# Stacknet Based Decision Fusion Classifier for Network Intrusion Detection

Isaac Kofi Nti
Department of Computer Science and Informatics, University of Energy and Natural Resources, Ghana
isaac.nti@uenr.edu.gh

Owusu Narko-Boateng
Department of Computer Science and Informatics, University of Energy and Natural Resources, Ghana
owusu.nyarko-boateng@uenr.edu.gh

Adebayo Felix Adekoya
Department of Computer Science and Informatics, University of Energy and Natural Resources, Ghana
adebayo.adekoya@uenr.edu.gh

Arjun Remadevi Somanathan
School of Computer Science and Engineering Vellore Institute of Technology, Vellore- 632 014, India
arjun.r@vit.ac.in

**Abstract:** *Network intrusion is a subject of great concern to a variety of stakeholders. Decision fusion (ensemble) models that combine several base learners have been widely used to enhance detection rate of unauthorised network intrusion. However, the design of such an optimal decision fusion classifier is a challenging and open problem. The Matthews Correlation Coefficient (MCC) is an effective measure for detecting associations between variables in many fields; however, very few studies have applied it in selecting weak learners to the best of the authors' knowledge. In this paper, we propose a decision fusion model with correlation-based MCC weak learner selection technique to augment the classification performance of the decision fusion model under a StackNet strategy. Specifically, the proposed model sought to improve the association between the prediction accuracy and diversity of base classifiers. We compare our proposed model with five other ensemble models, a deep neural model and two stand-alone state-of-the-art classifiers commonly used in network intrusion detection based on accuracy, the Area Under Curve (AUC), recall, precision, F1-score and Kappa evaluation metrics. The experimental results using benchmark dataset KDDcup99 from Kaggle shows that the proposed model has a identified unauthorised network traffic at 99.8% accuracy, Extreme Gradient Boosting (Xgboost) (97.61%), Catboost (97.49%), Light Gradient Boosting Machine (LightGBM) (98.3%), Multilayer Perceptron (MLP) (97.7%), Random Forest (RF) (97.97%), Extra Trees Classifier (ET) (95.82%), Different decision (DT) (96.95%) and, K-Nearest Neighbor (KNN) (95.56), indicating that it is a more efficient and better intrusion detection system.*

**Keywords:** *Network intrusion detection, stacknet, ensemble learning classifier.*

*Received April 8, 2022; accepted April 28, 2022*
*https://doi.org/10.34028/iajit/19/3A/8*

## 1. Introduction

Computer applications and network technologies have become an essential part of our daily lives, with organisations and individuals relying on them for data storage and communications such as Person-To-Person (P2P) and Business-To-Business (B2B). Therefore, an excellent and secure computer network solution is vital for business. However, the fears of individuals and organizations are about the security and privacy of their online activities. As various corporations and institutions across the globe continue to be distracted by the new COVID-19, cybercriminals continue to devise more terrifying ways to bring them down [4]. Also, as remote workforces increase globally, forgoing the security of a well-developed IT setup due to the ongoing pandemic, cybercriminals have discernible vulnerable workers as the target of choice. That is, businesses and workers were forced to innovate and adapt to remote working and operate off cloud-based platforms so quickly, leaving security behind, making them vulnerable to the cyber threats spreading across the globe. With the pandemic serving as a catalytic agent, cybercrimes are anticipated to soon become the world's 3rd largest economy [10, 32].

The rise in cyberattacks and threats has been getting much attention recently from academics and professionals in the field. As a result, cybersecurity has become a vital tool for attenuating network intrusion [32]. In the past, using technologies and security policies like firewalls, antivirus, malware programs, and user authentication could have offered enough protection against these attacks [12, 32]. Even more so, modern cyberattacks like exploiting operating system flaws, social engineering, brute-force attacks, phishing, and spear-phishing make it hard for traditional tactics to protect system users [10]. Furthermore, these strategies are vulnerable to current assaults, according to the following research [8, 9, 36], due to their inability to

identify new attacks by learning from previous observations.

An anomaly-based Intrusion Detection System (IDS) functions by detecting any observation, pattern, or behaviour that deviates from the norm. As a result, employing models constructed to classify normal and abnormal events based on prior observations, IDSs may identify unauthorised requests and intrusion [6, 8, 24] Consequently, various resources, new technologies, and techniques, such as artificial intelligence and machine learning, are being deployed to secure existing Internet-based networks from potential assaults or aberrant activity [6, 9, 12, 23, 24, 36]. For example, Jiang *et al.* [13] suggested a deep hierarchical network for intrusion detection with a classification accuracy of 77% to 83.58%. Similarly, in Sornsuwit *et al*. [32], a hybrid artificial neural network was developed for identifying cybersecurity risks, with an experimental accuracy of 99.8%. Fitni and Ramli [8] suggested an ensemble machine learning architecture for anomaly-based IDSs and obtained accuracy, recall, F1 (97.9%), and precision of 98.8%, respectively. Other studies, such as [6, 33], have attempted to identify network infiltration using existing technology, with promising results

Despite the current enhancements in IDS performance as discussed above, Injadat *et al*. [12] argue that there is more room for further enhancement. This is especially true given the huge volume, noisy tagged, high dimensionality, and class unbalanced nature of real-world network traffic data. To put it another way, the traffic data contains millions of samples that are unevenly distributed, with infrequent abnormalities and too much typical traffic data. Furthermore, having a large number of features, i.e., having unwanted or inconsequential characteristics, might have a detrimental impact on a network intrusion detection system's detection capability, since it delays the model training process [12]. Furthermore, due to the high imbalance nature of the network traffic dataset, several studies [24, 33, 39] sought to build Multiple Classifier Systems (MCS) or ensemble models for network intrusion detection. However, ensemble learning has been shown to outperform individual classifiers in the literature [36].

However, this is not always the case since it depends on many things, such as base classifiers, voting methods, etc. Notably, the base learners in most of these experiments were chosen randomly. The question is whether the chosen base learners impact the prediction model's overall accuracy.

Our proposed method [22], used Mathews correlation coefficient based model for network intrusion detection. The findings showed 99.73% accuracy. In the current study, we extend this by executing Support Vector Machine (SVM) and deep neural network Multilayer Perceptron (MLP) algorithms and comparing the quantitative results of simulation. The overall contribution of this work is summarised as given below:

1) An experimental comparison of single classifiers, deep learners, and ensemble learners for network intrusion detection.
2) An extension of [22] uses a deep decision fusion stacknet classifier to enhance the classification power of the ensemble learner for network intrusion detection.
3) A novel mix of One-Side Selection (OSS) and the Synthetic Minority Over-Sampling Method (MOTE) was used to lower the majority instances and raise the minority examples.

We organised the remaining section of this paper as follows: Section 2 explains the review of literature relevant to the proposed work. Section 3 describes the methodology adopted and experimental setup. Section 4 explores the results of study and discussion. Finally, section 5 ends with the conclusion and future scope of research.

## 2. Related Studies

We review some of the related research in the following section; grouping them into two main categories i.e., machine learning techniques used:

1. Single and hybrid classifiers
2. Ensemble learning.

### 2.1. Single and Hybrid Classifiers

Different Decision Tree (DT) algorithms were adopted for network intrusion detection and report moderate classification accuracies [7]. A SVM predictive model was proposed [17] and applied to network traffic data collected with a modern honey network. Ajdani *et al.* [1] proposed an SVM based classifier for detecting network intrusion. Anomaly-Based intrusion detection with SVM was presented in [16]. Likewise, K-Nearest Neighbor (KNN) has been applied extensively for detecting network intrusion [15]. Hybrids techniques for identifying malicious traffic in networks has have been suggested. A genetic algorithm combined with a neural network for increasing classification accuracy in network intrusion detection systems [37]. In another work [18], a combination of DT and Particle Swarm Optimization (PSO) was used as a network instruction detection model, and experiments were carried out on the KDD99Cup dataset.

