Deep Learning Network Intrusion Detection with the Conv1d-Lstm Model: Integrating CNN and LSTM For Superior Performance

Cikambasi Ciza Lukogo[1*], Lawrence Mwenda Muriira[1], Robert Mutua Murungi[1]

[1]*Kenya Methodist University, Po Box 267-60200, Meru, Kenya*

**Correspondence email:* chrispinciza01@gmail.com

Abstract

Increased cases of cyber-attack and the rising levels of sophistication presents a significant threat to corporate networks, resulting in potential data breaches, financial losses, and reputational harm. Traditional Intrusion Detection Systems, which rely on predefined signatures and rules, have proven inadequate due to high false positive and false negative rates. This study introduces an innovative AI-based intrusion detection model to enhance corporate network security leveraging on deep learning techniques. The objective was to propose a Conv1d-LSTM Model, integrating convolutional neural networks (CNN) and recurrent neural networks (RNN) to analyze network traffic data from the CSE-CIC-IDS-2018 dataset, which encompasses a wide array of attack types, and provides a realistic representation of modern network traffic. This deep learning model effectively detects complex patterns and temporal dependencies in the data. The performance of the innovated model was evaluated using precision, accuracy, recall, and F1 score, to demonstrate its superior detection capabilities compared to conventional Intrusion Detection Systems (IDS). Additionally, a comparative analysis of CNN and RNN performance on the same dataset was conducted, highlighting the strengths and limitations of each approach. This research underscores the importance of integrating advanced AI methodologies into IDS frameworks to protect corporate networks from cyber threats.

Keywords*: Deep Learning, Intrusion Detection, Conv1D-LSTM Model, Convolutional Neural Networks, Recurrent Neural Networks, Cyber Security, CSE-CIC-IDS-2018 Dataset (A Realistic Cyber Defense Dataset)*

IJPP 12(4); 41-49

1.0 Introduction

In today's digital era, network security is important to an organization that relies on IT infrastructure to function. Malicious cyber-attacks can jeopardize the confidentiality, integrity and availability of information, leading to substantial financial losses and reputational damage. For example, the 2018 Facebook breach exposed data from 50 million users (Rehman, 2019). Incidentally, the average cost of a data breach in 2020 was $3.86 million, with breaches taking about 280 days to identify and contain (Ponemon, 2020).

Detecting and preventing unauthorized activities and intrusions presents a primary challenge in network security. Techniques such as denial-of-service, infiltration, SQL injection, and malware pose significant threats (Contributor, 2019). Traditional Intrusion Detection Systems (IDS) use predetermined rules to identify known threats but struggle with modified attacks, often resulting in false alarms (Khraisat et al., 2019). Recent studies have shown that these limitations necessitate more adaptive solutions in intrusion detection.

To address these challenges, AI-based IDS have emerged, leveraging ML and DL techniques to adapt to evolving threats without relying on predefined rules (Park et al., 2022). These systems aim to reduce false positives and enhance detection accuracy by learning from diverse data sources (Fortinet, n.d.).

Recent advancements in AI have spurred the development of AI-based IDS models. For instance, while Liang et al. (, 2019) combined clustering with SVM to detect attacks, achieving a 99.450% accuracy on the NSL-KDD dataset, Kanimozhi et al. (2022) introduced a method for cloud intrusion detection using the oppositional fuzzy C-means algorithm, attaining an 80% accuracy rate on the CICIDS2017 dataset. Deep learning approaches have shown promising results in intrusion detection. Ashwaq et al. (2022) leveraged Recurrent Neural Networks (RNN) to secure IoT environments, achieving an 87% accuracy on the NSL-KDD dataset. Xiao et al. (2019) used CNN to identify intrusion in the network, The model achieved a 94% accuracy on the KDDcup99 dataset.

