jamk.fi

PLEASE NOTE! THIS IS PARALLEL PUBLISHED VERSION /
SELF-ARCHIVED VERSION OF THE OF THE ORIGINAL ARTICLE

This is an electronic reprint of the original article.
This version *may* differ from the original in pagination and typographic detail.

Author(s): Kokkonen, Tero; Samir, Puuska; Alatalo, Janne; Heilimo, Eppu; Mäkelä, Antti

Title: Network Anomaly Detection based on WaveNet

Version: author's accepted manuscript (AAM)

Copyright: © Springer Nature Switzerland AG 2019

Please cite the original version:
Kokkonen, T., Samir, P., Alatalo, J., Heilimo, E., & Mäkelä, A. (2019). Network Anomaly Detection based on WaveNet. In: Galinina O., Andreev S., Balandin S., Koucheryavy Y. (eds) Internet of Things, Smart Spaces, and Next Generation Networks and Systems. NEW2AN 2019, ruSMART 2019. Lecture Notes in Computer Science, vol 11660. Springer, Cham

DOI: 10.1007/978-3-030-30859-9_36

URL: https://doi.org/10.1007/978-3-030-30859-9_36

Network Anomaly Detection based on WaveNet

Tero Kokkonen, Samir Puuska, Janne Alatalo, Eppu Heilimo, and Antti Mäkelä

Institute of Information Technology, JAMK University of Applied Sciences,
Jyväskylä, Finland
{tero.kokkonen, samir.puuska, janne.alatalo, eppu.heilimo,
antti.mäkelä}@jamk.fi

Abstract. Increasing amount of attacks and intrusions against networked systems and data networks requires sensor capability. Data in modern networks, including the Internet, is often encrypted, making classical traffic analysis complicated. In this study, we detect anomalies from encrypted network traffic by developing an anomaly based network intrusion detection system applying neural networks based on the WaveNet architecture. Implementation was tested using dataset collected from a large annual national cyber security exercise. Dataset included both legitimate and malicious traffic containing modern, complex attacks and intrusions. The performance results indicated that our model is suitable for detecting encrypted malicious traffic from the datasets.

Keywords: Intrusion Detection · Anomaly Detection · WaveNet · Convolutional Neural Networks

1 Introduction

Intrusion detection systems (IDS) are divided into two categories: anomaly-based detection (anomaly detection) and signature-based detection (misuse detection). Anomaly-based-detection can be applied without pre-recorded signatures for unknown attack patterns and even for encrypted network traffic, however the weakness for anomaly detection is the high amount of false positive detections [3, 13].

Machine learning techniques have recently been applied successfully to network anomaly detection and classification [6]. Bitton and Shabtai in [1] have studied machine learning based IDS for Remote Desktop Protocols (RPD). Different machine learning techniques have been applied, e.g. Wiewel and Yang used Variational Autoencoder in their study [28], Chen et al. used Convolutional Autoencoder [2] while Long Short-Term Memory (LSTM) and Gated Recurrent Unit methods are used in the paper [6]. Paper [23] presents technique for increasing detection accuracy with feedback.

In our earlier study [19], we used Haar wavelet transforms and Adversarial Autoencoders (AA) [10] for implementing unsupervised network anomaly detection based IDS. Our earlier model, described in [19], had reasonable good operational characteristics; in this study we strived to improve it using alternative modeling approach. As argument of efficiency, numerical results are compared with the earlier results using the same dataset from Finland's National

Cyber Security Exercise [12]. Performance characteristics are also accomplished using publicly available reference intrusion detection evaluation dataset (CICIDS2017) [27].

Our study presents state-of-the-art network anomaly detection based intrusion detection system that exploits deep learning method WaveNet [15]. First, in section 2, this paper describes implemented anomaly detection method including feature extraction and analysis method. Then, in section 3, we introduce experimental results for the performance characteristics of our model and finally there are conclusions with found future research topics.

2 Anomaly Detection Method

2.1 Dataset

According to Nevavuori and Kokkonen [14], a network anomaly detection data set must (i) include network traffic data and (ii) host activity data, (iii) multiple scenarios, (iv) be representative of real-world circumstances, and (v) the format of the data must be usable.

