Comparative Performance Measures of Machine Learning Algorithms in PHR Security

Uyinomen O. Ekong
Department of Cybersecurity, Faculty of Computing, University of Uyo, Uyo, Nigeria

Samuel B. Oyong
Department of Computing, Topfaith University, Mkpatak, Nigeria

Victor E. Ekong
Department of Software Engineering, Faculty of Computing, University of Uyo, Uyo, Nigeria

Edith O. Abengowe
African Centre for Excellence on Enhanced Technology, National Open University of Nigeria, Abuja, Nigeria

ABSTRACT
Personalized health records (PHRs) are digital health records managed by patients to monitor their health information online and potentially share the information with trusted individuals such as physicians, nurses, or pharmacists. Unfortunately, digital health records have become highly valuable on the dark web, an illegal marketplace for stolen health information and related services. This poses a significant threat to patient privacy and security, increasing the risk of malware attacks and exposing individuals to potential embarrassment, ridicule, or even litigation against healthcare institutions for ethical breaches. Common attack agents include viruses, worms, Trojans (e.g., ransomware), key loggers, and rootkits. Types of attack include denial of service (DOS), Probe, remote to local (R2L) and user-to-root (U2R) exploits. To address these threats, this study used machine learning (ML) models such as Random Forest, Decision Tree, K-Nearest Neighbor, Naïve Bayes, and Logistic Regression, leveraging bagging, an ensemble learning technique. The performances of the trained models were evaluated and compared. NSL-KDD dataset was sourced from Kaggle and categorized into normal and attack classes. The dataset was imbalanced with fewer attack samples. To improve model performance, Synthetic Minority Oversampling Technique (SMOTE) was employed, with features extracted using information gain, normalization, principal component analysis, and one-hot encoder. The models learned normal patterns in the dataset to classify malware from normal applications, achieving accuracies of 98% (Random Forest), 98% (Decision Tree), and 96% (K-Nearest Neighbor). This study enhances data security, reduces privacy threats, and fosters patient trust in sharing health records with trusted medical staff.

General Terms
Random Forest (RF), Decision Tree (DT), K-Nearest Neighbor (K-NN), Gaussian Naïve Bayes (GNB), Logistic Regression (LR)

Keywords
Electronic Health Records, Health Security, Machine Learning, Personalized Health Records (PHRs).

1. INTRODUCTION
The era of COVID-19 pandemic held the world to a standstill in lockdown and social distancing measures**.** This informed the need to use patient portals and Telehealth systems to reach out to the sick, especially in rural communities, and those seeking medical information on their health issues. Personalized Health Records (PHR) are electronic health records that are managed by the patient to track their health information online and possibly share the information with trusted few like physicians, nurses, and pharmacists [1, 2, 3], while Telehealth system provides healthcare to rural communities. It was initiated and mostly used during COVID-19 lockdown and social distancing measures. It reduces stigmatization, the anxiety of patients, and the guilt of addiction of in-patients when meeting face-to-face with healthcare professionals [4]. Similarly, patient portals, also called tethered PHRs, enable patients to view part of their medical records in EHRs [5]. Physicians and healthcare providers are selective on what part of a patient's health record can be made available to the patient. This action is seriously frowned upon by [4].

Like all digitized health records, PHR are vulnerable to cyber-attacks and data breaches; more so, patients often lack access to and control over their medical records. This hinders the portability and interoperability of their data across healthcare providers [6, 2]. To promote patient-centered healthcare system, electronic personalized health records (ePHRs) are adopted, which enable patients to have absolute control over their health histories, allergies, treatments, medications, and even share their record information with trusted few [7, 6, 4]. The sensitive nature of personal health information makes it an attractive target for cybercriminals. As such, security breaches are mostly on patient's privacy. More so, the average cost of a breached record in the dark web is $219, but that of a healthcare record is $429, as of 2019 [8]. According to [6], one of the biggest challenges in healthcare settings is the lack of interoperability. Many healthcare systems work independently and store their data in databases (proprietary) using different standards that do not enable communication with each other. This makes it difficult to share patients' information between different providers, leading to poor coordination. ePHRs are used by patients who may not be well-schooled in digital information and communications technology (ICT) and deserve to be protected.

This article intends to design and develop a framework that uses machine learning tools, such as Random Forest (RF), Decision Tree (DT), K-Nearest Neighbor (K-NN), Gaussian Naïve Bayes (GNB), and Logistic Regression (LR), to train

classifiers (models) using National Science Laboratory – Knowledge Discovery in Databases (NSL-KDD) dataset. Machine learning (ML) is an aspect of artificial intelligence (AI) that focuses on training computers and computer programs to learn from past experiences without being programmed [9]. The trained models use the acquired knowledge to detect or classify cyber intrusions (malicious applications) from normal applications on ePHRs without human intervention. The performances of the machine learning tools will be compared to ascertaining the best performing model in this research work.

The remaining sections include section 2, which discusses the related works in medical healthcare system, Section 3 provides the steps used in solving the intrusion problem; and section 4 provides the results, confusion matrix analysis and evaluation of models performances, while section 5 concludes the research with suggestions for further works.

2. LITERATURE REVIEW

The digital health records system is the engine room of the electronic health system (EHS). The data records are used for various purposes, from reading the patient's history to analyzing the patient's disease condition in the laboratory, and to taking effective decisions by stakeholders. Timely access to these records in healthcare facilities promotes efficient healthcare delivery. The critical nature of DHRs notwithstanding, they are constantly being attacked by malware (malicious software) to steal information, disrupt operations, and inflict pain on the innocuous user. The stolen information is sold in the dark web (the illicit market for hackers) as they are highly priced [10, 11]. To curb the menace of malware and internal intruders (persons working within the organization), the research community is putting up a fight by using ensemble learning or hybridization to combine ML tools and counter these attacks, safeguard the data records, and reduce morbidity and mortality rates.

In [12], a comparison of threat modeling methods to determine their suitability for identifying and managing healthcare-related threats in cloud computing technology (CCT) is discussed. The article identified threat modeling in pervasive (TMP) computing to be the best method to use in healthcare security because it can be combined with attack tree (AT), attack graph (AG), and practical threat analysis (PTA) to identify cloud-related threats for healthcare.

Similarly, [13] presented a tele-clinical diagnostic system for effective delivery of medical services to patients in an academic environment (within a rural or isolated community). However, the model was not secure enough as it used password-based authentication to control access to the data. In [14] a cloud-based electronic health record framework is proposed, that is capable of automating storage, retrieval, updating, and maintaining patients' medical records in Nigeria. Although ML tools such as RF, DT, KNN, and NB were used to build the framework, the article did not mention the type of software developed and the programming language used. In [11], the security challenges faced by healthcare workers are determined while performing their duties. The article also attempted to investigate anomaly practices in healthcare systems using big data and ML techniques. The study identified that the healthcare sector experienced 503 data breaches and 15 million records were compromised in 2018 [11].

