IJDI-ERET

INTERNATIONAL JOURNAL OF DARSHAN INSTITUTE ON ENGINEERING RESEARCH AND EMERGING TECHNOLOGIES
Vol. 14, No. 2, 2025
www.ijdieret.in

DOI 10.32692/IJDI-ERET/14.2.2025.250

(Research Article)

A Stacked Hybrid Classifier for an Intrusion Detection System in Bank Networks

Abraham Danlami[1*], E. J. Garba[2], Y. M. Malgwi[3], S. E. Dogo[4]

[1*]*Department of Computer Science, Faculty of Computing & Artificial Intelligence, Federal University, Wukari,* Taraba State*, NIGERIA*
[2,3]*Department of Computer Science, Faculty of Computing & Artificial Intelligence, Modibbo Adama University, Yola, Adamawa State, NIGERIA*
[4]*Department of Computer Science, Faculty of Computing & Artificial Intelligence, Taraba State University, Jalingo, Taraba State, NIGERIA*

Abstract

A bank network is an interconnected system designed to facilitate financial transactions, minimize customer waiting times, and reduce the risk of errors caused by automated systems or bots. However, over the past three decades, cyberattacks have emerged as a significant threat to the security of these networks. The tools like penetration testing, machine learning classifiers, multi-factor authentication, network traffic analysis, and bot detection systems are reactive rather than proactive. As a result, these measures may fail to detect or prevent sophisticated attacks in real time. Such vulnerabilities can be exploited by attackers to gain unauthorized access to banking networks, posing serious risks to data integrity and customer privacy. In this study, we aim to enhance the performance of an anomaly-based Intrusion Detection System (IDS) within a banking environment by developing a Stacked Hybrid Classifier. To achieve this, the study employed the CICIDS2017 dataset, which contains realistic traffic data simulating various types of attacks and benign behaviors. The dataset's inherent class imbalance common in intrusion detection scenarios was addressed using the Synthetic Minority Oversampling Technique (SMOTE), ensuring more balanced training and improved sensitivity to minority attack classes. The model was implemented using Python, stacking multiple base classifiers including Random Forest, Decision Tree, Logistic Regression, and K-Nearest Neighbors. The predictions from these models were combined to form a meta-model, resulting in a more robust and generalized detection capability. The study was evaluated using stratified k-fold cross-validation to ensure consistent and unbiased performance assessment across different data partitions. The results show that the stacked Hybrid classifier achieved the best accuracy of 99.76%, along with superior precision, recall, and F1 scores when compared to the individual classifiers. This demonstrates the effectiveness of Hybrid learning, data balancing, and rigorous validation in improving IDS performance in banking networks.

Keywords: Machine Learning, Cyberattacks, Bank Networks, Stacked Hybrid, Intrusion Detection System, SMOTE.

1. Introduction

Bank network security has become increasingly complex due to recent technological advancements, the rapid evolution of cybersecurity, and the exponential growth of internet data driven by big data technologies. Additionally, the rises in cybercrime, the massive volume of online data, and expanded network connectivity have significantly increased the vulnerability of computer systems to attacks. (1). globally, recent technological developments expose consumers to a range of hacks in which private information is compromised. (2). There are various network and technology assessment methods to detect cyberattacks. Although network tactics raise user awareness, they are unable to detect cyberattacks that require technology assisted techniques. The infrastructure of bank networks is vulnerable to both physical and network threats. According to (3), a physical attack results in the vandalism of network infrastructures, which makes the services ineffective and increases network traffic or congestion. One of the primary threats to bank networks is botnet attacks, which are orchestrated by malicious actors known as Botmasters. A Botmaster controls a "botnet"a network of compromised devices used to execute various harmful activities, such as data breaches, DDoS attacks, and malware distribution. Although numerous security measures are in place to protect bank networks including firewalls, user authentication, machine learning classifiers, multi-factor authentication OTP, the Zero Trust Security Model, bot detection tools, web application firewalls, and penetration testing these defenses are largely reactive. As a result, they often fail to proactively detect and prevent sophisticate and evolving cyber threats like botnet attacks. These security solutions stop a lot of attacks, but they don't do a thorough packet analysis, so they can't give the network of the company the

Corresponding Author: e-mail: abrahamdanlami@fuwukari.edu.ng, Tel-+2347033168016, +2348133231560
ISSN 2320-7590 (Print) 2583-3863 (Online)

protection it needs (4). To address these problems, a stacked hybrid classifier for an intrusion detection system in banking networks has been developed in this study to detects, and minimizes botnets activities and unauthorized access of the intrusion system that could attack bank network infrastructures.

2. Botmaster

An individual who manages the command and control of botnets for remote process execution is known as a "Botmaster." Usually, different types of remote code installation are used to deploy the botnets on compromised computers (4). The Botmaster often uses proxies to hide their identify so that investigators and law enforcement cannot find their IP address. The bots are configured to authenticate with the command and control station using either keys or a password in order to enable remote control. Occasionally, multiple Botmasters collaborate to administer a shared botnet. Furthermore, it is not unusual to obtain access to another Botmaster's botnets in another manner or to take advantage of their credentials. Botmasters have the ability to target networks of compromised devices and grant botnets access to do a variety of destructive actions, such as disabling network functions and disseminating malware. They are consequently essential for planning and executing bot attacks in the following ways (5).

2.1 Recruitment and compromise: The task of recruiting and infecting a significant number of computers or devices falls to the botmasters. They accomplish this by luring users into inadvertently installing harmful software, using malware, or taking advantage of software flaws. A gadget joins the botnet after becoming infected.

2.2 Botnet formation: These infected devices are gathered by a botmasters into a network known as a "botnet." Depending on the goals of the Botmaster, this network may contain a few hundred, hundreds of thousands, or even millions of bots.

2.3 Control and command: The botnet is under the control of the botmaster. To communicate with and control the compromised devices in the botnet, they usually set up a command-and-control server or other infrastructure. These guidelines may describe how to conduct assaults or engage in other violent acts.

