

## A Machine Learning Approach to Intrusion Detection in Cloud-Based e-Health Systems

Ogwo Eme<sup>1\*</sup>, Fergus Uchenna Onu<sup>2</sup>, Augustine Chidiebere Onuora<sup>1</sup>  
and Chibuike Ezeocha Madubuike<sup>1</sup>

<sup>1</sup> Department of Computer Science, Akanu Ibiam Federal Polytechnic, Unwana, Nigeria

<sup>2</sup> Department of Computer Science, Ebonyi State University, Abakaliki, Nigeria

\*Corresponding Author: eogwo@akanuibiampoly.edu.ng

**ABSTRACT** The integration of digital technologies and web-based services has significantly transformed healthcare delivery in recent years. However, this digital evolution has also increased the vulnerability of e-health systems to a wide range of cyber-attacks, posing serious risks to the confidentiality and integrity of patients' electronic health records. This study proposes a machine learning-based intrusion detection approach specifically designed to secure cloud-based e-health systems. The Object-Oriented Analysis and Design Methodology (OOADM) was employed for system design and implementation. Two machine learning algorithms—Random Forest (RF) and K-Nearest Neighbor (KNN)—were applied to classify and detect various intrusion types, including General Practitioner attacks, Consent attacks, and uncorrelated threats. Simulations were conducted using Python and a medical dataset sourced from the Kaggle repository, partitioned into 80% training and 20% testing subsets with fine-tuned hyperparameters with the desired error value set to 0.000005 tolerance levels. The results of the model show that the Random Forest algorithm achieved a detection accuracy of 92%, significantly outperforming KNN algorithm, which achieved an accuracy level of 42%. The receiver operating characteristic (ROC) graph of KNN and RF classifiers indicated that the RF ROC curve has a value of 0.972 which is closer to 1 or top-left corner of the graph and is perfect and performed better than KNN model with a ROC value of 0.8892. These findings demonstrate the effectiveness of the RF algorithm in accurately identifying intrusions within cloud-hosted e-health systems, underscoring its potential for enhancing cybersecurity and safeguarding sensitive health information.

**KEYWORDS:** Cyber-attacks, Digital Technologies, e-Health Systems, Intrusion Detection Systems, Machine Learning

### I. INTRODUCTION

The incorporation of digital technologies and online platforms into healthcare delivery has significantly enhanced patient care quality. e-Health, which refers to the utilization of Information and Communication Technology (ICT) within the health sector, is emerging as a rapidly expanding area in modern healthcare [1]. These e-Health applications are generally user friendly, interactive, and capable of disseminating reliable healthcare information across various devices and platforms [2]. As most e-Health systems operate on cloud infrastructure—enabling instant access to medical data and facilitating the exchange of patients' Electronic Health Records (EHRs) between institutions—they face a wide array of security vulnerabilities [3]. Although cloud-based e-Health systems offer numerous advantages, including convenience and scalability, they also present serious privacy and cybersecurity risks that must be addressed for widespread and effective use. To tackle these risks, several security strategies—both cryptographic and non-cryptographic—have been implemented to safeguard sensitive patient data. These methods help ensure confidentiality and integrity within healthcare systems. This study investigates how non-cryptographic techniques, particularly Machine Learning (ML), can be employed to identify and respond to intrusions within cloud-based e-Health platforms.

ML algorithms are increasingly being used in intrusion detection systems to identify and mitigate threats. Technologies such as Artificial Intelligence (AI), robotics, and computer vision have begun to reshape the healthcare landscape. ML models like Convolutional Neural Networks (CNN), Decision Trees (DT), K-Nearest Neighbors (KNN), and Support Vector Machines (SVM) are being integrated with security frameworks to enhance classification accuracy. Despite the potential of healthcare networks, their sensitive data makes them prime targets for cyber-attacks, as confirmed by recent cybersecurity research. Therefore, there is an urgent need to build intelligent, real-time Intrusion Detection Systems (IDS) that can protect medical data and support hospitals in delivering secure healthcare services. The specific research problems tackled are:

- (a). Patient EHRs hosted on cloud-based systems face persistent threats of unauthorized access and privacy breaches. Malicious users often exploit system vulnerabilities to access this confidential information. A robust and secure mechanism is needed to detect and prevent such intrusions.
- (b). Literature review reveals that existing approaches primarily address various attacks on e Health systems, yet there is limited research on ML-based IDS models capable of identifying specific threats such as General Practitioner attacks, Consent attacks, and Uncorrelated Documents Attacks.
- (c). Many existing models are inadequate—they lack automation, fail to alert administrators in real time, and cannot detect poisoning attacks, which could seriously compromise patient care. Most models rely on manual expert oversight and do not function in real-time environments. These shortcomings underscore the necessity of developing ML-powered IDS that operates in the background of e-Health systems, enhancing data security and aiding hospitals in maintaining quality healthcare delivery [4-6].