In another development, [15] conducted a systematic review of security threats, procedures, and controls associated with cloud storage. Other concerns associated with cloud storage include poor data visibility, storage sinks without protective pointers, and enormous data spills. The article summarized security risks in cloud computing technology (CCT). In[16] incidences of account hijacking, data sanitization, data control, and harmful insiders were reviewed.

In [17], it was observed that electronic health systems (EHS) have become popular targets for ransomware attacks, phishing, and insider threats. Studies in [18] observed that cyber-health in Nigeria is still in its infancy, mostly with healthcare systems and services fragmented, disjointed, and heterogeneous, with strong local economic content. The study aimed at developing a trust management system for guaranteed privacy and confidentiality of patient's health records across hospitals.

In [9], advances in EHRs and the proliferation of genomic data leading to personalized medicine were reviewed. Indeed, genomic sequencing has opened doors to various practices in medicine, such as personalized medicine, cancer detection and treatment, pre- and perinatal testing, in-patient management of critically ill infants, and more.

In [5], the effects of patient-centered digital health records on critical and patient-reported outcomes, healthcare utilization, and satisfaction among patients with chronic diseases were studied. The authors concluded that the use of hospitalized individuals with chronic health issues is beneficial to healthcare utilization, treatment adherence, and self-management. In [19], a systematic review of evidence regarding factors that influence patients' use of ePHR was carried out. ePHRs (tethered) allow patients to view part of their health records in EHRs and can share them with trusted others.

In [20], a study on ePHRs and patient-centered services were x-rayed. The authors lamented the deliberate act of resistance to sharing notes with patients perpetrated by physicians. As such, physicians not only write illegibly but sometimes construct insulting phrases or words in the patients' record history. Poor documentation practices may always lead to patient safety risks, so the team observed and concluded that physicians should improve documentation practices. In [21], the impact of DHRs to personalized healthcare and public healthcare was discussed by presenting their roles, challenges, and potential future.

Another study in [3], discussed the healthcare delivery services in Nigeria, which are still practiced using the traditional paper-based approach. Hospitals and clinics are not following the global trends of inculcating ICT that will digitize patients' records and clinical services and stores them in the cloud. The authors rated the use of EHRs to be between 18% and 23%, and physicians' literacy level on ICT was rated at 15% only.

In [2], an assessment of patient's literacy level on ICT to effectively implement an electronic personalized healthcare record (ePHRs) system were performed. In the ePHRs system, patients will also upload scanned medical documents in PDF format concerning their diseases and treatments, for use by trusted few in decision-making. The article described the use of ePHRs in real-world pandemic cases like COVID-19, floods, and earthquakes, with restrained movement, lockdown, and social distancing measures.

The report in [7] explored the impact of EHRs on patients' care and outcomes. The article identified the benefits and challenges of EHRs and provided insights into their integration to clinical practices. The article further stressed on interoperability of EHRs to exchange and use patients' information within and between healthcare institutions. The absence of interoperability in EHRs can lead to fragmented care, duplicate test results, and medication errors. What is happening in Nigeria today is,

that different healthcare providers use different data standards, different storage formats, and fear of data corruption to limit seamless information exchange. In another development, [22] determined the true use of EHRs by staff and stakeholders as perceived by healthcare professionals concerning data quality and reuse, compared to the era of paper-based systems. Similarly, [23] sought to know why EHRs were not fully imbibed in developing countries like Nigeria. The article concluded that fear, education, anxiety, and infrastructure accessibility are contributing factors.

3. METHODOLOGY

In this section, Random Forest algorithm is used to train models (DT, LR, and GNB). KNN is used to measure the distance between each training dataset record in memory and the target object using Euclidean distance measure. These models will predict the class labels being a supervised learning problem. The predictions of the models are compared with the expected labels using confusion matrix. The models are then evaluated for performance using metrics.confusionMatrix() function. To develop the framework that will be used to detect the menace of malware threats, the data to train the models, which is the NSL-KDD dataset, was obtained from Kaggle data repository. Using SMOTE technique, the data types were balanced, and this balanced dataset was pre-processed using mini-max normalization, principal component analysis (PCA), and One_Hot_Encoder() function. The pre-processed dataset was split into a training dataset (80%) and a test dataset (20%) using train_test_split() function, in Python programming language. Using the RandomForestClassifier() function, the base models (classifiers) such as NB, DT, and LR were trained using the fit() function. Similarly, using Euclidean measure, the nearest neighbour distances were measured, and the target object was assigned to the neighbour with the majority vote. The labels were predicted using each trained classifier with the model.predict() function. The predictions of the trained models and the predictions of KNN were aggregated to form a consensus classifier called soft vote. Soft vote was preferred to hard vote in this work because hard vote predicts the class label with the majority vote, whereas soft vote averages predicted probabilities from all models to choose the class with the highest average. Soft vote usually yields better performance comparatively. The soft vote classifier was trained and used to predict the test dataset and produce results. The predicted results (labels) of soft vote were compared with the actual labels using confusion matrix. The confusion matrix analysis generated true positive (TP), true negative (TN), false positive (FP), and false negative (FN) classifications; the standard metrics such as accuracy, precision, recall, f1-score, and area under the curve (AUC) were also computed. The reported alerts were sent in the form of Normal and attack types such as denial of service (DOS), Probe, remote to local (R2L), and user to root (U2R) to the user via the Hospital, which is the control mechanism, and the results were displayed on the desktop, laptop, or smartphone. Figure 1 depicts the architecture of the processing steps used in the proposed Framework.

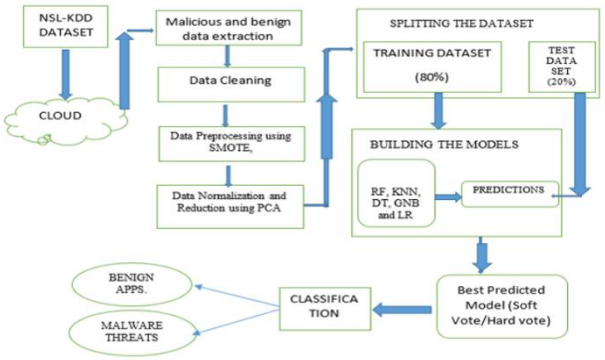

Figure 1: Architecture of the proposed framework used to classify malware

3.1 Data Preprocessing

3.1.1 One-Hot Encoder transforms Categorical Variables

To process the categorical features in the dataset, Label Encoding was applied to convert the 'protocol_type', 'service', and 'flag' columns into numerical values. Following this, One-Hot Encoding was utilized to further encode these categorical features, generating binary columns (0, 1) for each category within 'protocol_type', 'service', and 'flag'. The resultant encoded features were then concatenated with the original dataset as a new feature set, which increased the dimension of the dataset to 126 columns, and the initial categorical columns were subsequently removed to finalize the feature engineering process. Table 1 depicts the dataset after the encoding process. The problem with one-Hot-Encoding is that table columns are increased, thereby increasing the overhead, memory usage, and overfitting.