Singh *et al*. [31] proposed a cross-layer based model for detecting wireless network intrusion. While boosting the packet delivery ratio, their suggested approach achieved a low misdetection ratio and false positive rate. Similarly, Khan *et al*. [14] suggested a novel paradigm for wireless mesh network intrusion detection systems. An intrusion detection model based on the KNN classifier and fuzzy rough set feature selection was proposed by Senthilnayaki *et al*. [29]. The study's findings revealed that the suggested feature selection and classification algorithms are particularly

successful in detecting assaults and reducing false alarm rates. In Tabash *et al*. [35], a robust network intrusion detection system was built using two machine learning techniques (Naive Bayes and deep learning). According to the study, their results outperformed single- and hybrid-model outcomes.

In the same way, a combination of SVM and RF was utilised in detecting network intrusion [30]. Recently, network intrusion detection based on deep learning techniques has been proposed in several studies. For example, a deep neural network intrusion detection framework based on a combination of bidirectional long-term memory (BLSTM) and attention mechanism was proposed [34]. The proposed model was broadly assessed on the NSL-KDD dataset and achieved a classification accuracy of 84.25%. Similarly, Vinayakumar *et al*. [38] proposed a convolutional neural network-based network instruction detection model and experimented on KDDCup 99 with an accuracy of 97%. Finally, for intrusion detection, a behaviour profiling and statistical approach model were applied [5].

From the above, it is evident that machine learning algorithms such as SVM, DT, RF, and KNN have been extensively used in literature for detecting network intrusion [1, 3, 7, 15, 16, 17, 25, 26, 30, 40, 43]. However, shallow learning techniques often emphasise feature optimisation [30, 37]. Consequently, they have issues with feature selection. Hence, it cannot efficiently unravel the huge intrusion data classification problem as a single model, leading to low recognition accuracy and a high false alarm rate [34]. Therefore, finding an effective predictive model with low false-positive rates and high efficiency is still the focus of current work in this field.

### 2.2. Ensemble Learning

In most cases, various models provide varied detection rates when faced with an intrusion threat. In such instances, integrating many models rather than employing a single model in isolation can frequently result in improved prediction; Ensemble Learning (EL) is the method of training numerous models individually and merging their multiple outputs. Decision fusion is used in ensemble learning to combine the "decisions" of numerous base learners into a single "decision" concerning the forecasting goal. Fusion refers to combining data or information from several sources. Fusion can be divided into three levels of hierarchy: data fusion, feature fusion, and decision fusion. As a result, decision fusion refers to the merging of various projections provided by base learners.

Fusion for network intrusion detection market prediction entails a variety of methodologies, including artificial intelligence and data fusion, and there is no one-size-fits-all strategy to using fusion techniques. Because of the wide variety of network intrusion detection models, the mechanism for decision fusion is generally chosen based on the detection job and personal taste. Nonetheless, the basic idea is that the final forecast should be based on the situational knowledge that has been seen. Only by fully using various predictors can more meaningful intrusion detection be achieved at the decision level.

Bagging (bootstrap aggregation), mixing, boosting, error-correcting output coding, stacking, and other approaches can all be used to create an ensemble. EL has been employed in numerous sectors to obtain optimal accuracy due to the accuracy it provides when compared to single classifiers. Several studies in network intrusion detection have adopted ensemble learning techniques (see Table 2). Table 2 shows that most previous research used heterogeneous base learners who were chosen randomly.

On the other hand some significance tests, on the other hand, must be treated as a neutral benchmark comparing classification algorithms employed as base learners for ensemble classifiers [36]. Many well-known intrusion datasets, such as KDDcup99 and UNSW-NB15, are also unbalanced. As a result, it has an influence on machine learning performance, necessitating a statistical analysis of the performance significance of many base learning prior to ensemble. As a result, these tests have become routine in machine learning investigations, which often involve a large number of algorithms. In Table 1 Summarizes relevant studies for intrusion detection based on ensemble learning approaches.

## 3. Methodology

The proposed model's framework is shown in Figure 1, and the framework's succeeding subsections are described in depth.

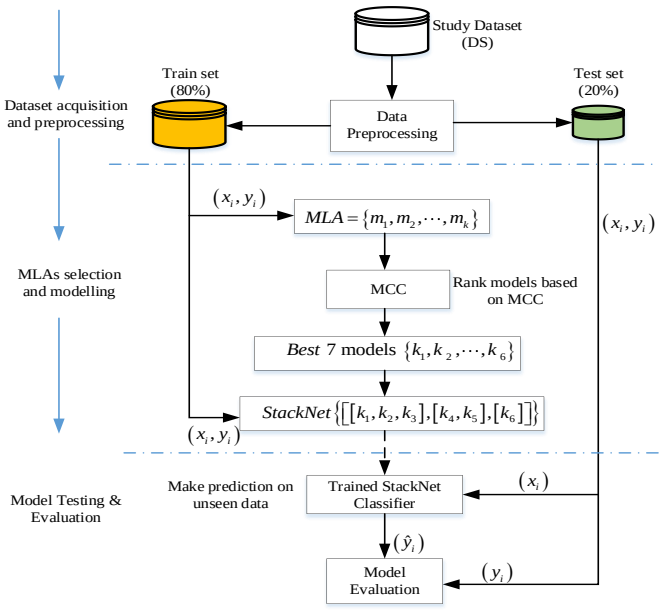


Figure 1. Conceptual framework of the proposed (MCC-StackNet) network intrusion-detection model.

Table 1. A Summary of related studies based on ensemble learning methods for intrusion detection.

| Ref | Base learners | Base learners' selection technique | Ensemble technique | Dataset | Metrics |
|---|---|---|---|---|---|
| [39] | Bayesian network and Random Tree | Based on literature | Simple voting | KDDcup99 | AUC 0.995 – 0.999 |
| [33] | Naïve Bayes, Decision Tree, SVM, Neural Network and KNN | Randomly | Adaboost | KDD Cup99 | Sensitivity = 0.7600 and specificity = 0.9905 |
| [24] | KNN, RF, Logistic regression and SVM | Randomly | Stacking | UNSW-NB15 | Acc = 0.94, precision = 0.96, recall = 0.93, AUC = 0.99, F1 score = 0.95 |
| [38] | Bayesian ridge, RF, Extra trees, Gradient boosting machine | Randomly | Stacking | SNMP-MIB | Acc = 99.3% |
| [40] | decision trees | Randomly | Bagging | KDD99 | Acc = 98.49% |
| [22] | RF, Catboost, ET, Lightgbm, Xgboost DT, KNN | Mathew's correlation coefficient | StackNet | KDDcup99 | Acc = 99.7% |
| Our study | RF, Catboost, ET, Lightgbm, Xgboost, DT, KNN and MLP | Mathew's correlation coefficient | StackNet | KDDcup99 | Acc = 99.8% |

*Acc = Accuracy

## 3.1. Dataset Acquisition and Preparation

We use Kaggle's publicly accessible intrusion detection dataset in this research.