Various deep learning architectures, such as RNNs, CNNs, Generative Adversarial Networks (GANs), and auto-encoders, have been combined to enhance IDS. For instance, Chawla et al. (2019) employed Gated Recurrent Units (GRUs) combined with CNNs for anomaly detection, demonstrating faster convergence and higher true positive rates compared to traditional methods. Zhang et al. (2019) combined CNN and GcForest techniques, achieving a 99.24% accuracy on the UNSW-NB15 and CICIDS2017 datasets. Selvarajan et al. (2023) suggest a new approach to cyber-attack detection and prevention, which combines LSTM-CNN with a fully connected neural network, incorporating hypermeter optimisation for intrusion detection.

These studies highlight the importance of combining deep learning techniques to improve intrusion detection accuracy and efficiency. Our research builds on the foundation established by these studies by integrating CNN and RNN networks to develop a Conv1d-LSTM model. This developed model aims to analyse network traffic data from the CSE-CIC-IDS-2018 dataset, providing a comprehensive solution to sophisticated cyber-attacks

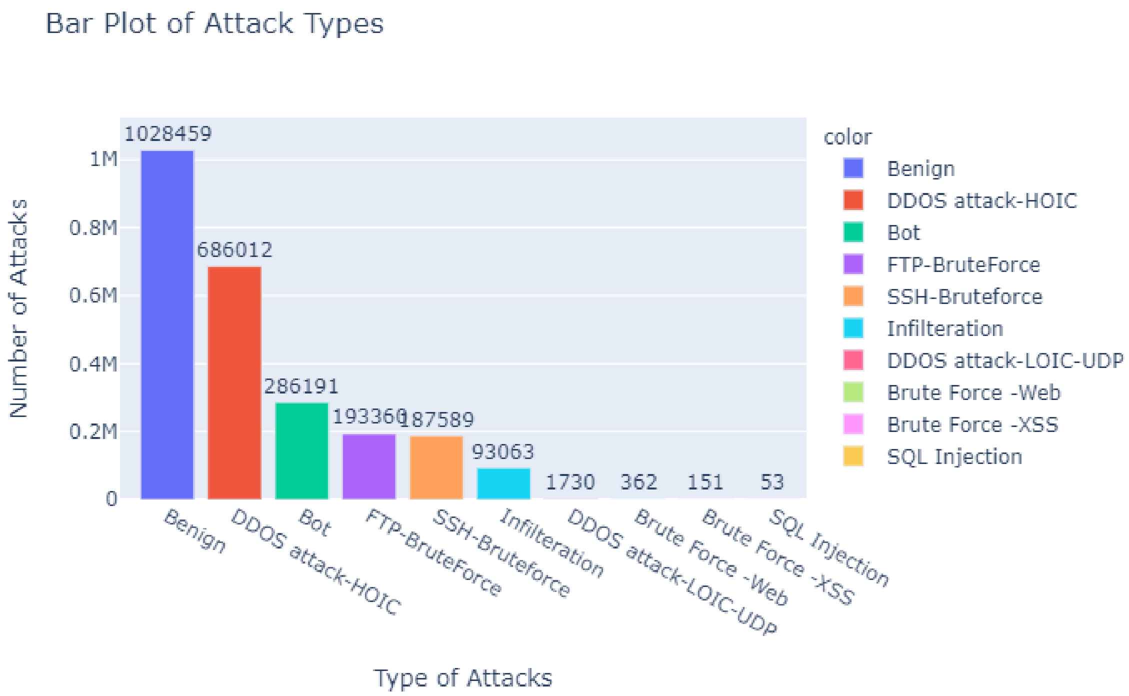
2.0 Materials and Methods

Dataset

The CSE-CIC-IDS-2018 dataset, which includes various types of attacks, was used for this study. In this research, the CSE-CIC-IDS-2018 data was obtained from the official website of the Canadian Institute for Cybersecurity (CIC). The institute provided access to the dataset on their official website (UNB, n.d.). It provides a realistic representation of modern network traffic, Fig 1 presents the features in the dataset.