Table 1: A subset of the dataset after one-hot-encoding

protocol_type_tcp	protocol_type_udp	flag_REJ	flag_RSTO	flag_RSTR	flag_S0	flag_S1	flag_S2	flag_S3	flag_SF	flag_SH
1	0	0	0	0	0	0	0	0	1	0
0	1	0	0	0	0	0	0	0	1	0
1	0	0	0	0	0	0	1	0	0	0
1	0	0	0	0	0	0	0	0	1	0
1	0	0	0	0	0	0	0	0	1	0

On the other hand, the target variables (labels) were manually encoded such that each class is mapped to a numerical value {0: ‘DOS’, 1: ‘PROBE’, 2: ‘R2L’, 3: ‘U2R’, and 4: ‘NORMAL’}.

3.1.2 Balancing the Dataset

The imbalanced class dataset underwent further processing to achieve a balance, which is crucial to avoid bias in model performance. Using the Synthetic Minority Over-sampling Technique (SMOTE) from the `imbalance` library, the minority classes were oversampled to match the number of instances in the majority class. Initially, the class distribution showed a significant imbalance, with the 'normal' and 'DOS' classes having a much higher count compared to 'PROBE', 'R2L', and 'U2R'. After applying SMOTE, the dataset was balanced, resulting in an equal number of instances (67,342) for each class. This balanced dataset is crucial for training a model that performs well across all classes. Figure 2 provides a visual representation of the class distribution before and after the balancing process.

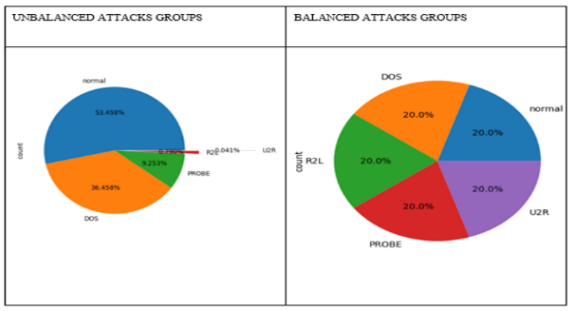

Figure 2: Chart of balanced and imbalanced data types

3.2 Principal Component Analysis (PCA)

Given the high-dimensional dataset, which includes redundant features, missing features, outliers, errors and noise, Principal Component Analysis (PCA) was applied to reduce the dataset to a much smaller set. The used set was generated after setting a threshold or cut off point of 95%. The Eigen vector with the highest Eigen value forms the first principal component, amongst those within the threshold. The explained variance ratio was analyzed to determine the number of components required to capture at least 95% of the total variance. A plot of the cumulative explained variance against the number of principal components was generated to visualize this relationship. Figure 3 illustrates the explained variance against the number of principal components, highlighting the point where 95% of the variance is captured.

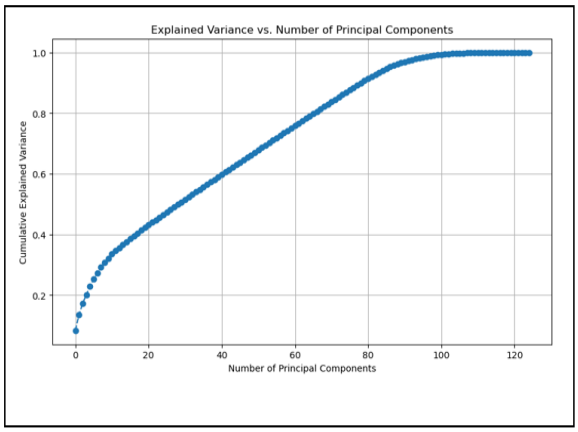

Figure 3: Plot of explained variance versus principal components

4. RESULTS AND DISCUSSIONS

4.1 Results

The results are predicted from test dataset and matched with the actual data classes such as NORMAL, DOS, PROBE, U2R, and R2L. As a supervised learning problem, the actual values of this test dataset are known [24]. Subsequently, principal component analysis (PCA) was applied to reduce the number of features, and the resulting principal components were used to create a new DataFrame with records and columns as depicted in Table 2. Given the new dataFrame, the original features have been replaced with the derived principal component, PC1 to PC90, including the labels. The predicted labels of the soft vote model will be matched with the expected (or actual or known) class types for analysis by confusion matrix. The extract presented in Table 2 has only ten records out of 125,973 records, for want of space. Observe that extracts from the results sheet are presented in sets of eight columns and ten row s each with labels as "class" and "predicted" as depicted in the last group of columns in the first ten (10) rows or records in Table 2. For want of space, only the first group and the last group of records will be displayed in Table 2. This is a convenient way of presenting the ninety columns of the set of records extracted, including the label columns.

Table 2: Extracts of results presented in groups of Records and Columns

SN	PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8
5188	-1.43341	0.596405	-0.44467	-1.14996	2.509284	1.120042	-1.56106	0.849499
110012	1.208092	0.588724	0.065598	-2.68224	3.190255	1.539098	-2.00343	1.376987
289664	4.76409	-4.17956	-0.4102	0.516971	0.221604	0.315511	-0.40474	0.2378
11733	-3.80554	0.220176	-6.16195	0.148521	-1.61079	0.749816	-0.13229	-0.60043
112761	4.891938	-4.20786	-0.46565	0.473168	0.239561	0.282929	-0.35332	0.18615
91777	-1.45234	-0.159	1.378551	-0.01186	0.491553	-0.4992	0.621893	-0.73715
241358	-2.60999	0.093091	0.456637	1.193181	0.238186	-1.14077	0.08406	1.828881
18724	-1.99626	-0.54995	-6.02778	-4.54614	0.817659	0.947084	1.188159	-4.17533
37521	-0.59901	0.710094	0.353775	-0.04459	-0.22459	0.124985	0.222975	-0.80542

PC81	PC82	PC83	PC84	PC85	PC86	PC87	PC88	class	Predicted
0.042053	-0.24301	-0.06745	0.331163	-0.51075	0.21299	-0.04361	1.33562	normal	normal
-0.05209	-0.6451	0.110284	0.252581	-0.59601	0.269111	0.228987	1.851491	DOS	DOS
-0.05794	-0.81644	0.267012	0.230452	-1.01476	0.258604	0.561143	-0.09836	U2R	U2R
-0.0068	-0.04942	-0.01854	0.010824	-0.13111	0.027481	-0.0334	-0.12842	DOS	DOS
-0.06084	-0.81933	0.265786	0.247251	-1.06038	0.265955	0.583791	-0.2435	normal	normal
-0.01655	-0.05213	-0.04446	-0.04408	0.168358	-0.02896	0.016122	-0.00182	normal	normal
-0.00846	-0.05918	0.0466	0.036568	-0.05721	-0.00211	0.010228	0.008848	R2L	R2L
-0.03255	-0.58336	0.127527	0.312746	-0.14551	0.064929	0.105298	0.175536	DOS	DOS
0.004521	0.308484	-0.03548	-0.19775	0.293882	-0.10385	-0.08376	0.216704	PROBE	PROBE

4.2 Confusion Matrix

The confusion matrix depicts the relationship between the actual (or expected) values and the predicted values. The confusion matrices depicted in this work are multi-class confusion matrices.

