Detecting Ransomware within Real Healthcare Medical Records Adopting Internet of Medical Things using Machine and Deep Learning Techniques

Randa ELGawish[1]
Mohamed Hashim[4]
Faculty of Computer and Information Sciences
Ain Shams University, Cairo, Egypt

Mohamed Abo-Rizka[2]
College of Computing and Information Technology
Arab Academy for Science and Technology, Cairo, Egypt

Rania ELGohary[3]
Artificial Intelligence Technology Center, Misr University for Science and Technology,
Cairo, Egypt

***Abstract*—The Internet of Medical Things was immensely implemented in healthcare systems during the covid 19 pandemic to enhance the patient's circumstances remotely in critical care units while keeping the medical staff safe from being infected. However, Healthcare systems were severely affected by ransomware attacks that may override data or lock systems from caregivers' access. In this work, after obtaining the required approval, we have got a real medical dataset from actual critical care units. For the sake of research, a portion of data was used, transformed, and manifested using laboratory-made payload ransomware and successfully labeled. The detection mechanism adopted supervised machine learning techniques of K Nearest Neighbor, Support Vector Machine, Decision Trees, Random Forest, and Logistic Regression in contrast with deep learning technique of Artificial Neural Network. The methods of KNN, SVM, and DT successfully detected ransomware's signature with an accuracy of 100%. However, ANN detected the signature with an accuracy of 99.9%. The results of this work were validated using precision, recall, and f1 score metrics.**

Keywords*—Artificial neural networks; deep learning; healthcare system; internet of things; machine learning; supervised learning

I. Introduction

The Internet of Medical Things (IoMT) is a collection of medical devices and applications that use networking technologies to connect to clinical information systems. It can reduce unnecessary hospital visits and the burden on healthcare systems by connecting patients to their medical practitioners and allowing their medical data to get transferred over a secured network. According to Frost & Sullivan, the global IoMT market was worth $22.5 billion in 2016 and was expected to be worth $72.02 billion by 2021, at a compound annual growth rate of 26.2 % [1].

According to NBC News, ransomware malware severely infected a major hospital chain in September 2020. Its impacts caused all employees and medical staff to be forced to use the traditional pen and paper method to monitor the patient's status over the weekend. This cyber-attack became the most significant in the history of the United States, as it has affected over 400 locations [2].

Ransomware is a form of malware designed to encrypt data partially or as a whole, causing the systems that rely on them to become unusable. Ransomware exists in two types; crypto and locker. According to Kaspersky [3], crypto-ransomware encrypts valuable files on a computer, making them inaccessible to the user.

Cybercriminals who carry out crypto-ransomware generate profit by demanding victims pay a ransom to recover their files. However, paying the ransom never guarantees the recovering of the victim's files. In crypto-ransomware, files are not encrypted by locker ransomware, but instead, it locks the victim out of their device(s), making it inoperable. Once locked out, cybercriminals will start executing inside attacks pressuring the victim to pay a ransom to unlock their device(s).

Crypto-ransomware can get subcategorized into other types, including payload ransomware which is the research focus in this paper. Payload ransomware encrypts values randomly stored within valuable files on a computer. In healthcare systems, encrypting values, especially for intensive care units (ICU), could lead to the loss of lives. IoMT has served the caregivers and healthcare during the covid 19 pandemic heavily as it has minimized the contact between the hospital staff and their patients. However, they need an indeed security against such attacks. In this paper, our primary focus is to detect payload ransomware's signature that has infected our medical data named Mimic III v1.4 [4]. Section IV will reveal details about this medical data and how the appropriate access is granted to researchers. Machine and deep learning techniques became viral tools that have attracted the attention of researchers in data analytics and cyber-security domains. However, the data collected from the perception layer is accompanied by numerous complications, such as dynamic data changes, their large volume accompanied by noises. These challenges require developing and implementing efficient methods to validate, visualize and extract knowledge from this immense amount of data.

