Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 185 (2021) 239–247

Procedia Computer Science

www.elsevier.com/locate/procedia

Complex Adaptive Systems Conference Theme: Big Data, IoT, and AI for a Smarter Future
Malvern, Pennsylvania, June 16-18, 2021

Network Intrusion Detection System using Deep Learning

Lirim Ashiku[1] Cihan Dagli

Department of Engineering Management and Systems Engineering
Missouri University of Science and Technology, Rolla, MO 65401, USA

Abstract

The widespread use of interconnectivity and interoperability of computing systems have become an indispensable necessity to enhance our daily activities. Simultaneously, it opens a path to exploitable vulnerabilities that go well beyond human control capability. The vulnerabilities deem cyber-security mechanisms essential to assume communication exchange. Secure communication requires security measures to combat the threats and needs advancements to security measures that counter evolving security threats. This paper proposes the use of deep learning architectures to develop an adaptive and resilient network intrusion detection system (IDS) to detect and classify network attacks. The emphasis is how deep learning or deep neural networks (DNNs) can facilitate flexible IDS with learning capability to detect recognized and new or zero-day network behavioral features, consequently ejecting the systems intruder and reducing the risk of compromise. To demonstrate the model's effectiveness, we used the UNSW-NB15 dataset, reflecting real modern network communication behavior with synthetically generated attack activities.

Keywords: Cybersecurity; Zero-day attacks; Deep learning; Intrusion detection systems

[1]* Corresponding author. Tel.: +1-608-333-7520.
E-mail address: lahnr@mst.edu

10.1016/j.procs.2021.05.025

Nomenclature

1.	IDS	Intrusion Detection System
2.	ICT	Information and Communication Technologies
3.	NIDS	Network Intrusion Detection System
4.	HIDS	Host Intrusion Detection System
5.	CNN	Convolutional Neural Network
6.	RNN	Recurrent Neural Network
7.	LSTM	Long Short Term Memory
8.	GRU	Gated Recurrent Unit

1. Introduction

Advances and widespread use of interconnectivity and interoperability of information and communication technologies (ICT) have become necessary to reshape our relations to daily activities. The vibe of reliance on ICT has enhanced individuals and organizations' posture allowing real-time global business continuity that continuously evolves to offer convenience-related interoperability frontier solutions [1]. The exchange of digital information across networks has opened a path to exploitable vulnerabilities that may have detrimental effects on both individuals and organizations, thus deeming an effective network security solution crucial to maintaining confidentiality, integrity, and availability [2]. Among the layered defensive mechanisms that address different attack vectors, network security controls are recognized as the first defense line. An intrusion detection system (IDS) scans network traffic to identify and report a violation based on the preconfigured customized detection levels. Early detection will deter an intrusion and eject it from the system before any damage to the data. IDS assumes that intrusions behavioral features differ from legitimate users' behavior; therefore, IDS quantifies intrusion behavior in terms of its features. However, an exact distinction cannot be deciphered, creating an overlap between normal and abnormal behavior that can be more obvious by deploying an intelligent intrusion detection system [3].

The main types of intrusion detection systems include:

A network intrusion detection system (NIDS) may consist of both hardware (sensors) and software (console) to control and monitor network traffic packets at multiple locations for a potential intrusion or anomaly.

Host intrusion detection system (HIDS) resides on a particular computer or server, identified as the host, and monitors activity only on that system. Although tapered to only one system, it offers higher capabilities than NIDS, as it can access encrypted information traversing the network, including system configuration databases, registries, and file attributes.

A Cloud intrusion detection system is a combination of cloud, network, and host layers. The cloud layer provides a secure authentication into the demand-based access to a shared group or application programming interface (API). Similarly, it will create a bridge between existing IDS and hypervisors.