4.2.1 K-Nearest Neighbour (KNN) Classification

Figure 4 depicts the confusion matrix of KNN. These predictions were evaluated using a custom_evaluate_model() function, with performance visualized using metrics.confusionMatrix() function and the detailed standard metrics displayed or printed using classification_ Report() function, highlighting metrics such as accuracy, precision, recall, F1-score and area under the curve (AUC).

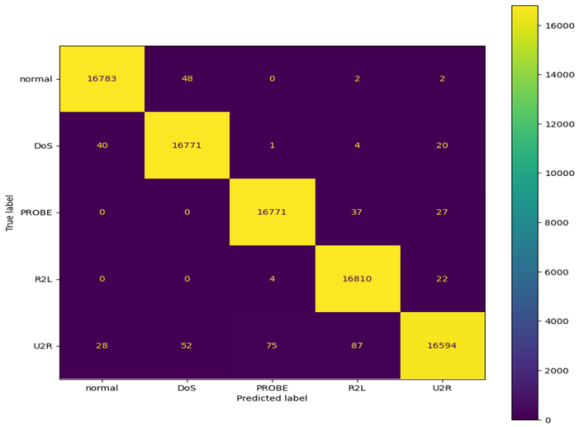

Figure 4: Plot of Confusion Matrix for KNN

From Figure 4, KNN predicted true positive (TP) records as depicted in the main diagonal of the confusion matrix. Other predictions include true negative (TN), false positive (FP), and false negative (FN). Others are true positive rate (TPR) and false positive rate (FPR). In malware detection problems, TP refers to the number of normal features predicted as normal;

TN refers to the number of malicious records predicted as malware, FP refers to the number of true positive records predicted as malicious; while FN refers to malicious records predicted as normal records. TPR is defined as the number of true positives divided by the total number of true positives and false negatives, as depicted in Equation 1.

$$\text{TPR} = \frac{TP}{TP+FN} \quad (1)$$

FPR is defined as the number of false positives divided by the sum of false positives and true negatives, as depicted in Equation 2.

$$\text{FPR} = \frac{FP}{FP+TN} \quad (2)$$

Precision is defined as true positive numbers divided by the sum of positive numbers and false positive numbers, as depicted in Equation 3.

$$\text{Precision} = \frac{TP}{TP+FP} \quad (3)$$

Recall is defined as true positive numbers divided by the sum of true positive numbers and false positive numbers, as depicted in Equation 4.

$$\text{Recall} = \frac{TP}{TP+FN} \quad (4)$$

Accuracy is defined as the sum of true positive numbers and true negative numbers divided by the total number of instances, as depicted in Equation 5.

$$\text{Accuracy} = \frac{TP+TN}{TP+FP+FN+TN} \quad (5)$$

The KNN model demonstrated strong performance with an Accuracy of 0.98816, Precision of 0.988158, Recall of 0.988170, F1-Score of 0.988152, and an overall receiver operating characteristic- area under the curve (ROC-AUC) score of 0.99740. Table 3 depicts KNN performance metrics.

Table 3: KNN Performance Metrics

Model	Accuracy	Precision	Recall	F1-Score	ROC-AUC
KNN	0.98816	0.988158	0.988170	0.988152	0.9991

The ROC-AUC scores for individual class labels were also impressive, with Normal at 0.9980, DOS at 0.9974, PROBE at 0.9982, R2L at 0.9986, and U2R at 0.9949. Figure 5 depicts the ROC-AUC plot depicting the relationship between false negative rate (FNR) and false positives rate (FPR) [25].

ROC-AUC is a performance metric for binary classification problems that measures a model's ability to distinguish between classes across all classification thresholds (it can also be used for multi-class classification using the One-vs-All (OvA) approach. It plots the true positive rate (TPR) against the false positive rate (FPR), with an AUC of 1.0 indicating a perfect model, while 0.5 suggests random guessing as shown in Figure 5. Unlike accuracy, ROC AUC evaluates how well the model separates classes regardless of the specific probability threshold chosen, making it robust for imbalanced datasets.

The results indicate that the KNN model is highly capable of accurately detecting and distinguishing between different types of network attacks, making it a reliable choice for network intrusion detection. However, because KNN does not learn a model during training but instead stores all training data, it must calculate the distance between a new point and every existing data point during prediction. This makes it extremely slow with large datasets or real-time applications. Also, KNN's performance degrades rapidly as the number of features (dimensions) increases.

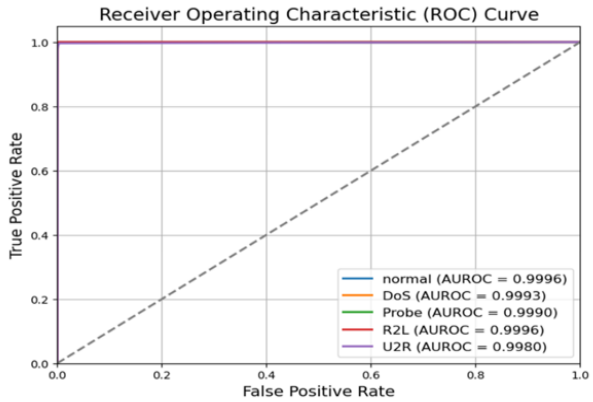

Figure 5: The ROC-AUC curve of KNN classifier

4.2.2 Random Forest Classification

To train and evaluate a Random Forest model, it was initialized with 500 estimators to ensure robust and stable model performance by aggregating the predictions from multiple decision trees, thus reducing over fitting. Entropy (the degree of impurity in a dataset) was used as the criterion to measure the quality of splits, focusing on maximizing the information gained (and reducing the entropy) at each split. The model was trained on the training dataset (80%), and predictions were made on the test dataset (20%). Figure 6 depicts the confusion matrix of RF.

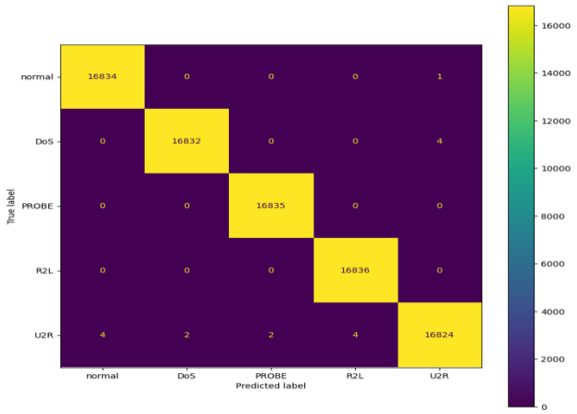

Figure 6: Confusion Matrix for Random Forest (RF)

Figure 6 shows that RF correctly predicted TP records as depicted in the main diagonal of the confusion matrix. To assess the model's performance, a custom evaluate model() function was used, and a confusion matrix was used to compare the predictions with the actual or expected values. Table 4 depicts the performance metrics for RF.

