Journal of Universal Computer Science, vol. 26, no. 11 (2020), 1422-1434
submitted: 28/8/20, accepted: 2/10/20, appeared: 28/11/20

Detection of Cyberattacks Traces in IoT Data

Vibekananda Dutta*
(UTP University of Science and Technology, Bydgoszcz, Poland
dutta.vibekananda@utp.edu.pl)

Michał Choraś
(UTP University of Science and Technology, Bydgoszcz, Poland
chorasm@utp.edu.pl)

Marek Pawlicki
(UTP University of Science and Technology, Bydgoszcz, Poland
marek.pawlicki@utp.edu.pl)

Rafał Kozik
(UTP University of Science and Technology, Bydgoszcz, Poland
rkozik@utp.edu.pl)

Abstract: Artificial Intelligence plays a significant role in building effective cybersecurity tools. Security has a crucial role in the modern digital world and has become an essential area of research. Network Intrusion Detection Systems (NIDS) are among the first security systems that encounter network attacks and facilitate attack detection to protect a network. Contemporary machine learning approaches, like novel neural network architectures, are succeeding in network intrusion detection. This paper tests modern machine learning approaches on a novel cybersecurity benchmark IoT dataset. Among other algorithms, Deep AutoEncoder (DAE) and modified Long Short Term Memory (mLSTM) are employed to detect network anomalies in the IoT-23 dataset. The DAE is employed for dimensionality reduction and a host of ML methods, including Deep Neural Networks and Long Short-Term Memory to classify the outputs of into normal/malicious. The applied method is validated on the IoT-23 dataset. Furthermore, the results of the analysis in terms of evaluation matrices are discussed.

Key Words: Cybersecurity, Anomaly detection, Neural Networks, Dimensionality reduction, Deep learning.

Category: I.2.0, I.2.1

1 Introduction

The digital world excerpts a massive influence on modern life; never before has this fact been this clear. The recent events connected to the global pandemic emphasised the role of the cyber world in contemporary society. The domain of cybersecurity is rising in importance year after year [Bobowska et al., 2018], [Bieniasz et al., 2019].

For years now, and even more so with the recent events in the picture, cybersecurity has been a significant field of research.

The approaches of the cybersecurity domain offer a degree of protection against contemporary threats. Network Intrusion Detection Systems (NIDS) are a group of defense mechanisms that make substantial contributions in ensuring the protection of assets connected to a network [Zong et al., 2020]. The tools augmented by machine learning have been gaining traction for many years now. In cybersecurity, the premise of automating the effective detection of network traffic abnormalities causes research to gravitate to the use of those methods.

The fast-paced evolution of the leading-edge technologies, such as cloud computing or the Internet of Things (IoT) [Dhanabal and Shantharajah, 2015], spawns novel hazards. A multitude of research works have been conducted in the domain of intelligent IDS for different kinds of applications [da Costa et al., 2019].

At present, in concurrence with the newest trends of ML research in other fields, the state-of-the-art advancements in neural network technology are applied in network intrusion detection, for their potential to construct accurate models from difficult data.

In network intrusion detection, the volume, velocity and variety of data in modern networks (also known as the V's of Big Data) are all challenges that need to be handled. This predicament spurred a number of solutions - some of the contemporary big data handlers utilise Apache Kafka. We have investigated the use of Kafka to perform effective intrusion detection for streaming data in [Kozik and Choraś, 2017] and [Komisarek et al., 2020]. The developed solution is a scalable data processing framework which is well suited for processing Big Data workloads. The key element of this framework is the capability to integrate multiple machine learning models [Kozik and Choraś, 2017]. The framework has so far been equipped with a range of state-of-the-art IDS mechanisms. In this paper, an approach featuring several deep learning algorithms is tested on a brand new IoT benchmark dataset with the intention of augmenting the developed solution with a stronger detection method.

1.1 Contributions and structure

In general, ML algorithms are capable of inferring the non-linear relationships in large sets of observations [Arel et al., 2009]. This prompts cybersecurity research to apply various deep learning methods to evaluate if the obtained detection results can be further improved [Roy et al., 2017]. Therefore, this work attempts to facilitate an efficient deep learning mechanism to detect network anomalies and an attentive study of a recently published reference-point dataset *IoT-23*. The reported results are promising and this research will be further continued within the H2020 InfraStress project [1].