To examine network traffic flow, IDSs deploy various detection techniques. Most common detection methods include:

- *Signature-based detection*, also known as knowledge-based, examines network traffic to identify patterns that match existing or known signatures. This detection method looks at existing attacks associated with a distinct signature, yet it requires continuous updates to account for unknown attack patterns.
- *Anomaly-based detection,* also known as behavior-based detection, examines network traffic to identify patterns that deviate from normal or baseline behavior. This type of detection samples network traffic and deploys statistical methods to scrutinize deviations; once a threshold is exceeded, it will alert the administrator of an anomaly. Anomaly-based detection can detect new anomalies, but at the same time, require much more processing power to compare behavior patterns continuously, and even a minute change from baseline may trigger an alarm increasing false positives.
- *Stateful protocol analysis*, unlike signature-based, compares recognized protocol profiles to network traffic. It uses predetermined, vendor-provided profiles to identify a random sequence of commands in both the network and application layer.

Nevertheless, each detection method associates weaknesses when dealing with the overlap between normal and abnormal traffic patterns; some drawbacks lead to false positives, false negatives, slow networks, increased CPU

usage, etc. To overcome some of the detection method limitations, deployment of traditional machine learning techniques such as Naïve Bayes, Decision Trees, Support Vector Machines, etc., are sought. These methods have significantly contributed to improved detection accuracy yet require expert knowledge and involvement to process the extensive data. These techniques or shallow classifiers discover algorithms that permit machines to learn without human intervention eventually may deliver suboptimal performance for multiclass problems with an increased number of features [5]. Research has further advanced self-learning intrusion detection systems to detect and classify recognized and zero-day intrusions; such detection methods facilitate proactive measures to identify and deter malicious network traffic. Deep learning is characterized as the complex model, or advanced subset, of machine learning algorithms to conquer some of the shallow networks' limitations. Deep learning algorithms have demonstrated significance in speech recognition results, image processing, natural language processing, and many other domains [6].

This paper focuses on the effectiveness of deep learning architectures for network security solutions that scans network traffic to identify and report a violation based on the intrusions behavioral features identified in the dataset UNSW-NB15 that reflect real modern normal behavior with synthetically generated attack activities [7]. Section 2 introduces related literature for IDS models followed by the approach utilized in this problem. Nest, the profound learning model results are presented. And finally, section 5 summarizes the conclusions and future directions.

2. Background

Attributable to technology advancements for enhancing the digital society's lifestyle, intruders use the advances for exploitation purposes. Exploitations include attempting to remotely access a network and cause a compromise of integrity, confidentiality, or accessibility. The exploitations cause a detrimental effect to the affected and leverage for capital gain for the intruder. Network IDS aims to analyze network traffic to identify network traffic patterns that can cause damage to systems or networks. Although recognized practices are incorporated in the detection methods' configurations, filtering traffic that measures up to rules may cause a decline in communication speed, exploring options for deep learning approaches.

Deep neural networks consist of simultaneous multi-layer feature extraction that has gained popularity and have significantly impacted the world of science [8]. Existing work within the domain of network IDS offers an improved detection accuracy across a range of datasets. However, due to privacy and security issues, there is a lack of openly available intrusion detection labeled datasets. Such limitation influences researchers to simulate network traffic with real-life feature representativeness, and as a result, the collection of KDDCup, a DARPA project, was the pioneer in IDS datasets. Lincoln Lab-created KDDCup using tcpdump using closed network generated traffic with manual injected attacks to simulate traffic in U.S. Air Force bases. Failure to validate that the simulated dataset represented real network traffic and many redundant records, especially for attacks, led to the creation of the next version NSL-KDD [9]. Although with a significant redundancy improvement, the revised dataset is still being criticized for real traffic representation, yet continually referenced as the most used and useful benchmark dataset related to NIDS research. Other network IDS datasets include NSL-KDD, Kyoto, WSN-DS, CICIDS, etc. The most recent publicly available dataset, UNSW-NB15, is created by Moustafa and Slay [7, 10] at the University of New South Wales, Australia. The dataset was replicated to the KDDCup dataset using Association Rule Mining (ARM) approach in feature selection but with very few standard features making it difficult for dataset comparisons. UNSW-NB15 identifies nine families of attacks discussed in [7], which reflect common vulnerabilities and exposures (CVE); CVE is a dictionary of publicly disclosed types of attacks that evolves upon discovery of potential vulnerabilities [11]. Like earlier IDS datasets, UNSW-NB15 offers more than 40 features to represent real network traffic with nine modern attack types elaborated on [12] with over 2.5 million records. The architects provide abridged training and testing datasets commonly utilized for research. If not directly accessed through university websites, some sources have reversed the partitions of training and testing datasets and have used the smaller dataset for training, which resulted in reduced classification accuracy due to small instances on some of the attack types. The partitioned datasets have been amended and processed to data cleansing, editing, reduction, and wrangling techniques to accommodate use requiring only minimal deep learning pre-processing.

