CFS-AE: Correlation-based Feature Selection and Autoencoder for Improved Intrusion Detection System Performance

Seiba Alhassan[1*], Dr. Gaddafi Abdul-Salaam[2], Asante Micheal[3], Yaw Marfo Missah[4], Dr. Ernest D. Ganaa[5] and Alimatu Sadia Shirazu[6]

[1*]Lecturer, Department of ICT, Dr Hilla Limann Technical University, Ghana. salhassan@dhltu.edu.gh/alhseiba@gmail.com, https://orcid.org/0000-0003-4734-0240

[2]Senior Lecturer, Department of Computer Science, Kwame Nkrumah University of Science and Technology, Ghana. gaddafi.ict@knust.edu.gh, https://orcid.org/0000-0001-9393-7780

[3]Associate Professor, Department of Computer Science, Kwame Nkrumah University of Science and Technology, Ghana. m.asante@knust.edu.gh/mickasst@yahoo.com

[4]Senior Lecturer, Department of Computer Science, Kwame Nkrumah University of Science and Technology, Ghana. ymissah.cos@knust.edu.gh, https://orcid.org/0000-0002-2926-681X

[5]Senior Lecturer, Department of ICT, Dr Hilla Limann Technical University, Ghana. gernest@dhltu.edu.gh, https://orcid.org/0000-0002-2161-7435

[6]PHD Candidate, Department of Computer Science, Kwame Nkrumah University of Science and Technology, Ghana. shiraz.sadia84@gmail.com, https://orcid.org/0000-0003-4073-0211

Received: October 12, 2023; Accepted: December 18, 2023; Published: February 29, 2024

Abstract

The major problem computer network users face concerning data – whether in storage, in transit, or being processed - is unauthorized access. This unauthorized access typically leads to the loss of confidentiality, integrity, and availability of data. Consequently, it is essential to implement an accurate Intrusion Detection System (IDS) for every information system. Many researchers have proposed machine learning and deep learning models, such as autoencoders, to enhance existing IDS. However, the accuracy of these models remains a significant research challenge. This paper proposes a Correlation-Based Feature Selection and Autoencoder (CFS-AE) to enhance detection accuracy and reduce the false alarms associated with the current anomaly-based IDS. The first step involves feature selection for the NSL-KDD and CIC-IDS2017 datasets which are used to train and test our model. Subsequently, an autoencoder is employed as a classifier to categorize data traffic into attack and normal categories. The results from our experimental study revealed an accuracy of 94.32% and 97.71% for the NSL-KDD and CIC-IDS2017 datasets, respectively. These results demonstrate improved performance over existing IDS systems.

Keywords*:* Autoencoder, Feature Selection, IDS, CFS, Anomaly, Intrusion, Train.

Journal of Internet Services and Information Security (JISIS), volume: 14, number: 1 (February), pp. 104-120.
DOI: *10.58346/JISIS.2024.I1.007*

*Corresponding author: Lecturer, Department of ICT, Dr Hilla Limann Technical University, Ghana.

1 Introduction

The number of computer network users, particularly on the internet, has been steadily increasing every day. This upward trend implies that sensitive personal, confidential, or highly classified business information becomes vulnerable to cybercriminals if not adequately protected. Cybersecurity professionals have implemented various security measures to safeguard such data during storage, transit, and processing. These measures include Cryptography, Access control methods, Authentication techniques, Antivirus, Intrusion Detection Systems (IDS), and more.

Among these security measures, the Intrusion Detection System is extensively explored in the literature as a technique to enhance security for individual users and information systems. Intrusion, defined as compromising the integrity, availability, and confidentiality of information on a computer network through unauthorized access, has been a subject of significant attention in the field (Maseno et al., 2022). As per Ali et al. (2018), an intrusion detection system serves as the primary security technique deployed to protect IT infrastructure against intentional and unintentional attacks. This system, implemented either as hardware or software, is designed to identify attempts by both authorized and unauthorized subjects to gain illicit access to an object.

The ability of intrusion detection systems to detect and report insiders (authorized users) and external users (unauthorized users) makes it an important research area in cyber security. The prospects of IDS have attracted several researchers in Machine learning and deep learning. For instance, deep learning has also been applied to increase the performance of IDS further (Siddique et al., 2019). These researchers and many others have made significant improvements in the performance of IDS. Nevertheless, current IDSs continue to grapple with many issues, rendering them ineffective in ensuring the security of computer networks. The problems are weaknesses directly associated with a particular type of intrusion detection system (Yang and Wan 2022).

In (Sadaf & Sultana, 2020), IDS is classified into host-based IDS and network-based Intrusion Detection Systems based on where it is implemented. Alhasan et al.(2021) have stated that a network-based intrusion detection system monitors data packets arriving at a network segment and determines if those packets are normal or attacks. At the same time, host-based IDS consists of a single computing device that monitors data packets for the presence or absence of intrusion. The strength and weakness of each type of IDS is shown below in Table 1.

The methods used to implement network and host-based intrusion detection systems can also be classified into anomaly-based IDS or Signature Intrusion Detection Systems. Anomaly-based IDS, according to Khraisat et al. (2019) is where a normal model of a computer system is created using a statistical-based, machine learning, and knowledge-based system, and any deviation from data traffic from the model is deemed as an anomaly and hence an intrusion. A signature-based intrusion detection system keeps a profile of all known attacks on a computer system, and any data traffic that matches the existing signature is considered an anomaly and, if otherwise, normal traffic (Alhasan et al., 2021). Table 1 below shows the strengths and weaknesses of signature-based IDS and anomaly-based IDS.