## II. RELATED WORKS

In previous studies [7], researchers have explored various security solutions for e-Health systems. One study introduced a blockchain-based system for managing EHRs, maintaining data integrity by logging all access events. Another solution leveraged cryptographic hash functions within a decentralized blockchain network to facilitate secure and rapid information sharing among healthcare providers [8]. In another approach [9], researchers used statistical models and ML techniques like Logistic Regression and SVM to identify unusual EHR access patterns. Their iterative labeling and model training process resulted in high accuracy and sensitivity, outperforming rule-based systems. In another research [4], a different method employed collaborative filtering to detect unauthorized access by analyzing user behavior. This method incorporated explicit and implicit features of patients and staff, with evaluation metrics such as RMSE and AUC showing improved results over traditional ML algorithms. In [5], an unsupervised learning model was proposed for insider threat detection in a collaborative environment using access logs called a community-based anomaly detection system. The approach proposed by the authors is hybrid; they use singular value decomposition, a special case of PCA, to infer communities from relational networks of users, and then they use KNN to create a set of nearest neighbors. The developed model demonstrated high accuracy using access logs from real-world EHR systems. Another hybrid IDS dual-module system that utilized agents that monitor system logs was proposed by these researchers [6]. The misuse detection module addressed known attacks, while the anomaly detection module, using DT, neural networks, and K-means, detected unknown (zero-day) threats. The system used a voting system and expert analysis ensured reliable results. In a research work conducted by [10], the authors used SVM and Exponential Moving Average (EMA) in a framework to detect privacy breaches and denial of-service threats. After testing with synthetic data, SVM showed superior performance in anomaly detection, and EMA effectively spotted message flow disruptions.

### III. METHODOLOGY

The proposed ML-based IDS was developed using several structured phases: data collection, transformation, feature extraction, model development (training and testing), intrusion detection, and performance evaluation.

#### A. Data Collection

The training dataset was sourced from the Kaggle Machine Learning repository. It included 58,976 entries with attributes such as NumCPTevents, AdmitProcedure, diagnosis, and NumProcs.

#### B. Data Transformation

This phase involved converting raw inputs into a format suitable for ML algorithms using Load, Transform, and Extract (LTE). Numeric attributes were normalized to improve model performance and enable non-linear algorithms like Random Forest (RF) and KNN to perform optimally.

#### C. Feature Extraction

Relevant features were derived and compiled into a structure known as a feature vector. This vector served as the input for classification and regression models to perform accurate predictions.

#### D. Model Development

Model training was conducted using KNN and RF algorithms. The dataset was split into 80% for training and 20% for testing. Python's sklearn library was employed to train and evaluate the models. KNN was chosen for its ability to handle missing values and prevent overfitting. The RF algorithm, an ensemble method that mitigates high variance in single decision trees, was implemented using the RandomForestRegressor class. Visualization was done using matplotlib.

#### E. Model Training

To ensure statistical stability, randomized training was applied. This approach maintained consistent model performance regardless of dataset variations. The training process was visualized and supported by Python code to illustrate the underlying mechanisms. The model development involved training and testing processes. The dataset was split into 80% training and 20% testing subsets. The KNN and RF algorithms were selected for this process due to their strengths in handling complex data structures and minimizing overfitting. The implementation was done using Python programming language. The KNN model was initialized using the KNeighborsClassifier, where the number of neighbors was specified. The training was carried out with `model.fit(X_train, Y_train)`, and predictions were made using `model.predict(X_test)` from the Scikit-learn library. For the Random Forest approach, an ensemble of decision trees was built using the RandomForestRegressor class from the sklearn.ensemble module. This ensemble method helped overcome high variance often seen in single decision trees by averaging results across multiple trees. The forest was configured with a maximum depth of 8 and a fixed random state of using the code:

```
reg_forest = RandomForestRegressor(max_depth=8, random_state=17)
```

The training and testing were executed using:

```
reg_forest.fit(X_train, y_train)
```

It was evaluated through prediction. Visualization of results was performed using Python's matplotlib library.

## F. Evaluation of the Machine Learning Model

This stage represents the final phase of the project, aimed at assessing how effectively the Random Forest and K-Nearest Neighbor algorithms perform in comparison to existing solutions. Several performance metrics were employed to measure the accuracy and reliability of the models using the testing dataset. A confusion matrix was used as the primary tool for performance evaluation and comparison. To generate these metrics, the confusion matrix must first be constructed. It is created by applying the trained ML model to classify the instances in the testing dataset. The matrix compares the actual labels of the test samples with those predicted by the model. Four key parameters are derived from this comparison: True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN). TP and TN correspond to the correctly classified attack and normal cases, respectively, while FP and FN represent misclassifications—where normal data is wrongly flagged as an attack, or an actual attack is missed. Using these values, additional performance indicators such as precision, recall, F1-score, and accuracy can be computed to quantify the model's effectiveness. After development, the trained ML model can be integrated into an existing e-Health infrastructure to facilitate real-time detection of intrusions, ensuring swift response to any unauthorized activities.

## G. Analysis of the Proposed ML Model

Intrusions within e-Health systems typically occur across three primary layers: the Data Collection Layer, the Data Transmission Layer, and the Data Storage Layer. For instance, data manipulation may occur during collection, denial-of-service (DoS) attacks may arise during transmission, and unauthorized access often happens at the storage level. This study specifically focuses on storage-level attacks, namely:

- (a). General Practitioner (GP) Attacks
- (b). Consent Attacks
- (c). Uncorrelated Documents (UD) Attacks

The developed ML model incorporates an Intrusion Detection Module (IDM) designed to identify and classify these types of security threats. The IDM plays a central role in detecting potential breaches by applying a rule-based logic to assess and flag specific conditions:

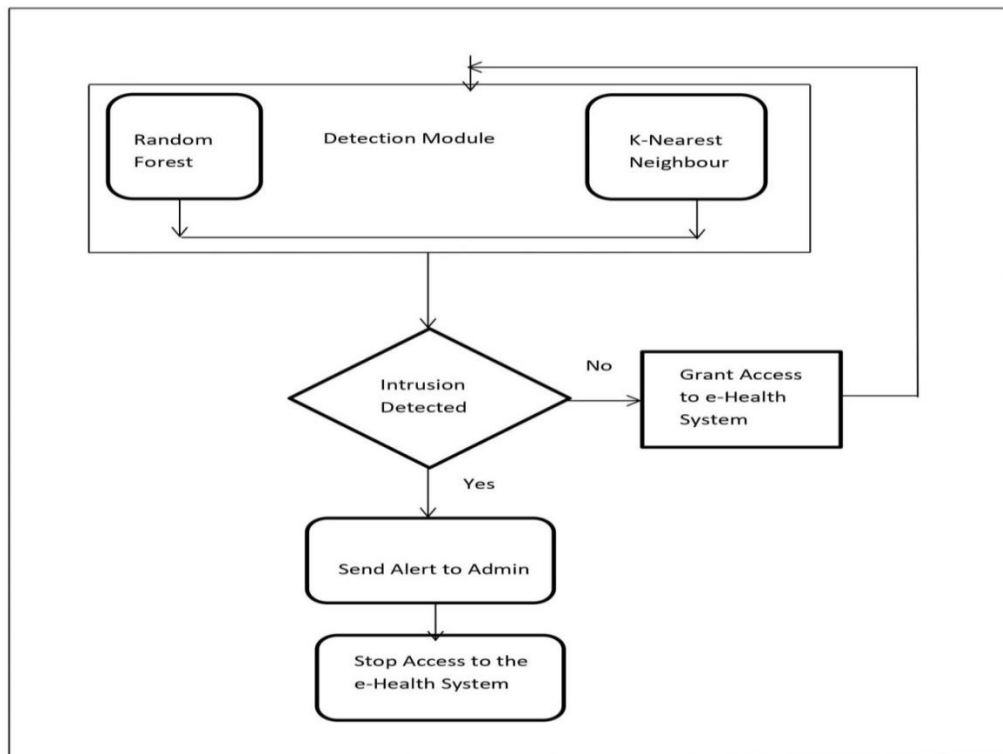
- (a). GP Attack: Occurs when a healthcare provider accesses patient data that they are not authorized to view.
- (b). Consent Attack: Happens when patient records are accessed without the necessary permissions or approvals.
- (c). UD Attack: Arises when a user retrieve disjointed or unrelated medical data from different patients.

To carry out this detection, the system utilizes both Random Forest and K-Nearest Neighbor classifiers. The detection module outputs one of two possible states—normal or anomalous. It applies a majority voting strategy to determine whether an input is an intrusion. If an attack is identified, the system classifies the type of attack accordingly. Non-threatening inputs are labeled as "normal", while harmful ones are categorized under the appropriate intrusion type (GP, Consent, or UD). The classification process begins with the loading of a medical dataset that includes both normal and attack-related records. This dataset is analyzed by both ML classifiers. Based on the outcome of this analysis, when a request is made to access EHRs, the system decides if the request is legitimate or malicious. Upon confirming the presence of an attack, the system activates a real-time voice alert to notify an automated system administrator. This mechanism instantly blocks the attacker's access to the protected EHRs. Furthermore, every intrusion attempt is logged and recorded within the system.

A visual representation of this ML-powered Intrusion Detection System is shown in Figure 1, which highlights the detection module as the central unit of the architecture. This module capitalizes on the predictive power of ML algorithms to identify both known and previously unseen threats that could jeopardize the security of e-Health platforms.

#### IV. RESULTS AND DISCUSSION

The results of our developed model are discussed subsequently. We start the discussion on the training and testing dataset used, and concludes the discussion on the accuracy of the ML algorithms used for the detection of intrusions on the e-Health system.



**Figure 1:** Block diagram of the ML-Based Intrusion Detection Model for e-Health Systems

#### Training and Testing Datasets

Figure 2 depicts the total dataset used in this work, as well as the percentages used for training and testing. The dataset was divided into 80%  $((80 \times 58,976) / 100 = 47,180)$  training and 20%  $((20 \times 58,976) / 100 = 11,796)$  testing sets amounting to total of 58,976 records. That means that 80% of the total dataset was used for training (47,180) and 20% of the total dataset was used for testing (11,796). The train part consisted of 80% of the dataset and the ML algorithms were trained and evaluated with this part. On the other hand, the test part consisted of 20% of the dataset and was held back for further evaluation of the models with unseen data. The percentage split of 80% train and 20% test was determined according to Géron (2017) as the best ratio to avoid the problem of overfitting.

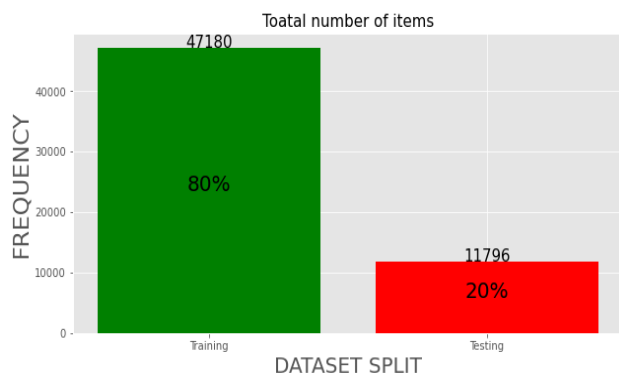
#### KNN cross validation test

Figure 3 illustrates the learning curve graph of KNN, which shows that the cross-validation score of KNN continuously increases while the training score falls along varied neighbor values. The training and validation curves converge from points 2 to 4 and across additional points, resulting in an accuracy rate of 94% to 92% at k-values ranging from 6, 7, 8, and 9.