2.4 Botnet maintenance: Botmasters must control their botnets to ensure their uninterrupted functioning and avoid detection. This include removing compromised or non-functional bots, updating the virus on compromised devices, and adding new bots

2.5 Bot attacks: Botmaster utilize their botnets to perform a number of harmful tasks such as Distributed Denial of Service (DDoS) Attacks, Spam and Phishing Bots can be used to spread malware, Data Theft, Cryptocurrencies Mining: Cryptocurrency mining can be done by bots, depleting the victim's resources while profiting the Botmaster, Click Fraud.

2.6 Evasion and anonymity: Botmaster usually take steps to hide their identity and location, making it difficult for law enforcement and security professionals to track them down. They can use proxy servers, anonymization techniques, or compromised systems spread across multiple jurisdictions to cover their traces.

2.7 Profit and motivation: Botmaster may launch these attacks for a number of reasons, including financial gain, espionage, or simply to cause chaos and disrupt online services.

2.8 Countermeasures and detection: For cybersecurity professionals, detecting and preventing bot attacks is a never-ending task. It entails locating and removing hacked computers, keeping an eye on network traffic for unusual activity, and, if required, pursuing legal action against botmasters.

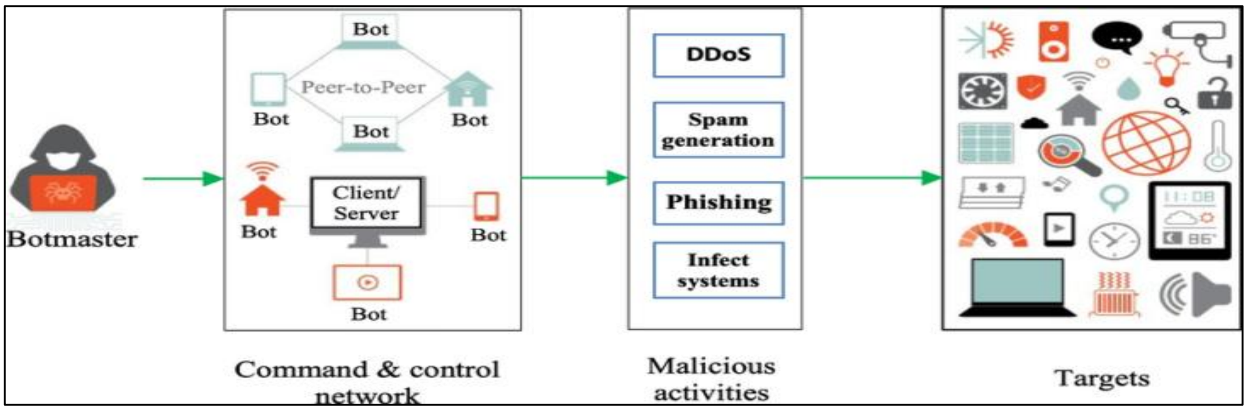


Figure 1. Botmasters command and control network (5)

3. Related Work

The use of learning systems, machine learning, or a stacked classifier for intrusion detection models for detecting botnet attacks in banking networks has received a lot of attention and focus from the research community in the past few years. These enormous efforts have produced some fascinating results that served as a basis for the development of detection schemes for these attacks. The following are included in the review:

(6) Discussed the detection system of a DDoS attack within CICIDS2017 by evaluating and applying some machine learning methods. It makes use of popular feature selection filter model techniques known as information gain, which gauges each attribute's significance according to the class's greater entropy. Out of the 80 features in the CICIDS2017 dataset, it chooses the top 10 significant properties. Then, using the C5.0, Naive Bayes, SVM, and Random Forests algorithms, four machine learning models were tested and compared based on accuracy, recall, and precision, Detection Rate were based on the False alarm. (7) Stated that bank clients, mobile banking offers a handy and appealing way to do remote banking from any location at any time. Nevertheless, the aim of the study was to strengthen user authentication and end-to-end security mechanisms to raise the mobile banking networks' dependability, integrity, and secrecy using SMS. This paper suggests end-to-end encryption with user-managed randomly generated question data that uses pre-saved question and answer on the server for authentication method to improve data protection over mobile networks. The study addresses concerns of data confidentiality against phishing or data breaches, user authentication, and message integrity, but the study was only Focus on predicting vulnerable users on the mobile banking networks. (8) Built a network-based method for detection and prevention uses processes based on anomaly and signature detection. RF, DT, & LR Classifier and the researchers face challenges during the process of Implementation

(9) Suggested a detection model that uses a dataset's botnet properties in conjunction with PSO and CFS algorithms. The suggested model was tested using several data sets, including UNSWNB15, and KDD Cup99. Three metrics Accuracy, True Positive Rate, and False Negative Rate were used to assess the performance of the three classifiers that were tested: Naive Bayes, K-NN, and SVM. In the KDD Cup99 dataset, the accuracy was impacted by the model results utilizing SVM, K-NN, and Naïve Bayes classifiers by 79.9%, 69.9%, and 58%. (10) Developed detection of unsuitable tweeter accounts. RF classifier applied to the study, Detects base on the fake accounts not bot (11) use Host Intrusion Detection Systems to detects aberrant activities in Internet of Things devices with 89.75% , DT, KNN, LR, NB and Bag of Words (BOW) model were used, Difficult to detect bots but uses other terminologies not captured by BOW in larger dataset. (12) Developed a botnet attack detection in the Internet of Things using machine learning models, DT and RF) & unsupervised learning (Deep Learning), Detected attacks based on the abnormality of the Network.

4. Methodology

The research work intends to adopt the analytical and experimental research methodologies. Observations and a review of the literature in the investigation of concepts and models within the current system have the procedures and methodology flow. The model was developed using the analytical research method:

4.1 Problem definition: The existing security methods, including machine learning classifiers, multi-factor authentication, OTP, Zero Trust Security Model, and penetration testers are not proactive.

4.2 Data gathering: This is where the collection of diverse datasets containing network traffic logs, system events and botnet-related activities from banking networks and relevant sources.