Table 1: Strengths and Weaknesses of IDS based on where it is Implemented

Type of IDS	**Strength**	**Weakness**
Network	• Does not burden the operations of the host • No delay in detecting attacks • Can detect attacks on a broad range of network protocols	• Traffic cannot be analyzed • Can only detect attacks from a segment of the network • Dedicated hardware is required • Difficult to detect insider attacks
Host	• Can analyze encrypted traffic • Additional hardware is not required • Attack on any part of the host can be detected	• HIDS is a burden on the host machine • Delay in reporting of attacks • The HIDS system needs to be installed on each host
Type	**Strength**	**Weakness**
Anomaly-based IDS	• Capable of detecting new types of attacks • Does not require database updates	• False alarm rate is high • Low detection accuracy
Signature Based IDS	• Low false alarm rate • High detection accuracy for known attacks	• Cannot detect new types of attack • Requires constant database update

The weakness with the various types of IDS, as shown in Tables 1 has prompted several researches in the literature aimed at finding solutions to these challenges. Most of these studies have identified feature selection as a technique that can potentially improve the performance of existing IDS. When it is detected that some features contribute to the low performance of classifiers, feature selection is taken more seriously (Ren et al., 2019). Sadaf & Sultana (2020) carried out a study with the main goal of improving the performance of IDS in a FOG environment using an autoencoder and isolation forest as a dimensionality reduction technique.

Other studies ranging from machine learning to deep learning using various methods have proposed similar strategies. Even though these studies achieved significant improvement in terms of accuracy, more work needs to be done for further improvement. This study seeks to propose a correlation-based feature selection technique to improve the performance of an autoencoder IDS model using the NSL-KDD dataset and CIC-IDS2017. The study also investigates the impact of varying the number of neurons and the activation functions on the performance of the autoencoder.

2 Related Works

Feature selection is an important step in building any classification problem. An intrusion detection system that is built without feature selection may result in low model performance. Before applying the IDS method feature selection needs to be handled carefully (Saleh et al., 2019).

Correlation-based feature selection has been applied by intrusion detection system researchers including (Siddique et al., 2019). After training and validating the IDS model, they reported a reduction in the false alarm rate and an increase in detection accuracy. Since correlation-based feature selection is a filter method it those not interact directly with the classification algorithm and it treats every feature independently representing a major weakness of the proposed technique. The strength of their proposed technique however is the considerable reduction in computational cost as compared to a feature selection system that directly interacts with the classification algorithms.

Chi-square was also proposed by Thaseen & Kumar (2014) to overcome the challenges of false alarms and detention accuracy of anomaly-based intrusion detection systems. They argue that using feature selection such as chi-square provides superior performance compared to randomly selecting parameters. They further stated that the use of the chi-square has led to a reduction in the time required for training and testing their proposed model and also improves the generalization performance of their classifier. They concluded that the use of chi-square has helped to solve the phenomenon of the curse of dimensionality associated with large data. The weakness however is that since chi-square is a filter technique it has all the challenges associated with it as stated earlier.

Information gain has been used extensively by several Intrusion detection system researchers including (Gadal & Mokhtar, 2017). As with previous researchers, the results of the study clearly show that applying feature selection could help reduce the problem of false alarms and low detection accuracy. For instance, Mazumder et al. (2020) reported the accuracy of their proposed models before applying feature selection as follows: Decision Tree, AdaBoost, XGBoost, Random Forest, Gaussaian, Naïve Bayes: 75.93%, 76.42%, 75.78%, 75.61%, 79.023%, 77.55% respectively and after applying feature selection with Information gain, XGBoost increased to 76.72% and Random Forest to 75.65%.

2.1. Autoencoder

Autoencoder is a deep learning technique that receives data as input and produces an output that is similar to the input. It carries out its functions in two main stages: the encoding phase and the decoding phase. The encoding is where the data is compressed into a lower dimension known as a bottleneck. The decoding phase converts the bottleneck into a higher dimension that is similar to the input. Several researchers have taken advantage of the dimensionality reduction ability of autoencoders to improve their IDS model performance. These researchers applied different datasets to train and test their models.

Tao et al. (2016) put forward an intrusion detection system that uses autoencoders as a dimensionality technique and fusion approach which is based on the Fisher score. KDD Cup 99 dataset was used for training and testing their model. The results of their study suggested that applying an autoencoder as a feature selection technique had contributed to increasing their model's performance considerably. They specifically implemented and tested the accuracy of classification algorithms such as SVM, backpropagation, neural network and J48. The results of this model showed an increase in detection accuracy and reduced false alarm rate. The weakness of their proposed system is the use of old datasets such as KDD Cup which does not contain the new types of attacks. In line with this view, Siddique et al.(2019) stated that KDD Cup 99 which was perfect for evaluating network intrusion detection systems in the 90s and early 2000s has duplicate records as its shortcomings and therefore makes it inappropriate for evaluating NIDS models. Due to these shortcomings, researchers in the field of intrusion detection systems shifted to the use of the NSL-KDD dataset. NSL-KDD according to Tavallaee et al.(2009)has its redundancies removed for both the training and testing set.

In the same way Gavel et al. (2021) developed a network intrusion detection system with a symmetric deep autoencoder proving the feature selection functionality. They used the random forest as a classification algorithm. Their model was evaluated using two different types of datasets namely KDD Cup 99 and NSL-KDD dataset. Their experimental results attained an accuracy of 85.42% for KDD Cup 99 and 97.85% for the NSL-KDD dataset. Another study evaluated their model using the NSL-KDD dataset with a sparse autoencoder as a feature selection method is (Al-Qatf et al., 2018). In their study, a support vector machine algorithm was used to build an intrusion detection system model after the feature selection by the autoencoder. An f-score of 85.25% was obtained.

In the domain of IOT, Dutta et al. (2020) used a deep space autoencoder for a dimensionality reduction technique, a Deep Neural Network(DNN) and an LSTM and logistic regression as classifiers. The performance of their model was tested using IoT-23, LITNET-2020, and NETML-2020 which are common datasets found in the IoT environment. Evaluation results from their experimental study reveal an improvement in performance in terms of detection accuracy.

Apart from the autoencoder performing the function of dimensionality reduction, they can also be used as classifiers. In designing an intrusion detection system with an autoencoder as a classifier, Sadaf & Sultana(2020)proposed the use of an autoencoder as a classifier followed by an Isolation forest as an outlier detection technique. The strength of their proposed IDS system was an improved performance in metrics of evaluation but the weakness of their proposed system is their failure to disclose the performance of their autoencoder alone that serves as input for the isolation forest.