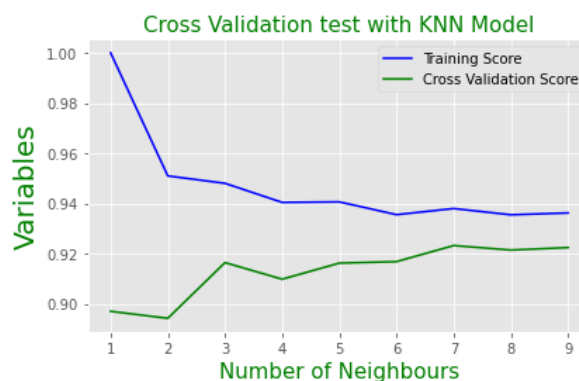
#### Random Forest Learning training curve

Figure 4 shows the plots representing training and validation score for the Random Forest model. The blue line represents the training curve while the green line represents the cross-validation curve. The R value is an

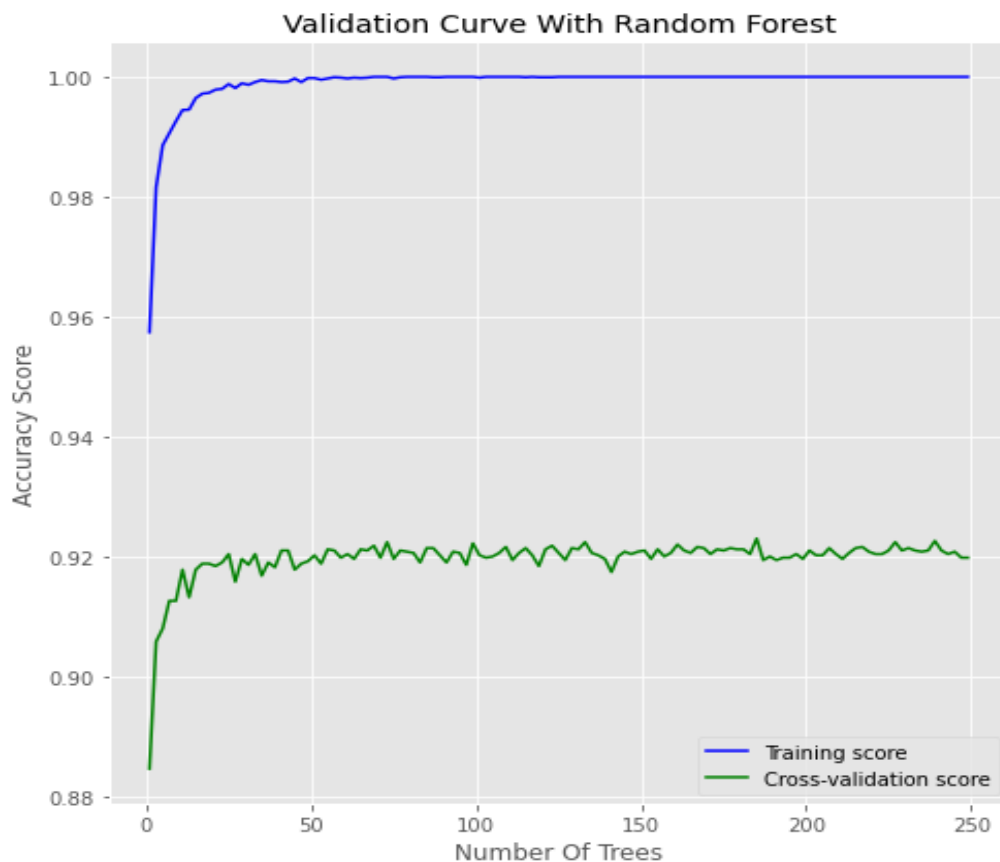
indication of the relation that exists between the outputs and the targets. Cross validation accuracy score is .92 which is 92 % while training accuracy score is 1.0 which is 100%. The training was perfect. The training and validation results show large R values which indicate a system performance in terms of training of the dataset and a good regression score. It could also be noticed that as the iteration of the trained dataset increases the number of trees increases thereby increasing the accuracy level of the training of the model.



**Figure 2:** Training and Testing Sets



**Figure 3:** KNN cross validation test



**Figure 4:** Random Forest Learning Training Curve

### A. Intrusion Detected Attacks

Figure 3 depicts the total number of attacks and free cases as obtained from the testing dataset by the KNN and random forest models. The test set's exact answer was more closely approximated by 3,343 normal instances produced by the KNN and 3,365 by the random forest model. The figure illustrates the different types of attacks that the KNN and RF identified respectively: 2,643 and 3,354 UD attacks, 1,113 and 1,649 consent attacks, and 2,452 and 2,813 GP attacks.