Table 4: Performance metrics for RF classifier

Model	Accuracy	Precision	Recall	F1-Score	ROC-AUC
Random Forest Classifier	0.99932	0.999322	0.988170	0.999319	1.0

The Random Forest Classifier demonstrated excellent performance metrics, achieving an Accuracy of 0.99932, Precision of 0.999322, Recall of 0.988170, F1-Score of 0.999319, and a perfect overall ROC-AUC score of 1.0. The ROC-AUC scores for each class label were outstanding with 1.0 in all the classes, as depicted in Figure 7. Specifically, the AUC curve establishes the relationship between FNR and FPR.

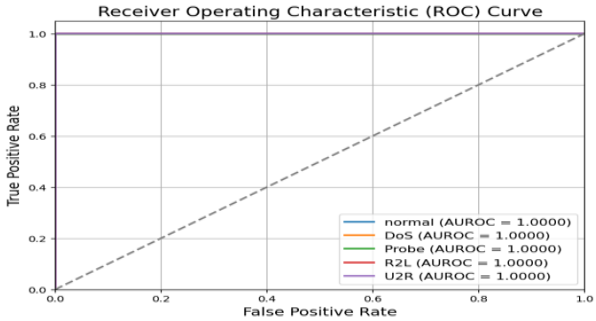

Figure 7: ROC-AUC curve for RF Classifier

Despite these high scores, there are drawbacks. The main drawbacks of Random Forest algorithm in ensemble learning classification are its lack of interpretability**,** high computational cost**,** and significant memory consumption, especially when dealing with large datasets, as evidenced in the confusion matrix with reference to 'DOS' and 'U2R' attack classes. The "black-box" nature is a significant disadvantage in organizations requiring transparent and explainable decision-making. Training many decision trees can be computationally expensive and time-consuming. The model requires significant memory to store all the individual decision trees and their training data specifics (like feature splits and leaf node predictions). Although generally robust to overfitting, it can still over fit if the data is particularly noisy or if there are too many trees that capture irrelevant patterns in the training data.

4.2.3 Naïve Bayes Classification

In training and evaluating a Naive Bayes model, the Gaussian Naive Bayes classifier was used due to its effectiveness in handling continuous data and its assumption of a normal distribution for the features. The model was trained on the training dataset (80%), and predictions were made on the test dataset (20%). Figure 8 depicts the confusion matrix of the Gaussian Naïve Bayes classifier.

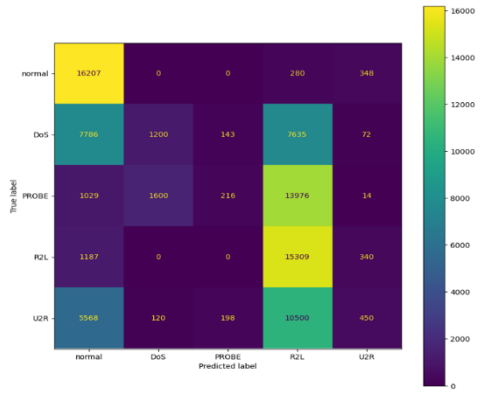

Figure 8: Confusion Matrix for Gaussian Naïve Bayes Classifier

Figure 8 depicts that the Naïve Bayes classifier correctly predicted TP records as indicated in the main diagonal of the confusion matrix. To assess the model's performance, the custom_evaluate_model() function was used and the parameters derived are displayed in Table 5.

Table 5: Naïve Bayes model performance evaluation result

Model	Accuracy	Precision	Recall	F1-Score	ROC-AUC
Naive Bayes Classifier	0.39624	0.410736	0.396438	0.268998	0.7419

The Naive Bayes Classifier displayed significantly lower performance metrics compared to other models, with an Accuracy of 0.39624, Precision of 0.410736, Recall of 0.396438, F1-Score of 0.268998, and an overall ROC-AUC score of 0.7419. The ROC-AUC scores for individual class labels were class 0 (Normal) at 0.9032, class 1 (DoS) at 0.4746, class 2 (PROBE) at 0.6181, class 3 (R2L) at 0.8777, and class 4 (U2R) at 0.8448 as depicted in Figure 9. Figure 9 compares the FPR and TPR to derive the area under the curve (AUC).

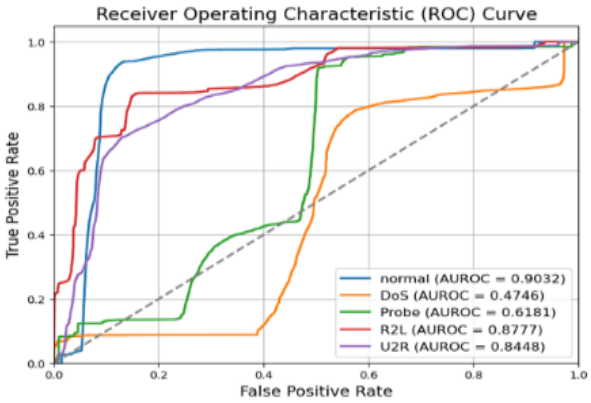

Figure 9: Comparison of FPR and TPR of NB Classifier

The ROC-AUC curves highlight the classifier's challenges in correctly identifying the different types of network attacks. The high number of misclassifications in Figure 9, especially among 'DoS', 'PROBE', and 'U2R' attacks, suggest that the Naive Bayes Classifier struggled with the complexity of the dataset and the variability within attack classes. These results indicate that, while Naive Bayes provides some insight, it may not be the best choice for accurately classifying network intrusions in this context. Other reasons could be because of its assumption that all features are independent of each other given the class label. In real-world data, features are rarely independent; they are often correlated. This misrepresents the combined influence of features, leading to skewed probabilities and reduced accuracy. While it works well with smaller datasets, its performance is heavily reliant on the quality and distribution of the training data. If the training data is highly skewed or imbalanced, the model will struggle, often favoring the majority class.

4.2.4 Decision Tree Classification

To train and evaluate a Decision Tree (DT) model, entropy was used to initialize it as the criterion to measure the quality of splits, focusing on maximizing the information gained at each split. The model was trained on the training dataset (80%), and predictions were made on the test dataset (20%). Figure 10 depicts the confusion matrix of DT.

