L2 were also applied to enhance the experimental results.

4 Experimental Results and Analysis

The training and testing processes were conducted repeatedly through different sets of parameters. The two best sets of results are achieved with window size of 5 and 9. The Detection Rate (DR) which is calculated as the ratio of successfully detected abnormal events and the total number of abnormal events, and the False Alarm Rate (FAR) which is the average of the false positive rate and the false negative rate, are shown in Fig. 3. The temporary maximum DR is 61.18%, achieved through running ADFA-LD dataset with window size of 9. Meanwhile, the best DR value from NGIDS-DS is 60.46%, but with window size of 5. In conjunction with the detection rate of 60.46% for NGIDS-DS is the FAR 20.64%. Besides, the best DR from ADFA-LD is 61.18%. These results are somehow better than the counterpart of the work in [37] with DR of 60% and FAR of 20%. In addition, the kMC technique used in that literature is now considered obsolete due to two following reasons: First, in the Big Data era, the rapid growth in size of data significantly increases the computational cost for kMC. Secondly, as new data appears, the kMC-based algorithm needs remodelling, which means that training must be conducted again. The Neural Network approach, on the other hand, demonstrates its advantage in dealing with large-scale data, while is maintaining the similar, competitive experimental results.

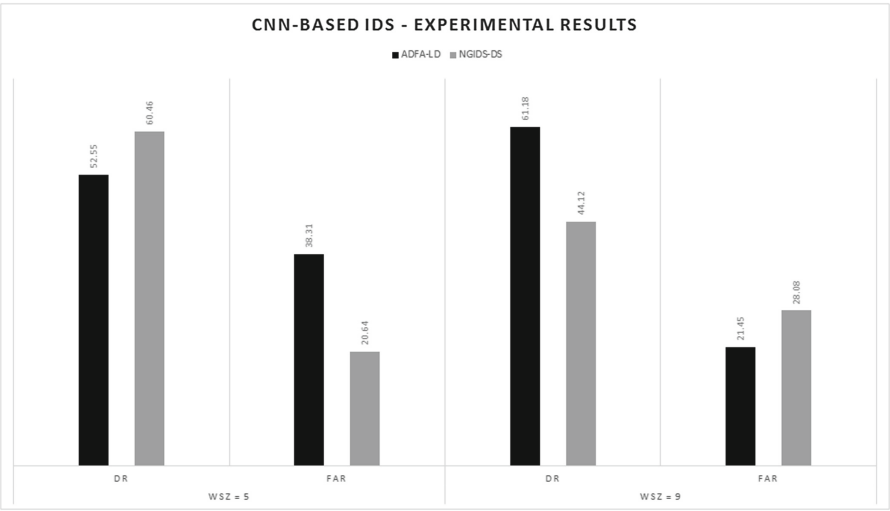

Fig. 3. The CNN-based IDS experimental results with different window sizes

5 Conclusion

The preliminary work in this paper has introduced a novel yet feasible approach method for Host-based Intrusion Detection System design. The design product worked well with large-scale raw input data, provided several decent experimental results from an extremely simple yet minimalistic architecture. This research has extended the field of application for Convolution Neural Network to a completely new regime, which, at first is seemed to be irrelevant. The detection rate as well as false alarm rate would possibly be improved by applying a more complicated model for CNN. As the neural network is inherently about feature extraction, a complex model would possibly produce more unique features, which in turn could certainly improve the classification results.

References

1. A Guide to TF Layers: Building a Convolutional Neural Network. https://www.tensorflow.org/tutorials/layers. Accessed 08 Mar 2017
2. A path to unsupervised learning through adversarial networks. https://code.facebook.com/posts/1587249151575490/a-path-to-unsupervised-learning-through-adversarial-networks/. Accessed 03 Mar 2017
3. Ahmed, M., Mahmood, A.N., Hu, J.: A survey of network anomaly detection techniques. J. Netw. Comput. Appl. **60**, 19–31 (2016)
4. Ashfaq, R.A.R., et al.: Fuzziness based semi-supervised learning approach for intrusion detection system. Inf. Sci. **378**, 484–497 (2017)
5. Canzanese, R., Mancoridis, S., Kam, M.: System call-based detection of malicious processes. In: 2015 IEEE International Conference on Software Quality, Reliability and Security (QRS), pp. 119–124. IEEE (2015)
6. Ciregan, D., Meier, U., Schmidhuber, J.: Multi-column deep neural networks for image classification. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3642–3649. IEEE (2012)
7. Ciresan, D.C., et al.: Convolutional neural network committees for handwritten character classification. In: 2011 International Conference on Document Analysis and Recognition (ICDAR), pp. 1135–1139. IEEE (2011)
8. Collobert, R., Weston, J.: A unified architecture for natural language processing: deep neural networks with multitask learning. In: Proceedings of the 25th International Conference on Machine Learning, pp. 160–167. ACM (2008)
9. Convolutional Neural Networks Matlab Documentation. https://au.mathworks.com/help/nnet/convolutional-neural-networks.html. Accessed 08 Mar 2017
10. DARPA Intrusion Detection Data Sets. https://www.ll.mit.edu/ideval/data/. Accessed 28 Feb 2017
11. Egmont-Petersen, M., de Ridder, D., Handels, H.: Image processing with neural networks—a review. Pattern Recogn. **35**(10), 2279–2301 (2002)
12. Fan, S., et al.: A dynamic on-line sliding window support vector machine for tunnel settlement prediction. In: 2013 3rd International Conference on Computer Science and Network Technology (ICCSNT), pp. 547–551. IEEE (2013)
13. Forrest, S., Hofmeyr, S., Somayaji, A.: The evolution of system-call monitoring. In: Annual Computer Security Applications Conference, ACSAC 2008, pp. 418–430. IEEE (2008)