Since many publicly available datasets already exist [20], we decided to utilize them in this research. Although notable public datasets, such as the KDD99 [25] and DARPA datasets [7–9] exist and are used in many existing network intrusion detection research, they are very old, and many researches have directed a lot of criticism against them [11, 24]. The main problem is that datasets do not include modern threat and attack patterns with required statistical characteristics nor sophisticated and modern architectures [14, 26, 4, 22]. In many datasets the raw data is already processed into network flows losing the information of individual packet timings. Fortunately, in addition to the processed flow data, some datasets include the raw packet captures.

The Intrusion Detection Evaluation Dataset (CICIDS2017) by the Canadian Institute for Cybersecurity [27] is one of the more modern publicly available datasets. Although the dataset was created with a traffic generator, it was modeled after modern real-world network traffic. It includes benign HTTPS network traffic and therefore is suitable for research concerning encrypted communication. Unfortunately, the dataset does not include many TLS based attacks, which form a sizable amount of modern malware control channels.

We decided to use the benign traffic from the CICIDS2017 dataset as clean traffic during the model development and testing, but because the anomalous traffic in the dataset was not large enough, more anomalous traffic was required. We generated additional anomalous traffic in our own environment using Empire PowerShell post-exploitation agent [1] and Cobalt Strike [2]; both are adversary simulation frameworks that use real-world malware characteristics. A small amount of benign traffic was also generated in the environment. The benign traffic was

[1] https://www.powershellempire.com/
[2] https://www.cobaltstrike.com/

generated by controlling Windows virtual machine using a scripted bot that operated normal GUI software with virtual mouse and keyboard aided by computer vision. This data was used in the evaluation to make sure that the environments are compatible enough so that our generated benign traffic is not classified as anomalous with the model that is trained with the CICIDS2017 benign data.

In addition to the CICIDS2017 dataset and the self-generated dataset, the final model was also tested with the Finland's National Cyber Security Exercise dataset (FNCSE2018), also used in our previous publication [19]. This dataset was used to get comparable results to our previous research. RGCE Cyber Range (Realistic Global Cyber Environment) is used for research and development or training and exercises. In the RGCE Cyber Range main structures and services of the real Internet are modeled with the realistic user traffic patterns of users. RGCE offers tailored organization environments with real assets [5]. Finland's National Cyber Security Exercise is conducted annually in the RGCE Cyber Range. Network data from the real Cyber Security Exercise conducted in the RGCE Cyber Range includes realistic complex environment and legitimate network traffic mixed with modern attack patterns for testing the capabilities of Intrusion Detection System capability. [12] In this study we were authorized to use the traffic captures from Finland's National Cyber Security Exercise of 2018.

2.2 Feature Extraction

Our research focused on finding the anomalies based on packet timing patterns. This choice was made to accommodate encrypted command and control channels modern malware use. Traditional deep inspection techniques and statistical analyses that utilize payloads are incompatible with modern security landscape, made e.g. decrypting proxies obsolete due to various certificate pinning features. In this project we used a modified version of Suricata IDS software [18] to process the raw packet capture files into parsed network data. The modification in the software allowed the packet timings information to be extracted from packet capture files along with the parsed data.

The CICIDS2017 dataset includes the raw packet captures in addition to labeled processed flow data. Since the processed flow data does not include packet timings, the raw data had to be reprocessed to flow data with the modified Suricata software. The processed flows were then labeled by joining the flows to the CICIDS2017 flow labels by matching flow timestamps, IP addresses and network ports. The result was labeled flows from the CICIDS2017 dataset including packet timings. Because our system used different software for packet capture to network flow conversion from the one used in CICDS2017, the resulting flows did not match exactly, resulting in lost flows. Only the flows that matched correctly between Suricata processed flows and CICDS2017 labeled flows were retained in the dataset. Based on the flow label, the dataset was then split to anomaly and benign flows. All the flows that did not have *benign* label were treated as anomalies. The final processed CICDS2017 dataset included 1,425,742 flows, of which 1,107,695 were labeled as benign flows, and 318,047 flows were labeled as