[1] This research work is a continuation of our previously published paper [Dutta et al., 2020a] within the framework of the H2020 InfraStress project

The effort described in the following pages attempts to fulfil the objective of offering a capable anomaly detection approach, following the characterisation disclosed below:

- The work proposes leveraging deep learning algorithms: DAE, LSTM and Multi Layer Perceptron (MLP/DNN) to detect network anomalies;
- The authors implement a dimensionality reduction approach in the form of a Deep AutoEncoder (DAE) in order to decrease the number of features in the vector used to train the classifier, as it was formulated in [Dutta et al., 2020b];
- LSTM cells are utilised to enhance classification effectiveness following the premise of distinguishing time-related relationship;
- The approach is tested on the recently released and publicly available IoT-23 [Agustin et al., 2020] dataset which incorporates lifelike, modern traffic data from IoT devices;
- Finally, the described method is thoroughly evaluated via a series of experiments, noting the improvements in a range performance metrics.

The remaining part of the paper discusses these contributions in-depth. In section 2, the recent works in the domain have been discussed. Section 3 offers a summary of the suggested approaches. Section 4 delineates the details of the setting of the study and the obtained results. Eventually, section 5 ends with conclusions, and the possibilities of future research.

2 Literature Review

'Conventional' machine learning (sometimes referred to as 'shallow' models) has been investigated at length in the literature. NIDS commonly incorporate Random Forest (RF), Support Vector Machines (SVM), K-Nearest Neighbour (KNN), Naïve Bayes (NB), logistic regression (LR), Decision Trees (DT), and clustering methods [Liu and Lang, 2019].

Ektefa et al. [Ektefa et al., 2010] contemplate network intrusion detection using several machine learning techniques and stipulate that the conducted experiments clearly display that the proposed classifier, based on Decission Trees, outperformed the SVM.

In [Goeschel, 2016], Goeschel et al. suggested an aggregated approach that incorporates SVM, DT, and NB algorithms, respectively. Initially, they trained an SVM to classify the data into in a binary fashion. Thereafter, for the malicious datapoints, the authors employed another approach (i.e., decision tree) to investigate the distinct malicious labels. The applied approach can distinguish only the categories that were included in the training set. Therefore, the authors decided to apply a Naïve Bayes (NB) classifier to identify unrecognised malicious samples. Hence, applying the strengths of the chosen classifiers, the offered aggregated approach reaches the accuracy of 99.22% on the KDD99 dataset. Similarly, Panda et al. [Panda et al., 2012] ponder a double-level network intrusion detection algorithm. In the preliminary stage, the authors employed a balanced nested dichotomy. This is followed by a random forest classifier. The following setup achieved a superior classification rate, lowering the false detection rate at the same time. The method uses principal components to lower the number of dimensions.

Deep learning algorithms leverage non-linear modelling in network traffic data through the use of multiple hidden layers [Gao et al., 2020]. The process of proper feature engineering can become problematic for detection mechanisms in NIDS. To alleviate this problem, some contemporary neural network algorithms offer a direct process for raw features, allowing them to fit to the unprocessed data and then perform classification [Liu and Lang, 2019].

B. Abolhasanzadeh [Abolhasanzadeh, 2015] uses a deep autoencoder for dimensionality reduction as part of intrusion detection on the NSL-KDD dataset. Potluri et al. [Potluri et al., 2018] evaluates a convolutional neural network-based network intrusion detection mechanism. The research employed the NSL-KDD and the UNSW-NB15 benchmark datasets. Then, as part of the process, the datasets are converted into an image-like format. The authors build a three-layer convolutional neural network (CNN) to label the attacks. The work is compared the state-of-the-art networks (e.g., ResNet50 and GoogLeNet), and the designed CNN method offered the finest outcomes, with accuracy in the range of 94.9% on the UNSW-NB15 and 91.14% on the NSL-KDD, respectively. In [Al-Qatf et al., 2018], Al-Qatf et al. merge a sparse autoencoder with a SVM supported by self-taught learning (STL) framework to detect network intrusions.