Research publications on IDS machine learning begin with the original DARPA datasets initiating machine learning research with shallow neural networks such as Decision Trees, Support Vector Machines, Random Forests, etc., yielding high accuracy but significantly failing on underrepresented attack types [13,14]. Like related IDS datasets, the UNSW-NB15 dataset offers binary classification demonstrating whether an instance is considered

benign (normal) or malignant (attack) and categorical classification where the type of attack is classified. Most researchers develop two models to account for both binary and categorical classifications. In addition to the said models, the KDDCup dataset offers sub-categorical classification (24 attack types); however, it is not commonly disclosed in research due to training dataset misrepresentation discrepancy of sub-categories, while testing dataset accounts for all classes. One way to handle such class imbalances is to merge the data and deploy sampling approaches that divide the data for training and testing purposes. A revised LaNet-5 model to classify network threats for the NSL-KDDD dataset is discussed by [15]. The image-reshaped NSL-KDD dataset's cross-validation scheme showed significant improvements for both consequential types of intrusion threats and attack sub-categories. Upwards dimension reshape is not a common practice, but in this case, it proved influential to the results observed with the originally shaped dataset [15]. Time-series modeling of the network traffic events for the one-dimensional data is discussed in [16]. Time-series modeling included a hybrid of both CNN and recurrent neural network approaches such as LSTM and GRU. Open-source data flow engine TensorFlow, with a single Nvidia Tesla k40 GPU, was used for the KDDCup dataset. Hyperparameter tuning and many network topologies with a sampling of underrepresented classes showed significant improvements for the given network dataset [16]. A deep neural network (DNN) was staged by [17], indicating that the ReLU activation function helps overcome the vanishing gradient and works faster than other nonlinear functions. Although many network traffic datasets are discussed in the literature, UNSW-NB15 is considered an inclusive dataset of lower-level network attacks using modern network traffic to meet real-world network activities. The binary classification for the UNSW-NB15 yielded 78.4% accuracy, whereas the multiclass classification model yielded 66% accuracy [17]. The binary classification was best with a single hidden layer, whereas the multiclass was similar for both 2 and 3 layers DNN. An increase in the number of layers caused performance degradation using the hyperparameters discussed in [17]. A deep learning model with ten hidden layers, each with ten neurons per layer, with ten-fold cross-validation is carried out under three experiments [21]. The initial investigation is searching for the best activation function to use for the ten-fold cross-validation.

In contrast, the second experiment is searching for principal features, and finally, the model is utilized and achieved significant accuracy improvements. However, the authors in [21] do not disclose whether they experimented on a binary or multiclass classification model, and neither did they show the proposed architecture. A deep random neural network for UNSW-NB15 presents the multiclass classification's best performance using five hidden layers [22]. Compared to other deep learning models with an 87%, 91%, and 95% respectively presented by [23-25], the proposed scheme of deep random neural network yielded a 99% accuracy with the lowest detection rate of 97% for the underrepresented classes.

3. Approach

The proposed deep learning model includes CNN, with regularized multi-layer perceptron, instead of a fully connected feed-forward neural network (FNN). As opposed to FNN, CNN uses convolution as a mathematical operation instead of multiplication or dot product. Convolution operation involves custom hyperparameters such as dimensions of filter, number of filters, and strides for generating the output matrix. To handle diminishing tensor dimensions as the input propagates through multiple convolutional layers, we introduced input padding. The pooling layer is used between successive convolutional layers to reduce the or downsample feature dimensions across the layers. Finally, a fully connected layer with regularization is remarked, followed by the classification output layer. The dataset used to experiment with our model, as denoted, will be UNSW-NB15 primarily selected for its representation of real-world network traffic designated to the common vulnerabilities and exposures. The dataset has been examined through numerous models yielding suboptimal results conceding to opportunities for model improvements. Although raw dataset reaches beyond two million instance simulations, the architects characterize imputed datasets for training and testing purposes with nine attack families. Table I depicts the types of attacks followed by a short description for each attack.