This research paper is organized as follows. A literature review of other researchers detecting cyber-attacks in IoT is presented in Section II, followed by the methodologies used to detect the encrypted medical records within the dataset in Section III. Section IV briefly explained the medical dataset

used and prepared for the detection by the supervised learning techniques after feature selection, manifestation, clustering, and preprocessing. The detection of ransomware-infected records by machine learning and deep learning is explained in Section V. The evaluation of results and discussion are described in Section VI. Lastly is the conclusion in Section VII.

II. Related Work

M. M. Rashid, J. Kamruzzaman, M. Hassan, T. Imam, and S. Gordon, in [5], implemented decision trees, random forest, linear regression, support vector machine, and artificial neural network to detect cyber-attacks at fog nodes within a distributed rather than centralized system to track network traffic using two different datasets UNSW-BC15 and CICI052017. Their experimental results showed DT and RF performed better in terms of accuracy than the other algorithms.

In another study [6], M. Hasan, M. Islam, I. Zarif and M.M.A Hashem used a publicly accessible IoT dataset [7] and proposed a data analysis method to identify and prevent systems from attacks that cause abnormal behavior. They used DT, RF LR, SVM, and ANN; however, RF scored the best accuracy.

In [8], A. A. Diro and N. Chilamkurti proposed a deep learning model versus a shallow neural network model to detect normal, DoS, probe R2L and U2R traffic using the NSL-KDD dataset. The two models scored the following accuracies respectively, 99.2 % versus 98.27 % for binary classification and 95.22 % versus 96.75 % for multi-class classification.

R. Doshi, N. Apthorpe and N. Feamster in [9] trained binary classifiers to differentiate between benign and denial of service (DoS) attack traffic generated by mirai botnets. The results showed a good performance in detecting well-known signature attacks compared to new or unknown ones.

In [10], B. Ingre and A. Yadav applied ANN to NSL-KDD dataset to analyze binary and five class classification. The detection accuracy was 81%, in addition to 79 % in detecting the attack classification type.

Moreover, a hybrid learning model proposed by M. M. Lisehroodi, Z. Muda and W. Yassin in [11], implemented ANN and K-means methods for clustering within their design for an advanced network intrusion detection system. Their results showed a successful detection accuracy of 99 %.

Another hybrid model [12] was proposed by S. Peddabachigari, A. Abraham, C. Grosan, and J. Thomas, in which DT and SVM are implemented and compared to each one of them individually. Their experimental results showed DT has equal or slightly better performance in comparison to SVM and DT-SVM.

J. Zhang and M. Zulkernine in [13] implemented RF in network intrusion detection systems against DoS attacks to overcome imbalanced intrusions and reduce the error rate from 1.92 % to 0.05 %.

In this study [14], S. Mukkamala, G. Janoski and A. H. Sung applied a strategy that uses ANN and SVM to detect network traffic. ANN had an accuracy of detection of 99 %; however, the training time was 30 minutes and further 30 minutes for testing. While SVM had a lower accuracy score, it took around 52s to 211s for training and other 1s to 16s for testing.

In addition to another research where A. Azmoodeh, A. Dehghantanha, M. Conti and K.Choo target the detection of crypto-ransomware via these learning methodologies and monitor the power consumption of internet-connected devices or android devices [15].

Our work measures the accuracy of detecting payload ransomware infected patients' records within our obtained medical dataset using supervised machine learning and deep learning techniques.

However, the availability of such medical datasets is highly restricted and inaccessible unless approved by the appropriate parties after fulfilling the Health Insurance Portability and Accountability Act (HIPAA) standards and signing a data user agreement (DUA).

Therefore, studying some of the impacts of ransomware on medical data is highly demanded, significantly when this malware negatively impacts them.

III. Methodology Framework

In the following Fig. 1 is an illustrated overview of our suggested IoMT network. In this network, devices are portable and dispersed within a network-defined range, with the edge router serving as a coordinator between the control and IoMT environment zones. Wireless communication protocols are used by devices to communicate to the server in the control zone.