It comprises a variety of network intrusion scenarios that are simulated on a Local Area Network (LAN) to get raw TCP/IP dump data. Multiple known intrusion attacks were used to target the LAN design, which was designed to simulate a genuine environment.

Data flow from a source IP address to a target IP address that follows a well-defined protocol is classified as either normal or an attack with a single attack type. Normal and attack data yield forty-one qualitative and quantitative characteristics for each TCP/IP connection (3 qualitative and 38 quantitative features). The class feature has two labels:

1. Normal (1)
2. Anomalous (0).

Thus, the dataset size was (47,736, 42) our dataset was pre-processing by applying cleaning, encoding, scaling techniques and feature selection. Firstly, cleaning refers to handling miscellaneous data, missing values and data inconsistency. During this step of the pre-processing, 251 records were successfully deleted in this research. The eliminated records were, however, insignificant in comparison to the remaining records due to the large data size employed in this study. Second, we reduced the majority examples while increasing the minority examples using two well-known data imbalance correction approaches (OSS and SMOTE). Finally, all nominal data characteristics were encoded using Python's label encoder (i.e., converting all nominal data to numeric form). Fourthly, using the max-mum function as defined in Equation (1), we scale our data between 0 and 1. The aim was to ensure that features having a bigger numeric range do not dominate those in smaller numeric ranges.

Finally:

1. We reduce the noise in our dataset and created an accurate depiction of the dataset by applying the Principal Component Analysis (PCA),
2. Eliminating the feature dimensions with the low variance among subjects, and
3. Pick out 21 independent feature dimensions with the optimal correlations. Thus, our dataset feature dimensions were reduced from 42 to 22.

$$x' = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \tag{1}$$

## 3.2. Model Building

This section presents a brief description of the selected base learners and the decision fusion model.

### 3.2.1. Selection of Base Models

From a literature survey by Salo *et al.* [27] and our partial review of previous studies presented, nineteen (19) MLAs were identified as commonly used for network instruction detection. Out of the 19, we adopted seven (17) machine learning algorithms comprising of single classifiers, deep learners and ensemble learner. Since we deal with some imbalanced datasets, we consider the Matthews Correlation Coefficient (MCC) to statistically examine the performance significance between these 17 algorithms and use the best eight (8) for our decision fusion ensemble classifier.

Thus, to the best of our knowledge, this study, if not the first, is among the few to use MCC metrics in evaluating base learners' algorithms in intrusion detection modelling. We discuss in brief the ten weak learners in earlier below:

1. SVM: this is an optimum margin-based classification approach in machine learning. SVM is a binary linear classifier that has been extended to non-linear data using Kernels and multiclass data using techniques

like one-versus-one, one-versus-rest, Crammer Singer SVM, Weston Watkins SVM and Directed Acyclic Graph SVM (DAGSVM) etc. SVM has been changing since its conception, and academics have presented several problem formulations, solvers, and methodologies for solving SVM.

2. Artificial Neural Networks (ANNs): are computer networks that are inspired by biology. In this paper we adopted the MLPs using backpropagation learning methods, among the numerous forms of ANNs. MLP is a supervised deep ANN that learns a function $f(.):R^m \rightarrow R^o$ by training on a dataset ($DS_{Train}$), where *(m)* is the dimensions of DS and *(o)* is the output dimensions. It contains at least three (3) layers of nodes:
    1. The input layer for receiving the input signal.
    2. The output layer for making a judgment or forecast about the input.
    3. Hidden layer(s) sandwiched between 1and 2 for all MLP calculation; this layer can have an indefinite number of nodes.
3. Nearest Centroid (NC): is one of the most underappreciated and underused MLAs, despite the fact that it is relatively powerful and extremely efficient for certain ML applications. It works in a similar way to the KNN.
4. Ridge Classifier (Ridge): the ridge classifier, based on the Ridge regression method, converts the label data into (-1, 1) and solves the problem with the regression method. The highest value in prediction is accepted as a target class, and multi-output regression is applied.
5. Random Forest (RF): random Forest classifiers are part of the larger category of ensemble-based learning techniques. They are easy to set up, operate quickly, and have a long track record of performance in various fields. The essential principles underpinning the random forest technique are the building of numerous “simple” decision trees in the training stage and the majority vote (mode) across them in the classification stage. Random forests, like nave Bayes and k-nearest neighbour algorithms, are well-known for their ease of use and typically high performance.
6. Quadratic Discriminant Analysis (QDA) and Linear Discriminant Analysis (LDA): LDA and quadratic discriminant analysis are examples of discriminant analysis methods that may be used for both classification and dimensionality reduction QDA. LDA is a dimensionality reduction approach and a classifier, while QDA is a version of LDA that allows for non-linear data separation. Furthermore, Regularised Discriminant Analysis (RDA) is a hybrid of LDA and QDA.
7. Naive Bayes (NB): NB is a graphical Bayesian model with nodes for each column or characteristic. It’s dubbed naïve because it disregards the prior distribution of parameters and believes that all features and rows are independent. NB assumes two things:
    1. that all columns are independent of each other and solely rely on the label,
    2. That all rows are independent of each other.
8. Logistic Regression (LR): a procedure of modeling the likelihood of a discrete result given an input variable is known as logistic regression. The most popular logistic regression models have a binary result (true/false, yes/no, etc). Multinomial logistic regression can be used to model situations with more than two discrete outcomes. Logistic regression is a useful analytical tool for classification issues, such as determining if a fresh sample belongs in a specific group.
9. Light Gradient Boosting Machine (LightGBM): light GBM is a gradient boosting framework based on a decision tree algorithm that may be used for ranking, classification, and a variety of other machine learning applications. It splits the tree leaf-wise with the greatest fit since it is based on decision tree algorithms, unlike other boosting methods split the tree depth or level-wise rather than leaf-wise. As a consequence, while growing on the same leaf in Light GBM, the leaf-wise approach may minimize more loss than the level-wise technique, resulting in significantly higher accuracy than any of the existing boosting strategies. The name 'Light' refers to the speed with which it computes.
10. KNN: the KNN classifier is a supervised machine learning method that addresses classification and regression issues. It's simple to set up and comprehend, but it has the disadvantage of being substantially slower as the amount of data in use rises.
11. Gradient Boosting Classifier (GBC): GBC is a collection of machine learning algorithms that combine a number of weak learning models to produce a more robust prediction model. When conducting gradient boosting, decision trees are commonly employed. Gradient boosting models are gaining popularity as a result of their ability to categorise complex information.
12. Extreme Gradient Boosting (Xgboost): the Extreme Gradient Boosting Method is a decision-tree-based ensemble MLA based on the GB framework, which is an improvised version of the GBM algorithm, similar to the LightGBM. It may be used for both regression and classification machine learning problems. The key distinction between RF and GB Machines (GBM) is that RF builds trees separately, whereas GBM adds a new tree to complement existing ones.
13. Extra Trees Classifier (ET): to get its classification result, this sort of ensemble learning approach integrates the outcomes of several de-correlated

decision trees collected in a "forest." It is conceptually identical to a Random Forest Classifier, with the exception of how the decision trees in the forest are constructed.