Figure 1

Features in the "CSE-CIC-IDS-2018" dataset



The proposed Model

The proposed model combines CNN and RNN architectures to detect both spatial and temporal aspects of network traffic data. While CNNs are effective in feature extraction, RNNs excel at modelling sequential dependencies, allowing the model to capture complex patterns in the data. For model development, the research utilized Azure Machine Learning. Azure Machine Learning offers the advantage of training and testing the model on GPU platforms for rapid experimentation, as deep learning requires high computational capability for optimal performance. The proposed architecture begins with an input layer that accepts 850-dimensional vectors with a single channel. This is followed by a series of Conv1D layers, each applying a set of filters with a ReLU activation function.

"The study developed a proposed hybrid model that combines the strength of CNN and RNN model which when tested, it achieved an accuracy of 99.97%, demonstrating the best convergence and

These layers are interspersed with MaxPooling1D layers to reduce the spatial dimensions, BatchNormalization layers to stabilize and accelerate training, and Dropout layers to prevent overfitting.

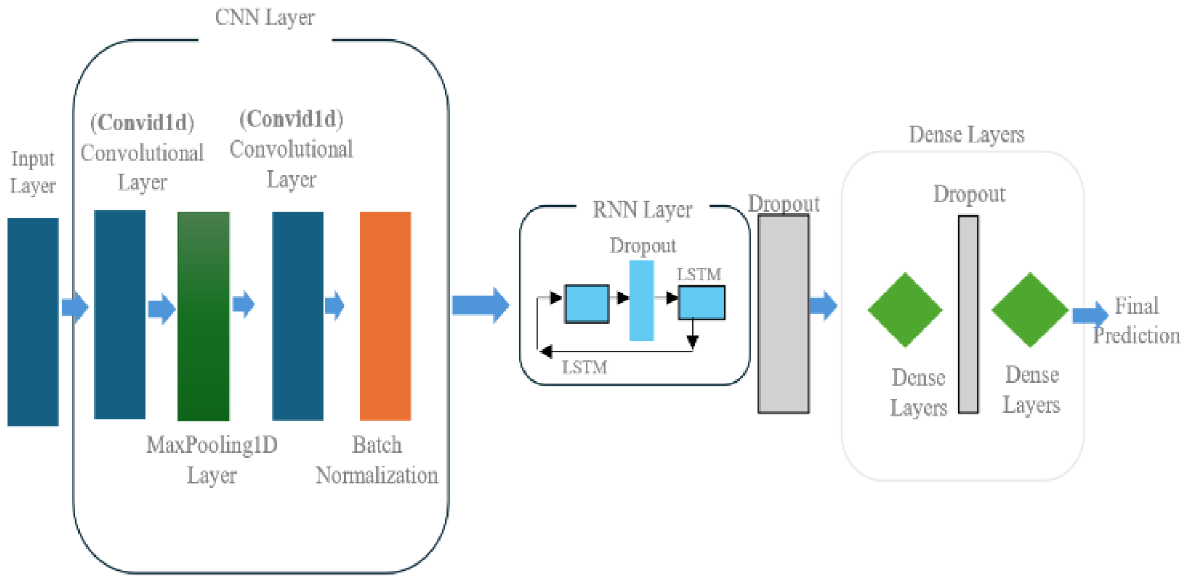
Specifically, the initial Conv1D layer uses 64 filters with a kernel size of 3, producing feature maps with an output shape of (None, 80, 64). This is followed by MaxPooling, BatchNormalization, and Dropout layers, maintaining an output shape of (None, 40, 64).

A second Conv1D block applies 128 filters, further refining the feature extraction process. Subsequent MaxPooling, BatchNormalization, and Dropout layers continue to reduce dimensionality and prevent overfitting, resulting in an output shape of (None, 20, 128).

To capture temporal dependencies, the model incorporates an LSTM layer with 64 units, transforming the feature maps into sequences with an output shape of (None, 20, 64). A final Dropout layer ensures robust training by mitigating overfitting. The architecture concludes with a Dense layer, yielding a final output shape of (None, 10), suitable for classification or regression tasks. Fig 2 presents the model architecture.