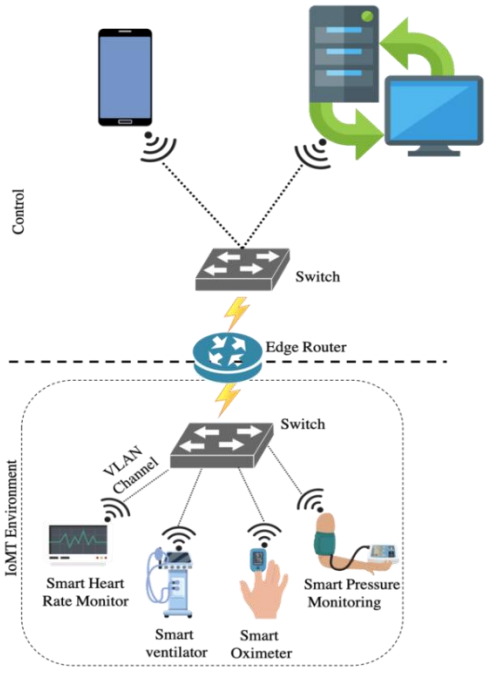

Fig. 1. System Overview.

A. Proposed Detection System

This section will discuss the steps taken to get the dataset prepared for being labeled after manifestation. This step is followed by the detection phase using the previously mentioned machine and deep learning techniques. The proposed mechanism works as illustrated in Fig. 2. Our dataset is composed of more than 300 different physiological tests available to be performed on patients during their ICU stay period.

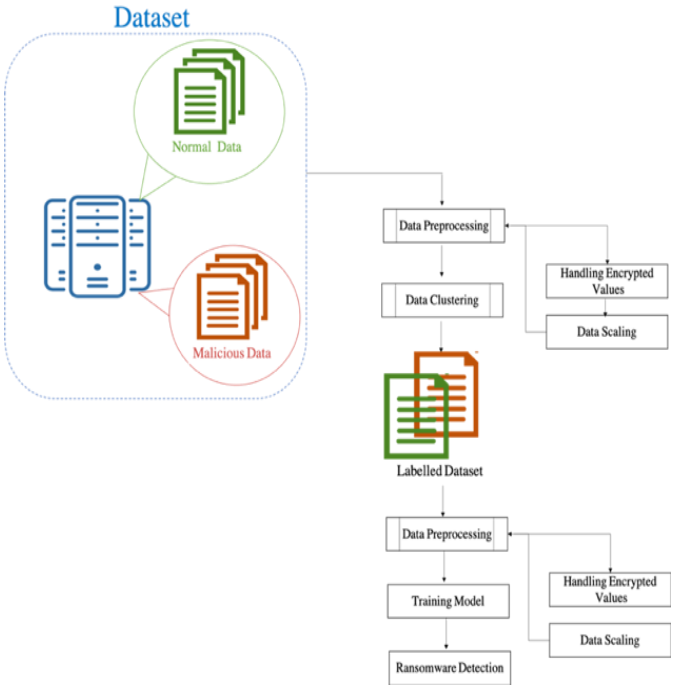

Fig. 2. Proposed System for Detection.

Therefore, we have chosen the most frequent tests performed in the ICU as our features. As the rest of the tests are not in demand or necessary to be conducted, they are performed depending on their patients' medical condition.

These tests are eligible to be captured via sensors and bedside monitors to orchestrate an IoMT environment. After transformation, the dataset becomes manifested using payload ransomware that is described in the following section. This step involves selecting random records and encrypts some of their respective features at random.

Any record that has any of its fields encrypted is considered infected. The records are either labeled as infected by number 1 or 0 for the normal ones in an additional column. This label column is later used to evaluate the supervised machine learning and deep learning techniques implemented in this work.

IV. Dataset Preparation

A. Original Dataset

The clinical dataset, MIMIC-III v1.4, used in this work is a credential actual medical data obtained from Beth Israel Deaconess Medical Center, Boston, MA, United States. This data was accessed after becoming a credentialed user on PhysioNet.

This step involves the completion of a training course on human subject research. The training received were "Human Resource: Data or Specimens Only Research" and "Human Research Data or Specimens Only Research (Course Learner Group 2).