Torres et al. [Torres et al., 2016] utilises recurrent neural networks to classify the network traffic after having converted the feature vector from the dataset into characters. In their work, K. Jihyun et al. [Kim et al., 2016] employed LSTM with the KDD Cup'99 dataset. The proposed LSTM-RNN got up to 96.93% accuracy with 98.88% recall [Hodo et al., 2017].

The reviewed literature suggests that flow-based methods are finding their way to network intrusion detection. The lack of novel and up-to-date cybersecurity datasets is remains a problem, just as finding the most appropriate collection of features.

As machine learning shows a lot of promise, it has been carefully studied, taking into consideration dozens of various possible applications, and the outcomes of the experiments proved to be fairly impressive [D'Angelo et al., 2019] [D'Angelo et al., 2020].

For instance, Kim et al. [Jin Kim et al., 2017] have constructed a Deep Neural Network (DNN), containing 4 computational layers. The network was trained on a subset of the KDD99 benchmark dataset, using the ADAM (adaptive moment estimation) optimisation algorithm, and produced very good results.

In Yan and Han's [Yan and Han, 2018] work, a Stacked Sparse AutoEncoder (SSAE) was applied for detecting attacks that the NSL-KDD dataset contains. After having reorganized the original data into a number of subsets and conducting the experiment on them, the accuracy of the system reached 98.63%.

In the paper by Dutta et al. [Dutta et al., 2020b], a hybrid anomaly detection system was presented, where a Deep Neural Network (DNN) incorporated a Deep AutoEncoder (DAE) and was trained on the UNSW-NB15 dataset, achieving 91.29% accuracy. Obtaining this score was the result of the improved identification of the bahaviour type, achieved by making the generalization capabilities more efficient.

Yan et al. [Yan and Han, 2018] have scrutinised a set of deep learning methods, namely a Multi-Layer Perceptron (MLP), Restricted Boltzmann Machine (RBM), Sparse AutoEncoder (SAE), and a MLP with feature embeddings. The four models have been trained on two datasets containing intrusion data, i.e., NSL-KDD and UNSWNB15. The study was limited, as it did not take into consideration all the widely used models (e.g., recurrent neural networks) and the authors did not use the most recent datasets containing intrusion data. The provided results encompass accuracy, precision and recall.

In their paper, Ferrag et al. [Ferrag et al., 2020] have investigated the way seven contemporary artificial neural network methods perform when trained on the BoT-IoT and the CICIDS-2018 datasets and have provided the details on overall accuracy, training time and per-class detection rate.

Dutta et. al [Dutta et al., 2020a] presented an ensemble method which utilizes deep models, such as a DNN and LSTM, with a meta-classifier (logistic regression). The authors followed the stacked generalization principle. In order to increase the effectiveness of the presented solution, when approaching network anomalies, they apply a two-step process. Data pre-processing constitutes the first phase, in which a Deep Sparse AutoEncoder (DSAE) is in the preprocessing stage. The other stage, consisting in classification, utilizes a stacking ensemble learning approach. Then, the presented method is examined using the IoT-23, LITNET-2020 and NetML-2020 datasets; they are up-to-date and heterogeneous, they also contain data sourced from IoT environments.

In their work, Lopez-Martin et al. [Lopez-Martin et al., 2017] characterise a number of network classifiers in the context of various combinations of convolutional (CNN) and recurrent neural networks (RNN). Firstly, the authors scrutinize the coupling of the convolutional and fully-connected neural networks (CNN-NN) and the recurrent and fully connected neural networks (LSTM-NN).

Afterwards, the configuration of CNN-LSTM-NN is analyzed and, when tested on the RedIRIS dataset, it proves to have the highest accuracy, i.e. 96%.