Figure 2

Proposed AI-based Model



Why Combining The cnn andrnn Algorithms

CNNs use pooling layers and convolutional filters to recognize characteristics in images and other forms of data. They can detect different types of network intrusions by categorizing data from network monitoring into distinct groups. RNNs are good at processing sequential input and detecting temporal patterns. They can detect a variety of network intrusions by extracting attributes from network monitoring data and classifying them.

Thus, integrating CNNs and RNNs results in a more robust and versatile model that can process both spatial and temporal aspects of network data. This combination allows the models to learn patterns in the dataset, thereby reducing the false positive and negative rates produced by traditional intrusion detection methods.

Data Preprocessing

Preparing data is a crucial phase in the data mining process, aimed at enhancing the quality and utility of data to achieve better outcomes in analysis. This phase not only improves data quality, but also facilitates the extraction of meaningful and useful information, preparing the data for subsequent stages and ensuring its integrity and relevance (Tawakuli et al., 2022).

Initially, the dataset undergoes filtering to remove redundant rows representing class instances. Following this, an analysis is conducted to identify any ‘NAN’ (Not A Number) or ‘INF’ (Infinite Value) values, which are treated as missing data. These missing values, particularly in the ‘Flow Bytes’ column, are addressed by filling them with a specific value such as the mean. This approach maintains the data's integrity while preserving the significance of the ‘Flow Bytes’ feature.

Feature Selection

Feature selection aims to identify the most significant attributes for effective classification or prediction, thereby reducing the problem's dimensionality and resource requirements (storage, computation). This process can also enhance machine learning algorithms' performance by speeding up training, reducing overfitting, and sometimes improving prediction accuracy (Tawakuli et al., 2022). While it may appear that feature selection leads to information loss, it is not usually the case when dealing with redundant or irrelevant information. Redundant features often duplicate information found in other attributes or can be derived from other features.

Descriptive statistics revealed columns with zero values, such as “Bwd PSH Flags”, “Bwd URG Flags”, and others, which do not contribute discriminative information for distinguishing attack classes. These columns were removed to prevent suboptimal results and improve model performance. Additionally, features like ‘IP Address’ and ‘Timestamp’ were deemed irrelevant for network attack characteristics and were excluded to focus on network traffic-related features.

Feature Encoding

The dataset contains several categorical features that require encoding. For example, the ‘Flow Packets/s’ column, which is converted to a numeric format. The ‘Label’ column indicates the class of each instance, was transformed using ‘One-Hot Encoding’. This method converts each category into its distinct column, with a value of 1, which means that the instance belongs to that class and 0 otherwise. This conversion allows the machine learning model to utilize categorical data effectively without losing any information.

Feature Scaling

Standardization is a critical aspect of feature scaling, which involves transforming input data; so that each feature in the data has a mean of zero with a standard deviation of one. This improves the model-building phase by accelerating convergence during training and potentially introducing a regularization effect. Standardization mitigates the influence of varying feature scales, enhancing the accuracy of data analysis and machine learning models.

3.0 Results and Discussion

This study introduces a hybrid model combining CNN and RNN. First, the three models were trained and tested separately using the CIC-IDS2018 dataset. Several hyperparameters, including the number of neurons, layers, batch size, and iterations, were fine-tuned for the models.

The CSE-CIC-IDS-2018 dataset was divided into 80% data to train the modes, and 20% data for models testing. Each model was trained for 20 epochs. The CNN model achieved 99.86% accuracy, RNN model had 99.87% accuracy, while the hybrid model attained 99.97% accuracy. Both CNN and RNN models converged to a minimum loss value, showing similar learning and evaluation loss values. The learning and validation accuracy for all models increased consistently from the start to the end of training, approaching a maximum value of 1. Additionally, the loss value decreased significantly during training and evaluation, reaching a minimum value close to 0. This indicates that the models improved their predictive performance with each optimization epoch.