14. DT: in the form of human-understandable tree rules, decision trees extract predictive information. A Decision Tree is a useful approach for various classification issues that uses human-readable "If.... Then..." rules to describe the model's rationale.
15. CatBoost Classifier (Catboost): CatBoost is an open-source toolkit for gradient boosting on decision trees that is very fast. It enhances training outcomes by allowing non-numeric elements to be used instead of pre-processing data or wasting time and effort converting it to numbers.
16. AdaBoost Classifier (Ada): ada-boost, also known as Adaptive Boosting, is an ensemble boosting classifier that combines numerous classifiers to improve Meta classifier accuracy. AdaBoost is an iterative ensemble approach for constructing a strong classifier by merging many low-performing classifiers to produce a high-accuracy strong classifier. The primary idea behind Adaboost is to train the data sample and set the weights of classifiers in each iteration to provide accurate predictions of uncommon observations. Any machine learning method that takes weights on the training set can be used as a basic classifier.
17. GBC to create the final forecasts, combine the estimations from numerous DTs. It's one of the most effective MLAs for creating regression and classification models. RF develops an ensemble of deep autonomous DTs, whereas GBM's ensemble shallows DTs progressively, with each DT learning and improving on the previous one.

### 3.3. StackNet Ensemble Classifier Design

There are several decision fusion techniques for combining weak learners. However, in this study, we adopted a unique variant to the stacking technique called StackNet. StackNet [39] is an ML technique that looks like a feed-forward NN; it uses the stacked generalisation technique of Wolpert [41] at numerous levels (layers) to curtail regressor error or enlarge the accuracy of classifiers. However, unlike the feed-forward NN that uses backward propagation for training, StackNet is constructed iteratively one layer at a time (using stacked generalisation), with each layer using the final target as its target. StackNet can be achieved in two ways:

1. Every single layer makes use of the predictions from only one preceding layer,
2. Every single layer makes use of the predictions from all preceding layers in addition to the input layer.

The latter is referred to as restacking. They usually give better accuracy than the single optimal model in each layer. However, their performance always depends on putting together a good mix of heterogeneous base learners to get the best meta-classifier. We adopted EL because they offer several advantages in classification tasks [21]. That is:

1. Better prediction and model stability.
2. EL helps in enhancing classification accuracy by merging many single learners. Thus, using many single heterogeneous learners helps lead to higher prediction accuracy.

Specifically, we adopted the StackNet ensemble for the study based on:

1. Due to the success in literature [2, 19] in different fields.
2. Including more single heterogeneous classifiers that have similar or diverse prediction performance offer better meta-classifiers.
3. Offers the ability to place single learners with optimal performance on a higher layer.
4. The ability to increase diversity in each layer.

Figure 2 shows the architecture of the proposed decision fusion StackNet ensemble classifier. It consists of three (3) layers with eight (8) heterogeneous classifiers, including RF, Catboost, ET, Lightgbm, Xgboost, DT, MLP and KNN. The first layer has two ensembles (Lightgbm and ET) and two single (DT and KNN) classifiers, the second layer contains two ensemble classifiers (CatBoost and RF) and a deep neural network (MLP), and the last layer has one ensemble classifier Xgboost. It's worth mentioning that deep learning models were not considered in our earlier study [22]. However, in this paper, we added an extra node (a deep neural network) to create deeper Stacknets, to improve accuracy.

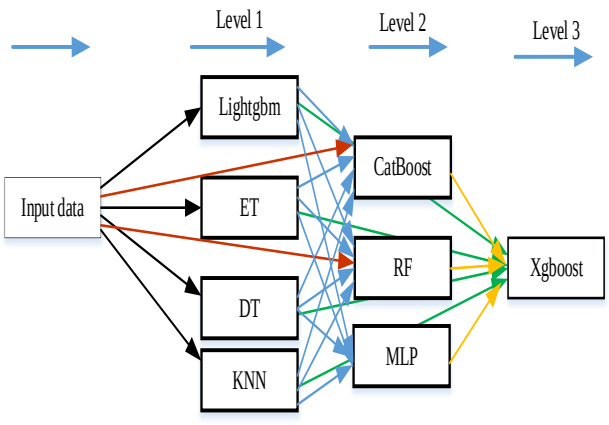


Figure 2. Architecture of the three-layer proposed MCC-StackNet, an ensemble classifier.

*Algorithm 1: Pseudocode of proposed model*

*Setup:*

1. *Input: Training dataset {$DS_{Train}$=($x_i$, $y_i$),i= 1,2, …k} consisting of 21 independent features and one dependent feature*
2. *Output: StackNet model [0, 1]*

*First Phase (Training all initially selected base learners): (Each initially selected classifier participates in the m-fold CV)*

3. *Randomly Divide( $D\ S_{Train}$ ) in M instances in same sizes training ($D_1$, $D_2$, …,$D_M$)*
4. *Let $D_m$ and $D^{(-m)}$ (m= 1,2, …M) represent $m^{th}$ testing and training sets, respectively*
5. *Let L be the initial classifiers and $h_r^{(-m)}$ (r = 1,2, …L) be the $r^{th}$ classifier in $D^{(-m)}$ and $h_r$ ($x_i$) be the corresponding output for ($x_i$) in $D_m$ using classifier $h_r^{(-m)}$*
6. *Get the prediction of each L along with the corresponding actual labels*
7. *Calculate each L MCC's scores and rank*
8. *Rank L concerning their MCC score in increasing order, from the best performer to the worst performer and select top seven (T)*

*Train the proposed ensemble classifier:*

9. *Layer 1: Train each of the first four{$t_1$, $t_2$, …, $t_4$} base learners*
10. *Perform 10-fold cross-validation on each base classifier {$t_1$, $t_2$, …, $t_4$}*
11. *Collect the outcome of their predictions {$t_1$, $t_2$,… ,$t_4$} along with the original label, leading to a new training data set*

$$D_{new1}=\left\{\left(y_1(x_i),t_2(x_i),\cdots t_4(x_i),\hat{y}_i\right),i=1,2,\cdots,k\right\}$$

12. *Layer 2: Train the second layer of the StackNet with $D_{new2}$ and collect the prediction output of laye 2 in addition to layer one along with the original dataset as $D_{new2}$*
13. *Layer 3: Train the final layer with $D_{new2}$*

*Prediction on unseen testing sets:*

14. *Apply the train StackNet on the unseen dataset*
15. *Measure model's performance*

### 3.4. Evaluation Metrics

The current study aimed to improve the prediction accuracy of intrusion detection predictive framework; hence we adopted the MCC as defined in Equation (1) to evaluate the performance of 15 base learners and pick the best seven for ensemble classifier using StackNet. Six (6) well-known evaluation metrics frequently used for network intrusion detection are also considered. They include Accuracy (ACC), F1-score, The Area under Curve (AUC), Precision Equation. (5), Recall and Kappa, as defined in Equations. (2), (3), (4), (5), (6), (7), (8), respectively.