These training courses ensure that the researchers seeking the use of this database would treat it with care and respect since it contains detailed clinical care information about patients, in addition to signing a DUA.

MIT Laboratory for Computational Physiology removed patients' critical health information such as diagnostic reports or text fields from the database. They fulfilled the HIPAA standards by eliminating all data elements such as the patient's name, phone number, and address. A random offset was used to shift the dates into the future. In addition to hiding the patients' actual age by moving it, some of this data showed the patients over 300.

The dataset was obtained using two different clinical information systems CareVue and MetaVision. Some of the tests were recorded but under other names according to their associated clinical information system.

After inspecting the dataset, it was found that some tests were monitored by one of the clinical information systems while the other did not. Therefore, it was necessary to manually go through the dataset to choose the features, especially those under shorthand terms.

The dataset consists of 61,532 ICU stays, of which there are 53,432 stays for adult patients and 8,100 for neonatal patients; however, this doesn't affect the features chosen for this work. The whole dataset is composed of 29 distinct tables.

The table of interest stores all of the medical tests conducted during the patient's ICU stay. As previously mentioned, the medical tests were under shorthand terms. Therefore, using a table provided in the dataset that provides the complete form of these tests aided in identifying the concept measured.

The choice of the features depends on its frequency; the number of the concept was carried out by devices. In addition to its eligibility to be captured remotely without any external factor, i.e. caregiver, to promote an actual remote monitoring environment. The number of records in this table was over 310 million records. Therefore, as a proof of concept, one million records were used for this investigation.

B. Feature Selection

Computing the frequency of each medical concept measured in our table of interest has resulted 353 different medical tests. Their frequencies range from 1 to 8263. Therefore, the chosen medical features are the features that have number of occurrences above 5000.

In contrast, the rest of the features had frequencies ranging from 1 to 2000. In Table I, the chosen six features are presented.

TABLE I. LIST OF FEATURES SELECTED

Feature n	*Feature Name*
1	Heart Rate
2	Respiratory Rate
3	O_2 Saturation Pulseoxymetry
4	Non-invasive blood pressure systolic
5	Non-invasive blood pressure diastolic
6	Non-invasive blood pressure means

The heart rate had a frequency of 8263. The respiratory rate was 8213. 8148 was for o_2 saturation pulseoxymetry was 8148. The non-invasive blood pressure systolic, diastolic and means are; 5526, 5524 and 5559. In this dataset, a patient could have had any of these features numerous times a day, e.g. 30 and extremely few had the first three features measured only. For those who had certain features tested innumerable times, the means of these values were computed for every single day instead.

Note that each record is associated with additional fields such as date, time and ICU ID. The ICU ID column stores the unique numbers given to the patients once admitted into the ICU. The feature's string name was used, such as heart rate in the record itself, to represent that it has been captured along with its value in the CSV file. Therefore, we have rearranged the table into transpose, and all of these features became the headers of the CSV file. This transformation has led to a decrease in the number of records within the file to 149, 360 records.

C. Data Manifestation and Labeling

The data was split into two portions (51 % and 49%). The more significant portion was fed into the manifestation process. Some of the fields' values in the patients' records were encrypted randomly regardless of their data type using the algorithm shown in Fig. 3, hiding their valid values under the signature "□ PayMeLocker Decrypt □".

Note that the malware did not make the whole record encrypted; some of its fields were encrypted, including the date, time and ICU ID. A record is considered infected if one or more fields are encrypted. Thus, this step was followed by clustering to label the record as 1 or 0 depending on its manifestation case.

Both manifested, and regular records were merged again at random; ready to be labeled. A threshold-based method was used to label the records.

We have handled the encrypted values by replacing them with unique constant numbers depending on the feature infected within the respective record. The constant value for each feature is shown in Table II.