In the paper by D'Angelo et al. [D'Angelo and Palmieri, 2021], an innovative architecture of an autoencoder-based deep neural network was discussed. In it, numerous autoencoders have been embedded with CNNs and RNNs, in order to gain the information on the relations between the basic features (spatial) and the way they evolve over time (temporal features). The authors have scrutinised both the theoretical background of the combinations and their actual performance, during the phase of testing them based on datasets containing real traffic. Having stacked the autoencoder with a fully connected neural network, the authors made the traffic classifier's average accuracy increase by 28%, when compared with the recent ML approaches. At the same time, the proposed solution's accuracy was 10% better than that of the pure convolutional and recurrent stacked neural networks, and 18% higher than that of the pure feedforward networks.

Finally, the survey paper by Berman et al. [Berman et al., 2019] gives a detailed overview of the deep learning methods that may prove useful as cybersecurity solutions, e.g., for detecting intrusions. The authors indicate that as far as the cybersecurity domain is concerned, it is certainly worth applying such tools as DNNs (e.g., multilayer perceptron, RNNs, autoencoders and restricted Boltzmann machines.

3 Strategies Used

This section provides an overview of the applied mechanisms for detecting network anomalies. The conceptual overview of the proposed solution is illustrated in Figure 1.

The algorithm consists of these steps: 1) dataset selection, 2) feature engineering; the step incorporates data pre-processing and dimensionality reduction, followed by, 3) a classifier in which the LSTM cells are feeding the final cell state to a fully connected dense layer (the approach we called mLSTM) and 4) output of the classification. Those steps will be elaborated upon in the coming paragraphs.

The lack of balance of classes in the dataset may cause trouble for ML algorithms [Ksieniewicz and Woźniak, 2018]. To compensate for the class imbalance, a data balancing procedure on the IoT-23 dataset was implemented. In this work, the authors employed a Synthetic Minority Over-sampling Technique (SMOTE) in advance of the Edited Nearest Neighbors (ENN) method [Rendón et al., 2020] (Tab. 2).

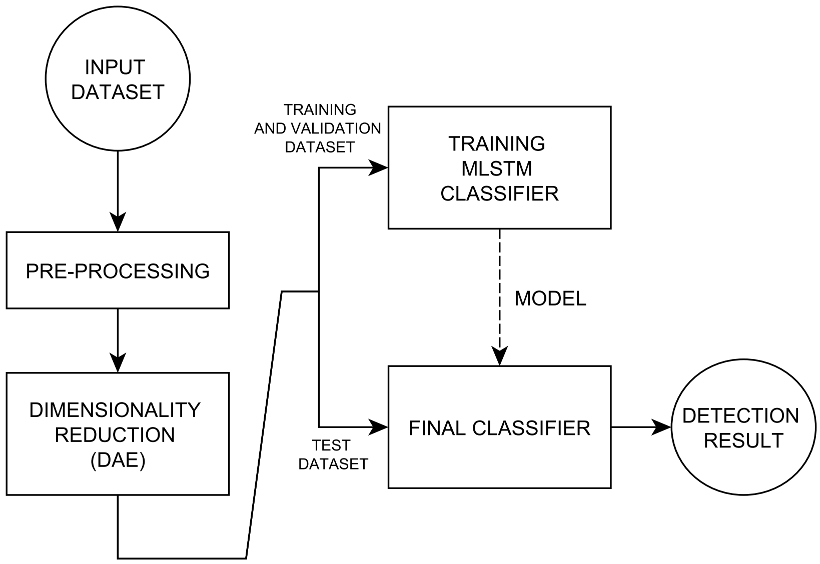

Figure 1: Overview of the concept.

3.1 Feature engineering

3.1.1 Dimensionality reduction

Removing the meaningless and inconsequential information is the foremost step to produce stronger inputs to the classifier, since the effectiveness of ML classifiers is in direct correlation to the grade of the selected characteristics. Principal Component Analysis (PCA), and AutoEncoder (AE) are efficient methods for dimensionality reduction. However, an autoencoder offers the ability to acquire non-linear relationships, a feat the PCA does not posses [Topolski, 2020].

Following [Dutta et al., 2020b], we have tested a Deep AutoEncoder (DAE) in conjunction with a multi-layer neural network to decrease the number of features. The autoencoder has the capability to formulate a latent representation the inputs [Zhang et al., 2018].