Table I. Attack categories and descriptions

Attack	Short Description
Normal	Benign network traffic
Fuzzers	Malignant traffic related to spams, penetrations or port scans
Analysis	Attack related to intercepting and or examining network traffic through penetrations or scans
Backdoors	Attack pertaining to use of mechanism designed to bypass security measures
DoS	Attack aimed at flooding network resources making it inaccessible
Exploits	Exploitations through security holes in O.S. or other software applications
Generic	Attacks related to block-cipher, brute force or cryptanalysis
Reconnaissance	The target system is observed for vulnerabilities
Shellcode	Short code 'payloads' to navigate through the system and gain control
Worms	Malicious code that replicates itself to spread through the network

The model uses Keras library as a prototype working on top of the TensorFlow framework. Similarly, the framework offers comprehensive and flexible tools and libraries that support deep learning architectures such as CNN and RNN while enabling seamless exhibition for CPU, GPU, and TPU usage [18]. Google Colab serves as a free cloud-based Jupyter notebook environment that supports training machine learning or deep learning models using their computing units. To increase optimization momentum, we trained our deep learning model on GPU enabled framework of Google Colab. The network IDS dataset required some pre-processing to convert objects into numerical vectors to serve as input into the network. Data encoding was used to convert object features into vectors and simultaneously created a distinct feature-category for missing data. The features were normalized and reshaped for CNN input, while the labels were one-hot encoded to represent the number of classes. The multiclass model characterizes ten classes, nine related to various attacks and then the normal traffic flow. Semi-dynamic hyperparameter optimization included regularization techniques, learning rate, optimization algorithm, and batch size. Similar to the hyperparameter optimization on the fly [26], we start midway from grid-search to establish a baseline. Then we advance into trajectories that generate candidate hyperparameters from the search space. The next dominant model replaces the baseline model and continues into the course until a decline in performance.
In addition to the hyperparameter tuning, we considered callback functions such as EarlyStopping and ModelCheckpoint. These functions expedite search space and eliminate process continuity when the model converges, hence preserving weights of best model performance. Fig. 1 illustrates the semi-dynamic approach that iterates through sample space of the dropout rate, batch size, and learning rate.

Optimization Functions: F_k (f_1,… f_k)
Gradient Descent Optimization Algorithm: Ga (g_1,….g_a)
Dropout Rate: D_r (d_1,….dr)
Batch Size: Bs (b_1,…b_s) each consecutiuve $b_i = 2*b_{i-1}$
Learning Rate: Lj (l_1,…l_j) each consecutive $l_j = .8\ l_{j+1}$

```
Assign a midway dropout rate, batch size and learning rate from given sample space
    for i =1 to k (Fk)
      for j=1 to a (Ga)
        run models, save as baseline models (deploy EarlyStopping, ModelCheckpoint)
        increment dropout rate, batch size, learning rate one at a time
          if (new_model < baseline) switch sample space direction else update baseline
      end for
  end for
```

Fig. 1. A semi-dynamic hyperparameter optimization approach

On the other hand, the proposed model architecture realized by trials when blended with the hyperparameter optimization approach. The architecture includes a double stacked Convolutional layer, Max Pooling, and Dropout. Double stacked Convolutional architecture supported accuracy improvements, whereas Dropout after each Max Pooling layer reduced overfitting. Fig. 2 depicts the proposed classification architecture with Dropout between hidden layers and an increase in Dropout with Dense layers as a regulator for overfitting. Hidden layers use ReLu as a nonlinear activation function followed by a Softmax activation function in the output layer.