3 Methodology

We proposed anomaly-based IDS for computer network environments to protect sensitive data from losing its confidence, availability, and integrity. We intend to achieve this aim with our proposed model using Correlation-based feature selection and autoencoder (C-AE). A base model consisting of classical autoencoder (AE) without feature selection is first implemented. Here, the autoencoder receives the preprocessed NSL-KDD and CIC-IDS2017 datasets classifies it into normal and attack. Correlation-based Feature Selection (CFS) and Autoencoder (CFS-AE) are implemented next. First, we use a correlation-based feature selection method to select an efficient set of features. The second phase is the classification stage, where the autoencoder is trained with the selected dataset traffic to differentiate between normal traffic and an attack. The autoencoder as a dimensionality reduction technique can produce better IDS performance by applying the feature selection technique (correlation-based feature selection technique), which will further increase the model performance. Figure 1 below shows the flow of the proposed system.

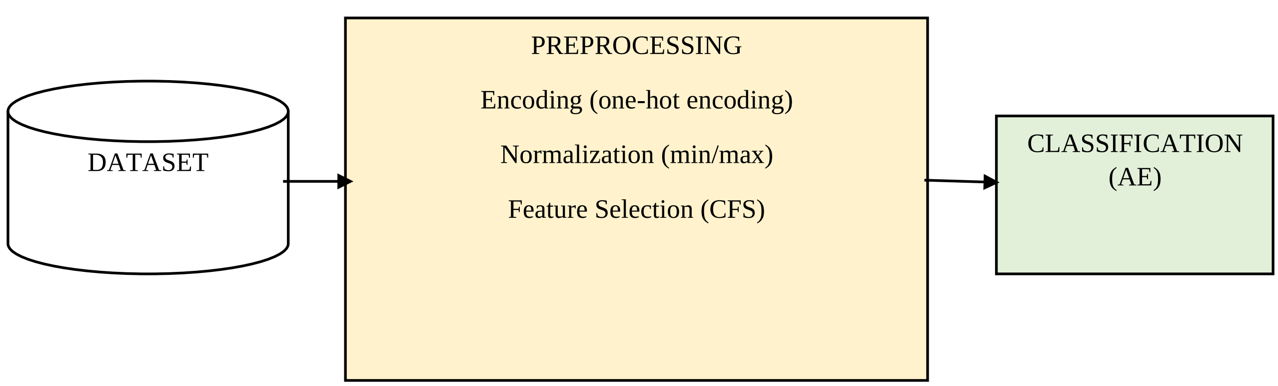


Figure 1: Flow of CFS-AE IDS

3.1. Autoencoders

Autoencoders are unsupervised feature selection techniques that accept input and produce a low-dimensional version. The low-dimensional version, the bottleneck, can reconstruct a similar version of the input. The basic autoencoder consists of the encoder, the process between the input and the bottleneck. The role of the encoder is to map a high dimensional vector H = (h1, h2, h3, h4,…..hn) to a lower dimensional vector L = (l1, l2, l3, l4,….ln). The lower dimension represents a compressed version of the input, which is known as the bottleneck. Wang et al.(2015) established that the encoding phase can be formulated as follows.

$$L = f\,(WH+b) \qquad (1)$$

Also, the decoder represents the process between the hidden and output layers (Sadaf & Sultana, 2020). The encoder task is to take the lower dimensional representation, the bottleneck L, and reconstruct a high dimensional representation similar to the input H. The decoder is represented as follows.

$$X = g(W'L+b') \qquad (2)$$

From the above equations:

F and g are activation functions

W denotes the weight of the encoder

b denotes the encoder's bias

W' denotes the weight of the decoder

b' stands for the bias of the decoder.9ycfd

3.2. Preprocessing

One-Hot-Encoding

The autoencoder used in these studies cannot directly process the NSL-KDD and CIC-IDS2017 datasets without resulting in errors. One-hot encoding is applied to convert non-numeric or categorical features into numeric ones before applying an autoencoder. NSL-KDD has 38 numeric and three non-numeric features. The three non-numeric categories are "Service," "Flags," and "protocol-type." These 3 non-numeric features are first converted into numeric features using the one-hot-encoding. The encoding is carried out as follows.

Step 1:

The protocol type has icmp, tcp,udp as attributes and, therefore, encoded into numeric features as seen in Table 2.

Table 2: Encoding NSL-KDD dataset

TCP	UDP	ICMP
1	0	0
0	1	0
0	0	1

Step 2:

Service has 70 non-numeric features, and the flag has 11 non-numeric features that are all converted onto numeric features using the same approach as in step 1. After encoding, the service is mapped to 70 distinct dimensional features, and the flag is also mapped to 11 distinct dimensional features. A total of 122 features were obtained after the encoding. CIC-IDS2017 dataset, one of the latest IDS datasets resembling real-world network data. Seven files containing benign and attack were consolidated into one file for preprocessing following the same procedure above.

3.3. Normalization

The datasets are first normalized to enhance the performance and reliability of our model by converting all numeric columns to a common scale. Equation three (3) below shows how the min-max technique performs the normalization task.

$$y=x-min/max-min \tag{3}$$

where

y = new value of each entry

Min = minimum value for each data point

Max = maximum value for each data point

3.4. Correlation-based Feature Selection

According to Wang et al. (2015) correlation-based feature selection is used to select the best features by integrating with search algorithms such as backward elimination, forward selection, genetic search, directional search, and best-fit search, and it is also the most widely used feature selection technique. The mathematical representation of correlation-based feature selection is shown below.

$$R_{xc} = \frac{\mathrm{n}\mathrm{R}_{xi}}{\sqrt{n+n(n-1)R_{ii}}} \tag{4}$$

where R_{xc} represents the correlation that exists between the summed features subsets and class variables, n is the features subset, Rxi is the average correlation that exists between the features subset and the class variables, and Rii is the inter-correlation average that exists between attributes subsets (Zhou et al., 2020).

4 Experimental Setup

Our study used a 1.10GHZ Intel(R) Celeron(R) N4020 processor with 4GB RAM on the Windows 10 Operating System. The model was developed using several packages, Tensorflow, Pandas, Numpy, Matplotlib, and Sckitlearn, from a Python programming language. The system is developed using the Google Collaboration platform. Complete detail of the experimental setup is shown in Table 3.