non-benign flows. From the 1,107,695 benign flows, 307,771 were TLS flows. Originally the Suricata processed CICIDS2017 packet capture files included 1,956,363 flows, so 530,621 flows did not find matching flow in the CICIDS2017 flow label files. This can be almost certainly accounted on the poor quality of the flow label files in CICIDS2017 dataset. The files include a duplicate entry for most of the flows and the flow timestamps are recorded in a minute accuracy with an ambiguous 12-hour clock format.

The FNCSE2018 dataset and our self generated datasets were processed in the same way. The labels were assigned by hand based on known origin and destination addresses of the attacks. The FNCSE2018 dataset included 715,158 benign TLS flows, and 653 non-benign TLS flows. The self generated dataset included 15,124 benign flows and 7,991 non benign flows.

The resulting flows were then further processed by calculating timing differences between packets. The final features for one packet in a flow were: *packet direction*, *time difference to next received packet*, *time difference to next transmitted packet* and *packet size*. The timing differences varied from microseconds to minutes with most of the differences being very small. Because our model required quantization of the input data, the timing differences were scaled with the common logarithm to better utilize the reduced quantization precision. The packet sizes were scaled in similar way for the same reason. This choice is warranted, because in network traffic large delays are often the result of an unrelated problem, and not an inherent feature of the protocol in question. Although many protocols, including malware command channels, may use delays and timers, there usually is no reason to keep using the same flow. Packet sizes follow the same scaling principle, the maximum size being the MTU of the path. Small packet sizes and the variation therein are are likely to be indicative of the intrinsic properties of the protocol, unlike the variation near the MTU. This is especially apparent in many malware communication protocols, which often use fixed size binary messages. The aforementioned adversary simulation frameworks also exhibit this phenomenon.

2.3 Multi-feature WaveNet

The network traffic was analyzed with a deep neural network model based on the WaveNet [15] architecture, illustrated in the Figure 2. WaveNet was chosen as a basis for our model for its capability to directly interface with variable length sequential data. This enables us to feed complete and unreduced sequences to the model. We utilized this trait to predict network traffic connections of varying length packet by packet.

The primary task of the model is to predict the next sample by using prior samples. The core network structure consists of a variation of the WaveNet architecture configured for multiple features. The modified WaveNet is extended to utilize two-dimensional dilated causal convolutions; input data is arranged into a two-dimensional lattice, discrete time steps forming the first dimension and individual sample features along the other dimension. Dilated convolutions expand the receptive field of the network exponentially [29], giving the model a

potential to observe long term temporal dependencies. Dilation of convolutions is only performed along the time axis of the data, as the receptive fields are exceedingly large and thus not optimal for the relatively small fixed length feature axis. The causality aspect of the convolutions is used to assert an ordered time-dependency on the input data: predicted samples may only depend on preceding input samples. We implemented the causality by padding the beginning of the sequence by the filter size in the first layer and by (filter size$-1)\times$dilation rate in the subsequent layers, effectively shifting the convolution operations. The causal layer stack is visualized in Figure 1.

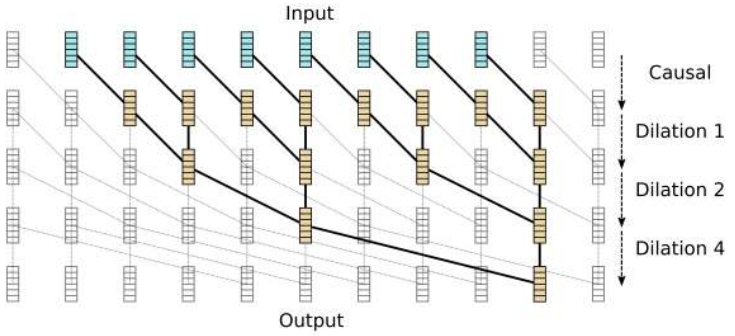


Fig. 1: Visualization of the models two-dimensional dilated causal layers and the first causal layer.