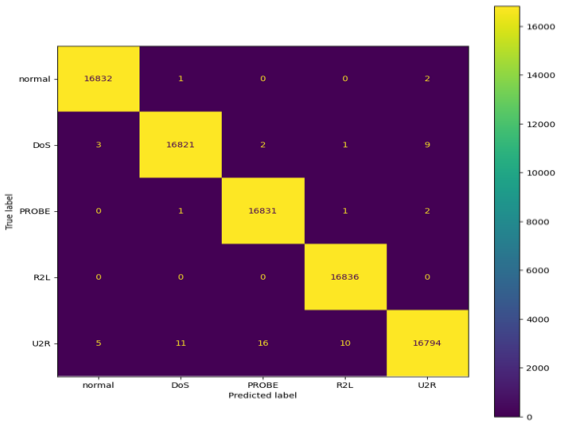

Figure 10: Confusion Matric for DT Classification

The confusion matrix depicts TP records in the main diagonal. The low number of false positives and false negatives across all classes further underscores the model's reliability and accuracy in classifying network intrusion types, making it a robust choice for this task. To assess the model's performance, metrics.confusionMatrix() function was used to compute the metrics. The resulting performance metrics are recorded in Table 6 as accuracy, precision, recall, F1-score, and ROC-AUC

Table 6: Performance metrics for DT classifier

Model	Accuracy	Precision	Recall	F1-Score	ROC-AUC
Decision Tree Classifier	0.99804	0.410736	0.998042	0.998041	0.9988

The Decision Tree Classifier demonstrated impressive performance metrics, achieving an Accuracy of 0.99804, Precision of 0.410736, Recall of 0.998042, F1-Score of 0.998041, and an overall UAC-ROC score of 0.9988 as depicted in Table 6. The ROC-AUC scores for individual class labels were class 0 (Normal) at 0.9999, class 1 (DoS) at 0.9995, class 2 (PROBE) at 0.9997, class 3 (R2L) at 0.9999, and class 4 (U2R) at 0.9987 as depicted in Figure 11.

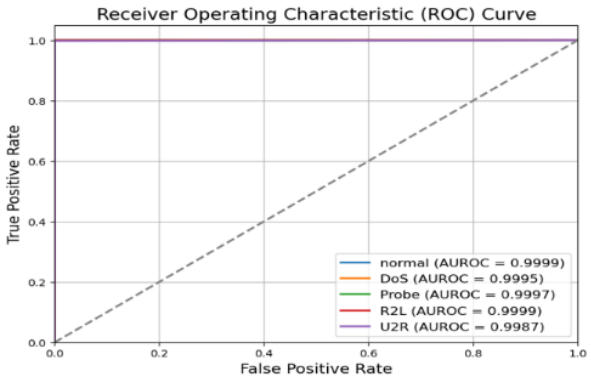

Figure 11: ROC-AUC curve for DT classifier

Despite the impressive performance, decision trees have drawbacks such as high variance (instability) as small, minor changes in the training data can result in a completely different tree structure, making individual models unreliable; and a tendency to over fit training data with complex trees, particularly deep ones that learn noise in the training set rather than just the underlying patterns, leading to poor performance on new data. They are greedy, selecting the best split at each step. This local optimization does not guarantee the overall best tree, and struggles to model linear relationships, leading to poor generalization when used alone.

4.2.5 Logistic Regression Classification

In training and evaluating Logistic Regression model, the classifier was initialized with a maximum of 1000 iterations to ensure convergence. The model was trained on the training dataset, which comprised 80% of the total data, and predictions were made on the test dataset, which comprised the remaining 20%. Figure 12 depicts the confusion matrix of LR.

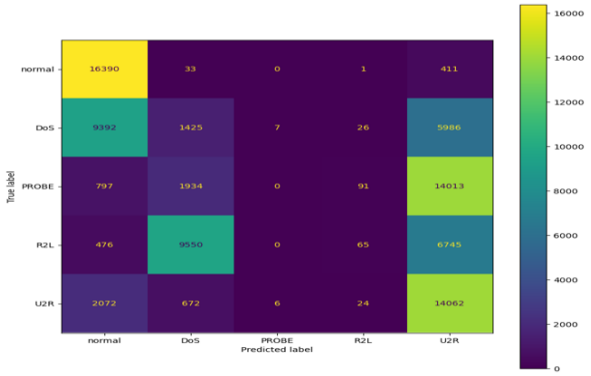

Figure 12: Confusion Matric of LR classifier

From Figure 12, Logistic Regression true positive predictions are lined in the main diagonal of the confusion matrix, while the misclassifications are outside the main diagonal. To assess the model's performance, metrics.confusionMatrix() function was used to derive the metrics. The resulting performance metrics are recorded in Table 7 as accuracy, precision, recall, F1-score, and AUC of ROC.

Table 7: Performance evaluation of LR classifier

Model	Accuracy	Precision	Recall	F1-Score	AUC
Logistic Regression	0.37945	0.26451	0.379460	0.259775	0.7421

In evaluating the Logistic Regression model using the NSL-KDD dataset, the performance metrics highlight the model's strengths and weaknesses. The model achieves a moderate overall accuracy of 37.945%, with low recall and precision resulting in an F1-Score of 25.9775% and a ROC of 0.7421. The ROC-AUC scores for individual classes labels were class 0 (Normal) at 0.9330, class 1 (DoS) at 0.2893, class 2 (PROBE) at 0.6684, class 3 (R2L) at 0.9354, and class 4 (U2R) at 0.8845 as depicted in Figure 13. Specifically, the AUC curve establishes the relationship between FNR and FPR [25]. Other reasons for its poor performance include Logistic regression assumes a linear relationship between the independent variables and the log-odds of the dependent variables. If the underlying relationship is non-linear, the model will underperform. It can only create linear decision boundaries. If the data is not linearly separable such as in complex datasets with intricate, non-linear interaction logistic regression will fail to capture the patterns effectively. If one class significantly outnumbers the other, the model may become biased toward the majority class, performing poorly on the minority class.

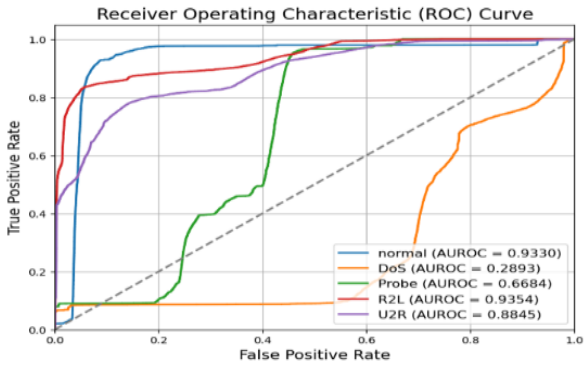

Figure 13: ROC-AUC curve for LR classification

4.2.6 Ensemble Learning (Soft Vote) Classification

The ensemble learning approach employed in this implementation leverages the strengths of multiple individual classifiers to enhance predictive performance [26, 27]. The voting classifier is the core of this strategy, combining five distinct models: K-Nearest Neighbors (KNN), Random Forest (RF), Naive Bayes (GNB), Decision Tree (DT), and Logistic Regression (LR). Soft voting is utilized, where the predicted class probabilities from each model are averaged, and the class with the highest average probability is selected. This ensemble method ensures a more balanced and robust prediction. The weights assigned to each model slightly favoring KNN and Decision Tree reflect their relative contributions to the ensemble, fine-tuning the overall performance. The process involves defining the ensemble model with these estimators, training it on the dataset, and predicting outcomes using the test dataset. Figure 14 depicts the confusion matrix of the soft vote classifier, comparing the existing or expected labels with the predicted ones, and generating TP, FP, TN, and FN values of the comparison

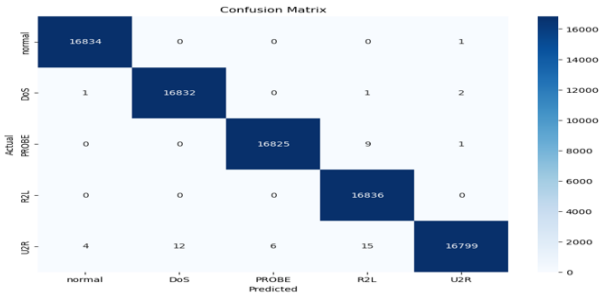

Figure 14: Confusion matric of Soft Vote classifier

From Figure 14, the Soft Vote classifier's true positive predictions are depicted in the main diagonal of the confusion matrix, while misclassifications, which were minimal, are scattered in other locations. Its performance was evaluated using metrics.confusionMatrix() function, with resulting metrics such as accuracy, precision, recall, F1-Score, and ROC-AUC as depicted in Table 8.