Table 3: Specification for the Implementation Environment

Element	**Description**
Packages	Numpy, pandas, Matplotlib, Sklearn, Keras, Tensorflow
RAM	4GB
Operating System	Windows 10
Processor	1.10GHZ Intel(R) Celeron(R) N4020 processor
Development environment	Google Collaboration

4.1. Metrics of Evaluation

We evaluated and compared the performance of our proposed system based on precision, accuracy, F1-score, and Recall.

These metrics are obtained from four main elements of Confusion Metrics.

True positive: correctly classified attacks in a data sample

True Negative: Normal traffic in a data sample that has been correctly classified as Normal

False positive: Normal traffic in data sample wrongly classified as an attack

False negative: Malicious traffic in a data sample that has been wrongly classified as Normal

The metrics are calculated as follows: **Accuracy measures:** the total number of data samples correctly classified as true positive or negative. A higher accuracy for the balanced dataset is an indication of good performance. Equation 5 below shows how accuracy is calculated.

$$\textbf{Accuracy (ACC)} = \frac{TP+TN}{TP+TN+FP+FN} \tag{5}$$

Recall, also called true positive rate, is the proportion of correctly predicted positive instances of a class to the overall instance of the same class.

A higher recall rate that ranges from 0 to 1 indicates a better model performance. Equation 6 below shows how the Recall is calculated.

$$\textbf{Recall} = \frac{TP}{TP+FN} \tag{6}$$

Precision is the ratio of positive instances correctly predicted to the ratio of all predicted samples for a class. Recall and Precision are always paired when evaluating model performance. Equation 7 below shows how the Precision is calculated.

$$\textbf{Precision} = \frac{TP}{TP+FP} \tag{7}$$

F1-score is computed by taking the harmonic mean of precision and recall. F1-score normally calculates the tradeoff between precision and recall. F1-score is calculated as shown in equation 8 below

$$\textbf{F1-score} = 2 * \frac{Precision*Recall}{Precision*Recall} \tag{8}$$

NSL-KDD as a dataset contains many unnecessary features for an accurate intrusion detection system. Our model addresses this challenge by selecting features using the correlation-based feature selection. These reduce the number of features and avoid the curse of dimensionality. The second issue addressed in our experimental setup is handling the imbalance of the NSL-KDD dataset. An autoencoder as a classifier can handle the issue of an imbalanced dataset of NSL-KDD to some degree. One-hot encoding is then applied to the dataset, followed by normalization. After normalization, the dataset is split into training and testing sets in the ratio of 80%: 25%. The detail of the proposed system is shown in Figure 2.