14. Forrest, S., et al.: A sense of self for unix processes. In: Proceedings of 1996 IEEE Symposium on Security and Privacy, pp. 120–128. IEEE (1996)
15. Graves, A., Mohamed, A., Hinton, G.: Speech recognition with deep recurrent neural networks. In: 2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 6645–6649. IEEE (2013)
16. Hoang, X.D., Hu, J., Bertok, P.: A multi-layer model for anomaly intrusion detection using program sequences of system calls. In: Proceedings of 11th IEEE International Conference. Citeseer (2003)
17. Hofmeyr, S.A., Forrest, S., Somayaji, A.: Intrusion detection using sequences of system calls. J. Comput. Secur. **6**(3), 151–180 (1998)
18. Horng, S.-J., et al.: A novel intrusion detection system based on hierarchical clustering and support vector machines. Expert Syst. Appl. **38**(1), 306–313 (2011)
19. Introducing DeepText: Facebook's text understanding engine. https://code.facebook.com/posts/181565595577955/introducing-deeptext-facebook-s-text-understanding-engine/. Accessed 03 Mar 2017
20. Intrusion Detection System. https://en.wikipedia.org/w/index.php?title=Intrusion_detection_system. Accessed 30 Nov 2016
21. Jaradat, M., et al.: The internet of energy: smart sensor networks and big data management for smart grid. Procedia Comput. Sci. **56**, 592–597 (2015)
22. Kaneda, Y., Mineno, H.: Sliding window-based support vector regression for predicting micrometeorological data. Expert Syst. Appl. **59**, 217–225 (2016)
23. Karpathy, A., et al.: Large-scale video classification with convolutional neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1725–1732 (2014)
24. KDD Cup 1999 Data. http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html. Accessed 28 Feb 2017
25. Khan, L., Awad, M., Thuraisingham, B.: A new intrusion detection system using support vector machines and hierarchical clustering. VLDB J. Int. J. Very Large Data Bases **16**(4), 507–521 (2007)
26. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems, pp. 1097–1105 (2012)
27. Liao, Y., Vemuri, V.R.: Use of k-nearest neighbor classifier for intrusion detection. Comput. Secur. **21**(5), 439–448 (2002)
28. Moustafa, N., Slay, J.: UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). In: 2015 Military Communications and Information Systems Conference (MilCIS), pp. 1–6. IEEE (2015)
29. Mukkamala, S., Janoski, G., Sung, A.: Intrusion detection using neural networks and support vector machines. In: Proceedings of the 2002 International Joint Conference on Neural Networks, IJCNN 2002, vol. 2, pp. 1702–1707. IEEE (2002)
30. Mukkamala, S., Sung, A.H.: Detecting denial of service attacks using support vector machines. In: The 12th IEEE International Conference on Fuzzy Systems, FUZZ 2003, vol. 2, pp. 1231–1236. IEEE (2003)
31. Next Generation Intrusion Detection Systems Data Set (NGIDS-DS): Overview. https://research.unsw.edu.au/sites/all/files/facultyadmin/ngids-ds_overview_final.pdf. Accessed 28 Feb 2017
32. NSL-KDD Data Set. http://www.unb.ca/cic/research/datasets/nsl.html. Accessed 28 Feb 2017
33. Rectifier (neural networks). https://en.wikipedia.org/wiki/Rectifier_(neural_networks). Accessed Mar 2017

34. Suzuki, Y., et al.: Proposal to sliding window-based support vector regression. Procedia Comput. Sci. **35**, 1615–1624 (2014)
35. System Call Definition. http://www.linfo.org/system_call.html. Accessed 01 Feb 2017
36. The ADFA Linux Dataset (ADFA-LD). https://www.unsw.adfa.edu.au/australian-centre-for-cyber-security/cybersecurity/ADFA-IDS-Datasets/. Accessed 28 Feb 2017
37. Xie, M., Hu, J., Yu, X., Chang, E.: Evaluating host-based anomaly detection systems: application of the frequency-based algorithms to ADFA-LD. In: Au, M.H., Carminati, B., Kuo, C.-C.J. (eds.) NSS 2014. LNCS, vol. 8792, pp. 542–549. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-11698-3_44
38. Zhang, Y., Wallace, B.: A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification. In: arXiv preprint arXiv:1510.03820 (2015)
39. Zuech, R., Khoshgoftaar, T.M., Wald, R.: Intrusion detection and big heterogeneous data: a survey. J. Big Data **2**(1), 3 (2015)