The DAE determines its outputs $\hat{x}_i \in \Re^n$ from the input vector x_i.

The learning algorithm attempts to update the weights W, and biases b to reduce the value of the cost function.

$$h_i = f_\theta(x_i) \tag{1}$$

The sub-network decoder aims at reconstructing the encoding back to its initial delineation,

$$\hat{x}_i = g_\theta(h_i) \tag{2}$$

f_θ and g_θ are encoding and decoding parameters. The sets of parameters for the f_θ and g_θ are learned simultaneously by diminishing the *loss* during the reconstruction task.

The work contained herein uses an auto-encoder of three hidden-layers. The layers utilise the *sigmoid* activation function. The input is built of n neurons, depending on the selected dataset after the initial pre-processing phase.

Once the dimensionality reduction process (i.e., autoencoder learning) has ended, the outcomes of the trained autoencoder are forwarded to the classifier. PCA orders the data samples on a plane defined by the pre-defined number of principal components (PCs) that convey a certain percentage of variance. In contradiction to PCA, the autoencoder has all the information from the original data compressed in to the reduced layer without mislaying the structure of the data samples.

3.2 Building the model

The LSTM is a type of RNN equipped with a gating mechanism, which allows the network to infer relationships over longer sequences [Zhao et al., 2017]. The tested mechanism uses the final state of the LSTM cells as input to a fully connected dense layer.

The authors provide the results of two baseline neural networks - a DNN and an LSTM on the dataset to provide a reference point.

The characteristics of the networks are disclosed below:

– $DNN_{3-layer}$ - hidden layers nodes (20, 16, 12); chosen optimizer: Adam; activation function: relu, Sigmoid; applied batch size and epochs: 512, 500; loss function: binary cross-entropy.

– $LSTM_{3-layer}$ - hidden layers nodes (20, 16, 12); chosen optimizer: Adam; activation function: tanh, Sigmoid; applied batch size and epochs: 512, 500; loss function: binary cross-entropy.

– $mLSTM$ (proposed) - hidden layers nodes (20, 16, 12, 8); chosen optimizer: Adam; activation function: relu, tanh, Sigmoid; applied batch size and epochs: 512, 500; loss function: binary cross-entropy.

4 Experiments and Results

This section delineates the selected dataset and experiment results of the proposed mechanism built on the mLSTM classifier to detect network anomalies. For evaluation of the tested methods, the following metrics have been used: accuracy score, precision, recall, MCC and geometric mean (g-mean). It is worth mentioning that this work considers binary classification: normal and malware

(0, 1) for both baseline algorithms and proposed mLSTM framework, respectively.

4.1 Dataset description

IoT-23 [Agustin et al., 2020] is a newly established dataset which contains 20 different malicious software classes, and three separate captures of background traffic. This dataset was released in January 2020. The set incorporates network features categorized into (a) flow features, (b) basic features, (c) time features, and (d) content features. The dataset was created to provide real-life, labelled traffic captures for ML research.

4.2 Results

The following paragraphs disclose the evaluation of the results obtained by the proposed framework $mLSTM$ classifier against state-of-the-art : RF, SVM, MLP [Barut et al., 2020] and baseline classifiers: $DNN_{3-layer}$, $LSTM_{3-layer}$, respectively.

The authors employ the IoT-23 benchmark, since it is an up-to-date and relevant dataset, to assess the efficiency of the classifiers. Each algorithm has undergone the identical training using the same training set. The results of the experiments are discolsed in Table 1.