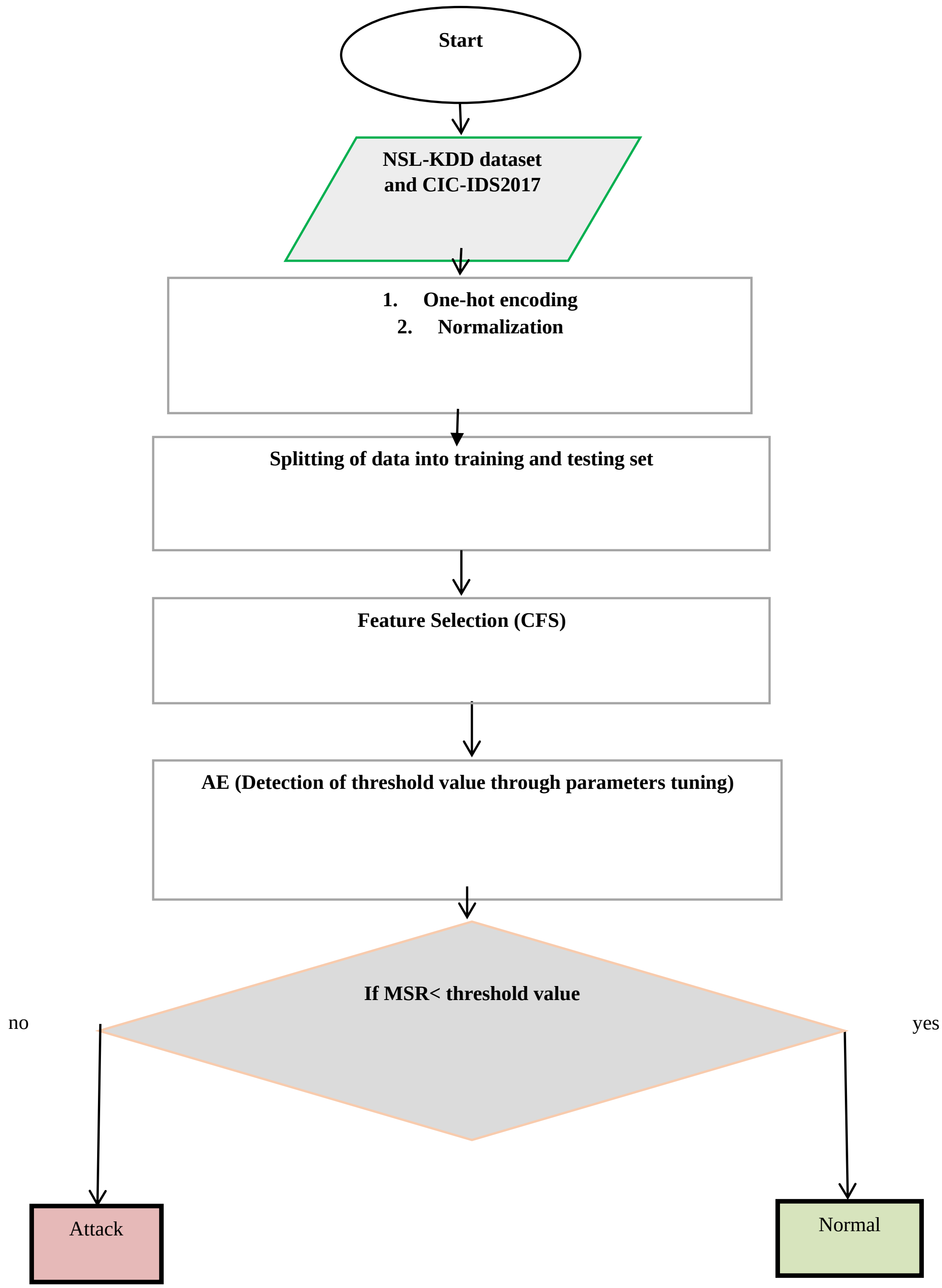


Figure 2: IDS based on correlation-based FS and AE

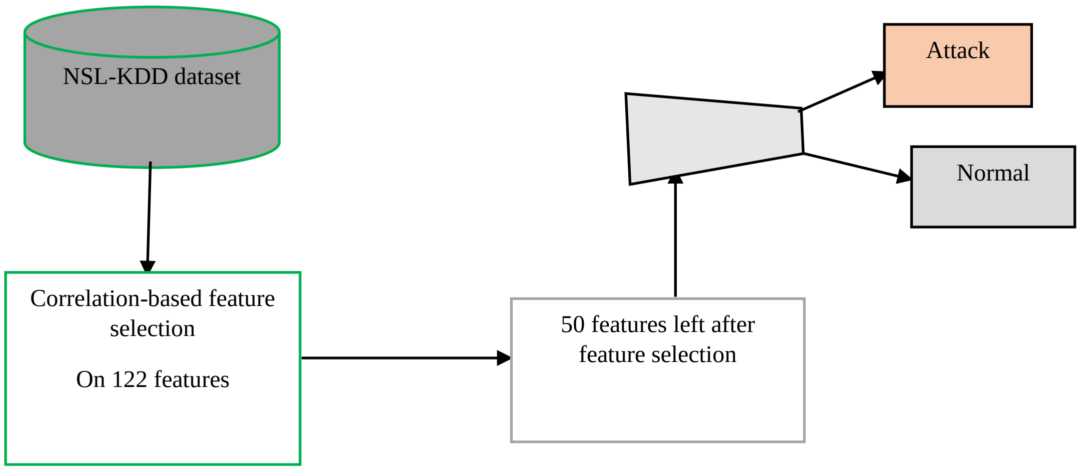


Figure 3: Feature Selection and Classification for the NSL-KDD Dataset

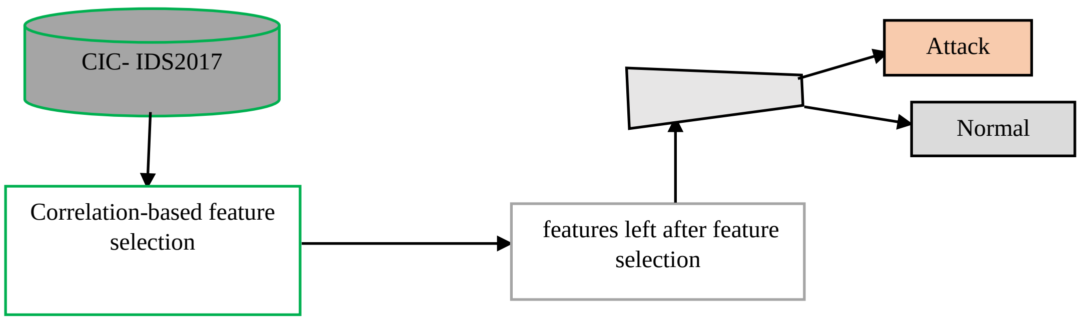


Figure 4: Feature Selection and Classification for CIC-IDS2017 Dataset

The Correlation-Based Feature Selection reduced redundant and irrelevant features from 122 to 50 that serve as input for the autoencoder. We achieve this by

1. Initially eliminating the high feature-to-feature correlation and maintaining a subset with a low feature correlation.
2. The selected features are further prone down to 50 based on the features with a strong correlation with classes.
3. A similar process is also carried out to reduce the number of features in the CIC-IDS2017 dataset from r 82 features to 38.

Figures 3 and 4 above show the processes involved from preprocessed dataset to classification. The base model and the proposed system were set up and trained to obtain data to determine the impact of CFS on the performance of the autoencoder using specific parameters. These parameters included 100 epochs, batch size of 500, and splitting of data into the ratio 25 for testing and 75 for training. The activation functions used in the study include the relu and softmax activation functions.

5 Results and Discussions

This section offers the experimental results based on the guidelines listed below.

1. The performance of the autoencoder without feature selection (AE)
2. The performance of the proposed system when feature selection was applied (CFS-AE)
3. The performance of the model with the changing number of neurons for the autoencoder

The proposed system without feature selection (AE) exhibited the same behavior as compared to CFS-AE when the number of neurons in the encoding phase was increased. For example, consider the suggested system's accuracy in the absence of feature selection for the input neurons. 50,100,150,200,250 and 300 were 88%, 89%, 90% 91%, 91%, 91% respectively as show in Figure 5. Figure 5 clearly shows that the performance of the model in terms of accuracy increased with an increasing number of neurons and remains steady at a point such that a further increase does not affect the accuracy of the model.