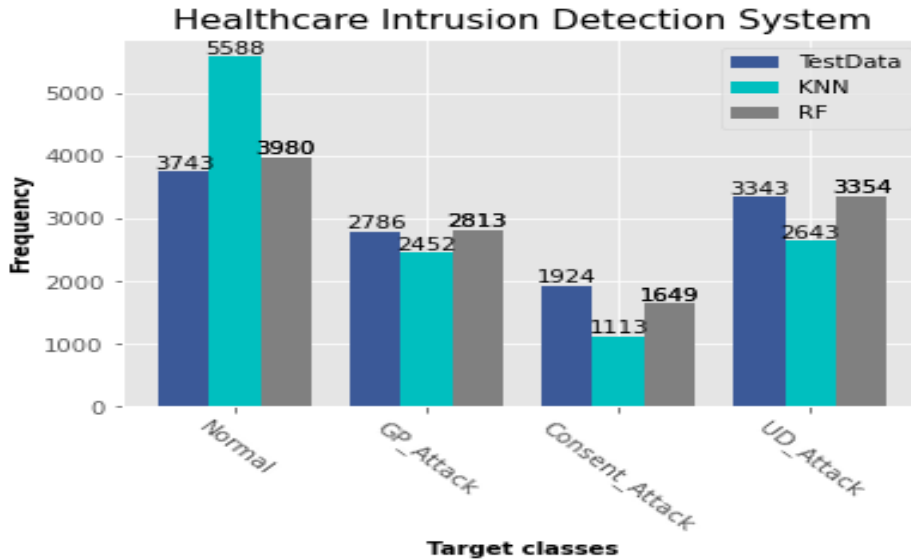


Figure 5: Intrusion Detected Attacks

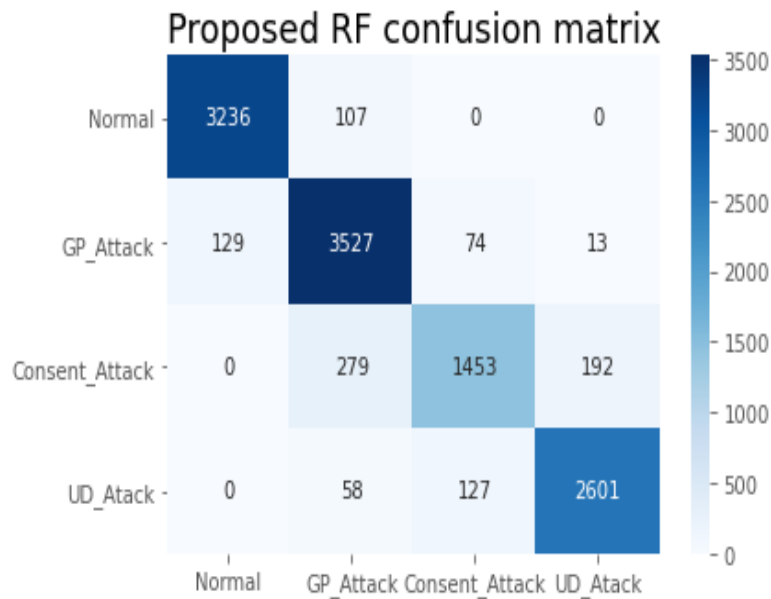


Figure 6: Confusion Matrix of RF



### B. Confusion Matrix of RF

The Confusion Matrix is an evaluation metrics used to assess the performance of the ML algorithms used in the model. Two ML algorithms were used namely Random Forest and K Nearest Neighbor. Figure 5 depicts the proposed RF model's confusion matrix at the testing stage, with correct classification presented at the secondary diagonal and incorrectly classified values recorded above and below the main diagonal. The total number of correct classified attacks are  $3,236+3,527+1,453+2,601=10,817$  and the number of incorrect classified attacks are  $(129+279+127+58+0+0) + (107+74+192+13+0+0) = 593+386 = 979$ . This is presented in figure 5, where TP is true positive, FP is false positive, FN is false negative, and TN is true negative.

### C. Confusion matrix of KNN

The confusion matrix of the KNN model is shown in figure 5, with the leading diagonal elements or values indicating the total number of properly predicted values that are equal to the actual or true values, and the off-diagonal components indicating the incorrectly predicted values. The greater the diagonal values, the more accurate the prediction. According to the confusion matrix, the total number of accurate predictions are  $2200+1401+236+1275=5,112$  and the total number of bad predictions are  $(864+1063+1461+345+237+284+258) + (357+382+804+1063+1461+345+349+258) = 6,684$ .

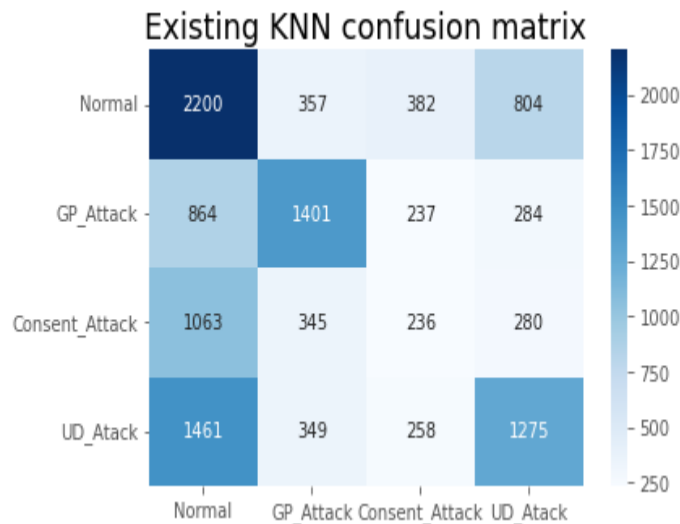


Figure 7: Confusion Matrix of RF

### D. RF Classification Report

Figure 8 shows the RF classifier classification report, which includes the precision, recall, and f1 score accuracy of true and false data instances. Normal, GP\_attack, consent\_attack and UD\_attack types produced a precision accuracy of score of 0.96, 0.89, 0.88 and 0.93, while recall 0.97, 0.94, 0.76, and 0.93. The f1-score recorded 0.74 for true and 0.32 for false cases, with an f1-score of 0.96, 0.91, 0.81 and 0.93 for the normal and selected attack types respectively. The overall performance accuracy of the RF algorithm is .92 which is 92%.