4.3 Data wrangling: This is the stage of data cleansing, data filtering, and Principal Company Analysis (PCA) where important feature is selected.

4.4 Model design: This is where the architecture of an intelligent intrusion detection model is designed; incorporating with advanced machine learning algorithms specifically tailored for detecting botnet activities in banking network environment.

4.5 Model training: Teaching the machine learning model to act intelligently by using cross validation method so that Implementation and optimization of the machine learning algorithms will give accurate and efficient results. The stage uses 70% of datasets for training in order to obtain accurate results.

4.6 Testing/validation: This is stage that used 30% of datasets in checking the performance of the machine learning model and used cross validity.

4.7 Evaluate: The performance evaluation of the IDS using appropriate metrics such as detection accuracy, false positive rate, false negative rate, computational overhead and provides a robust framework for detecting cyber threats in real-time deployment and dynamic banking environments.

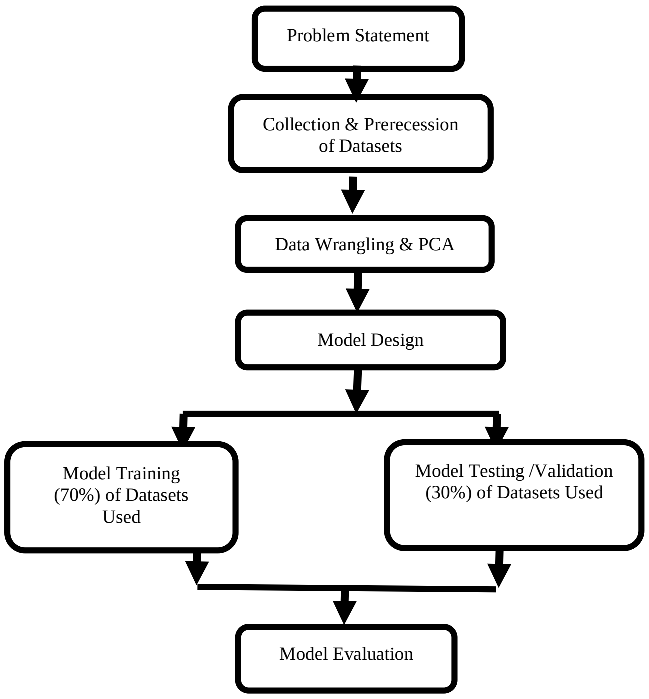
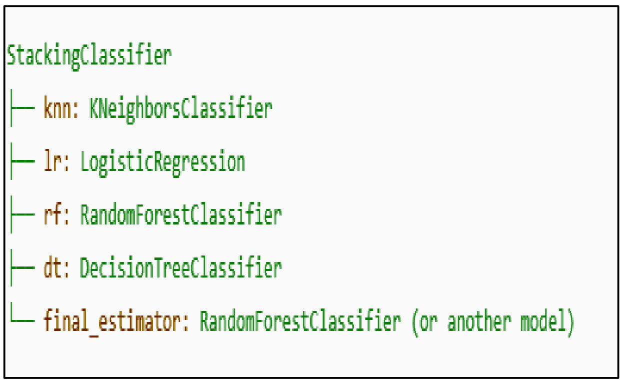

Problem Statement
Collection & Prerecession of Datasets
Data Wrangling & PCA
Model Design
Model Training (70%) of Datasets Used
Model Testing /Validation (30%) of Datasets Used
Model Evaluation

Figure 2. Analytical design method

The figure 2 illustrates the structure of a Stacking Classifier, which is an ensemble learning technique used in machine learning. This method combines the predictions of multiple different models, known as base estimators, to improve the overall performance of the system. In this case, the base estimators include a K-Neighbors Classifier, Logistic Regression, Random Forest Classifier, and Decision Tree Classifier. Each of these models is trained in parallel using the same dataset. Once these base models have made their predictions, a final estimator often another model like a Random Forest Classifier is trained on their outputs. The role of this final estimator is to learn how to best combine the predictions from the base models in order to make a more accurate final prediction. The main advantage of using a Stacking Classifier is that it leverages the unique strengths of different models. By combining them, the final output is usually more robust and accurate than relying on any single model alone.



```
StackingClassifier
├─ knn: KNeighborsClassifier
├─ lr: LogisticRegression
├─ rf: RandomForestClassifier
├─ dt: DecisionTreeClassifier
└─ final_estimator: RandomForestClassifier (or another model)
```

Figure 3. Stacking classifier

The construction of a stacking classifier, a potent ensemble learning technique frequently employed the CICIDS2017 dataset, which contains realistic traffic data simulating various types of attacks and benign behaviors. The dataset's inherent class imbalance common in intrusion detection scenarios was addressed using the Synthetic Minority Oversampling Technique (SMOTE), ensuring more balanced training and improved sensitivity to minority attack classes. The model was implemented using Python in the Anaconda Navigator environment, especially with the scikit-learn library, was shown in Figure 3. In order to provide a more accurate final prediction, a stacking classifier combines the predictions of multiple machine learning models, sometimes referred to as base learners. This ensemble method minimizes the shortcomings of each model while maximizing the strengths of KNN, RF, LR and DT models. The Stacking Classifier is fundamentally composed of two primary layers. The foundation learners, also known as level-0 classifiers, are part of the first layer. The same training dataset is used to independently train these models. The base learners include the following in the provided diagram:

- The KNeighbors Classifier (KNN), which uses distance-based voting to produce predictions based on how close data points are to one another.
- A popular linear model for binary classification issues is Logistic Regression (LR), which calculates probabilities using a linear decision boundary.
- The Random Forest Classifier (RF) is an ensemble model consisting of many decision trees that were bagged and trained on random subsets of features and data.
- Based on feature splits, the Decision Tree Classifier (DT) builds a tree-like structure to make choices.