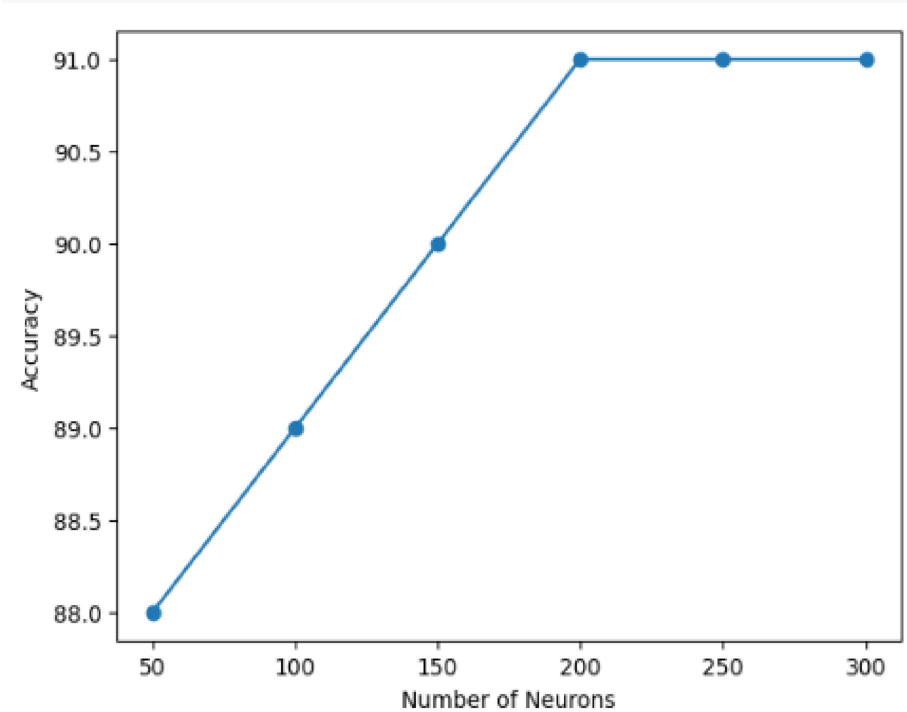


Figure 5: IDS Accuracy vr Number of Neurons for AE

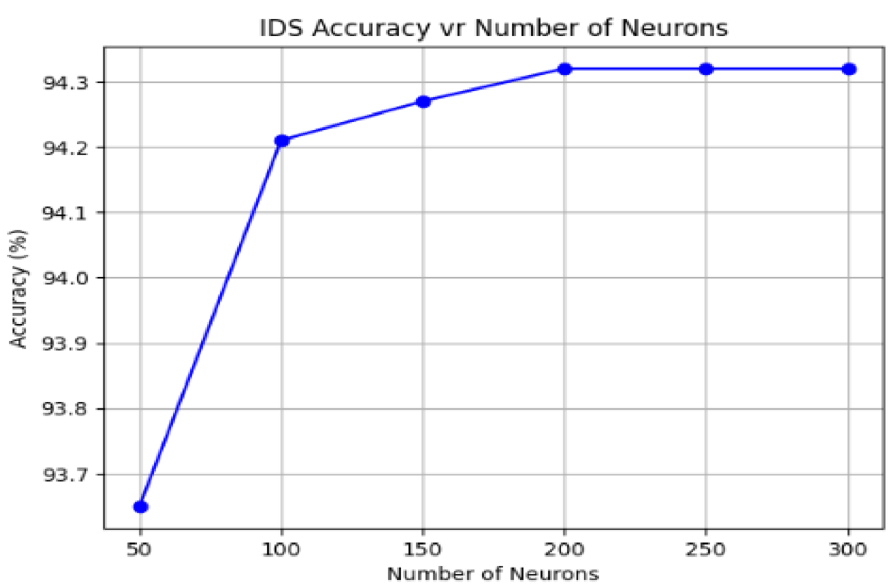


Figure 6: IDS Accuracy vr number of neurons for CFS-AE

Similarly, as in AE, the results of our proposed system, thus CFS-AE, reveal interesting observations starting from the preprocessing, training, and testing stages. Feature selection played a major role in the results obtained in our study in that, without feature selection, the autoencoder recorded low performance in terms of accuracy; however, when feature selection was carried out, there was an increase in performance. During the training stage, it was observed that the activation function used at the encoding stage influences the model's performance. Using a nonlinear activation function, such as tanh or sigmoid leads to degraded performances. Conversely, employing a linear activation function, such as linear or relu activation functions enhances the model's performance. Nevertheless, the relu activation function outperformed the linear activation function.

Similarly, the activation function of the decoder layer worked best with nonlinear activation functions such as sigmoid, softmax, tanh, etc. It has also been observed that increasing the number of neurons increases the detection accuracy, as shown in Figure 6. For instance, when 50 neurons were used, the highest accuracy recorded was 93.65%; however, when 100 was used, 94.21% was recorded, and when the number of neurons was again increased to 150, the accuracy increased to 94.27%. However, when the numbers of neurons were further increased to 200,250 and 300, a constant accuracy of 94.32% was recorded.

This means the number of neurons impacts the performance of our proposed system. Because of this phenomenon, the study settled on 200 input neurons since the best performance is observed around this figure.

5.1. Performance of CFS-AE FOR NSL-KDD and CIC-IDS2017 Datasets

The result from our experimental study registered a superior performance in four main evaluation metrics: accuracy, Precision, recall, and F1_Score. For instance, the detection accuracy was recorded at 94.32%, Precision = 76.00%, F1_score = 86.00%, and recall of 99% for the NSL-KDD dataset and an accuracy of 97.71%, Precision = 89.00%, F1_Score = 95% and recall of 99.70% for CICIDS2017.

The results between NSL-KDD / CFS-AE and CIC-IDS2017 are diagrammatically represented in Figure 7. From the figure 7, it is clear that our proposed model, which is CFS-AE, performed better compared with the based model, which is the classical Autoencoder with feature selection. These results confirmed that the performance of the Autoencoder IDS has been improved with CFS as a feature selection technique. It is also clearly shown in Figure 7 that the CIC-IDS2017 produced the best accuracy, Precision, and F1-Score.

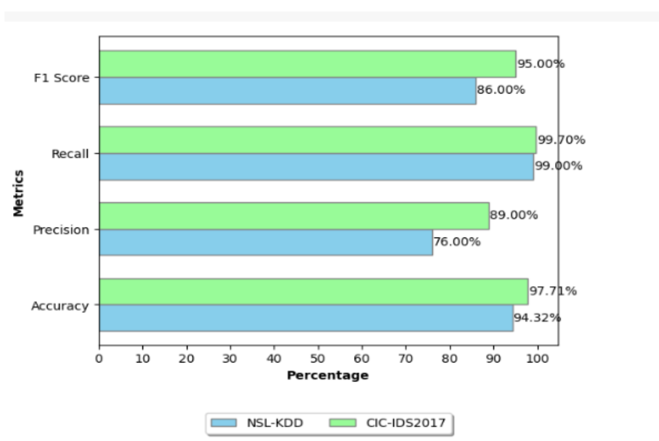


Figure 7: The performance of CFS-AE with NSL-KDD and CIC-IDS2017 datasets

The accuracy of our proposed Intrusion detection system has also been compared with previous state-of-the-art research. The results from Table 4 below vividly show that our proposed system performed better as compared to previous studies that have been considered. However, our study did not consider the model's performance for a multiclass classification.