### E. KNN Classification Report

The classification report of the KNN model is shown in Figure 9, with precision, recall, and f1 score classification accuracy. The precision recorded 0.42, 0.39, 0.22 and 0.63 measured against normal, GP\_attack, consent\_attack and UD\_attack types. Recall yielded 0.53, 0.48, 0.12 and 0.43 while f1-score gave 0.47, 0.43, 0.15 and 0.52 against the selected types of attacks. The overall performance accuracy of the KNN algorithm is .42 which is 42%.



	precision	recall	f1-score	support
Normal	0.96	0.97	0.96	3343
GP_Attack	0.89	0.94	0.91	3743
Consent_Attack	0.88	0.76	0.81	1924
UD_Attack	0.93	0.93	0.93	2786
accuracy			0.92	11796
macro avg	0.91	0.90	0.91	11796
weighted avg	0.92	0.92	0.92	11796

Figure 8: RF Classification Report

	precision	recall	f1-score	support
Normal	0.42	0.53	0.47	3343
GP_Attack	0.39	0.48	0.43	3743
Consent_Attack	0.22	0.12	0.15	1924
UD_Attack	0.63	0.43	0.52	2786
accuracy			0.42	11796
macro avg	0.41	0.39	0.39	11796
weighted avg	0.43	0.42	0.42	11796

Figure 9: KNN Classification Report

#### F. Detected Attack Types

The different attack types detected by the Machine Learning-Based Model and the accuracy of these ML algorithms are shown in Figure 10. The ML algorithms used were KNN and RF algorithms. The RF algorithm was able to classify correctly GP\_Attack as GP\_Attack while the KNN algorithm wrongly classifies GP\_Attack as Consent\_Attack. Also, the RF algorithm was able to classify UD\_Attack as UD\_Attack while the KNN wrongly predicted it as Consent\_Attack. This shows that the RF algorithm is more accurate for the classification of the selected attacks. In other instances, both RF and KNN attacks rightly classified GP\_Attack and UD\_Attack respectively.

TestData_Attacks	KNN_Pred	RF_Pred
40028 GP_Attack	Consent_Attack	GP_Attack
38143 UD_Attack	Consent_Attack	UD_Attack
14276 Normal	Normal	Normal
49801 Normal	Normal	Normal
21549 GP_Attack	GP_Attack	GP_Attack
...	...	...
50194 UD_Attack	Consent_Attack	UD_Attack
41920 UD_Attack	UD_Attack	UD_Attack
22400 Consent_Attack	GP_Attack	Consent_Attack
40485 GP_Attack	GP_Attack	GP_Attack
23098 UD_Attack	UD_Attack	UD_Attack

Figure 10: Predicted Attack types

### G. Accuracy of KNN and RF

Figure 11 demonstrates the accuracy of the KNN and RF models' performance. The RF model provided the maximum classification accuracy of 91%, while the KNN model produced the lowest forecast accuracy of 42%. This indicates that the RF algorithm performs better than The KNN algorithm in detection the selected attacks (GP attack, consent attack and UD attack).

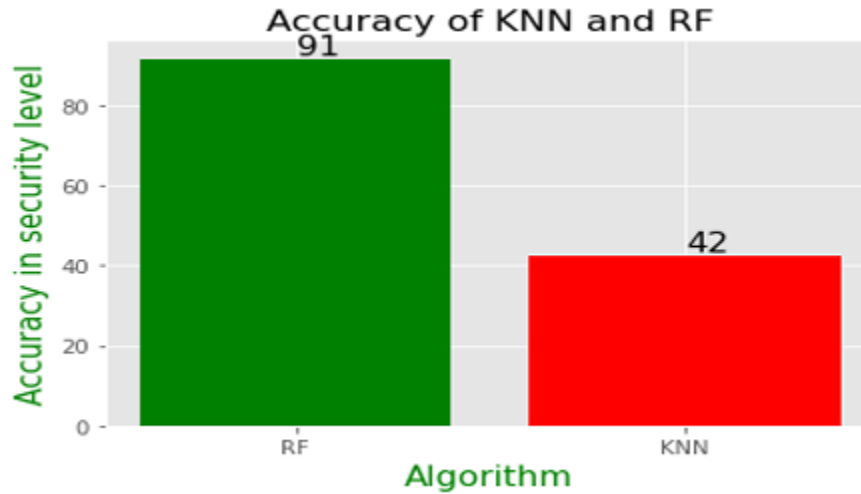


Figure 11: Accuracy of KNN and RF

### Receiver Operating Characteristic (ROC) of KNN and RF

Figure 12 is the receiver operating characteristic (ROC) graph of KNN and RF classifiers showing the trade-off between sensitivity or true positive rate and specificity (1-FPR). The RF ROC curve (0.972) which is closer to 1 or top-left corner of the graph and is perfect and performed better than KNN model. The KNN curve (0.8892) which is a bit away from the top x and y axis with respect to the number of false positive rate (FPR) and true positive rate (TPR) as shown in the ROC graph. The proposed gave points lying along the diagonal (True Positive Rate=False Positive Rate) as expected.