Evaluation Metrics for the Models

Precision (PPV): The percentage of network attacks identified as TP attacks from all predicted examples as attacks can be calculated as follows:

$$\text{precision} = \frac{TP\ Attack}{(TP + FP\ \text{Benign.}\ )} \qquad (1)$$

Recall (TPR.): The proportion of network attacks that are correctly identified as true positives (TP) out of all actual attacks present in the dataset.

$$\text{Recall} = \frac{TP\ Attack}{(True\ positive\ + FN\ Attack)} \qquad (2)$$

F1-score (F1): The weighted harmonic means of precision and recall.

$$\text{F1} - \text{Score} = 2 * \frac{Precission * recall}{precision + recall} \qquad (3)$$

Accuracy (Acc): "The total of correctly predicted to the total number of predictions" (Selvarajan et al., 2023).

$$\text{Accurancy} = \frac{Number\ of\ correct\ prediction}{Total\ number\ of\ predictions} \qquad (4)$$

Model Comparison

The CNN model exhibited high overall accuracy, with most classes demonstrating near-perfect precision, recall, and F1 scores. However, minor misclassifications were observed in a few classes. For example, the Label_Benign class had a high precision of 0.999 and recall of 0.996735, but a slightly lower F1-score due to some benign samples being misclassified. The Label_SSH-Bruteforce class also showed high performance with a few misclassifications. The CNN model effectively differentiated between most types of network traffic, though it struggled with subtle attack vectors.

The RNN model also achieved high overall accuracy with similar strengths and weaknesses compared to the CNN model. For instance, the Label_Bot and Label_SQL Injection classes had perfect precision and recall. However, the RNN model exhibited more significant misclassifications in certain classes. The Label_Infiltration class had a lower recall, indicating that some infiltration attacks were misclassified. The Label_SSH-Bruteforce class showed more misclassifications compared to the CNN model. Despite these issues, the RNN model performed well, particularly in identifying straightforward attack types.

The hybrid model combined the strengths CNN and RNN algorithm, resulting in exceptional performance across most classes. The confusion matrix for the hybrid model presented an overall accuracy of 99.97%, with near-perfect precision, recall, and F1 scores across most classes.

The hybrid model addressed some of the misclassification issues observed in the CNN and RNN models. For instance, the Label Infiltration class had a recall of 0.9873, significantly better than the RNN model's performance. The hybrid model also demonstrated improved precision for the Label SQL Injection class compared to the CNN model, the Label_SSH-Bruteforce class shows high precision at 0.9977, and recall at 0.9995, with an F1-score of 0.9986 for 3990 samples. The Label FTP-BruteForce class also shows no misclassifications, with recall, F1-score and precision at 1.0000 for 3978 samples. These results suggest that the hybrid model effectively leverages on the strengths of both algorithm, CNN and RNN, to attain superior performance.

In summary, the hybrid model outperformed both the CNN and RNN models. The CNN model showed strong performance in identifying various attack types, but struggled with subtle distinctions. On the other hand, RNN model had similar strengths but more significant issues with certain classes. The hybrid model combined the advantages of both architectures, leading to near-perfect classification across most types of network traffic.