Table 1: Results of evaluated classifier performance reported for each fold of the 5-fold CV (the finest achievements are emphasised in bold)

Fold	$DNN_{3-layer}$	$LSTM_{3-layer}$	$mLSTM$ (proposed)
f1	**99.97%**	99.95%	**99.97%**
f2	99.92%	99.94%	**99.98%**
f3	98.65%	99.39%	**99.98%**
f4	99.97%	**99.98%**	**99.975%**
f5	96.93%	99.96%	**99.99%**
Avg Acc	99.688%	99.844%	**99.98%**
Std Dev	1.35	0.254	**0.006**
SEM	0.595	0.113	**0.002**

The effectiveness of the models w.r.t. the state-of-the-art classifiers is also disclosed using well-established evaluation metrics, such as:

$$Acc = \frac{tp + tn}{tp + tn + fp + fn} \tag{3}$$

$$MCC = \frac{tp * tn - fp * fn}{\sqrt{(tp + fp) * (tp + fn) * (tn + fp) * (tn + fn)}} \quad (4)$$

Matthews Correlation Coefficient (MCC) considers all four fields of the binary confusion matrix; a high value (close to 1) means that both classes are predicted well ($MCC \in [0, 1]$), even if one class is disproportionately under-(or over-) represented.

This work used the $g - mean$ definition contained in [Espíndola and Ebecken, 2005] to investigate the classifier performance for the datasets displaying class imbalance; the metric can be formulated as follows:

$$g - mean = \sqrt{Precision \times Recall} \quad (5)$$

Where $Precision(Pr) = \frac{tp}{tp+fp}$, and $Recall(Re) = \frac{tp}{tp+fn}$. Table 2 presents the overall accuracy, MCC and g-mean score respectively.

Table 2: Performance evaluation on IoT-23 testing set (bold font used for emphasis on the finest performance)

Method	Accuracy	MCC	g-mean
RF [Dutta et al., 2020b]	0.897	0.891	–
SVM [Barut et al., 2020]	0.871	0.864	–
MLP [Barut et al., 2020]	0.903	0.897	–

g-mean score obtained only if a data balancing approach (SMOTE+ENN) is employed

The showcased performance of the proposed method indicates improvement when compared to the baseline and the newest methods found in the literature in terms of overall classification accuracy (99.9%), MCC (99.2%), and g-mean score (97.1%). Following results prove experimentally that the proposed framework improves classification performance.

5 Concluding Remarks and Future Work

For large and heterogeneous datasets like the IoT-23, DAE proves useful in learning and provides feature representation suitable for ML algorithms. The conducted experiments indicated that the use of the DAE as a feature extractor in conjunction with deep learning in the form of the proposed mLSTM in the role of a classifier offers improvements when the performance on the IoT benchmark is measured.

The experimental results and statistical significance tests illustrate that the proposed method is an improvement over the individual baseline classifiers including Random Forest and Support Vector Machine on the used novel IoT benchmark, reaching the accuracy of 99.9% and g-mean score 97.1%, respectively.

Considering future work, our study will attempt to apply a lifelong learning approach to deep learning algorithms to facilitate better detection of novel attacks, and its is expected to be further extended to conduct experiments on modern and more sophisticated datasets. Finally, security of the deep learning model against adversarial attacks needs to be provided [Pawlicki et al., 2020].

Acknowledgement

This work is funded under InfraStress project, which has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 833088 .

References

[Abolhasanzadeh, 2015] Abolhasanzadeh, B. (2015). Nonlinear dimensionality reduction for intrusion detection using auto-encoder bottleneck features. In *2015 7th Conference on Information and Knowledge Technology (IKT)*, pages 1–5. IEEE.

[Agustin et al., 2020] Agustin, P., Sebastian, G., and Maria Jose, E. (2020 (accessed February 3, 2020)). Stratosphere laboratory. a labeled dataset with malicious and benign iot network traffic. `https://www.stratosphereips.org/datasets-iot23`.

[Al-Qatf et al., 2018] Al-Qatf, M., Lasheng, Y., Al-Habib, M., and Al-Sabahi, K. (2018). Deep learning approach combining sparse autoencoder with svm for network intrusion detection. *IEEE Access*, 6:52843–52856.

[Arel et al., 2009] Arel, I., Rose, D., and Coop, R. (2009). Destin: A scalable deep learning architecture with application to high-dimensional robust pattern recognition. In *2009 AAAI Fall Symposium Series*.

[Barut et al., 2020] Barut, O., Luo, Y., Zhang, T., Li, W., and Li, P. (2020). Netml: A challenge for network traffic analytics. *arXiv preprint arXiv:2004.13006*.