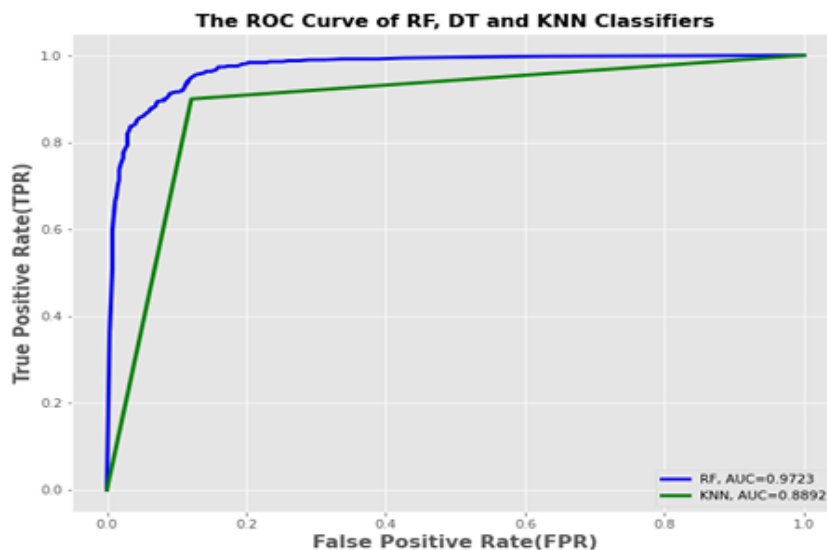


Figure 12: ROC graph of KNN and RF classifiers

#### H. Comparison of Our Results against Other Studies

A summary of the comparison of related works on Machine Learning using Intrusion Detection on e-Health systems is shown in Table 1. The table shows the ML methods with their authors, the aims, the ML methods adopted and the weaknesses of their research works.

**Table 1:** Summary of results of related works

Authors	Aims	ML Methods Adopted	Weaknesses
Menon et al., (2014)	Detection of privacy breaches resulting from inappropriate access to EHR	Collaboration Filtering Using Latent and Explicit Features	a) The model was on not real time. b) The solution was not applied to more than one institution. c) Used only a small dataset for training d) All metrics were not used. e) The model is not fully automated
Malin and Bradley, (2014)	Detecting insider Threat in collaboration environment using access logs and Community based Anomaly detection System	Adopted Unsupervised Learning Technique (KNN and singular value decomposition)	a) It is not fully automated; b) The value of k used in the KNN cluster changes from one system to another system c) This method cannot detect attackers that imitate legitimate of another user or legitimate behavior of a group.
Marwan et al., (2018)	Securing image data processing in a cloud environment based on machine learning	Combination of SVM and FCM	It uses a small dataset (just two images)
Sicuranza and Paragliola (2020)	Used Hybrid IDS for the detection of cyber-attacks on EHR system	Decision Tree, Neural Network and K-Means	a) The time of detection is not indicated. b) The result of the misuse detection module is not reported

			c) The solution was not applied to more than one institution. d) The proposed model was not fully automated as presence of experts was needed to resolve any conflict.
McGlade and Scott-Hayward (2019)	Detection of confidentiality and availability issues on EMR systems	SVM and EMA	a) All metrics were not used. b) The integrity of the system EMR was not checked through the adopted ML method.

Although many researchers have done some works on the detections of intrusions on e-health systems as summarized on table 3, gaps still exist. Our Machine Learning Model for detection of intrusion in e-health System is different from the existing Intrusion Detection model (Sicuranza and Paragliola, 2020) because our IDS model incorporates novel features different from existing models. These features are:

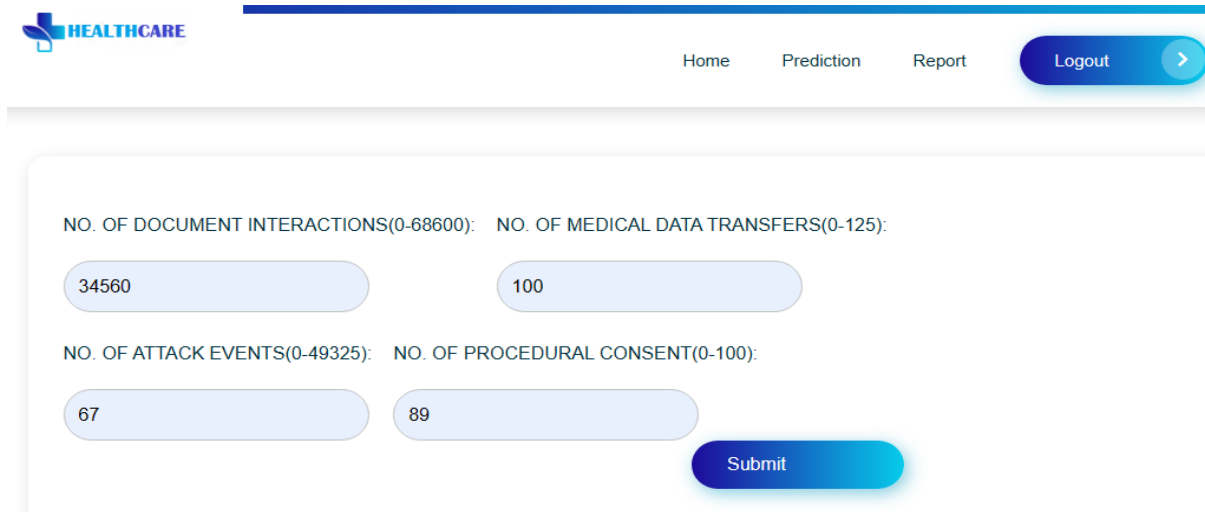
- 1) It is fully automated as the presence of experts is not physically needed to detect any anomaly intrusion when it occurs.
- 2) The administrator is notified with an audio voice when an intrusion is detected.
- 3) It detects poisoning data attacks in the train parameters of the medical dataset.
- 4) Our Intrusion Detection System uses optimal hyperparameters for its simulation testing and the results of the evaluation of our ML model also show high level performance rate as against other existing models [11].
- 5) We developed and incorporated web application software for predicting attacks that occur in the e-health System using features obtained from their dataset.