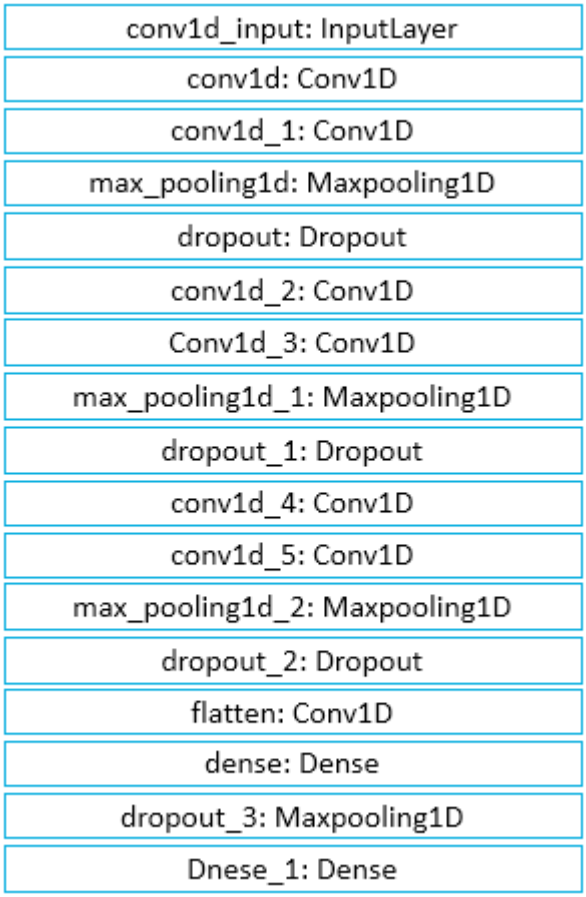

Fig. 2. Proposed classification architecture

Although class discrepancy is not much of an issue for a binary model, the ten-class model suffers from class imbalance with 56K instances on the uppermost class and only 130 cases on the bottom extreme. Class imbalance causes significant model downgrade, thus validating the need for bootstrapping to supplement the under-sampled classes. However, due to data redundancy and duplication of the training data and a fair comparison to the existing models, we opted to remain with the original datasets. In addition to the original datasets, we investigated a second option of merging the datasets and then splitting on 70-30 training and testing sets. For the remainder of the paper, we reference the second option as *user-defined* datasets.

1. Results and Discussions

The architecture coupled with the semi-dynamic hyperparameter tuning approach is used for readily available pre-partitioned UNSW-NB15 train and test datasets and the user-defined datasets.

A. Model testing for the pre-partitioned UNSW-NB15 datasets

The benchmarks on the pre-partitioned UNSW-NB15 dataset with a full set of features using deep learning models range anywhere from 78.4% [17] to 98.47% on a wrapper-based approach discussed in [22]. The proposed architecture and hyperparameter approach provided significant model performance resulting in 94.4% accuracy for the testing dataset. The algorithm suggested using 'categorical cross entropy' with the Adam optimization algorithm with momentum to prevent overfitting. The approach offers a learning rate of 0.001 with a lower dropout rate for the first two double-stacked layers and an increase for the final layer. Fig. 3 presents the learning curve accuracy for

the multiclass classification model. On the other hand, Fig. 4 illustrates the detection rate for the types of attacks. As shown, the detection rate for the underrepresented classes was significantly lower.