Table 4: Accuracy of Proposed System Compared to Similar Previous Systems

Paper	**Dataset**	**Method**	**Accuracy**
(Ieracitano et al., 2019)	NSL-KDD	AE	84.24%
(Xu et al., 2021)	NSL-KDD	AE	90.61%
(Singh & Jang-jaccard, 2022)	NSL-KDD	MSCNN+LSTM+AE	93.30%
(Song et al., 2021)	NSL-KDD	AE	84.00%
Our Proposed Model	**NSL-KDD**	**CFS-AE**	**94.32%**

5.2. Implication of the Results

Intriguingly, the autoencoder-based IDS has practical uses due to its high precision and recall. The measurements show that the system can accurately recognize and classify attacks, making it a valuable tool in cybersecurity. Organizations can integrate this technology into their security infrastructure to enhance threat detection capabilities. Examples of useful applications include real-time network traffic monitoring, prompt response to potential threats, and integration with more comprehensive cybersecurity frameworks to create a robust defense system.

The improved accuracy on the CIC-IDS2017 dataset demonstrates how well the model reduces false positives. It lessens needless hiccups or alerts for regular network operations, improving the system's acceptability and usage in practical situations.

The model can potentially reduce computational costs due to the dimensionality reduction by the autoencoder encoder and the CFS technique employed in this study. Apart from that, other factors that make the proposed system computationally inexpensive are parameter sharing by the autoencoder for both the encoding and decoding phase, the autoencoder's ability to work with large unlabeled data samples, and parallel processing by the neural network of the autoencoder. The proposed system's low computational cost also implies that it is scalable because it can handle the high volumes of data traffic coming from current computer networks. The CFS, a filter method, does not depend on the learning algorithm and is, therefore, less computationally expensive. The practical application of our proposal in an actual environment is realized in the following steps:

1. **Dimensionality Reduction**: Autoencoder through CFS-AE helps to improve the generalization of our proposed system by reducing the dimensionality of incoming computer network traffic and, therefore, makes our model computationally efficient
2. **Feature Selection**: CFS in our proposed model selects an efficient set of features from incoming network traffic, and this helps the model to discriminate between normal and malicious traffic. This component enables the model to handle the high volumes of data traffic at a computer network.
3. **Anomaly detection**: Autoencoder is a powerful tool for anomaly detection by first modeling a normal environment and, based on that environment, identifying deviation as malicious or attack. The ability to perform this modeling correctly makes the model adaptable in a changing network environment.

4. **Improved Performance**: Dimensionality reduction and feature selection focus on only a relevant set of features, and this helps to improve the model's performance. Once the model's performance is improved, it can accurately detect intrusion, and the rate of false alarms is reduced. The performance of our proposed system, as shown in Figure 6, indicates that it will accurately detect intrusion when implemented in an actual network.

The effectiveness of the current intrusion detection system for handling the increasing network complexity and data volumes can only be achieved by a scalable and adaptable network IDS. The results of this study address these needs as stated below:

1. The high accuracy of 97.71% and a precision of 89% indicates scalability because a robust and accurate system is expected to maintain its effectiveness as data volumes increase.
2. On the CIC-IDS2017 dataset of the proposed system, the high recall (99.7%) and F1 score (95%) show a great capacity to identify true positive cases while keeping a decent balance between precision and recall. This indicates that the system can tolerate a rise in positive instances without losing its capacity to recognize them, which is significant for scalability.

6 Conclusion and Future Works

A proposed strategy for increasing the performance of intrusion detection systems has been proposed in this study. One major problem of intrusion detection system research is the presence of irrelevant and redundant features in the dataset that affect classification accuracy. To overcome this challenge, a feature selection technique known as the Correlation-based Feature Selection technique is used alongside an autoencoder as a dimensionality reduction technique and a classifier.

Our study then measured the performance of our proposed system in terms of accuracy, recall, precision, and F1 score. The experimental results reveal that increasing the number of neurons increases the classification accuracy until a point is researched where the model has fished learning. A further increase in neurons does not affect the model performance. Another observation is that the activation function used at the encoding and decoding phase of the autoencoder affects the model performance.

Our model also registered a superior performance compared to similar research on the problem. However, in the future, the issue of imbalance dataset and outlier detection needs to be handled to increase the performance of existing IDS further. Future studies should also consider multiclass classification since this study only looked at binary classification.

7 Funding

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

8 Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] Alhasan, S., Abdul-Salaam, G., Bayor, L., & Oliver, K. (2021). Intrusion Detection System Based on Artificial Immune System: A Review. *Proceedings - 2021 International Conference on Cyber Security and Internet of Things, ICSIoT 2021*, 7–14.

[2] Ali, M.H., Fadlizolkipi, M., Firdaus, A., & Khidzir, N.Z. (2018). A hybrid particle swarm optimization-extreme learning machine approach for intrusion detection system. *In IEEE Student Conference on Research and Development (SCOReD)*, 1-4.

[3] Al-Qatf, M., Lasheng, Y., Al-Habib, M., & Al-Sabahi, K. (2018). Deep Learning Approach Combining Sparse Autoencoder with SVM for Network Intrusion Detection. *IEEE Access, 6*, 52843–52856.

[4] Dutta, V., Choraś, M., Pawlicki, M., & Kozik, R. (2020). A deep learning ensemble for network anomaly and cyber-attack detection. *Sensors*, *20*(16), 1–20.

[5] Gadal, S.M.A.M., & Mokhtar, R.A. (2017). Anomaly detection approach using hybrid algorithm of data mining technique. *In International Conference on Communication, Control, Computing and Electronics Engineering (ICCCCEE)*, 1-6.

[6] Gavel, S., Raghuvanshi, A.S., & Tiwari, S. (2021). A novel density estimation based intrusion detection technique with Pearson's divergence for wireless sensor networks. *ISA transactions*, *111*, 180-191.

[7] Ieracitano, C., Adeel, A., Morabito, F.C., & Hussain, A. (2020). A novel statistical analysis and autoencoder driven intelligent intrusion detection approach. *Neurocomputing*, *387*, 51-62.

[8] Khraisat, A., Gondal, I., Vamplew, P., & Kamruzzaman, J. (2019). Survey of intrusion detection systems: techniques, datasets and challenges. *Cybersecurity*, *2*(1), 1-22.

[9] Maseno, E.M., Wang, Z., & Xing, H. (2022). A systematic review on hybrid intrusion detection system. *Security and Communication Networks*, 2022.