We selected various evaluation metrics because any individual metric is insufficient to detect the model's efficiency For example, according to [28], the accuracy metric in situations where the dataset is skewed could lead to biased results in the performance indicator.

$$MCC=\frac{(TP\times TN)-(FP\times FN)}{\sqrt{(TP+FP)\times(TP+FN)\times(TN+FP)\times(TN+FN)}}\ \in\{-1,1\} \quad (2)$$

Where a value of $-1$ represents an entirely wrong prediction, and $+1$ indicates a perfectly correct prediction. If MCC=0 means that the base learner is no better than random guessing, there is no correlation between the learner's predicted

*(y)* and actual values*(y)*. *TP*=true positive, *TN*=true negative, *FN*=false negative, *FP*=false positive

$$ACC=\left\{\frac{(TP+TN)}{(TP+FP+TN+FN)}\right\}\ \in\{0,1\} \quad (3)$$

It defines the ratio of all correct predictions to the total number of predictions. Where values closer to *1* indicate better accuracy measures.

$$F1-\text{score}=\frac{2\times P\times R}{P+R}\ \in\{0,1\} \quad (4)$$

It displays the evenness among *(P)* and *(R)*, i.e., is the consonant mean *(P)* and *sensitivity (R).*

$$AUC=\int_0^1\left(\frac{TP}{TP+TN}\right)d\left(\frac{FP}{FP+FN}\right)=\int_0^1\frac{TP}{P}\,d\,\frac{FP}{N} \quad (5)$$

It is a graphic picture that shows the intrusion detection accuracy against the false positive rate. It is a well-known evaluation metric used to evaluate intrusion detection systems' performance [38].

$$\text{Precision}=\left\{\frac{TP}{TP+FP}\right\} \quad (6)$$

It measures the ability of a classification model to classify the positive class. A measure closer to one (1) is better.

$$\text{Recall}=\left\{\frac{TP}{TP+FN}\right\} \quad (7)$$

It explains how good a classification model is at classifying the positive class when the actual result is positive. Thus, a recall value closer to one (1) is better.

$$Kappa=\left\{\frac{\left(p_{(\hat{y})}-p_{(y)}\right)}{\left(1-p_{(y)}\right)}\right\} \quad (8)$$

It measures the chance of agreement between the *(y)* and the *(y)* classes, where $p_{(y)}$ is the predicted agreement and $p_{(y)}$ is the expected agreement.

## 4. Results and Discussion

We used the Python programming language and the Scikit-Learn, Matplotlib, pandas, and seaborn libraries to conduct all of the experiments in this work. This study employed a Lenovo (20EGS12E00) Intel® coreTM i5-4340M CPU At 2.90GHz (4 CPUs) with 12GB memory. The results are presented in the next section.

### 4.1. Selection of Base Learners

Table 2 Shows the performance (MCC score) of the fifteen initially selected base learners in this study; each base learner was trained using the 10-fold Cross-

Validation (CV) technique. We rank their performance from the best performer to the worst performer based on MCC score, i.e., a model with MCC score closer to one (1) the better than the one with MCC score close to zero (0). Of the ensemble learners, it was observed that RF, Catboost, ET, Lightgbm and XgBoost were the best ensemble classifiers. Thus, the ensembles gave better MCC scores compared with single classifiers. Furthermore, the DT was more accurate than Ridge, KNN, LDA, LR, SVM, and NB among the single classifiers; this shows why it is commonly used among machine learning practitioners [3, 7, 18] in intrusion detection studies. However, the deep learning classifier (MLP) outperformed all single classifiers used in this study. Also, looking at the DT's training time compared with the MCC scores of models, it can be inferred that the DT is computationally lesser than the top five models.

Table 2. Ranking of initially selected algorithms based on their MCC score.

| S/N | Symbol | Model | MCC | TT (Sec) |
|---|---|---|---|---|
| 1 | RF | Random Forest Classifier | 0.988 | 1.055 |
| 2 | Catboost | CatBoost Classifier | 0.987 | 5.668 |
| 3 | ET | Extra Trees Classifier | 0.986 | 1.029 |
| 4 | Lightgbm | Light Gradient Boosting Machine | 0.983 | 0.215 |
| 5 | MLP | Multi-Layer Perceptron | 0.981 | 2.027 |
| 6 | Xgboost | Extreme Gradient Boosting | 0.98 | 0.463 |
| 7 | GBC | Gradient Boosting Classifier | 0.98 | 0.812 |
| 8 | ADA | AdaBoost Classifier | 0.966 | 0.243 |
| 9 | DT | Decision Tree Classifier | 0.959 | 0.051 |
| 10 | KNN | K Neighbours Classifier | 0.94 | 0.094 |
| 11 | Ridge | Ridge Classifier | 0.936 | 0.253 |
| 12 | LDA | Linear Discriminant Analysis | 0.927 | 0.093 |
| 13 | LR | Logistic Regression | 0.819 | 0.159 |
| 14 | QDA | Quadratic Discriminant Analysis | 0.7 | 0.08 |
| 15 | SVM | SVM - Linear Kernel | 0.295 | 0.048 |
| 16 | NB | Naive Bayes | 0.254 | 0.048 |

Based on the results in Table 3, we selected the top four (4) ensemble learners, a deep learning model (MLP) and the top two (2), single learners, as our base learners for the StackNet ensemble. Table 3 shows the final selected algorithms.

Table 3. Final selected algorithms.

| S/N | Symbol | Model |
|---|---|---|
| 1 | RF | Random Forest Classifier |
| 2 | Catboost | CatBoost Classifier |
| 3 | ET | Extra Trees Classifier |
| 4 | Lightgbm | Light Gradient Boosting Machine |
| 5 | Xgboost | Extreme Gradient Boosting |
| 6 | MLP | Multi-Layer Perceptron |
| 7 | DT | Decision Tree Classifier |
| 8 | KNN | K Neighbours Classifier |

## 4.2. Feature Engineering

Computational complexity in machine learning is an issue of concern in designing a predictive model; to reduce this challenge, the PCA was adopted for dimensionality reduction in this study. Implementing dimensionality reduction enables the MLAs to train faster, improves accuracy if the correct subset is chosen, reduces overfitting, and makes interpreting the model easier. Figure 3 shows the learning curve of the PCA algorithms based on different feature subsets. For example, it can be seen that 20 features gave an accuracy score (99.1%) better than all 41 features combined. Therefore, based on the PCA output, we reduced the feature dimensions of our dataset from an initial value of 41 to 20 (i.e., more than 50% reduction) independent features and fed these features to our proposed model. Thus, it can be said that the computational complexity has indirectly been reduced by 50% approximately.

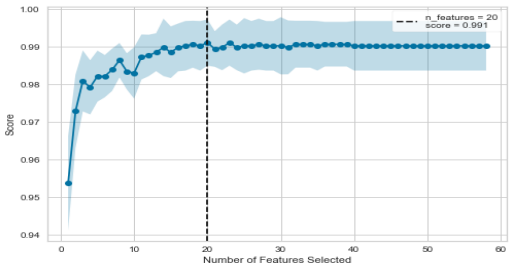


Figure 3. Feature selection with PCA.

## 4.3. Performance of Proposed Decision Fusion Classifier

The accuracy, AUC, recall, precision, F1-Score, Kappa and MCC values of the proposed model are 0.998, 0.9989, 0.9989, 0.9961, 0.9975, 0.9946 and 0.9946, respectively. The experiment was designed to study the classification performance of the proposed model. The prediction error measures how efficient samples are classified to the correct class in classification. Figure 4 shows the prediction error of our proposed model. Figure 5 summarises the prediction results (confusion matrix) of the proposed classifier. The results show that most of the class labels were adequately found by the proposed method. It also shows how effective the proposed decision fusion classifier is at detecting regular and aberrant network traffic. Figures 6, 7 depict the proposed model's ROC curve and precision-recall curve, respectively. The AUC score for the suggested model is 0.9989. The high scores demonstrate that the proposed model produced accurate outcomes (high precision). Similarly, the majority of the model results were good (high recall). Furthermore, the findings suggest that the proposed approach is robust and can effectively hand-pick base learners to give complementing data in a variety of ways. They do, however, have high predictive abilities, which improves the suggested model's classification performance.