The input variables are quantized to n bins, continuous and discrete variables alike, matching the practice used in WaveNet [15] as well as PixelRNN [16]. As the length of the input data varies with each example, a special end of sequence value is used to represent sequence termination. The network utilizes a discretized mixture of logistic distributions, as described in PixelCNN++ [21] and Parallel WaveNet [17]. We found this to perform slightly better when compared to a more classical soft-max layer.

The individual residual layers follow closely the structure present in WaveNet. Unlike the WaveNet architecture, we included a dropout layer before each dilated convolution layer as shown in Figure 2. Applying dropout inside each residual layer has been previously explored in PixelCNN++ [21] and Wide Residual Networks [30].

To distinguish anomalous data from benign data, an anomaly score is quantified from the network outputs with a single forward pass, effectively avoiding the downside of slow sampling of the WaveNet model. In our approach, we computed the training loss contributions for each sample in the input sequence. The overall anomaly score of the whole sequence was the mean of these loss values, with samples past the end of sequence marker masked out to account for different length of sequences.

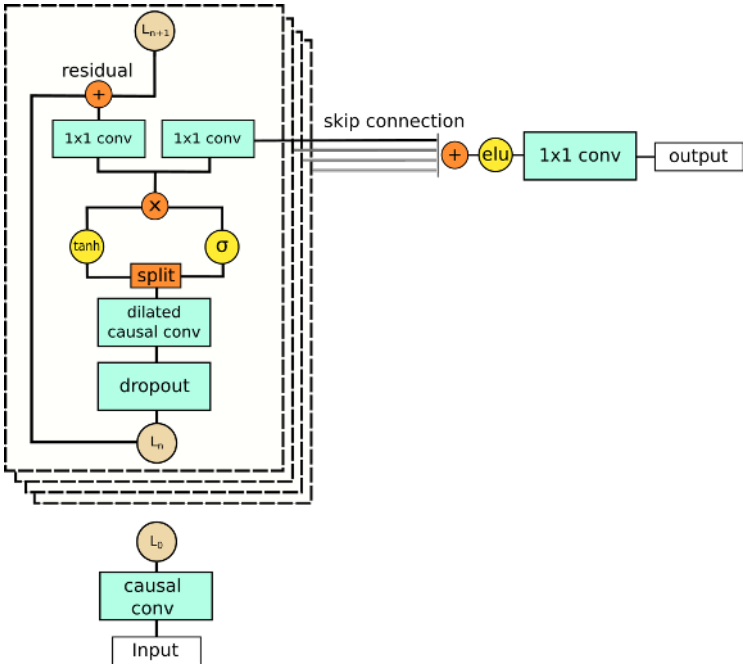


Fig. 2: The architecture is similar to the original WaveNet [15], with the exception of a dropout layer between all dilation layers and exclusive weights between residual and skip connections.

3 Experimental Results

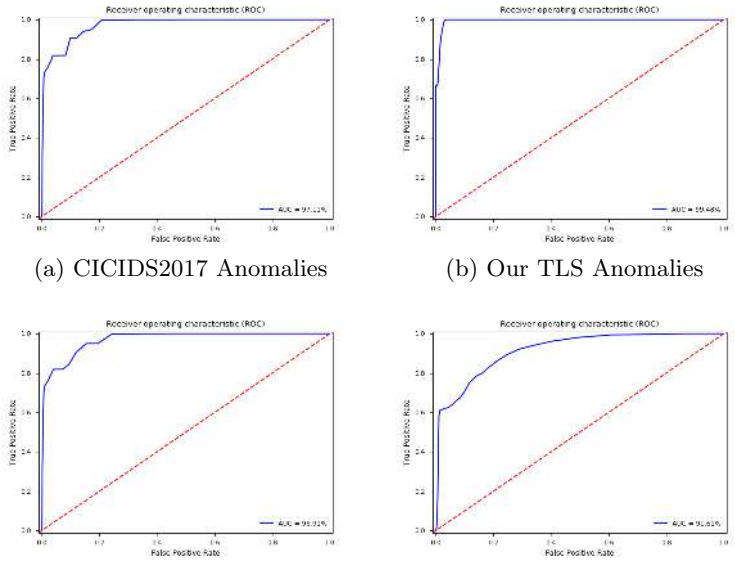
For the numerical results, we created receiver operating characteristic (ROC) curves by plotting the true positive rate (TPR) to y-axis and false positive rate (FPR) to x-axis. As a comparable score we also calculated the area under curve (AUC) from the ROC.