[Berman et al., 2019] Berman, D., Buczak, A., Chavis, J., and Corbett, C. (2019). A Survey of Deep Learning Methods for Cyber Security. *Information*, 10(4):122.

[Bieniasz et al., 2019] Bieniasz, J., Stepkowska, M., Janicki, A., and Szczypiorski, K. (2019). Mobile agents for detecting network attacks using timing covert channels. *J. UCS*, 25(9):1109–1130.

[Bobowska et al., 2018] Bobowska, B., Choras, M., and Wozniak, M. (2018). Advanced analysis of data streams for critical infrastructures protection and cybersecurity. *J. UCS*, 24(5):622–633.

[da Costa et al., 2019] da Costa, K. A., Papa, J. P., Lisboa, C. O., Munoz, R., and de Albuquerque, V. H. C. (2019). Internet of things: A survey on machine learning-based intrusion detection approaches. *Computer Networks*, 151:147–157.

[D'Angelo et al., 2020] D'Angelo, G., Ficco, M., and Palmieri, F. (2020). Malware detection in mobile environments based on Autoencoders and API-images. *Journal of Parallel and Distributed Computing*, 137:26–33.

[D'Angelo and Palmieri, 2021] D'Angelo, G. and Palmieri, F. (2021). Network traffic classification using deep convolutional recurrent autoencoder neural networks for spatial–temporal features extraction. *Journal of Network and Computer Applications*, 173:102890.

[D'Angelo et al., 2019] D'Angelo, G., Palmieri, F., and Rampone, S. (2019). Detecting unfair recommendations in trust-based pervasive environments. *Information Sciences*, 486:31–51.

[Dhanabal and Shantharajah, 2015] Dhanabal, L. and Shantharajah, S. (2015). A study on nsl-kdd dataset for intrusion detection system based on classification algorithms. *International Journal of Advanced Research in Computer and Communication Engineering*, 4(6):446–452.

[Dutta et al., 2020a] Dutta, V., Choraś, M., Pawlicki, M., and Kozik, R. (2020a). A Deep Learning Ensemble for Network Anomaly and Cyber-Attack Detection. *Sensors*, 20(16):4583.

[Dutta et al., 2020b] Dutta, V., Choraś, M., Pawlicki, M., and Kozik, R. (2020b). Hybrid model for improving the classification effectiveness of network intrusion detection. In *Conference on Complex, Intelligent, and Software Intensive Systems*. Springer.

[Ektefa et al., 2010] Ektefa, M., Memar, S., Sidi, F., and Affendey, L. S. (2010). Intrusion detection using data mining techniques. In *2010 International Conference on Information Retrieval & Knowledge Management (CAMP)*, pages 200–203. IEEE.

[Espíndola and Ebecken, 2005] Espíndola, R. and Ebecken, N. (2005). On extending f-measure and g-mean metrics to multi-class problems. *Sixth international conference on data mining, text mining and their business applications*, 35:25–34.

[Ferrag et al., 2020] Ferrag, M. A., Maglaras, L., Moschoyiannis, S., and Janicke, H. (2020). Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study. *Journal of Information Security and Applications*, 50:102419.

[Gao et al., 2020] Gao, M., Ma, L., Liu, H., Zhang, Z., Ning, Z., and Xu, J. (2020). Malicious network traffic detection based on deep neural networks and association analysis. *Sensors*, 20(5):1452.

[Goeschel, 2016] Goeschel, K. (2016). Reducing false positives in intrusion detection systems using data-mining techniques utilizing support vector machines, decision trees, and naive bayes for off-line analysis. In *SoutheastCon 2016*, pages 1–6. IEEE.

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

[Jin Kim et al., 2017] Jin Kim, Nara Shin, Jo, S. Y., and Sang Hyun Kim (2017). Method of intrusion detection using deep neural network. In *2017 IEEE International Conference on Big Data and Smart Computing (BigComp)*, pages 313–316. IEEE.