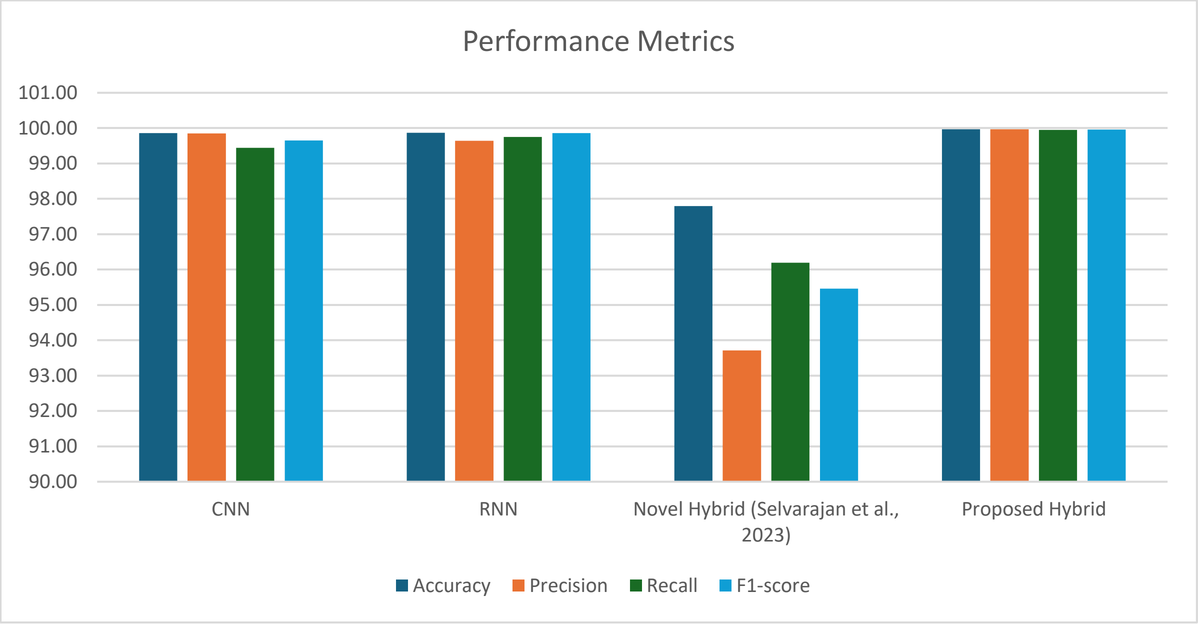
Table 1

Evaluation metrics

Algorithm	Evaluation Metrics			
	Accuracy	Precision	Recall	F1-score
CNN	99.86%	99.85%	99.4%	99.65%
RNN	99.87%	99.64%	99.7%	99.86%
Novel Hybrid (Selvarajan et al., 2023)	97.80%	93.71%	96.1%	95.46%
Proposed Hybrid	99.97%	99.97%	99.9%	99.96%

Figure 3

Model's Performance



The experimental results given in Table 1 show that the proposed hybrid model achieves the highest performance across all evaluated metrics compared to the existing algorithms. Specifically, the accuracy of the proposed model is 99.97%, which surpasses the existing Novel Hybrid (Selvarajan et al., 2023) at 97.80%, CNN at 99.86%, and RNN at 99.87%. In terms of precision, the proposed model maintains an exceptional score of 99.97%, significantly outperforming the Novel Hybrid

(Selvarajan et al., 2023) with 93.71%, CNN with 99.85%, and RNN with 99.64%. The recall for the proposed model is also superior at 99.95%, compared to the Novel Hybrid (Selvarajan et al., 2023) at 96.19%, CNN at 99.44%, and RNN at 99.75%. Further, the F1-score for the proposed model stands at 99.96%, highlighting its balanced and robust performance, while the Novel Hybrid (Selvarajan et al., 2023) achieves 95.46%, CNN 99.65%, and RNN 99.86%.

These results indicate that the proposed hybrid model exhibits superior performance among all evaluated classifiers, underscoring its effectiveness and reliability for network intrusion detection. The Conv1d-Lstm model demonstrates significant improvements: 99.97% in precision, 99.95% in recall, 99.97% accuracy, and 99.96% in F1 score. These results highlight the effectiveness of hybrid architecture, which leverages on the strengths of CNNs and RNNs. These findings underscore the critical role of advanced AI methodologies in safeguarding corporate networks against evolving cyber threats. The Conv1d-Lstm Model holds significant promise for enhancing network intrusion detection capabilities. However, further research in exploring the full potential of Conv1d-Lstm Model is necessary.

4.0 Conclusion

Several deep learning (DL) models have been developed for network intrusion detection. In this study, a hybrid model that combines the strength of CNN and RNN was proposed. The proposed model was trained using the CIC-IDS2018 dataset, with a focus on optimizing various hyperparameters, over 20 epochs. The model achieved the highest accuracy of 99.97%, demonstrating the best convergence and stability.