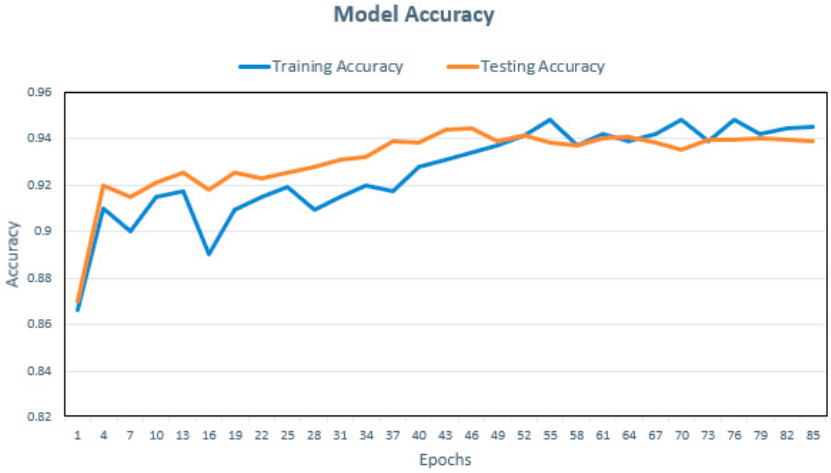

Fig. 3. Learning curve accuracy for the multiclass classification model

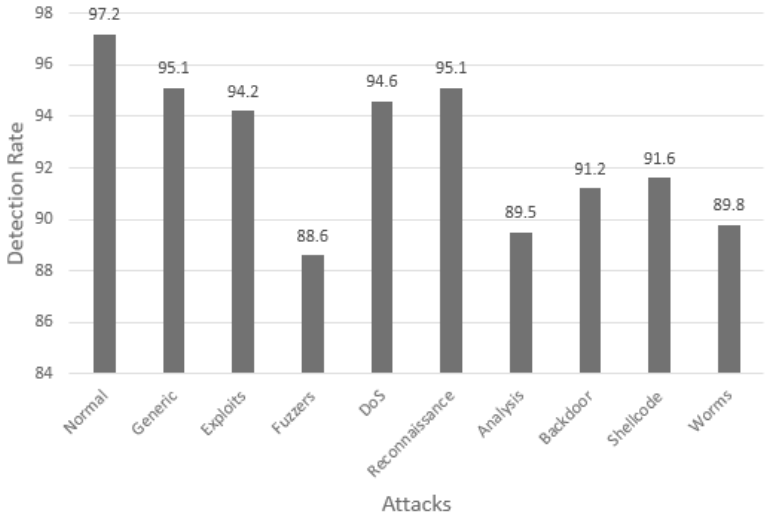

Fig. 4. Detection rates for the types of attacks

B. Model testing for the user-defined UNSW-NB15 datasets

The benchmarks on the user-defined partition of the UNSW-NB15 dataset with a full set of features using deep learning models are significantly higher than the pre-partitioned data. The proposed architecture and hyperparameter approach provided significant model performance resulting in 95.6% accuracy for the 25% testing dataset. Like the other datasets, the algorithm suggested using 'categorical cross entropy' with the Nadam optimization algorithm with momentum to prevent overfitting. The approach offers a slightly higher learning rate of 0.005 with the same dropout rate pattern. Fig. 5 presents the learning curve accuracy for the multiclass classification model. On the other hand, Fig. 6 illustrates the detection rate for the types of attacks. Similar to the pre-partitioned datasets model, the detection rate for the underrepresented classes was significantly lower.