Training Dataset	Evaluation Dataset	AUC
CICIDS2017	CICIDS2017	97.11%
CICIDS2017	Our TLS anomalies	99.48%
CICIDS2017	CICIDS2017 + Our TLS anomalies	96.81%
FNCSE2018	FNCSE2018	91.61%

Table 1: Area under curve scores for four different evaluation dataset combinations.

In order to model an anomaly detector we split the clean data from CICIDS2017 and FNCSE2018 datasets into training and evaluation parts using

80/20 ratio. We took 256 first packets from each flow and trained a model with 9 dilation layers (receptive field of 256), vertical filter size of 3 and horizontal 2, 128 filters each layer for $\sim$ 15 epochs while evaluating the model using the evaluation part of the dataset to keep the model from over-fitting. During and after the training we ran an evaluation where we included the anomaly data to validate the anomaly detection capability of the model. Since the CICIDS2017 dataset lacks TLS anomalies we ran the evaluation three times to validate the model against the included CICIDS2017 anomalies, our TLS anomalies and a mixture of both. The resulting AUC scores are listed in Table 1. The FNCSE2018 training and evaluation datasets include only TLS encrypted connections.



(a) CICIDS2017 Anomalies (b) Our TLS Anomalies

(c) CICIDS2017 and TLS Anomalies (d) FNCSE2018

Fig. 3: Receiver operating characteristic curves on the four datasets we used to evaluate the model.

From the results in Figure 3 we concluded that the model is capable of detecting anomalies in both datasets, while also retaining the capability of detecting anomalous connection with TLS encryption. The model also performs significantly better than our earlier model [19], which had 80% AUC whereas the new model got 91.61% AUC on the same dataset.

4 Conclusion

In this study we applied the WaveNet and PixelCNN models for constructing an IDS based on anomaly detection. For the feature extraction and data processing, an open source software -based data pipeline was constructed. We utilized network data from Finland's National Cyber Security Exercise as well as public reference dataset CICIDS2017. The combined dataset was relatively extensive, although further efforts should be made to include a more diverse selection of applications and web browsing activities.

Results suggest that the machine learning model is suitable for detecting malicious command and control channels from TLS encrypted connections. The model is able to circumvent issues arising from samples of various lengths, and quantize timing and packet size differences into ranges suitable for neural networks.

Future work includes a conditioned WaveNet, variational or adversarial encoder to self-condition the WaveNet, and further testing on possible anomaly scores. Furthermore, visualization methods of found network anomalies should be studied for achieving better situational awareness in operative environments.

Acknowledgment

This research project is funded by MATINE - The Scientific Advisory Board for Defence.

References

1. Bitton, R., Shabtai, A.: A Machine Learning-Based Intrusion Detection System for Securing Remote Desktop Connections to Electronic Flight Bag Servers. IEEE Transactions on Dependable and Secure Computing pp. 1–1 (2019). https://doi.org/10.1109/TDSC.2019.2914035
2. Chen, Z., Yeo, C.K., Lee, B.S., Lau, C.T.: Autoencoder-based network anomaly detection. In: 2018 Wireless Telecommunications Symposium (WTS). pp. 1–5 (April 2018). https://doi.org/10.1109/WTS.2018.8363930
3. Chiba, Z., Abghour, N., Moussaid, K., Omri, A.E., Rida, M.: A Clever Approach to Develop an Efficient Deep Neural Network Based IDS for Cloud Environments Using a Self-Adaptive Genetic Algorithm. In: 2019 International Conference on Advanced Communication Technologies and Networking (CommNet). pp. 1–9 (April 2019). https://doi.org/10.1109/COMMNET.2019.8742390
4. Creech, G., Hu, J.: Generation of a new IDS test dataset: Time to retire the KDD collection. In: IEEE Wireless Communications and Networking Conference, WCNC. pp. 4487–4492. IEEE (apr 2013). https://doi.org/10.1109/WCNC.2013.6555301
5. JAMK University of Applied Sciences, Institute of Information Technology, JYVSECTEC: Rgce cyber range. http://www.jyvsectec.fi/en/rgce/, accessed: 26 April 2019