[Kim et al., 2016] Kim, J., Kim, J., Thu, H. L. T., and Kim, H. (2016). Long short term memory recurrent neural network classifier for intrusion detection. In *2016 International Conference on Platform Technology and Service (PlatCon)*, pages 1–5. IEEE.

[Komisarek et al., 2020] Komisarek, M., Choras, M., Kozik, R., and Pawlicki, M. (2020). Real-time stream processing tool for detecting suspicious network patterns using machine learning. *ARES conference*.

[Kozik and Choraś, 2017] Kozik, R. and Choraś, M. (2017). Pattern extraction algorithm for netflow-based botnet activities detection. *Security and Communication Networks*.

[Ksieniewicz and Woźniak, 2018] Ksieniewicz, P. and Woźniak, M. (2018). Imbalanced data classification based on feature selection techniques. In *International Conference on Intelligent Data Engineering and Automated Learning*, pages 296–303. Springer.

[Liu and Lang, 2019] Liu, H. and Lang, B. (2019). Machine learning and deep learning methods for intrusion detection systems: A survey. *Applied Sciences*, 9(20):4396.

[Lopez-Martin et al., 2017] Lopez-Martin, M., Carro, B., Sanchez-Esguevillas, A., and Lloret, J. (2017). Network Traffic Classifier With Convolutional and Recurrent Neural Networks for Internet of Things. *IEEE Access*, 5:18042–18050.

[Panda et al., 2012] Panda, M., Abraham, A., and Patra, M. R. (2012). A hybrid intelligent approach for network intrusion detection. *Procedia Engineering*, 30:1–9.

[Pawlicki et al., 2020] Pawlicki, M., Choraś, M., and Kozik, R. (2020). Defending network intrusion detection systems against adversarial evasion attacks. *Future Generation Computer Systems*.

[Potluri et al., 2018] Potluri, S., Ahmed, S., and Diedrich, C. (2018). Convolutional neural networks for multi-class intrusion detection system. In *International Conference on Mining Intelligence and Knowledge Exploration*, pages 225–238. Springer.

[Rendón et al., 2020] Rendón, E., Alejo, R., Castorena, C., Isidro-Ortega, F. J., and Granda-Gutiérrez, E. E. (2020). Data sampling methods to deal with the big data multi-class imbalance problem. *Applied Sciences*, 10(4):1276.

[Roy et al., 2017] Roy, S. S., Mallik, A., Gulati, R., Obaidat, M. S., and Krishna, P. V. (2017). A deep learning based artificial neural network approach for intrusion detection. In *International Conference on Mathematics and Computing*, pages 44–53. Springer.

[Topolski, 2020] Topolski, M. (2020). Application of the stochastic gradient method in the construction of the main components of pca in the task diagnosis of multiple sclerosis in children. In Krzhizhanovskaya, V. V., Závodszky, G., Lees, M. H., Dongarra, J. J., Sloot, P. M. A., Brissos, S., and Teixeira, J., editors, *Computational Science – ICCS 2020*, pages 35–44, Cham. Springer International Publishing.

[Torres et al., 2016] Torres, P., Catania, C., Garcia, S., and Garino, C. G. (2016). An analysis of recurrent neural networks for botnet detection behavior. In *2016 IEEE biennial congress of Argentina (ARGENCON)*, pages 1–6. IEEE.

[Yan and Han, 2018] Yan, B. and Han, G. (2018). Effective feature extraction via stacked sparse autoencoder to improve intrusion detection system. *IEEE Access*, 6:41238–41248.

[Zhang et al., 2018] Zhang, C., Cheng, X., Liu, J., He, J., and Liu, G. (2018). Deep sparse autoencoder for feature extraction and diagnosis of locomotive adhesion status. *Journal of Control Science and Engineering*, 2018.

[Zhao et al., 2017] Zhao, R., Yan, R., Wang, J., and Mao, K. (2017). Learning to monitor machine health with convolutional bi-directional lstm networks. *Sensors*, 17(2):273.

[Zong et al., 2020] Zong, W., Chow, Y.-W., and Susilo, W. (2020). Interactive three-dimensional visualization of network intrusion detection data for machine learning. *Future Generation Computer Systems*, 102:292–306.