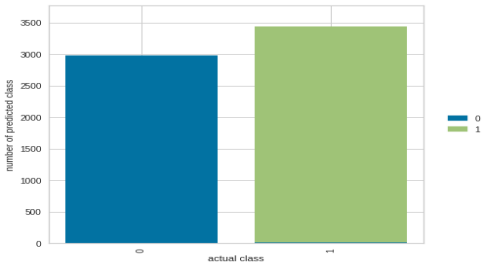


Figure 4. MCC-StackNet prediction Error.

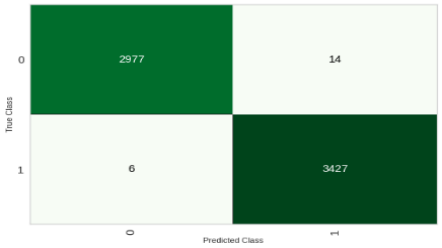


Figure 5. Confusion matrix yielded by the MCC-StackNet model.

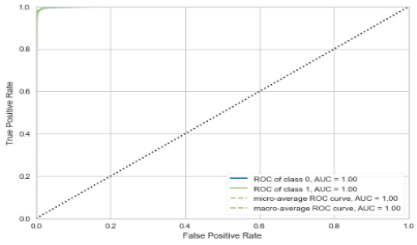


Figure 6. ROC curve of the MCC-StackNet classifier.

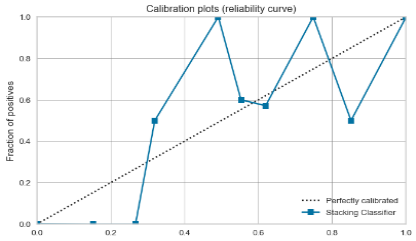


Figure 7. Calibration plot of the MCC-StackNet

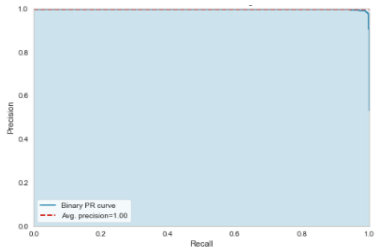


Figure 8. Precision–recall curve of the MCC-StackNet.

Figures 8, 9 depict the proposed model's class report and calibration charts, respectively. The decision boundary and cumulative gain curve, the lift curve, KS Statistic, the dimensions, and learning curve plots of the proposed model are shown in Figures 10, 11, 12, 13, 14, 15, respectively. The learning curve (see Figure 15) depicts our model's learning performance over time. The results reveal that the suggested model does not overfit nor underfit, indicating robustness of our proposed classifier. Figure 16 shows the manifold curve of the proposed model.

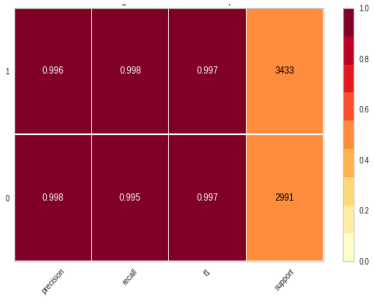


Figure 9. Class report of the proposed decision fusion model.

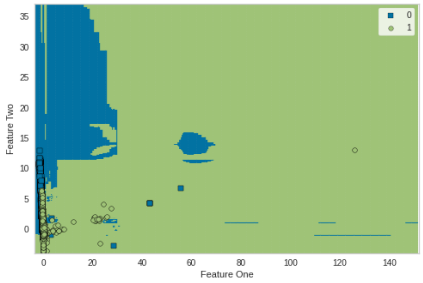


Figure 10. Decision boundary of the proposed decision fusion model.

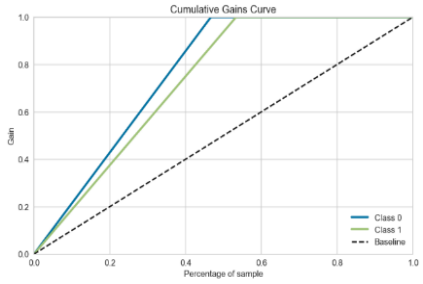


Figure 11. The cumulative gain curve of the proposed decision fusion model.

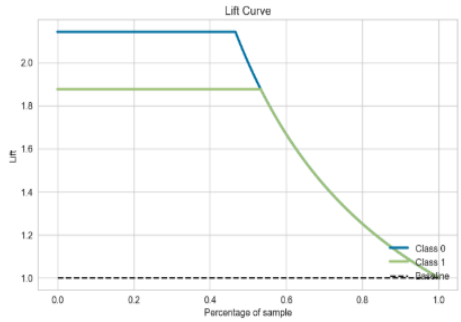


Figure 12. Lift Curve of the proposed decision fusion model.

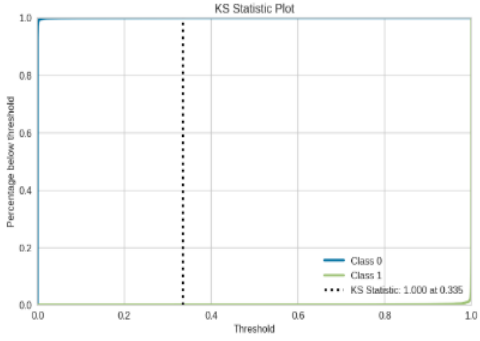


Figure 13. KS Statistic plot of the MCC-StackNet.

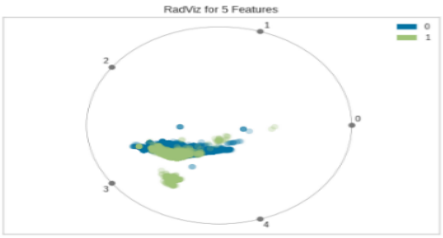


Figure 14. Dimensions of the proposed decision fusion model.

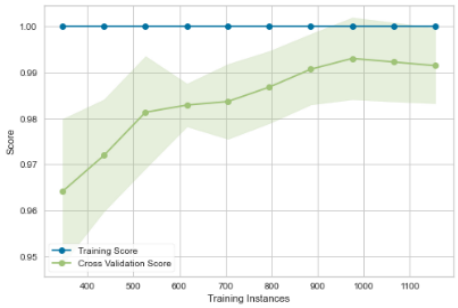


Figure 15. The learning curve of the proposed decision fusion model.

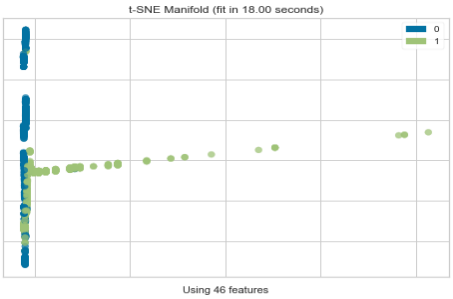


Figure 16. Manifold curve of the proposed decision fusion model.

### 4.4. A Comparative Analysis With State of the Art

We conducted a comparative experiment of our state-of-the-art model in order to objectively analyse the performance of the proposed model proposed in this work. The proposed model's accuracy, AUC, recall, precision, F1-Score, Kappa, and MCC values are 0.998, 0.9989, 0.9961, 0.9975, 0.9946, and 0.9946, respectively, as shown in Figure 17. The suggested model outperforms state-of-the-art ensemble models such as Xgboost, Catboost, LightGBM, RF, MLP, and Extra Trees Classifier, according to the results ET. As a result, the proposed model has a higher probability of correctly identifying unauthorised network traffic at 99.8 percent accuracy than Xgboost (97.61%), Catboost (97.49%), LightGMB (97.63%), RF (97.97%), MLP (97.7%) ET (95.82%), DT (96.95%) and KNN (95.56), making it an efficient and better intrusion detection model. In addition, we compared the current study to our earlier work [22] and discovered a 0.001 increase in accuracy. As a result, the results show that the proposed model can more effectively and painstakingly capture the hidden relationship between network traffic variables and use it to improve prediction accuracy. Furthermore, the kappa score, which indicates the level of agreement among the actual values ($y$) and the predicted values ($y'$), shows that the predictions by our model are highly close to the true values as compared with other models. Thus, the proposed classifier can offer a more significant advantage in detecting abnormal network traffic in different attack scenarios. Table 1 shows a comparison of the current study with other studies.