Algorithm 1: Payload Ransomware Manifestation

Require: Data in Transpose;

1: **function** INFECT (file)
2: *Data* ← *file*
3: **for each** *r* ∈ *Data* **do**
4: **for each** *f* ∈ *Data* **do**
5: Generate *state* /* *0 or 1 state*
6: **if** state = *0* **then**
7: *f*++
8: **else**
9: Data [*r*][*f*] = Encrypt the value stored in the *feature*
10: **end if**
11: **end for**
12: **end for**
13: **end function**

Fig. 3. Manifestation Algorithm.

Note that the specific number for each feature was decided by finding the maximum value found in the dataset for each feature and doubling it to ensure its differentiation compared to the rest of the values within the same record. The first three features are in Table II represent the date, time, and ICU ID.

This step is followed by data scaling. Data scaling is a technique that normalizes the data values of features within a given dataset into a particular range. After replacing the encrypted values with numerical values, the data within each column were normalized to floating numbers ranging from zero to 1.

TABLE II. LIST OF THE DISTINCT VALUES WITHIN EACH FEATURE

Feature n	*Maximum Value*	*Distinct Value*
1	26758	53516
2	23.5	47
3	299707	599414
4	200	400
5	265	530
6	193	386

This step was implemented using the library of preprocessing and MinMaxScaler (). If the value under the selected feature carries a numerical value equivalent to 1, the whole record becomes labeled as one, i.e. infected; otherwise, it is labeled as zero, i.e. normal.

V. DETECTION

This section will discuss the procedure used for training and detecting the infected records using the most commonly implemented supervised machine and deep learning methods in the internet of things.

A. Detection using Machine Learning Techniques

To implement KNN, SVM, DT, RF, and LR, we have used their python-based libraries of sklearn neighbors, SVM, tree, ensemble, and linear model. The training dataset was read, stored in a data frame, and converted into a matrix during the classification stage.

Furthermore, these datasets are divided into training and testing datasets. The training dataset being the larger is composed of 45, 034 bengin records and 47,466 infected ones. In comparison, the testing dataset is composed of 14369 bengin records and 15503 infected ones.

After the initial training of the machine learning models using the labeled training dataset, it was applied to the testing dataset with no binary label information. Note that the k-fold cross-validation procedure was applied as a part of this work using train/test split to avoid overfitting.

B. Detection using Deep Learning Techniques

The deep–learning model of ANN uses supervised training and binary classification for identifying the infected tuples. A four-layer deep learning model was created for this study, with one input layer, two hidden ANN layers, and one output layer, a binary classifier layer.

The input layer consists of nine neurons while the hidden layers; each consist of eight neurons, and lastly, the output layer comprises two neurons. Each neuron in the ANN layer is assigned with a weight parameter adjusted using the gradient descent method, Fig. 4.

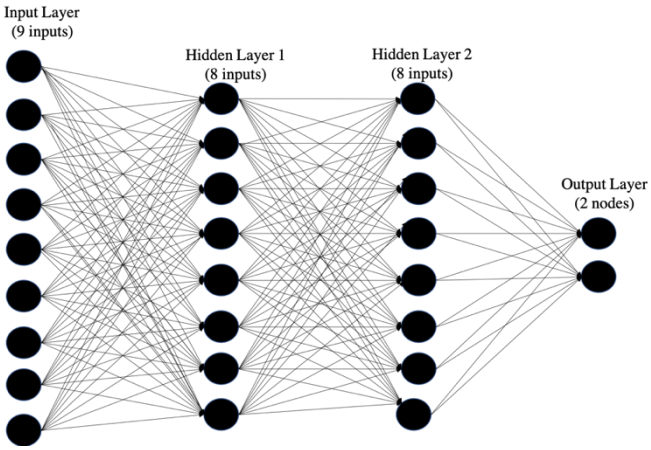

Fig. 4. Detection Mechanism using Neural Network.

Each tuple and its label information are fed into the ANN during the supervised training process, passing through the first hidden encode layer and being filtered out as x, the most significant features. The x features are passed into the second encode hidden layer, filtered and converted into y features.

Finally, the second encode layer sends them to the output layer, where they are classified as malicious or benign tuples. This process takes place at the lowest level when the feature vector fv is input into the ANN, and it passes through each layer of the deep neural network (DNN), Fig. 5.