5.0 Recommendations

Based on the findings in this study, the following recommendations can improve the effectiveness of DL models for network intrusion detection: continuously optimize hyperparameters like neurons, layers, batch size, and epochs, employing automated tuning techniques such as grid search or Bayesian optimization for optimal configurations. Additionally, feature extraction should be improved to reduce misclassifications, especially in classes with overlapping features, using advanced feature engineering and additional relevant features. Overfitting should also be mitigated with regularization methods like dropout, L1/L2 regularization, or data augmentation to improve model generalization, especially with imbalanced datasets. Continuous learning and adaptation mechanisms, such as online learning or periodic retraining with new data, to keep models updated with emerging attack patterns can also be implemented.

References

Chawla, A., Lee, B., Fallon, S., & Jacob, P. (2019). Host based intrusion detection system with combined CNN/RNN model. In *ECML PKDD 2018 Workshops: Nemesis 2018, UrbReas 2018, SoGood 2018, IWAISe 2018, and Green Data Mining 2018, Dublin, Ireland, September 10-14, 2018, Proceedings 18* (pp. 149-158). https://link.springer.com/chapter/10.1007/978-3-030-13453-2_12

Contributor, S. (2021). *What Is an Intrusion Detection System? Definition, Types, and Tools*. https://www.dnsstuff.com/intrusion-detection-system

Fortinet (n.d.). *Intrusion Detection System (IDS).* Fortinet. https://www.fortinet.com/resources/cyberglossary/intrusion-detection-system

Jayalaxmi, P. L. S., Saha, R., Kumar, G., Conti, M., & Kim, T. H. (2022). Machine and deep learning solutions for intrusion detection and prevention in IoTs: A survey. *IEEE Access*, *10*, 121173-121192. https://ieeexplore.ieee.org/abstract/document/9941074

Kanimozhi, P., & Aruldoss Albert Victoire, T. (2022). Oppositional tunicate fuzzy C-means algorithm and logistic regression for intrusion detection on cloud. *Concurrency and computation: practice and experience*, *34*(4), e6624. https://onlinelibrary.wiley.com/doi/abs/10.1002/cpe.6624

Khraisat, A., Gondal, I., Vamplew, P., & Kamruzzaman, J. (2019). Survey of intrusion detection systems: techniques, datasets and challenges. *Cybersecurity*, *2*(1), 1-22. https://link.springer.com/article/10.1186/s42400-019-0038-7

Liang, D., Liu, Q., Zhao, B., Zhu, Z., & Liu, D. (2019, October). A clustering-svm ensemble method for intrusion detection system. In *2019 8th International Symposium on Next Generation Electronics (ISNE)* (pp. 1-3). https://ieeexplore.ieee.org/abstract/document/8896514

Park, D., Kim, S., Kwon, H., Shin, D., & Shin, D. (2021). Host-based intrusion detection model using siamese network. *IEEE Access*, *9*, 76614-76623. https://ieeexplore.ieee.org/abstract/document/9436776

Ponemon, L. (2020). *Cost of a Data Breach Report 2019.* https://securityintelligence.com/posts/whats-new-in-the-2019-cost-of-a-data-breach-report/

Rehman, I. (2019). Facebook-Cambridge Analytica data harvesting: What you need to know. *Library Philosophy and Practice*, 1-11. https://digitalcommons.unl.edu/libphilprac/2497

Selvarajan, P., Salman, R., Ahamed, S., & Jayasuriya, P. (2023, February). Networks Intrusion Detection Using Optimized Hybrid Network. In *2023 International Conference on Smart Computing and Application (ICSCA)* (pp. 1-6). https://ieeexplore.ieee.org/abstract/document/10087611

Tawakuli, A., & Engel, T. (2022, December). Towards Normalizing The Design Phase of Data Preprocessing Pipelines For IoT Data. In *2022 IEEE International Conference on Big Data (Big Data)* (pp. 4589-4594). 10.1109/BigData55660.2022.10020312

Xiao, Y., Xing, C., Zhang, T., & Zhao, Z. (2019). An intrusion detection model based on feature reduction and convolutional neural networks. *IEEE Access*, *7*, 42210-42219. https://ieeexplore.ieee.org/abstract/document/8666014

UNB (n.d). *A Realistic Cyber Defense Dataset (CSE-CIC-IDS2018)* https://registry.opendata.aws/cse-cic-ids2018.