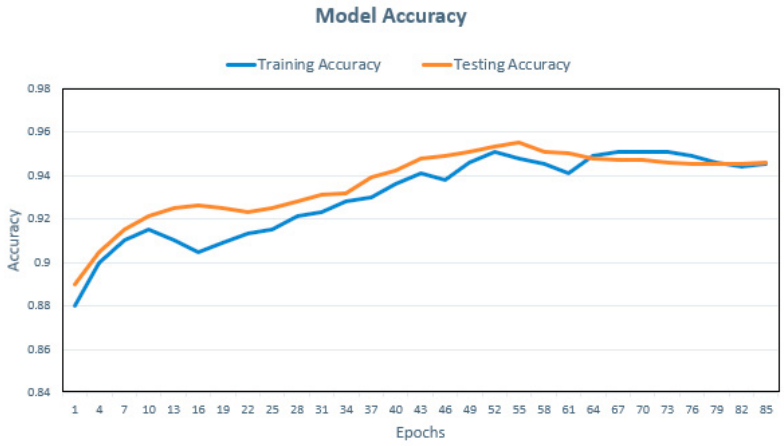

Fig. 5. Learning curve accuracy for the multiclass classification model

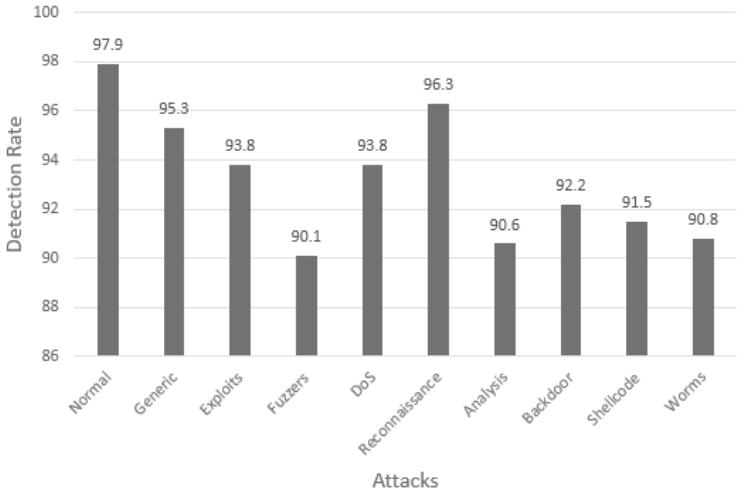

Fig. 6. Detection rates for the types of attacks

4. Conclusions and Future Work

This paper addressed network intrusion detection systems using the latest simulated network traffic dataset, incorporating relevant features and common cybersecurity vulnerabilities and exposures. The proposed deep-learning classification architecture coupled with the semi-dynamic hyperparameter tuning approach demonstrated significant improvements to multiclass models compared to the results of similar deep learning-based network IDSs. The models showed that our proposed approach obtained an overall accuracy of 95.4% and 95.6% for the pre-partitioned and user-defined multiclass classification.

Although the proposed models have achieved promising results, we acknowledge that there is room for improvements, mainly using feature reduction methods. Future work calls for transfer learning with relevant existing datasets to serve as a baseline for model classification improvements with the UNSW-NB15 dataset and extend our models' capability to handle zero-day attacks. In addition to transfer learning, bootstrapping techniques to produce a well-balanced dataset to train a multiclass classification model will be investigated. Deep learning anomaly detection models will amend cybersecurity architectures to develop adaptive and resilient network intrusion detection systems to detect common vulnerabilities and exposures correctly and zero-day network behavioral features reducing the risk of compromise.

References

[1] Ashiku, Lirim, and Cihan Dagli. (2019) "Cybersecurity as a Centralized Directed System of Systems using SoS Explorer as a Tool." *2019 14th Annual Conference System of Systems Engineering (SoSE)*, 140-145. IEEE.

[2] Duque, Solane, and Mohd Nizam bin Omar. (2015) "Using data mining algorithms for developing a model for intrusion detection system (IDS)." *Procedia Computer Science,* 61: 46-51

[3] Stallings, William, Lawrie Brown, Michael D. Bauer, and Arup Kumar Bhattacharjee.(2012) *Computer security: principles and practice*. Upper Saddle River, NJ, USA. Pearson Education.

[4] Whitman, Michael E., and Herbert J. Mattord.(2011) *Principles of information security*. Cengage Learning.

[5] Shone, Nathan, Tran Nguyen Ngoc, Vu Dinh Phai, and Qi Shi. (2018) "A deep learning approach to network intrusion detection." *IEEE Transactions on Emerging Topics in Computational Intelligence,* 2, no. 1: 41-50.

[5] L. Lipton, Zachary C., John Berkowitz, and Charles Elkan. (2015) "A critical review of recurrent neural networks for sequence learning." *arXiv preprint arXiv:1506.00019*.

[7] Moustafa, Nour, and Jill Slay. (2015) "UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)." *2015 military communications and information systems conference (MilCIS)*, 1-6. IEEE.

[8] Osken, Sinem, Ecem Nur Yildirim, Gozde Karatas, and Levent Cuhaci. (2019) "Intrusion Detection Systems with Deep Learning: A Systematic Mapping Study." *2019 Scientific Meeting on Electrical-Electronics & Biomedical Engineering and Computer Science (EBBT)*, 1-4. IEEE.

[9] McHugh, John. (2000) "Testing intrusion detection systems: a critique of the 1998 and 1999 darpa intrusion detection system evaluations as performed by lincoln laboratory." *ACM Transactions on Information and System Security (TISSEC)* 3, no. 4: 262-294.