Each DNN layer's neural nodes calculate an output using an activation function and generate a filtered result. In this work, we developed this model using a rectified linear unit (ReLU) activation function. The ReLU function is defined as follows:

$$f(x) = \max(0, x) \quad (1)$$

The smaller values in the matrix are set to zero with the input x (i.e. matrix), while the others remain constant. As a result, each hidden layer connects to the next hidden layer via a linear combination of outputs and feeds the filtered output generated by the ReLU activation function to the next layer.

The second encoded layer, like the first encoded layer, trains itself using labeled tuples. As a result, each layer of the ANN feeds on this data and maps it to a numerical value. Finally, the mapped values are normalized to 0 and 1, with 0 representing the benign tuple and 1 representing the malicious tuple.

The ANN model's objective function, a binary_crossentropy loss function, tries to minimize the total cost in the model, as shown in the following algorithm, Fig. 6. The ANN model must then be retrofitted for training and testing predictions.

The ANN was implemented using Keras, an open-source neural network library written in python, and the results are validated as well using the confusion matrix. During the classification stage, the training dataset was read, stored in a data frame in the same way as in the machine learning model. Furthermore, the same training dataset and testing dataset were used in this implementation to deduce the accuracy performance of this model with the previously conducted machine learning algorithms.

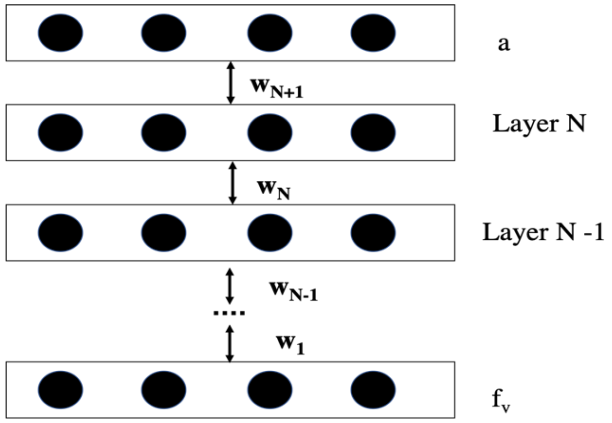

Fig. 5. DNN Structure.

Algorithm 2: Ransomware Detection using Deep-Learning model

Require: List of the 9 features

1: **function** DETECT (file)
2: *matrix* ← *file*
3: Extract *features* from matrix
4: Define $dataset_{train}$ & $dataset_{test}$
5: Initialize *sequential deep-learning model*
6: **if** *initialized* **then**
7: Compile *binary-crossentropy* classifier
8: classifier ← *sequential deep-learning model*
9: **end if**
10: Training: classifier: $dataset_{train}$
11: **if** *Training* is complete **then**
12: Prediction: classifier ← $dataset_{test}$
13: **if** *Predictions* are correct **then**
14: Re-Train the *model*
15: **end if**
16: **end if**
17: **end function**

Fig. 6. Detection Mechanism using Deep Neural Network.

A sequential 2 hidden layer was created and instantiated the ANN model, and the ReLU activation function was used to equip the processing units within each layer. Following that, the deep learning model was compiled and fitted with 100 runs, i.e. epochs and feature count. Finally, the deep learning model is assembled, and the classifiers are assigned and saved in the variable "prediction". Consequently, in every test, the results of the classifier are normalized into a binary value.

VI. Evaluation and Results

The performance metrics used to evaluate the detection model are; recall, precision and f1 score. Precision(P) and recall (R) are two essential metrics used to assess the accuracy of the detection process when there is an imbalanced classification.

These two metrics use true positive value as an outcome when the model predicts the positive class correctly. A true negative, on the other hand, is an outcome in which the model correctly predicts the negative class. A *false positive*, on the other hand, is an outcome in which the model mispredicts the positive class. A false negative is an outcome in which the model mispredicts the negative class. Lastly, a presented P-R curve refers to the composition of these two metrics.