A final estimator, also known as the level-1 model or meta-learner, is part of the second layer. The outputs (predictions or probability scores) of the base learners are used as input features in the training of this model. A Random Forest Classifier serves as the example's final estimator, but any other model, including KNeighbors Classifier, Random Forest, Decision Tree, and Logistic Regression, might be used in its place. The steps are as follows: First, the entire training data is used to train each base model. After then, each person's predictions are collected. The final estimator, which is trained to generate the final classification, uses these predictions as input. By doing this, the meta-learner finds trends in the basis models' predictions and discovers the ideal way to combine them for maximum effectiveness.

When opposed to employing a single model alone, this approach frequently results in superior accuracy and generalization. Because it enables the Stacking Classifier to balance out each basis model's unique biases and variances, the developed model finally deployed in a real-time environment to test its practical viability. Banks and financial institutions within monitored segments of their networks to assess real-time performance, latency, and responsiveness to it live threats.

5. Result and Discussion

On the modified botnet dataset, the effectiveness of the DT, LR, RF, KNN, and Stacked-Hybrid was assessed using a number of performance metrics, including the F1-Score, accuracy score, precision, recall, and AUC. The results of the DT, KNN, LR, RF, and Stacked classifiers are shown in Table 1 according to whether an instance is deemed an intrusion or not (0/1) as shown in the table headings. Percentages were used as the degree of measurement to assess the metrics result.

Table 1. The comparison of existing classifiers and stacked classifier on CIC-IDS2017 dataset

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)	ROC (%)
LR	89.37	86.02	89.37	87.48	69.38
KNN	96.93	97.04	96.93	96.93	97.12
DT	99.60	99.61	99.60	99.61	95.34
RF	99.75	99.75	99.75	99.75	98.70
Stacked classifier	99.76	99.75	99.76	99.75	99.22

From table 1, the Logistic Regression (LR) model achieves an accuracy of 89.37%, meaning it correctly classifies approximately 89 out of every 100 samples. Its precision is 86.02%, indicating that when it predicts a positive outcome, it is correct about 86% of the time. The recall is 89.37%, showing that it successfully identifies about 89% of all actual positive cases. The F1-score, which balances precision and recall, stands at 87.48%, reflecting moderate performance. However, the model's ROC-AUC is only 69.38%, suggesting it has a relatively weak ability to distinguish between classes compared to the other models. The K-Nearest Neighbors (KNN) model performs significantly better, with an accuracy of 96.93%, correctly classifying nearly 97% of the data. Its precision is slightly higher at 97.04%, meaning that its positive predictions are highly reliable. The recall matches the accuracy at 96.93%, indicating excellent detection of positive cases. It maintains a strong F1-score of 96.93%, reflecting a balanced and effective performance. Additionally, the model demonstrates excellent discriminative ability with a ROC-AUC of 97.12%. The Decision Tree (DT) model shows even stronger performance, boasting an accuracy of 99.60%, which implies that almost all samples are correctly classified. Its precision and recall are virtually identical at 99.61% and 99.60%, respectively, meaning it rarely makes incorrect positive predictions and almost never misses true positives. The F1-score is also extremely high at 99.61%, highlighting the model's outstanding performance. Its ROC-AUC is 95.34%, which, while slightly lower than KNN's, still indicates a high level of class separation. The Random Forest (RF) model slightly surpasses the Decision Tree in all metrics, achieving an accuracy of 99.75%, meaning it makes very few mistakes. Its precision, recall, and F1-score are all 99.75%, demonstrating a near-perfect ability to predict and detect positive cases with high consistency. The model also has a very high ROC-AUC of 98.70%, confirming its excellent capability in distinguishing between the classes.

Finally, an improved Stacked Model-Based Network stands out as the best-performing model overall. It achieves the highest accuracy of 99.76%, meaning it correctly classifies virtually every input. The precision is 99.75%, and the recall is 99.76%, showing both highly accurate and comprehensive detection of positive instances. The F1-score is also 99.75%, indicating outstanding balance and performance. Most impressively, the ROC-AUC reaches 99.22%, making Stacked Model the most effective model in terms of both classification accuracy and class discrimination.