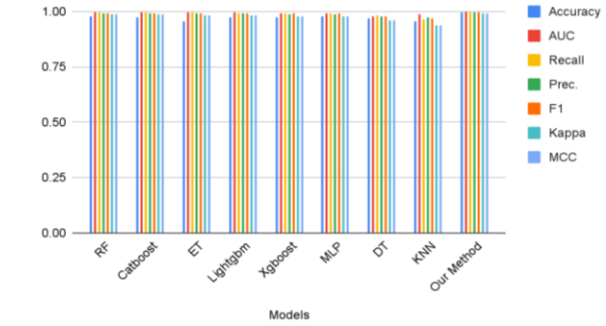


Figure 17. Performance evaluation indicators of different machine learning models and proposed decision fusion model.

## 5. Conclusions

Network intrusion is an issue of global concern; its economic impact affects the governments, organizations, and individuals. Of late, the advancement in internet technologies has increased its occurrence rate both in developed and developing economies. Recent works [11, 20] have used approximation matching algorithms run on parallelized processors and Teaching-Learning-Based Optimization Algorithm (NTLBO) approaches that show favourable results on various network types. However, in this era of big data, our future work will look at deeper Stacknets to achieve a successful training process through all layers, using $k$-fold cross-validation.

## Acknowledgment

We are grateful to all who supported this study in the computer science and informatics departments at UENR. The support from ACIT-2021 organizers, Sultan Qaboos University, and the editorial staff of IAJIT is greatly appreciated.

## References

[1] Ajdani M. and Ghaffary H., "Design Network Intrusion Detection System Using Support Vector Machine," *International Journal of Communication Systems*, vol. 34, no. 3, pp. e4689, 2021.

[2] Autee P., Bagwe S., Shah V., and Srivastava K., "StackNet-DenVIS: a Multilayer Perceptron Stacked Ensembling Approach for COVID-19 Detection Using X-ray Images," *Physical and Engineering Sciences in Medicine*, vol. 43, no. 4, pp. 1399-1414, 2020.

[3] Bhavani T., Rao M., and Reddy A., "Network Intrusion Detection System Using Random Forest

and Decision Tree Machine Learning Techniques," *in Proceeding of the Advances in Intelligent Systems and Computing*, pp. 637-643, 2020.

[4] Buil-Gil D., Miró-Llinares F., Moneva A., Kemp S., and Díaz-Castaño N., "Cybercrime And Shifts in Opportunities During COVID-19: A Preliminary Analysis in The UK," *European Societies*, vol. 23, pp. S47-S59, 2021.

[5] Devarajan R. and Rao P., "An Efficient Intrusion Detection System by Using Behaviour Profiling and Statistical Approach Model," *The International Arab Journal of Information Technology*, vol. 18, no. 1, pp. 114-124, 2021.

[6] Elmasry W., Akbulut A., and Zaim A., "Evolving Deep Learning Architectures for Network Intrusion Detection Using A Double PSO Metaheuristic," *Computer Networks*, vol. 168, pp. 107042, 2020.

[7] Ferrag M., Maglaras L., Ahmim A., Derdour M., and Janicke H., "RDTIDS: Rules and Decision Tree-Based Intrusion Detection System for Internet-of-Things Networks," *Future Internet*, vol. 12, no. 3, pp. 44, 2020.

[8] Fitni Q. Ramli K., "Implementation of Ensemble Learning and Feature Selection for Performance Improvements in Anomaly-Based Intrusion Detection Systems," *in Proceeding of the IEEE International Conference on Industry 4.0, Artificial Intelligence, and Communications Technology*, Bali, pp. 118-124, 2020.

[9] Gamage S. and Samarabandu J., "Deep learning Methods in Network Intrusion Detection: A Survey and an Objective Comparison," *Journal of Network and Computer Applications*, vol. 169, pp. 102767, 2020.

[10] Hawdon J., Parti K., and Dearden T., "Cybercrime in America amid COVID-19: the Initial Results from a Natural Experiment American," *Journal of Criminal Justice*, vol. 45, no. 4, pp. 546-562, 2020.

[11] Hnaif A., Jaber K., Alia M., and Daghbosheh M., "Parallel Scalable Approximate Matching Algorithm for Network Intrusion Detection Systems the International Arab," *Journal of Information Technology*, vol. 18, no. 1, pp. 77-84, 2021.

[12] Injadat M., Moubayed A., Nassif A., and Shami A., "Multi-Stage Optimized Machine Learning Framework for Network Intrusion Detection," *IEEE Transactions on Network and Service Management*, vol. 18, no. 2, pp. 1803-1816, 2021.

[13] Jiang K., Wang W., Wang A., and Wu H., "Network Intrusion Detection Combined Hybrid Sampling with Deep Hierarchical Network," *IEEE Access*, vol. 8, pp. 32464-32476, 2020.

[14] Khan S., Loo K., and Din Z., "Framework for Intrusion Detection in IEEE 802.11 Wireless Mesh Networks," *The International Arab Journal of Information Technology*, vol. 7, no. 4, pp. 435-440, 2010.

[15] Kilincer I., Ertam F., and Sengur A., "Machine Learning Methods for Cyber Security Intrusion Detection: Datasets and Comparative Study," *Computer Networks*, vol. 188, pp. 107840, 2021.

[16] Krishnaveni S., Vigneshwar P., Kishore S., Jothi B., and Sivamohan S., "Anomaly-Based Intrusion Detection System Using Support Vector Machine," *Artificial Intelligence and Evolutionary Computations in Engineering System*, pp. 723-731, 2020.

[17] Mahfouz A., Abuhussein A., Venugopal D., and Shiva S., "Network Intrusion Detection Model Using One-Class Support Vector Machine," *Advances in Machine Learning and Computational Intelligence*, pp. 79-86, 2021.

[18] Malik A. and Khan F., "A Hybrid Technique Using Binary Particle Swarm Optimisation and Decision Tree Pruning for Network Intrusion Detection," *Cluster Computer*, vol. 21, no. 1, pp. 667-680, 2018.

[19] Michailidis M., "StackNet, Meta Modelling Framework," https://github.com/ kaz-Anova/StackNet, Last Visited, 2022.

[20] Aljanabi M. and Ismail M., "Improved Intrusion Detection Algorithm Based on TLBO and GA Algorithms," *The International Arab Journal of Information Technology*, vol. 18, no. 2, pp. 170-179, 2021.

[21] Nti I., Adekoya A., and Weyori B., "A Comprehensive Evaluation of Ensemble Learning for Stock-Market Prediction," *Journal of Big Data*, vol. 7, no. 1, pp. 1-40, 2020.

[22] Nti I., Nyarko-Boateng O., Adekoya A., and Arjun R., "Network Intrusion Detection with StackNet: A Phi Coefficient Based Weak Learner Selection Approach*," in Proceeding of the 22nd International Arab Conference on Information Technology*, Abu Dhabi, pp. 10-11, 2021.