The precision is referred to as the positive predictive value and outlines how good a model predicts the positive (anomaly) label. To calculate precision, we use the following formula:

$$P=TP/(TP+FP) \quad (2)$$

The recall is the ratio between the number of true positive labels divided by the sum of the true positive values and the false negative values. To calculate recall, we use the following formula:

$$R=TP/(TP+F) \quad (3)$$

The f1 score is a measure of a test's accuracy. It depends on the values of precision and recall. To calculate the F1 score, we use the formula below:

$$F1=(2(P)(R)) /((P+R)) \quad (4)$$

Table III shows the number of TP values in KNN, 15,453, while the true negative values are 14,419 with zero errors. These numbers mark the true actual percentage of the benign and malicious tuples within the dataset.

KNN has shown excellent performance in the detection processes with 100 % precision and recall, as shown in Table IV; however, these percentages are expected to decrease with the increase in dimensionality. KNN can aid in the detection process if accompanied by a principle component analysis (PCA) algorithm.

SVM has reached an overall precision and recall of 96 % compared with the rest of the methods in this work. Thus, SVM can show excellence when the training dataset is not large, which is not the case in biomedical data.

DT and RF did reach 100 % in terms of accuracy; however, it was computationally expensive in the word of memory space. LR has the lowest precision and recall rates, marking the worst percentage compared to the other algorithms.

ANN has scored 99.9 % in precision and recall and is expected to reach 100% as the dataset dimensionality increases. This score proves that ANN can be used and furtherly developed to be able to detect ransomware signatures.

TABLE III. Table of TP, TN, FP and FN Values for the Detection Techniques Implemented

Technique	***TP***	***TN***	***FP***	***FN***
KNN	15453	14419	0	0
SVM	14558	14012	563	739
DT	15304	14568	0	0
RF	15331	14541	0	0
LR	14693	13322	1273	584
ANN	15305	14545	11	0

TABLE IV. TABLE OF PRECISION, RECALL AND F1 SCORE VALUES FOR THE DETECTION TECHNIQUES IMPLEMENTED

Technique	*Precision*	*Recall*	*F1*
KNN	100 %	100 %	100 %
SVM	96.0 %	96 %	96 %
DT	100 %	100 %	100 %
RF	100 %	100 %	100 %
LR	92.0	96.7	94.0%
ANN	99.9%	100 %	99.5%

The following P-R curve compares the detection performances of ransomware detection using the same dataset and same training and testing percentages, Fig. 7.

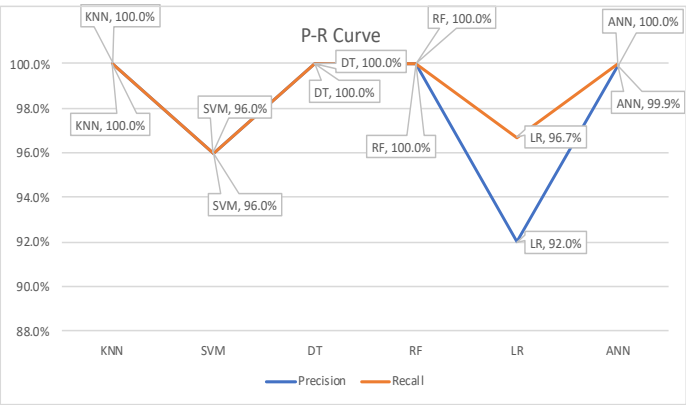

Fig. 7. P-R Curve.

In this work, we have detected infected ransomware tuples using supervised machine and deep learning techniques of KNN, SVM, DT, RF, LR, and ANN. The encryption signature of the malware used is evident within the dataset and very different from the original values stored under their respective features. Therefore, the precision score of KNN, DT, RF was 100 %, and ANN had it almost there.