5.1 Confusion matrix for stacked hybrid classifier:

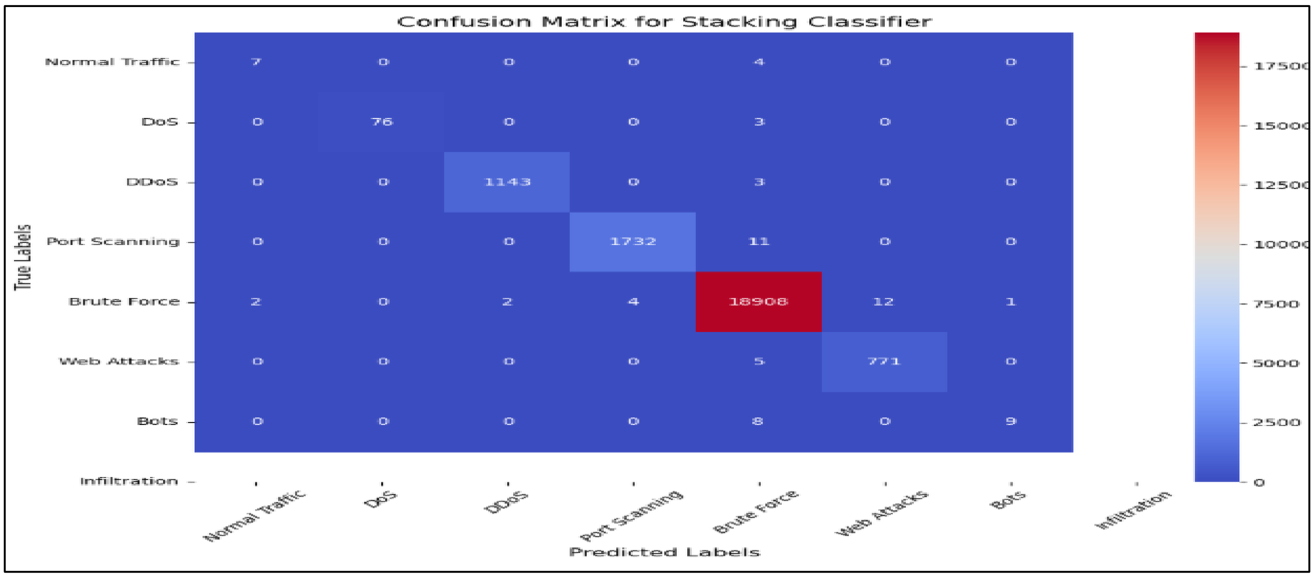


Figure 4. Confusion matrix for hybrid stack classifier

The figure 4 represents a confusion matrix for a Stacking Classifier, which evaluates the performance of an ensemble learning model that combines RF, DT, LR and K-NN base classifiers in either to improve classification accuracy. The true positives (correct classifications) appear on the diagonal of the matrix. Brute Force: 18,908 correctly classified instances. Port Scanning: 1,732 correctly classified instances. DDoS: 1,143 correctly classified instances. Web Attacks: 771 correctly classified instances. DoS: 76 correctly classified instances. Normal Traffic: 7 correctly classified instances. Bots: 9 correctly classified instances. These values suggest that the Stacking Classifier is performing well in correctly identifying attack types. Brute Force has minor misclassifications into: Port Scanning (4), Web Attacks (12), DDoS (2), Bots (7), DDoS is sometimes confused with Port Scanning (3), Port Scanning has small misclassifications (11 instances in Brute Force). Normal Traffic is misclassified into Web Attacks (4). Bots are confused with Brute Force (8). Infiltration has a small misclassification issue (9 instances predicted as Bots). The dataset appears highly imbalanced; with Brute Force dominating the dataset (18,908 samples).Infiltration has very few correctly classified instances, indicating difficulty in detecting this attack type. Despite the imbalance, the model performs well across most classes, with relatively few misclassifications.

Table 2. Comparison of hybrid stack classifier with RF, DT, LR and KNN

Model	Strengths	Weaknesses
RF	High accuracy, handles class imbalance well	Computationally expensive
DT	Explainable, performs well on structured data	Prone to overfitting
LR	Simple, fast	Poor in handling complex relationship
KNN	Works well on some classes	Struggles with high-dimensional data
Stacked classifier	Combines RF, LR, DT, KNN models to improved it accuracy	More complex, requires more tuning

The Stacked-Hybrid Classifier outperforms other models, including Random Forest and Decision Trees, by reducing misclassification rates and handling multiple attack types effectively. The Stacking Classifier is one of the best-performing models, showing high accuracy with minimal misclassification. Brute Force, Port Scanning, and DDoS detection are excellent. Areas for the Improvement: it improved minority class detection (especially Infiltration). Further optimize the model by adjusting base classifiers and hyperparameter tuning.

5.2 Receiver operating characteristics (ROC): Furthermore, on investigating the performance of the models inclusive of the hybrid, the ROC curve was considered. The curve is a graphical representation of the classification model's performance across different threshold values for a False Positive Rate (FPR). Essentially, the ROC curve shows the trade-off between sensitivity (true positive rate) and specificity (1 - false positive rate). AUC Values range from 0 to 1 with 1.0 indicating a perfect classifier, 0.5 suggesting random guessing (no discrimination capability) and a < 0.5 indicating worse than random guessing.

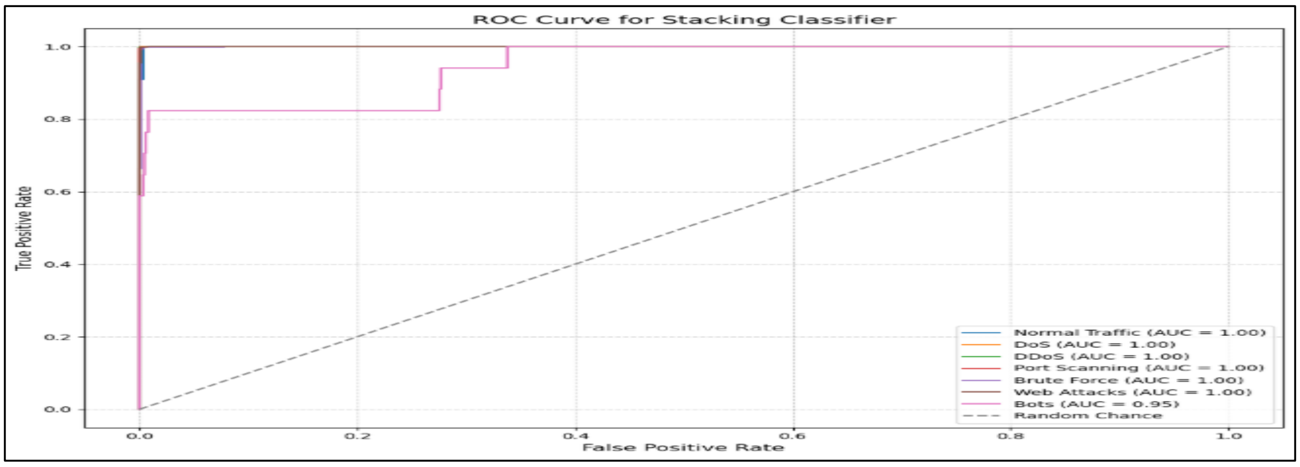


Figure 5. ROC for hybrid stack classifier

The figure 5 represents the ROC (Receiver Operating Characteristic) Curve for a Stacking Classifier, which evaluates the classifier's performance across different attack categories. The x-axis represents the False Positive Rate (FPR): the proportion of incorrectly classified negative instances. The y-axis represents the True Positive Rate (TPR): the proportion of correctly classified positive instances. The diagonal dashed line (labeled "Random Chance") represents a classifier that makes random predictions (AUC = 0.5). A good model shows a curve that moves towards the top-left corner, indicating a high true positive rate with a low false positive rate. The legend lists the AUC (Area Under the Curve) values for each class: Normal Traffic (AUC = 1.00) DoS (AUC = 1.00), DDoS (AUC = 1.00), Port Scanning (AUC = 1.00), Brute Force (AUC = 1.00), Web Attacks (AUC = 1.00), Bots (AUC = 0.95), AUC = 1.00 indicates perfect classification performance (no false positives, all true positives), Bots have the lowest AUC (0.95), meaning some misclassifications exist, but performance is still very high. Perfect Classification (AUC = 1.00), the classifier correctly identifies DoS, DDoS, Port Scanning, Brute Force, and Web Attacks with no false positives, slightly Lower Performance for Bots (AUC = 0.95), there are some false positives or false negatives in bot detection, but performance is still strong.