[23] Pawlicki M., Choraś M., and Kozik R., "Defending Network Intrusion Detection Systems against Adversarial Evasion Attacks," *Future Generation Computer Systems*, vol. 110, pp. 148-154, 2020.

[24] Rajagopal S., Kundapur P., and Hareesha K., "A Stacking Ensemble for Network Intrusion Detection Using Heterogeneous Datasets," *Security and Communication Networks*, vol. 2020, pp. 1-9, 2020.

[25] Relan N. and Patil D., "Implementation of Network Intrusion Detection System Using Variant Of Decision Tree Algorithm," *in Proceeding of the International Conference on*

*Nascent Technologies in the Engineering Field*, Navi Mumbai, pp. 1-5, 2015.

[26] Sahu S. and Mehtre B., "Network Intrusion Detection System Using J48 Decision Tree," *in Proceeding of the International Conference on Advances in Computing, Communications and Informatics*, Kochi, pp. 2023-2026, 2015.

[27] Salo F., Injadat M., Nassif A., and Essex A., "Data Mining with Big Data in Intrusion Detection Systems: A Systematic Literature Review," *arXiv preprint arXiv: 2005.12267*, 2020.

[28] Salo F., Injadat M., Nassif A., Shami A., and Essex A., "Data Mining Techniques in Intrusion Detection Systems: A Systematic Literature Review," *IEEE Access*, vol. 6, pp. 56046-56058, 2018.

[29] Senthilnayaki B., Venkatalakshmi K., and Kannan A., "Intrusion Detection System Using Fuzzy Rough Set Feature Selection and Modified KNN Classifier," *The International Arab Journal of Information Technology*, vol. 16, no. 4, pp. 746-753, 2019.

[30] Shah S., Muhuri P., Yuan X., and Roy K., Chatterjee P., "Implementing a Network Intrusion Detection System Using Semi-Supervised Support Vector Machine and Random Forest," *in Proceedings of the 2021 ACM Southeast Conference*, pp. 180-184, New York, 2021.

[31] Singh J., Kaur L., and Gupta S., "A Cross-Layer Based Intrusion Detection Technique for Wireless Networks," *The International Arab Journal of Information Technology*, vol. 9, no. 3, pp. 201-207, 2012.

[32] Sornsuwit P. and Jaiyen S., "A New Hybrid Machine Learning for Cybersecurity Threat Detection Based on Adaptive Boosting," *Applied Artificial Intelligence*, vol. 33, no. 5, pp. 462-482, 2019.

[33] Sornsuwit P. and Jaiyen S., "Intrusion Detection Model Based On Ensemble Learning for U2R and R2L Attacks," *in Proceeding of the 7th International Conference on Information Technology and Electrical Engineering*, Chiang Mai, pp. 354-359, 2015.

[34] Su T., Sun H., Zhu J., Wang S., and Li Y., "BAT: Deep Learning Methods on Network Intrusion Detection Using NSL-KDD Dataset," *IEEE Access*, vol. 8, pp. 29575-29585, 2020.

[35] Tabash M., Allah M., and Tawfik B., "Intrusion Detection Model Using Naive Bayes And Deep Learning Technique," *The International Arab Journal of Information Technology*, vol. 17, no. 2, pp. 215-224, 2020.

[36] Tama B. and Lim S., "Ensemble learning for Intrusion Detection Systems: A Systematic Mapping Study and Cross-Benchmark Evaluation," *Computer Science Review*, vol. 39, pp. 100357, 2021.

[37] Tian J. and Gao M., "Network Intrusion Detection Method Based on High Speed and Precise Genetic Algorithm Neural Network," *in Proceeding of the International Conference on Networks Security, Wireless Communications and Trusted Computing*, Wuhan, pp. 619-622, 2009.

[38] Vinayakumar R., Soman K., and Poornachandran P., "Applying Convolutional Neural Network For Network Intrusion Detection," *in Proceeding of the International Conference on Advances in Computing, Communications and Informatics*, Manipal, pp. 1222-1228, 2017.

[39] Wang Y., Shen Y., and Zhang G., "Research on Intrusion Detection Model Using Ensemble learning Methods," *in Proceeding of the 7th IEEE International Conference on Software Engineering and service science*, pp. 422-425, 2016.

[40] Wazirali R., "An Improved Intrusion Detection System Based on KNN Hyperparameter Tuning and Cross-Validation," *Arabian Journal for Science and Engineering*, vol. 45, no. 12, pp. 10859-10873, 2020.

[41] Wolpert D., "Stacked Generalisation," *Neural Networks*, vol. 5, no. 2, pp. 241-259, 1992.

**Isaac Kofi Nti** (Member IEEE) holds Ph.D. in Computer Science and Engineering and is a Lecturer at the Department of Computer Science and Informatics, University of Energy and Natural Resources (UENR), Sunyani Ghana. His research interests include artificial intelligence, energy system modelling, intelligent information systems, social and sustainable computing, business analytics and data privacy and security. He has widely published & reviews for refereed journals. ORCID: https://orcid.org/0000-0001-9257-4295.

**Owusu Nyarko-Boateng** holds HND Electrical and Electronic Engineering (2000), BSc Computer Science (2012), Postgraduate Diploma in Education - PGDE (2017), MSc Information Technology (2016) and PhD in Computer Science. He has also obtained some professional certifications. He has over ten years of working experience as a Transmissions & Operations Engineer with MTN-Gh and Huawei Technology (SA), Ghana. He is a lecturer at the University of Energy and Natural Resources, Sunyani-Ghana. His research interest is in Optical Technology, Submarine and Underground fiber optics cable transmission, 5G, data communication, intelligent transmission systems, and deploying machine learning in tracing faults, and IT Policy formulation and deployment. https://orcid.org/0000-0003-0300-2469.

**Adebayo Felix Adekoya** holds B. Sc. (1994), M. Sc. (2002), and Ph. D. (2010) in Computer Science, an MBA in Accounting & Finance (1998), and a Postgraduate Diploma in Teacher Education (2004). In addition, he has put in about twenty-five (25) years of experience as a lecturer, researcher and administrator at the higher educational institution levels in Nigeria and Ghana. A. F. Adekoya is an Associate Professor of Computer Science. Currently, he serves as the Dean, School of Sciences, and the Acting Pro-VC University of Energy and Natural Resources, Sunyani, Ghana. His research interests include artificial intelligence, business & knowledge engineering, intelligent information systems, and social and sustainable computing. ORCID: https://orcid.org/0000-0002-5029-2393.

**Arjun Remadevi Somanathan** is an Assistant Professor (Senior Grade) in School of Computer Science and Engineering at VIT Vellore. He has Ph.D. specialized in Information Systems from NITK, India. He holds a Masters in Software Engineering from CUSAT. His research interests are artificial intelligence, information systems, fintech, sustainable computing etc. Dr. Arjun is an Associate Editor for IJEBR, IGI Global and reviews for IJIM, Elsevier, IJIMAI, MDPI Electronics, European Alliance for Innovation and many more. His student feedback on teaching was 4.32/5.00 in 2021. He is rated as an excellent reviewer on Publons, Clarivate and was committee member/ reviewer for ICCSA 2019, AoM Annual meeting 2020-2021, FICTA 2020, ICIS 2021, ECIS 2022, IEEE WCCI 2022.ORCID: https://orcid.org/0000-0002-2770-6164.