6. Li, Z., Rios, A.L.G., Xu, G., Trajković, L.: Machine Learning Techniques for Classifying Network Anomalies and Intrusions. In: 2019 IEEE International Symposium on Circuits and Systems (ISCAS). pp. 1–5 (May 2019). https://doi.org/10.1109/ISCAS.2019.8702583
7. Lincoln Laboratory, Massachusetts Institute of Technology: 1998 DARPA Intrusion Detection Evaluation Dataset. https://www.ll.mit.edu/r-d/datasets/1998-darpa-intrusion-detection-evaluation-dataset, accessed: 29 April 2019
8. Lincoln Laboratory, Massachusetts Institute of Technology: 1999 DARPA Intrusion Detection Evaluation Dataset. https://www.ll.mit.edu/r-d/datasets/1999-darpa-intrusion-detection-evaluation-dataset, accessed: 29 April 2019
9. Lincoln Laboratory, Massachusetts Institute of Technology: 2000 DARPA Intrusion Detection Scenario Specific Datasets. https://www.ll.mit.edu/r-d/datasets/2000-darpa-intrusion-detection-scenario-specific-datasets, accessed: 29 April 2019
10. Makhzani, A., Shlens, J., Jaitly, N., Goodfellow, I.: Adversarial Autoencoders. In: International Conference on Learning Representations (2016), http://arxiv.org/abs/1511.05644
11. McHugh, J.: Testing Intrusion Detection Systems: A Critique of the 1998 and 1999 DARPA Intrusion Detection System Evaluations As Performed by Lincoln Laboratory. ACM Trans. Inf. Syst. Secur. **3**(4), 262–294 (Nov 2000). https://doi.org/10.1145/382912.382923, http://doi.acm.org/10.1145/382912.382923
12. Ministry of Defence Finland: The national cyber security exercises is organised in Jyväskylä - Kansallinen kyberturvallisuusharjoitus kyha18 järjestetään Jyväskylässä, official bulletin 11th of may 2018. https://valtioneuvosto.fi/artikkeli/-/asset_publisher/kansallinen-kyberturvallisuusharjoitus-kyha18-jarjestetaan-jyvaskylassa (May 2018), accessed: 26 April 2019
13. Narsingyani, D., Kale, O.: Optimizing false positive in anomaly based intrusion detection using Genetic algorithm. In: 2015 IEEE 3rd International Conference on MOOCs, Innovation and Technology in Education (MITE). pp. 72–77 (Oct 2015). https://doi.org/10.1109/MITE.2015.7375291
14. Nevavuori, P., Kokkonen, T.: Requirements for Training and Evaluation Dataset of Network and Host Intrusion Detection System. In: Rocha, Á., Adeli, H., Reis, L.P., Costanzo, S. (eds.) New Knowledge in Information Systems and Technologies. pp. 534–546. Springer International Publishing, Cham (2019)
15. van den Oord, A., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A.W., Kavukcuoglu, K.: WaveNet: A Generative Model for Raw Audio (2016), https://arxiv.org/pdf/1609.03499.pdf
16. van den Oord, A., Kalchbrenner, N., Kavukcuoglu, K.: Pixel Recurrent Neural Networks. In: Balcan, M.F., Weinberger, K.Q. (eds.) Proceedings of The 33rd International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 48, pp. 1747–1756. PMLR, New York, New York, USA (20–22 Jun 2016), http://proceedings.mlr.press/v48/oord16.html
17. van den Oord, A., Li, Y., Babuschkin, I., Simonyan, K., Vinyals, O., Kavukcuoglu, K., van den Driessche, G., Lockhart, E., Cobo, L.C., Stimberg, F., Casagrande, N., Grewe, D., Noury, S., Dieleman, S., Elsen, E., Kalchbrenner, N., Zen, H., Graves, A., King, H., Walters, T., Belov, D., Hassabis, D.: Parallel WaveNet: Fast High-Fidelity Speech Synthesis. CoRR **abs/1711.10433** (2017), http://arxiv.org/abs/1711.10433
18. Open Information Security Foundation (OISF): Suricata Open Source IDS / IPS / NSM engine. https://suricata-ids.org/, accessed: 7 May 2019