Table 3. Comparison of stacked-hybrid with RF, DT, KNN and LR model

Model	AUC performance
Stacking classifier	Almost perfect (AUC close to 1 for all classes)
Random forest	High but slightly lower than stacking
Decision tree	Moderate (prone to overfitting)
KNN	Lower, especially for imbalanced classes
Logistic regression	Struggles with complex decision boundaries

The Stacking Classifier outperforms all other models, achieving nearly perfect classification across all attack types. The Stacking Classifier is highly effective at detecting cyber-attacks, the AUC scores close to 1.00 indicate excellent predictive ability, Bots (AUC = 0.95) could be improved with further tuning or additional feature selection, overall, this model is highly reliable for cybersecurity attack detection, making it an excellent performance.

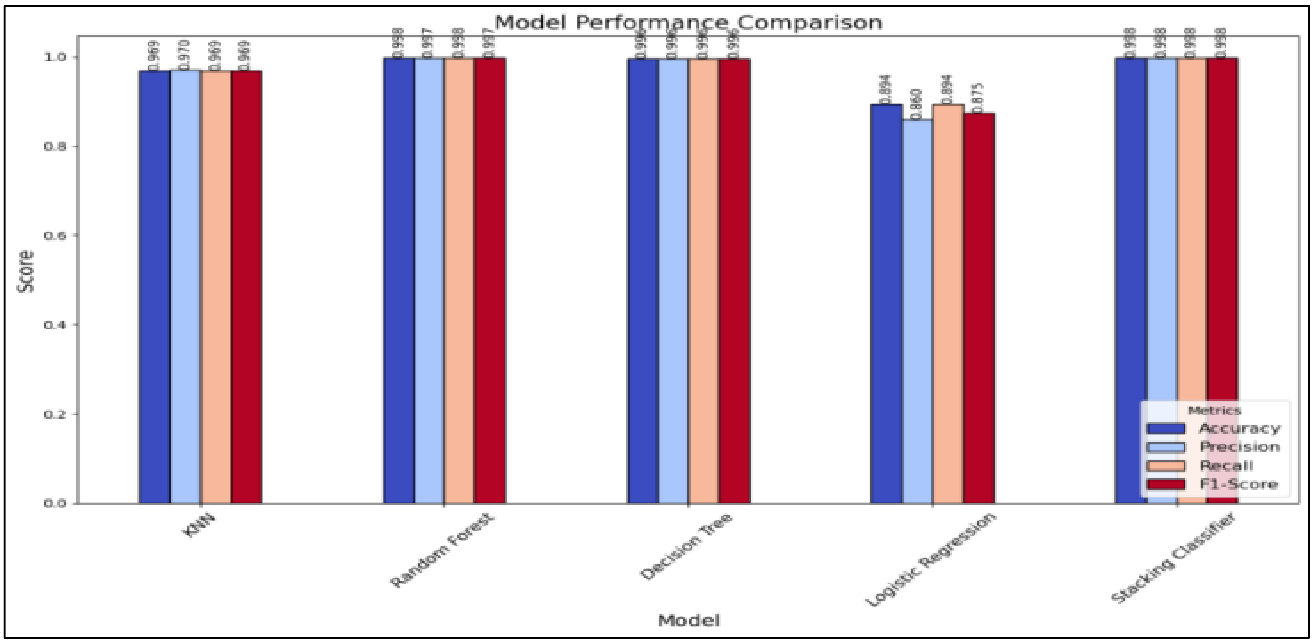


Figure 6. Graphical result presentation

The bar chart titled "Model Performance Comparison" illustrates the comparative effectiveness of five machine learning models—K-Nearest Neighbors (KNN), Random Forest, Decision Tree, Logistic Regression, and the Stacking Classifier—based on four key evaluation metrics: Accuracy, Precision, Recall, and F1-Score. Starting with the K-Nearest Neighbors (KNN) model, it demonstrates solid and consistent performance across all metrics. The values for accuracy, precision, recall, and F1-score are all approximately .96.90% to 97.00%, indicating that the model reliably classifies data with balanced precision and recall. This consistency suggests that KNN maintains stable predictive performance across different aspects of evaluation. The Random Forest model achieves near-perfect scores of around 99.80% in all four metrics. Such high and uniform performance indicates that Random Forest is highly accurate and dependable, with minimal discrepancy between how well it predicts positive cases and how many of those predictions are correct. This reflects a strong capability to generalize well on the data. Similarly, the Decision Tree model also delivers exceptional results, with accuracy, precision, recall, and F1-score all falling between 99.80% and 99.90%. Notably, its F1-score slightly surpasses that of Random Forest, which reflects its excellent balance between precision and recall. Overall, the Decision Tree proves to be extremely effective at classifying data correctly, with very few errors. In contrast, Logistic Regression shows the lowest performance among the models compared. Its accuracy is around 89.40%, while its precision, recall, and F1-score range from approximately 86.00% to 87.50%. This lower performance indicates that Logistic Regression is less effective at both correctly predicting and identifying positive cases. The noticeable gap between precision and recall further suggests a lack of balance in its predictions, leading to reduced reliability compared to the other models. The Stacking Classifier, an ensemble approach that combines multiple models, performs at the highest level alongside Decision Tree and Random Forest. It achieves consistent and exceptionally high scores, with all metrics hovering around 99.80%. The KNN model performs well, though it is slightly behind the top three. Logistic Regression, on the other hand, lags significantly behind the others, with lower and less balanced metrics, making it the least effective model in this comparison.