[10] Moustafa, Nour, and Jill Slay. (2015) "The significant features of the UNSW-NB15 and the KDD99 data sets for network intrusion detection systems." In *2015 4th international workshop on building analysis datasets and gathering experience returns for security (BADGERS)*, 25-31. IEEE.

[11] "Home." CVE. Accessed December 12, 2019. http://cve.mitre.org/about/index.html.

[12] Janarthanan, Tharmini, and Shahrzad Zargari. (2017) "Feature selection in UNSW-NB15 and KDDCUP'99 datasets." In *2017 IEEE 26th International Symposium on Industrial Electronics (ISIE)*, 1881-1886. IEEE.

[13] Pfahringer, Bernhard.(2000) "Winning the KDD99 classification cup: bagged boosting." *SIGKDD explorations,* 1, no. 2: 65-66.

[14] Sabhnani, Maheshkumar, and Gürsel Serpen. (2003) "Application of Machine Learning Algorithms to KDD Intrusion Detection Dataset within Misuse Detection Context." *MLMTA*, 209-215.

[15] Lin, Wen-Hui, Hsiao-Chung Lin, Ping Wang, Bao-Hua and Jeng-Ying Tsai. (2018) "Using convolutional neural networks to network intrusion detection for cyber threats." *2018 IEEE International Conference on Applied System Invention (ICASI),* 1107-110. IEEE.

[16] Vinayakumar, R., K. P. Soman, and Prabaharan Poornachandran. (2017) "Applying convolutional neural network for network intrusion detection." *2017 International Conference on Advances in Computing, Communications, and Informatics (ICACCI)*, 1222-1228. IEEE.

[17] Vinayakumar, R., Mamoun Alazab, K. P. Soman, Prabaharan Poornachandran, Ameer Al-Nemrat, and Sitalakshmi Venkatraman. (2019) "Deep Learning Approach for Intelligent Intrusion Detection System." *IEEE Access* 7: 41525-41550.

[18] "Keras: The Python Deep Learning Library." (2020) n.d. Home - Keras Documentation. Accessed January 11, 2020. https://keras.io/.

[19] Goodfellow, Ian, Yoshua Bengio, and Aaron Courville. (2016) *Deep learning*. MIT press.

[20] Jing, Dishan, and Hai-Bao Chen. (2019) "SVM Based Network Intrusion Detection for the UNSW-NB15 Dataset." In *2019 IEEE 13th International Conference on ASIC (ASICON)*, 1-4. IEEE.

[21] Zhiqiang, Liu, Ghulam Mohi-Ud-Din, Li Bing, Luo Jianchao, Zhu Ye, and Lin Zhijun. (2019) "Modeling Network Intrusion Detection System Using Feed-Forward Neural Network Using UNSW-NB15 Dataset." *2019 IEEE 7th International Conference on Smart Energy Grid Engineering (SEGE)*, 299-303. IEEE.

[22] Latif, Shahid, Zeba Idrees, Zhuo Zou, and Jawad Ahmad. (2020) "DRaNN: A Deep Random Neural Network Model for Intrusion Detection in Industrial IoT." *2020 International Conference on UK-China Emerging Technologies (UCET)*, 1-4. IEEE.

[23] Kasongo, Sydney Mambwe, and Yanxia Sun. (2020) "A deep learning method with wrapper based feature extraction for wireless intrusion detection system." *Computers & Security* 92: 101752.

[24] Choudhary, Sarika, and Nishtha Kesswani. (2020 "Analysis of KDD-Cup'99, NSL-KDD and UNSW-NB15 Datasets using Deep Learning in IoT." *Procedia Computer Science,* 167: 1561-1573.

[25] Tran Viet, Khoa, Saputra Yuris Mulya, Hoang Dinh Thai, Trung Nguyen Linh, Diep Nguyen N, Ha Nguyen Viet, and Dutkiewicz Eryk. (2020) "Collaborative learning model for cyberattack detection systems in IoT industry 4.0."

[26] Paul, Supratik, Vitaly Kurin, and Shimon Whiteson. (2019) "Fast efficient hyperparameter tuning for policy gradients." *arXiv preprint arXiv:1902.06583*.