## V. MODEL DEPLOYMENT

In order for the ML model system to be deployed on the web; and to enhance the prediction of attacks on our e-health system, a Web Application Interface of our intrusion system was designed, implemented and deployed. This deployment was done using the Random Forest algorithm running on the background because it has been found to perform better with higher accuracy rate than the K-Nearest Neighbour algorithm. This Web Application Interface (Figure 12) uses a Prediction Module for prediction of attacks on the e-health system and a Report module (Figure 13) for displaying information of predicted attacks on the e-Health system.

### A. Output Report

The output of our web application interface gives the results after processing the inputted data using the features from the dataset prediction. The outputs of prediction of the web software of our e-health system is a voice notification of the detected attack and the Report Module which shows the list of predicted attacks detected by our e-health system as shown in Figure 13.



HEALTHCARE

Home Prediction Report Logout

NO. OF DOCUMENT INTERACTIONS(0-68600): NO. OF MEDICAL DATA TRANSFERS(0-125):

34560 100

NO. OF ATTACK EVENTS(0-49325): NO. OF PROCEDURAL CONSENT(0-100):

67 89

Submit

**Figure 12:** Web Application Interface

## Attack Report

S/No:	Document Interactions	Medical Data Transfers	Attack Events	Procedural Consent	Attack Target	Attack Type
1	678	90	67	89	1	GP ATTACK
2	78980	100	13462	80	2	CONSENT ATTACK
3	5678	100	8785	60	1	GP ATTACK
4	768	100	13462	80	1	GP ATTACK
5	769	090	87.9	67	1	GP ATTACK
6	4	7	89	56	0	NO ATTACK
7	34560	45	453	46	1	GP ATTACK
8	34560	57	45000	45	2	CONSENT ATTACK

**Figure 13:** Predicted Attack Report

## VI. CONCLUSION

In this paper, we proposed a Machine Learning Based Model that detects different attacks on cloud-based e-Health Systems. The different attacks detected by the ML model were General Practitioner (GP) Attack, Consent Attack and Uncorrelated Documents Attack. The different attacks detected by the results from the simulation of the developed model using RF and KNN machine learning algorithms show that RF algorithm has a higher accuracy from the existing KNN algorithm and as such is more secure in detecting cyber and malicious attacks when compared to the existing e-Health systems. The results of the evaluation of our

developed model show that the RF algorithm has a higher performance rate than other existing models that uses the KNN algorithm for detection of intrusions on e-health systems. The RF also produced a higher accuracy performance of 92% as against 42% for the existing Intrusion Detection model that uses KNN technique. This result is in agreement with earlier work conducted by other researchers [11] which shows that RF performs better in detecting various security attacks than KNN. The results of the research work show that application of digital technologies in healthcare institutions hospitals assist healthcare providers to secure patients' electronic health records in cloud-based e-Health Systems. Future work will focus on the deployment of the developed machine learning model in e-health based cloud systems for detection of intrusion (attacks) one health systems.

## REFERENCES

- [1] Srivastava, S., Pant, M., Abraham, A & Agrawa, N. (2015). The Growth in e-Health Services. National Institutes of Health. doi: 10.1155/2015/894171 Technological
- [2] Bouamrane, M-M., and Mair, F. S. (2011). An Overview of Electronic Health Systems Development and Integration in Scotland. *Proceedings of the 1st international workshop on managing interoperability and complexity in health systems, Glasgow, Scotland*, 59-62.
- [3] Li, M., Yu, S, Zheng, Y. & Ren, K. (2012). Scalable and secure sharing of personal health records in cloudcomputing using Attribute-based Encryption. *IEEE Trans Parallel Distributed System*;1–14.
- [4] Menon, A. K., Jiang, X., Kim, Vaidya, J. and Ohno-Machado, L. (2014). Detecting inappropriate access to Electronic Health Records Using Collaborative Filtering. *Machine Learning* 95, 87–101.
- [5] Malin, Y. C. and Bradley, C. (2014) Detection of Anomalous Insiders in Collaborative Environments via Relational Analysis of Access Logs, *Bone* 23, 1–7. <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4257138/>
- [6] Sicuranza, M. and Paragliola, G. (2020). Ensuring Electronic Health Record Cyber-Security through a Hybrid Intrusion Detection System. <https://intranet.icar.cnr.it/wp-content/uploads/2020/05/RT-ICAR-NA-2020-01.pdf>.
- [7] Yang, G., Li, C. and Marstein, K.E. (2021). A blockchain-based architecture for securing electronic health record systems. *Concurrency and Computation: Practice and Experience* 33 (14), e5479.
- [8] Chelladurai, U. & Pandian, S. (2022). A novel blockchain based electronic health record automation system for healthcare. *Journal of Ambient Intelligence and Humanized Computing* 13 (1), 693-703.
- [9] Boxwala, A. A., Kim, J., Grillo, J. M. and Ohno-Machado, L. (2011). Using statistical and Machine Learning to Help Institutions Detect Suspicious Access to Electronic Health Records. *Journal of the American Medical Informatics Association* 18, 498–505.
- [10] McGlade, D. and Scott-Hayward, S. (2019). ML-based cyber-Incident Detection for Electronic Medical Record (EMR) Systems, *Smart Health* 12, 3–23.
- [11] Doshi, R.; Apthorpe, N. and Feamster, N. (2018). Machine Learning DDoS Detection for Consumer Internet of Things Devices. *In Proceedings of the 2018 IEEE Security and Privacy Workshops (SPW), San Francisco, CA, USA*. 29–35.