Table 4. Comparison between the existing authors and stacked model on CIC-IDS2017 dataset

Author/year	Classifiers algorithms	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)	ROC (%)
Stacked model	DT,RF,LR,KNN	99.76	99.75	99.76	99.75	99.22
(12)	DT,SVM,NB,LR	99.69	99.68	99.69	99.69	99.07
(4)	KNN,DT,LR	84.20	74.82	84.20	79.23	92.16
(9)	CNN,SVM,DB	98.50	98.23	99.02	98.62	87.00

The Stacked Classifier model, which combines Decision Tree (DT), Random Forest (RF), Logistic Regression (LR), and K-Nearest Neighbors (KNN), demonstrates the highest overall performance among all the models compared. It achieves an accuracy of 99.76%, indicating it correctly classifies almost all samples. Its precision and recall are both extremely high at 99.75% and 99.76%, respectively, showing that it makes very few false positive or false negative predictions. The F1-score, a harmonic mean of precision and recall, also stands at 99.75%, confirming a strong balance between precision and recall. Additionally, its ROC-AUC score is 99.22%, which indicates outstanding class separation capability and overall model reliability. In the study by (12), a different combination of classifiers was used Decision Tree (DT), Support Vector Machine (SVM), Naive Bayes (NB), and Logistic Regression (LR). This ensemble approach yielded an accuracy of 99.69%, which is very close to the stacked classifier. The precision, recall, and F1-score are all consistently high at 99.68%, 99.69%, and 99.69%, respectively. The ROC-AUC score is 99.07%, also indicating excellent discriminative ability, although slightly lower than the stacked model. The study by (4)) used a simpler combination of KNN, Decision Tree, and Logistic Regression, which resulted in a significantly lower accuracy of 84.20%. The precision is 74.82%, which suggests a relatively high rate of false positives. While the recall matches the accuracy at 84.20%, the F1-score of 79.23% reveals an imbalance between precision and recall. Interestingly, the ROC-AUC score is relatively high at 92.16%, which indicates the model still performs well in distinguishing between classes, even though its accuracy and other metrics are comparatively low.

Lastly, the study by. (9) Employed a deep learning approach using a Convolutional Neural Network (CNN), combined with SVM and Decision Boundaries (DB). This method achieved an accuracy of 98.50%, showing very strong predictive performance. The precision is 98.23%, the recall is 99.02%, and the F1-score is 98.62%, suggesting a well-balanced and highly accurate model. However, the ROC-AUC score is 87.00%, which, although still good, is lower compared to the top-performing ensemble methods. This suggests that while the model classifies well overall, it may not be as effective at distinguishing between positive and negative classes as the top ensemble models. The Stacked Classifier combining DT, RF, LR, and KNN is the best performing model across all metrics. (12) Also presented a highly effective ensemble approach with slightly lower but still excellent results. (4) showed weaker performance, particularly in precision and F1-score, although it's ROC-AUC was relatively strong. (9) Demonstrated the strength of deep learning methods in accuracy and recall, though slightly behind in ROC-AUC.

6. Summary, Conclusion, Recommendations and Further Studies

6.1 Summary: This research proposed a robust solution for enhancing anomaly-based Intrusion Detection Systems (IDS) within banking networks using a Stacked Ensemble Classifier. The model was developed using a hybrid stacking approach that integrated Random Forest (RF), Decision Tree (DT), Logistic Regression (LR), and K-Nearest Neighbors (KNN). The CICIDS2017 dataset was utilized to simulate realistic network attack scenarios. To address class imbalance, the Synthetic Minority Oversampling Technique (SMOTE) was applied. Model evaluation employed stratified k-fold cross-validation to ensure performance reliability. The results demonstrate that the Stacked Ensemble Classifier significantly outperforms individual models across all performance metrics. It achieved an average accuracy of 99.76%, with equally high precision, recall, F1-score, and ROC-AUC values. Comparative analysis with other research works confirmed the superior performance of the proposed model in intrusion detection, particularly in identifying complex and minority-class attack types.

6.2 Conclusion: The proposed stacked Hybrid model proves to be highly effective for intrusion detection in banking networks. By combining multiple classifiers and addressing data imbalance through SMOTE, the system achieved near-perfect classification performance across a range of cyberattack types. The stacked hybrid model's strength lies in its ability to leverage the complementary strengths of individual classifiers, thereby improving generalization and minimizing false positives and negatives. The study validates that ensemble learning, when combined with proper pre-processing, data balancing, and cross-validation, provides a robust framework for detecting cyber threats in real-time deployment and dynamic banking environments.

6.3 Recommendations: Based on the outcomes of this study, several recommendations are proposed to enhance cybersecurity in banking networks and guide future research:

6.3.1 Continuous model updating: As cyber-attack techniques evolve, IDS models must be regularly updated with new threat signatures and patterns. It is recommended that banking institutions establish pipelines for retraining and fine-tuning models with updated datasets to maintain effectiveness.

6.3.2 Hybrid model Optimization: Further tuning of hyperparameters and experimenting with alternative base and meta-classifiers should be pursued to refine detection accuracy. Grid search, random search, or Bayesian optimization can be applied to identify the most efficient configurations.

6.3.3 Collaboration for threat intelligence sharing: Banks should collaborate with other financial institutions and national cybersecurity bodies to share threat intelligence and datasets. This will enhance the robustness of detection systems by exposing them to a wider variety of attack vectors, help security analysts understand and verify the basis of each prediction or alert generated by the system.

6.3.4 Benchmarking with other datasets: While the CICIDS2017 dataset was highly effective for this study, future research should consider using additional datasets (e.g., NSL-KDD, UNSW-NB15) to evaluate the generalizability and robustness of the model.

6.4 Further studies: Future research can explore the following areas to enhance IDS capabilities further:

6.4.1 Deep learning integration: Incorporating LSTM, CNN, or Transformer-based models for feature extraction and temporal analysis.

6.4.2 Feature engineering: Application of advanced feature selection and dimensionality reduction techniques (e.g., autoencoders).