Table 8: Performance evaluation of Soft Vote Classifier

Model	Accuracy	Precision	Recall	F1-Score	ROC-AUC
Ensemble Learning (Soft Voting Classifier)	0.99788	0.99788	0.997889	0.99788	0.99996

The Voting Classifier demonstrates exceptional performance, with metrics indicating near-perfect predictive capabilities. The model achieves an accuracy of 0.99788, precision of 0.997879, recall of 0.997885, F1-Score of 0.997880, and a ROC-AUC score of 0.99996. The confusion matrix further supports these results, showing minimal misclassifications and indicating that most instances are correctly classified. The ROC-AUC scores for each class, normal (1.00000), DOS (1.00000), PROBE (1.00000), R2L (1.00000), and U2R (0.99999), highlight the model's excellent capability to distinguish between different attack types and normal traffic. The values are depicted in Figure 15.

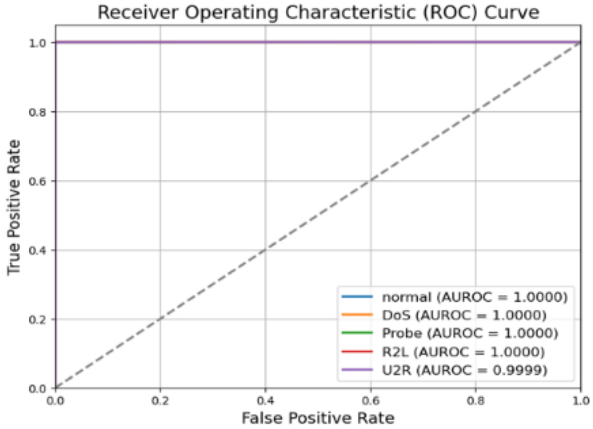

Figure 15: ROC-AUC curve for Soft Vote Classifier

This robustness in classification is critical for network intrusion detection, ensuring high precision and recall across all categories. The ensemble's ability to integrate the predictions from various models, balancing their strengths and weaknesses, leads to a superior overall performance, making it an ideal choice for this task. The strategic weighting of the models further enhances its effectiveness, providing a highly accurate and reliable solution for detecting network intrusions.

4.3 Discussions

In this section, the various models are analyzed to classify malware threats. The models trained and analyzed include; Random Forest (RF), Decision Tree (DT), K-Nearest Neighbor (KNN), Naïve Bayes (NB), and Logistic Regression (LR). The models were evaluated using the following metrics: Accuracy, Precision, Recall, F1-score, and Area under the ROC curve (ROC-AUC). Also used to plot the ROC curve are FPR and FNR values. Table 9 depicts the benchmark that contains analyzed values of the models trained and tested in this study.

4.4 Comparative Analysis of the ML Models

The comparative analysis of different models reveals significant differences in their performance metrics as depicted in Table 9. The Random Forest Classifier outperforms other models with the highest accuracy of 0.99932 and an exceptional ROC-AUC score of 1.0, indicating its superior ability to distinguish between classes. The KNN model also shows strong performance, with an accuracy of 0.98816 and a ROC-AUC of 0.99740. In contrast, the Naive Bayes classifier demonstrates much lower performance across all metrics, with an accuracy of 0.39624 and a ROC-AUC of 0.7419, suggesting it struggles with the dataset's complexity. The Decision Tree Classifier performs nearly as well as the Random Forest with an accuracy of 0.99804 and a ROC-AUC of 0.9988, though its precision score appears to be incorrectly reported, matching that of the Naive Bayes Classifier. Overall, the Random Forest Classifier and Decision Tree Classifier exhibit the best

performance, making them the preferred choices for this classification task.

Table 9: Comparative Analysis of the ML Models

Model	Accuracy	Precision	Recall	F1-Score	ROC
KNN Classifier	0.98816	0.988158	0.988170	0.988152	0.99740
Random Forest Classifier	0.99932	0.999322	0.988170	0.999319	1.0
Naive Bayes Classifier	0.39624	0.410736	0.396438	0.268998	0.7419
Decision Tree Classifier	0.99804	0.410736	0.998042	0.998041	0.9988
Logistic Regression	0.37945	0.26451	0.379460	0.259775	0.7421

5. CONCLUSION

The experimental results of this study indicate that ensemble learning methods do not always outperform single classifiers in network-based attack detection, as is commonly presented in literature. Both have their strengths and weaknesses. Ensemble approaches exhibit higher accuracy, precision, and low false positive rates than single classifiers. However, in terms of Recall, FNR, training, and prediction times, single classifiers perform better. Current data repositories are grossly imbalanced, containing only normal training and testing data classes. For instance, they have only 2180 and 928 web-based attack instances respectively and this may not be suitable for use to train and detect malware in the cloud. Consequently, the creation of novel, public, reliable web-based and network-based data repositories is of utmost need, if the proliferation of malware must be curbed. This study has shown that to improve the performance of trained models, the problem of class imbalance should be investigated, as malware are greatly outnumbered by normal class and DOS attack class. Attack classes mostly affected included PROBE, U2R, and R2L. Therefore, the dynamic analysis method detects malware from a single execution path at a time. Multiple execution paths should can be explored to be able to differentiate different behaviors displayed by suspicious executable files.

6. ACKNOWLEDGMENTS

The authors thank the network security experts consulted in the course of the research.

7. REFERENCES

[1] Davis, S., Roudsari, A. and Cpirtmeu. K. L.2017. Designing Personal Health Record Technology for Shared Decision Making. *Building Capacity for Health Informatics in the Future, F. Lau et al. (Eds.). DOI:10.3233/978-1-61499-742-9-75*

[2] Blazeska-Tabakovska, N; Bacevska, A., Jolevski, I., Beredimas, N., kilintzis, U., Maglaveras, N and Savaosk, S. 2021 Implementation of Cloud-Based Personal Health Record Integrated with IOMT. *Faculty of Information and Communication Technologies* – Bitola, Republic of North Macedonia.