VII. CONCLUSION

In this paper, machine and deep learning techniques were used to perform binary classification on a medical dataset infested with payload ransomware. The Healthcare system was not on the top priority for the security specialists a few years ago. Until the emergence of the Internet of Medical things that was heavily implemented during the pandemic of Covid 19 to enhance the infection control process. In addition to the immense increase of sensitivity of the data, it was transferring. When it was exposed to catastrophic attacks, especially ransomware, it was time to get experiment with the effectiveness of using these learning methods that have been used to famous attacks like DDoS and DoS, in detecting payload ransomware on real healthcare datasets. A subset of the dataset used was cleaned, transformed, and infested with the attack. The work implemented was evaluated by the recall, precision, and f1 score metrics. The results showed ANN showed 99.9% of accuracy in detection even while KNN, SVM, and DT were 100%. These results show that these methods can be considered to secure the data within medical or healthcare systems.

REFERENCES

[1] Internet of Medical Things Revolutionizing healthcare, Alliance of Advanced BioMedical Engineering. Available Online: https://aabme.asme.org/posts/internet-of-medical-things-revolutionizing-healthcare (accessed on 2017).

[2] K. Collier, NBC News. Available Online: https://www.nbcnews.com/tech/security/cyberattack-hits-major-u-s-hospital-system-n1241254 (accessed on September 2020).

[3] Ransomware Attacks and Types – How Encryption Trojans Differ, Kaspersky. Available Online: https://me-en.kaspersky.com/resource-center/threats/ransomware-examples (accessed on 2021).

[4] A. Johnson et al., MIMIC-III, a freely accessible critical care database. Scientific Data 2016**,** vol. 3, DOI: 10.1038/sdata.2016.35.

[5] M. M. Rashid, J. Kamruzzaman, M. Hassan, T. Imam and S. Gordon, Cyberattacks Detection in IoT-Based Smart City Applications Using Machine Learning Techniques. Int. J. Environ. Res. Public Health 2020, vol. 17, p. 9347.

[6] M. Hasan, M. Islam, I. Zarif and M.M.A Hashem, Attack and anomaly detection in IoT sensors in IoT sites using machine learning approaches. Internet Things 2019, vol. 7, p100059.

[7] C. C. Aggarwal, J. Han, J. Wang and P. Yu, A framework for clustering evolving data streams. In Proceeding VLDB Conference, Berlin, Germany, 2003; pp. 81–92.

[8] A. A. Diro and N. Chilamkurti, Distributed attack detection scheme using deep learning approach for Internet of Things. Future Gener. Comput. Syst.,2018,vol. 82, pp. 761–768.

[9] R. Doshi, N. Apthorpe and N. Feamster, Machine Learning DDoS Detection for Consumer Internet of Things Devices, IEEE Security and Privacy Workshops (SPW), San Francisco, CA, May 2018, pp. 29–35, DOI: 10.1109/SPW.2018.00013.

[10] B. Ingre and A. Yadav, Performance analysis of NSL- KDD dataset using ANN. In 2015 International Conference on Signal Processing and Communication Engineering Systems, Guntur, India, Jan 2015, pp. 92-96, DOI: 10.1109/SPACES.2015.7058223.

[11] M. M. Lisehroodi, Z. Muda and W. Yassin, A hybrid framework based on neural network MLP and K-means Clustering for Intrusion Detection System. In proceedings of the 4th International Conference on Computing and Informatics, Sarawak, Malaysia, 2013, pp. 305 – 311.

[12] S. Peddabachigari, A. Abraham, C. Grosan and J. Thomas, Modeling intrusion detection system using hybrid intelligent systems. Journal of network and computer applications 2007,vol. 30, pp. 114-132.

[13] J. Zhang and M. Zulkernine, Network Intrusion Detection using Random Forests, PST, 2005.

[14] S. Mukkamala , G. Janoski and A. H. Sung, Intrusion detection: support vector machines and neural networks. In proceedings of the IEEE - IJCNN, Honolulu, HI, USA, 2002, pp.1702–1707, DOI:0.1109/IJCNN.2002.1007774.

[15] A. Azmoodeh, A. Dehghantanha, M. Conti and K.Choo, Detecting crypto-ransomware in IoT networks based on energy consumption footprint. Journal of Ambient Intelligence and Humanized Computing, 2018, vol. 9, pp. 1141-1152, DOI: https://doi.org/10.1007/s12652-017-0558-5.