[10] Mazumder, A.M.R., Kamruzzaman, N.M., Akter, N., Arbe, N., & Rahman, M.M. (2021). Network intrusion detection using hybrid machine learning model. *In International Conference on Advances in Electrical, Computing, Communication and Sustainable Technologies (ICAECT)*, 1-8.

[11] Ren, J., Guo, J., Qian, W., Yuan, H., Hao, X., & Jingjing, H. (2019). Building an effective intrusion detection system by using hybrid data optimization based on machine learning algorithms. *Security and communication networks*, *2019*.

[12] Sadaf, K., & Sultana, J. (2020). Intrusion detection based on autoencoder and isolation forest in fog computing. *IEEE Access*, *8*, 167059-167068.

[13] Saleh, A.I., Talaat, F.M., & Labib, L.M. (2019). A hybrid intrusion detection system (HIDS) based on prioritized k-nearest neighbors and optimized SVM classifiers. *Artificial Intelligence Review*, *51*(3), 403–443.

[14] Siddique, K., Akhtar, Z., Aslam Khan, F., & Kim, Y. (2019). KDD Cup 99 Data Sets: A Perspective on the Role of Data Sets in Network Intrusion Detection Research. *Computer, 52*(2), 41–51.

[15] Singh, A., & Jang-Jaccard, J. (2022). Autoencoder-based Unsupervised Intrusion Detection using Multi-Scale Convolutional Recurrent Networks. *arXiv preprint arXiv:2204.03779*.

[16] Song, Y., Hyun, S., & Cheong, Y.G. (2021). Analysis of autoencoders for network intrusion detection. *Sensors, 21*(13), 1–23. https://doi.org/10.3390/s21134294

[17] Tao, X., Kong, D., Wei, Y., & Wang, Y. (2016). A big network traffic data fusion approach based on fisher and deep auto-encoder. *Information, 7*(2), 1–10.

[18] Tavallaee, M., Bagheri, E., Lu, W., & Ghorbani, A.A. (2009). A detailed analysis of the KDD CUP 99 data set. *In IEEE symposium on computational intelligence for security and defense applications*, 1-6.

[19] Thaseen, I.S., & Kumar, C.A. (2014). Intrusion detection model using fusion of PCA and optimized SVM. *Proceedings of 2014 International Conference on Contemporary Computing and Informatics, IC3I 2014*, 879–884.

[20] Wang, Y., Yao, H., Zhao, S., Wang, Y., Yao, H., & Zhao, S. (2016). Auto-Encoder based Dimensionality Reduction. *Neurocomputing*, *184*(5), 232-242.

[21] Xu, W.E.N., Jang-jaccard, J., Singh, A., & Sabrina, F. (2021). Improving Performance of Autoencoder-Based Network Anomaly Detection on NSL-KDD Dataset. *IEEE Access, 9*, 140136–140146.

[22] Yang, J., Wang, L., Lee, A., & Wan, P.J. (2022). Stepping-Stone Intrusion Detection via Estimating Numbers of Upstream and Downstream Connections using Packet Crossover. *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications (JoWUA)*, *13*(4), 24-39.

[23] Zhou, Y., Cheng, G., Jiang, S., & Dai, M. (2020). Building an efficient intrusion detection system based on feature selection and ensemble classifier. *Computer Networks*, 174. https://doi.org/10.1016/j.comnet.2020.107247

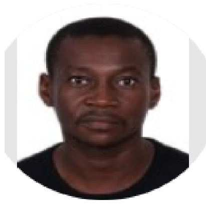
Authors Biography

Seiba Alhassan is currently a PhD candidate at the Kwame Nkrumah University of Science and Technology, Ghana. He graduated from the University of Cape Coast in Ghana with a Bachelor of Science in Information Technology and Kwame Nkrumah University of Science and Technology in Kumasi, Ghana, with a master's degree in Information Technology. He also works as a lecturer at the Dr. Hilla Limann Technical University in Ghana in the department of Information and Communication Technology. His research interest areas are in Computer security, Cybersecurity and Risk and Vulnerability assessment.

Dr. Gaddafi Abdul-Salaam received a Ph.D. degree from the Universiti Teknologi Malaysia, Johor, Malaysia, in 2017. He obtained his M.Sc. degree in Advanced ICT studies from the Institute for Advanced ICT Studies, Ghana, in 2009, and a B.Sc. degree in Computer Engineering from the Kwame Nkrumah University of Science and Technology (KNUST), Ghana, in 2005. He is currently a senior lecturer at the Department of Computer Science, KNUST, and serves as a reviewer for several Web of Science journals.

Professor Asante Micheal is currently a professor of computer science at the Kwame Nkrumah University of Science and Technology in Kumasi, Ghana. From 1992 to 1994, he attended London South Bank University in the United Kingdom, where he earned an MSc degree in Scientific Computing/scientific Information Technology. He then pursued his PhD in Systems Engineering at the University of Reading in the United Kingdom from 2003 to 2007.

Yaw Marfo Missah earned an MSC in Information Technology from Clark University in the United States and a PHD in Computer Science from Colorado Technical University in the United States. He is a Senior Lecturer at KNUST's Department of Computer Science. His research interests include network systems and artificial intelligence applications.



Dr. Ernest D. Ganaa is a lecturer in the ICT department of Hilla Limann Technical University, Ghana and holds a PhD in Computer Science from Jiangsu University, China. Ernest received his BSc. in Computer Science in 2008 and MSc. Information Technology in 2015, both from the Kwame Nkrumah University of Science and Technology (KNUST), Ghana. His research interest includes machine learning, pattern recognition and dimensionality reduction.

Alimatu Sadia Shirazu is currently a student of PhD in Information Technology at the Kwame Nkrumah university of Science and Technology (KNUST), Kumasi, Ghana. She obtained her M.Sc. degree in Information Technology from KNUST in 2021 and the Bachelor degree in Information Studies from the University of Ghana, Accra, Ghana in 2018. She is currently a tutor at the Department of Mathematics and ICT at the E. P. College of Education, Bimbilla. Alimatu Sadia's research interests include Data Networks and Communication, Wireless sensor networks, and Network Security.