[3] Salam, D. F., Kolade, I. O., Ohairi, B., and Babarimisa, O. 2023. Electronic Health Record: An Underutilized Tool in Nigeria's Healthcare System, *Continental Journal of Applied Sciences., 18(2):40 – 51.* DOI:10.5281/zenodo.8352514

[4] Hagglund, M., McMillan, B., Whittaker, R, and Blease, C. 2022. Patient Empowerment Through Online Access to Health Records. BMJ2022, 378:e071531. http://doi.org/10.1136/bmj.2022-071531

[5] Brands, M. R., Gouw, S. C., Beestrum, M., Cronin, R. M., Fijnvandraat, K., and Badawy, S. M. 2022. Patient-Centered Digital Health Records and their Effects on Health Outcomes: Systematic Review. *Journal of Medical Internet Research, 24(12):e43086.* DOI:10.2196/43036

[6] Agiwale, S., Panaskar, V., Tilekar, K., Solanke, K., and Dongare, S. 2023. Blockchain-Based Personalized Digital Health Record. *International Research Journal of Modernization in Engineering Technology and Science 05(03)*

[7] Adeniyi, A.O., Arowoogun, j. o., Chidi, R., Okolo, C. A., and Babawarun, O. 2024. The Impact of Electronic Health Records on Patient Care and Outcomes: A Comprehensive Review. *World Journal of Advanced Research and Reviews. 21(02):1446 – 1455.* https://doi.org/10.30574/wjarr.2024.21.2.0592

[8] Seh, A. H., Al-Amri, J. F., Subalu, A. F., Agrawal, A., Kumar, R., and Khan, R. A. 2021. Machine Learning Based Framework for Maintaining Privacy of Healthcare Data. *Soft Computing.* DOI: 10.32604/iasc.2021.018048

[9] Abul-Husn, N. S. and Kenny, E.E. 2019. Personalized Medicine and the Power of Electronic Health Records. *HHS Public Access, 177(1):58 – 69*, DOI:.10.1016/j.cell.2019.02.039

[10] Attah, A. O. 2017. Implementing the Electronic Health Record in Nigeria: Prospects and Challenges. A Master's Thesis in Telemedicine and E-Health (TLM-3902). The Arctic University of Norway.

[11] Yeng, P. K.., Nweke, L. O., Yang, B., Fauzi, M. A., and Snekkenes, E. A. 2021. Artificial Intelligence-Based Framework for Analyzing Health Care Staff Security Practice: Mapping, Review and Simulation Study. *JMIR Medical Informatics*, 9(120: e19250. https://medinform.jmir.org/2021/12/e19250

[12] Yeng PK, Fauzi MA, Yang B. 2020. Comparative analysis of machine learning methods for analyzing security practice in electronic health records' logs. 2020 Presented at: 2020 *IEEE International Conference on Big Data (Big Data*); December 10-13, 2020; Virtual p. 3856-3866

[13] Olaniyi, O. M., Alhassan, J. K., Abba, E., and Waziri, V. O. 2016. Threats modeling of Electronic Health Systems and Mitigating Counter Measures. *International Conference on Information and Communications Technology and its Applications (ICTA 2016), Nov. 28 – 30.* Federal University of Technology, Mina

[14] Okediran, O., Sijuade, A., Wahab, W., and Oladimeji, A. 2022. A Framework for a Cloud-based Electronic Health Record System for Nigeria. *LAUTECH Journal of Engineering and Technology, 1692):128 – 136*

[15] Syed, A., Purushotham, K. and Shidagani, G. 2020. Cloud Storage Security Risks, Practice and Measures: A Review. *IEEE International Conference for Innovation in Technology (INOCON). https://doi.org.sdl.idm.oclc.org/10.1109/INOCON50539.2020.9298281*

[16] Saeed, V. A. and Asaad, R.R. 2022. Cyber Security, Threats, Vulnerability, Challenges with Proposed

Solution. *Applied Computing Journal, 2(4):227 – 244.* https://doi.org/10.52098/acj.202260

[17] Fatima, A. and Colomo-Palacios, R. 2018. Security Aspects in Healthcare Information System: A Systematic Mapping. Procedia Computer Science, 138:12-9 https://doi.org/10.1016/j.procs.2018.10.003

[18] Jenyo, I., Amusan, E. A. and Emuoyibo-farhe, J. O. 2023. A Trust Management System for the Nigerian Cyber-Health Community. *International Journal of Information Technology and Computer Science, 1: 9 – 20.* DOI: 10.5815/ijitcs.2023.01.02

[19] Abd-Alrazaq, A.A., Bewick, b. m., Farragher, T., and Gardner, P. 2019 FactorsthatAffect the use of Electronic Personal Health Records among Patients: A SystematicReview.*International Journal of Medical Informatics, 126:164-175*

[20] Hagglund, M., Ceyander, A., Rexhepi, H. and Kane, B. 2022. Editorial: Personalized Digital Health and Patient-Centered Services. *Frontiers in Computer Science, 4:862358. DOI: 10.3389/fcomp.2022.862358*

[21] Chen, O. Y., and Roberts, B. 2021. Personalized Healthcare and Public Health in the Digital Age. *Frontiers in Digital Health, 3:595704. DOI: 10.3389/fdgth.2021.595704*

[22] Joukes, E. de Keizer, N. F., de Brijne, M. C., Abu-Hanna, A., Cornet. R. 2019. Impact of Electronic versus Paper-Based Recording Before EHR Implantation on Healthcare Professionals' Perceptions of EHR Use, Data, Quality, and Data Reuse. *Applied Clinical Informatics 10(2):199-209*

[23] Fan, M., Ezeudoka, B. C. and Qalati, S. A. 2024. Exploring the Resistance to e- Health Services in Nigeria: An Integrative Model Based Upon the Theory of Planned Behavior and Stimulus-Organism. Response. Humanities and Social Sciences Communication. https://doi.org/10.1057/5-4159.9-024-03090-6

[24] GitHub Inc. 2020. *NSL-KDD Dataset*. https://www.github.com

[25] Santos, I., Devesa, J., Brezo, F., Nieves, J., Bringas, P. G. 2013. OPEM: A Static-Dynamic Approach for Machine Learning-Based Malware Detection. In: Harrero, A., Conference CISIS'12-ICEUTE'12-SOCO'12 Special Sessions. *Advances in Intelligent Systems and Computing, vo.189* Springer, Berlin, Heidelberg. https://dio.org/10.1007/987-3-642-33018-6-28

[26] Chakir, O., Rehaimi, A., Sadqi, Y., Alaoui, E. A. A., Cridun, M., Gaba, G. S., and Gurtov, A. 2023. An Empirical Assessment of Ensemble Methods and Traditional Machine Learning Techniques for Web-based Attack Detection in Industry 5.0. *Journal of King Saud University - Computer and Information Sciences,* 35(2023):103–119. https://dio.org/10.1018/j.jksuci.2023.02.009

[27] Saini, N., Kasaragod, V. B., Prakasha, K., Das, A. K. 2023. *A Hybrid Machine Learning Model for Detecting APT Attacks Based on Network Behavior Anomaly Detection. Concurrency and Computation: Practice and Experience* John Wiley and Sons. https://dio.org/10.1002/cpe.7865