19. Puuska, S., Kokkonen, T., Alatalo, J., Heilimo, E.: Anomaly-Based Network Intrusion Detection Using Wavelets and Adversarial Autoencoders. In: Lanet, J.L., Toma, C. (eds.) Innovative Security Solutions for Information Technology and Communications. pp. 234–246. Springer International Publishing, Cham (2019)
20. Ring, M., Wunderlich, S., Scheuring, D., Landes, D., Hotho, A.: A Survey of Network-based Intrusion Detection Data Sets. Computers & Security **86**, 147 – 167 (2019). https://doi.org/10.1016/j.cose.2019.06.005
21. Salimans, T., Karpathy, A., Chen, X., Kingma, D.P.: PixelCNN++: Improving the PixelCNN with Discretized Logistic Mixture Likelihood and Other Modifications. In: 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017 (2017), https://openreview.net/references/pdf?id=rJuJ1cP_l
22. Shiravi, A., Shiravi, H., Tavallaee, M., Ghorbani, A.A.: Toward developing a systematic approach to generate benchmark datasets for intrusion detection. Computers and Security **31**(3), 357–374 (may 2012). https://doi.org/10.1016/j.cose.2011.12.012
23. Siddiqui, M.A., Stokes, J.W., Seifert, C., Argyle, E., McCann, R., Neil, J., Carroll, J.: Detecting Cyber Attacks Using Anomaly Detection with Explanations and Expert Feedback. In: ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 2872–2876 (May 2019). https://doi.org/10.1109/ICASSP.2019.8683212
24. Tavallaee, M., Bagheri, E., Lu, W., Ghorbani, A.A.: A Detailed Analysis of the KDD CUP 99 Data Set. In: Proceedings of the Second IEEE International Conference on Computational Intelligence for Security and Defense Applications. pp. 53–58. CISDA'09, IEEE Press, Piscataway, NJ, USA (2009), http://dl.acm.org/citation.cfm?id=1736481.1736489
25. The University of California Irvine (UCI): KDD Cup 1999 Data. http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html, accessed: 29 April 2019
26. Umer, M.F., Sher, M., Bi, Y.: Flow-based intrusion detection: Techniques and challenges. Computers and Security **70**, 238–254 (2017). https://doi.org/10.1016/j.cose.2017.05.009
27. University of New Brunswick, Canadian Institute for Cybersecurity: Intrusion Detection Evaluation Dataset (CICIDS2017). https://www.unb.ca/cic/datasets/ids-2017.html, accessed: 30 April 2019
28. Wiewel, F., Yang, B.: Continual Learning for Anomaly Detection with Variational Autoencoder. In: ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 3837–3841 (May 2019). https://doi.org/10.1109/ICASSP.2019.8682702
29. Yu, F., Koltun, V.: Multi-Scale Context Aggregation by Dilated Convolutions. CoRR **abs/1511.07122** (2016), https://arxiv.org/pdf/1511.07122.pdf
30. Zagoruyko, S., Komodakis, N.: Wide Residual Networks. In: Richard C. Wilson, E.R.H., Smith, W.A.P. (eds.) Proceedings of the British Machine Vision Conference (BMVC). pp. 87.1–87.12. BMVA Press (September 2016). https://doi.org/10.5244/C.30.87