6.4.3 Explainability: Integrating explainable AI (XAI) methods to improve trust and transparency in IDS predictions.

6.4.4 Multi-dataset generalization: Evaluating the model on additional benchmark datasets (e.g., NSL-KDD, UNSW-NB15) to validate generalizability employed the CICIDS2017 dataset, which contains realistic traffic data simulating various types of attacks and benign behaviors. The dataset's inherent class imbalance common in intrusion detection scenarios was addressed using the Synthetic Minority Oversampling Technique (SMOTE), ensuring more balanced training and improved sensitivity to minority attack classes. The model was implemented using Python in the Anaconda Navigator environment.

References

[1] P. Ioulianou, V. Vasilakis, I. Moscholios, and M. Logothetis, "A Signature-based Intrusion Detection System for the Internet of Things," in *Information and Communication Technology Form*, AUT: York, Jun. 2018. [Online]. Available: https://eprints.whiterose.ac.uk/id/eprint/133312/

[2] J. Bharadiya, "Machine Learning in Cybersecurity: Techniques and Challenges," *European Journal of Technology*, vol. 7, no. 2, pp. 1–14, Jun. 2023, doi: 10.47672/ejt.1486.

[3] H. A. Alghamdil and D. S. Alshamrani, "Developing A Security Scheme For Banking Networks Based On Honeypot Technology," *International Journal of Engineering Research & Technology*, vol. 12, no. 12, Dec. 2023, doi: 10.17577/IJERTV12IS120048.

[4] E. Etuh, F. S. Bakpo, and E. A. H, "Social Media Networks Attacks and their Preventive Mechanisms: A Review," 2022, *arXiv*. doi: 10.48550/ARXIV.2201.03330.

[5] A. A. Abdulrahman and M. K. Ibrahem, "Evaluation of DDoS attacks Detection in a New Intrusion Dataset Based on Classification Algorithms," *Iraqi Journal of Information & Communications Technology*, vol. 1, no. 3, pp. 49–55, Feb. 2019, doi: 10.31987/ijict.1.3.40.

[6] M. E. U. S. - and M. M. -, "Intrusion Detection Systems: Enhancing Network Security in the Digital Age," *International Journal For Multidisciplinary Research*, vol. 5, no. 5, p. 7083, Oct. 2023, doi: 10.36948/ijfmr.2023.v05i05.7083.

[7] J. P, J. Shareena, A. Ramdas, and H. A P, "Intrusion Detection System for IOT Botnet Attacks Using Deep Learning," *SN Computer Science*, vol. 2, no. 3, p. 205, May 2021, doi: 10.1007/s42979-021-00516-9.

[8] F. S. Alsubaei, "Detection of Inappropriate Tweets Linked to Fake Accounts on Twitter," *Applied Sciences*, vol. 13, no. 5, p. 3013, Feb. 2023, doi: 10.3390/app13053013.

[9] C. M. K. Ho, K.-C. Yow, Z. Zhu, and S. Aravamuthan, "Network Intrusion Detection via Flow-to-Image Conversion and Vision Transformer Classification," *IEEE Access*, vol. 10, pp. 97780–97793, 2022, doi: 10.1109/ACCESS.2022.3200034.

[10] F. L. De Caldas Filho *et al.*, "Botnet Detection and Mitigation Model for IoT Networks Using Federated Learning," *Sensors*, vol. 23, no. 14, p. 6305, Jul. 2023, doi: 10.3390/s23146305.

[11] Tewogbade Shakir Adeyemi and Ajasa Muhammed, "Botnet attack detection in IoT using machine learning models," *International Journal of Science and Research Archive*, vol. 12, no. 1, pp. 2221–2229, Jun. 2024, doi: 10.30574/ijsra.2024.12.1.0936.

[12] B. Düzgün, A. Çayır, U. Ünal, and H. Dağ, "Network intrusion detection system by learning jointly from tabular and text-based features," *Expert Systems*, vol. 41, no. 4, p. e13518, Apr. 2024, doi: 10.1111/exsy.13518.

Biographical notes

Abraham Danlami , Senior System Analyst at Federal University Wukari Taraba State; Nigeria, BSc Ed. (Computer Science Education, University of Jos, Plateau State Nigeria , MSc Information Technology at Open University of Nigeria , A PhD Student in the field of Cyber Security, Computer Science Department Faculty of Computing and Artificial intelligence Taraba State University Jalingo Taraba State Nigeria. His Specialization is Cyber Security.

Etemi Joshua Garba, Professor at Modibbo Adama University Yola, Nigeria. BSc (Computer Science, ITMO St. Petersburg, Russia), MEng Software Engineering (ITMO St. Petersburg, Russia), PhD (Computer Science, MAUTECH Yola, Yola). Specialization is Data Science (Multimedia Data Analytics), Software Engineering, Digital Economy, and Multimedia Data Analytics (in Data Science and Artificial Intelligence).

Dr. Yusuf Musa Malgwi is a Senior Lecturer and Head of Department, Computer Science at Modibbo Adama University, Yola, Adamawa State, Nigeria. He earned his Bachelor of Technology Degree with Honors in Computer Science from the Federal University of Technology, Yola Adamawa State, Nigeria, in 2006. He proceeded to obtain a Master of Science in Computer Science from Adamawa State University, Mubi, Adamawa State, Nigeria in 2014, and later completed his Ph.D. in Computer Science at Modibbo Adama University Adamawa State, Nigeria in 2019, with a specialization in Machine Learning and Medical Informatics.

Dogo Siyani Ezra, Lecturer I at Taraba State University, Jalingo Taraba State, Nigeria. BSc in Computer Science, Adamawa State University, Mubi, Adamawa, Nigeria , MSC Information Systems , Yola, Adamawa State, Nigeria, A PhD Student in the field of Artificial Intelligence, Computer Science Department , Faculty of Computing and Artificial intelligence at Taraba State University Jalingo, Taraba State Nigeria. She Specializes in the field of Artificial Intelligence.