IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received May 5, 2020, accepted May 21, 2020, date of publication June 5, 2020, date of current version June 18, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3000421

Intrusion Detection System for Healthcare Systems Using Medical and Network Data: A Comparison Study

ANAR A. HADY[1,2], (Member, IEEE), ALI GHUBAISH[1], (Graduate Student Member, IEEE), TARA SALMAN[1], (Graduate Student Member, IEEE), DEVRIM UNAL[3], (Member, IEEE), AND RAJ JAIN[1], (Life Fellow, IEEE)

[1]Department of CSE, Washington University in St. Louis, Saint Louis, MO 63130, USA
[2]Electronics Research Institute, Cairo 12622, Egypt
[3]KINDI Center of Qatar University, Doha, Qatar

Corresponding author: Ali Ghubaish (aghubaish@wustl.edu)

This work was supported in part by the National Priorities Research Program (NPRP) from the Qatar National Research Fund (QNRF) under Award NPRP-10-0125-170250 (a member of the Qatar Foundation), in part by the NSF under Grant CNS-1718929, in part by the United States Agency for International Development, Ministry of Higher Education, Egypt, and in part by Prince Sattam Bin Abdulaziz University, AlKharj, Saudi Arabia.

ABSTRACT Introducing IoT systems to healthcare applications has made it possible to remotely monitor patients' information and provide proper diagnostics whenever needed. However, providing high-security features that guarantee the correctness and confidentiality of patients' data is a significant challenge. Any alteration to the data could affect the patients' treatment, leading to human casualties in emergency conditions. Due to the high dimensionality and prominent dynamicity of the data involved in such systems, machine learning has the promise to provide an effective solution when it comes to intrusion detection. However, most of the available healthcare intrusion detection systems either use network flow metrics or patients' biometric data to build their datasets. This paper aims to show that combining both network and biometric metrics as features performs better than using only one of the two types of features. We have built a real-time Enhanced Healthcare Monitoring System (EHMS) testbed that monitors the patients' biometrics and collects network flow metrics. The monitored data is sent to a remote server for further diagnostic and treatment decisions. Man-in-the-middle cyber-attacks have been used, and a dataset of more than 16 thousand records of normal and attack healthcare data has been created. The system then applies different machine learning methods for training and testing the dataset against these attacks. Results prove that the performance has improved by 7% to 25% in some cases, and this shows the robustness of the proposed system in providing proper intrusion detection.

INDEX TERMS Healthcare monitoring systems, IoT, machine learning, security, healthcare dataset.

I. INTRODUCTION

Recent revolutionary advances in the construction of the Internet of Things (IoT) systems have made it possible to design healthcare monitoring systems using low power and low-cost sensors. These sensors have been used widely in recent years to facilitate remote monitoring of patients, alleviating the need for the physical presence of doctors in the field.

Recent trends in IoT and wireless communications can efficiently support a wide range of medical applications such as early diagnosis, real-time monitoring, and medical emergencies. The adaptation of secure and practical techniques for the rapid discovery of life-threatening emergency cases in real-time can minimize the dependency on caregivers and reduce healthcare costs. The innovation of smart decision-making techniques can enable early treatments resulting in favorable health outcomes and potentially saving lives in the community. To achieve such goals, continuous monitoring of the vital signs of community residents, which can be captured through wearable sensors, is required. Healthcare providers can then provide efficient remote healthcare communication for monitoring and diagnosis services to the residents of these smart communities. Any security threat to these systems may

The associate editor coordinating the review of this manuscript and approving it for publication was Quinn Qiao.

cause a serious problem, such as imposing a false diagnosis or delaying the interaction. This leads to a violation of patients' privacy, health issues, and even death in extreme cases [1].

Machine Learning (ML) is closely related to (and often overlaps with) computational statistics, and it has strong ties to mathematical optimization [2]. Over the last decade, ML has been introduced to cybersecurity applications for hybrid network analysis that includes both misuse detection and anomaly detection. Misuse detection is used to detect known attacks by using their signatures, while anomaly detection is used to identify any abnormal behavior in the network. Using ML for managing security issues in healthcare systems is the most promising technique to be used for previously unseen (also known as zero-day) attacks [3]. It can identify attacks simply by monitoring data alteration or by detecting changes in the network's traffic characteristics. Man-In-The-Middle (MITM) attacks on the system are example attacks where packet alteration is done on the fly [4]. Although ML may not be suitable for problems that require a formal descriptive solution, it can achieve robust results in areas and issues that we have difficulty in formalizing. Therefore, ML excels in fields as data clustering and classification, which are both main blocks in applications of data security. Most internet security models are based on making a list of harmful or malicious requests to block them. However, attackers are continually using creativity in improving and changing their techniques, which makes it impossible to predict their bad requests to be inserted in the black-list. A small tweak may allow an attacker to slip by undetected. This negative model – describing all potentially harmful requests and continuously updating the ruleset – is impractical and extremely resource-intensive. At this point, ML can play a significant role in learning the good requests; thus, creating a model of them such that requests that do not coincide with them are considered anomalies that are likely to be attacks [5].

We have built an Enhanced Healthcare Monitoring System (EHMS) testbed that utilizes the ML capability for managing security issues using a variety of healthcare sensors. The system includes a gateway for data gathering, an Intrusion Detection System (IDS) computer for monitoring the network traffic and detecting abnormal behaviors, an attacker to imitate a real attack threat to the system, and a server. The server is the endpoint of the system that stores the healthcare data and makes it available to the clinic. ML models are employed to detect data alteration and spoofing threats. This is done by analyzing the patients' biometric data and network traffic characteristics. If any traffic metric or biometric data is detected to be anomalous, the method reports a threat alert to the system managers. Different ML methods have been investigated in the literature to test their suitability for security approaches [3]. We have chosen four ML methods for attack detection: Random Forest (RF), K-Nearest Neighbor (KNN), Support Vector Machine (SVM), and Artificial Neural Networks (ANN). RF is a method that combines both decision trees and ensemble learning [6]. KNN is a machine learning method that measures the distance between two instances to estimate the similarity or the difference between them [7]. SVM is a classifier that separates these instances with a hyperplane [8]. ANN is biologically inspired computational networks that learn from given examples [9]. ML methods do not work without representative and reliable data [2].

The key contributions of this paper are as follows:

1. Designing a healthcare testbed has been presented in detail. Others can replicate it for further research in this area.
2. Collecting and analyzing a new dataset related to healthcare that combines network flow and biometrics information to build proper and realistic intrusion analysis.
3. A security system that resides in the IDS has been proposed. This system does not burden the sensors that have limited resources.
4. The system monitors the network flow metrics and patient's biometrics to determine if a security attack has taken place. The system compares the performance of different ML methods to detect such attacks using a different set of features.

We compared four ML methods and have shown that combining both network flow metrics and biometrics enhances the performance of the methods.

The rest of the paper is organized as follows. The related work is presented in Section II. Section III discusses the proposed framework architecture. Section IV describes the results gathered from the experiments on the system. Finally, Section V concludes the paper and provides future work.

II. RELATED WORK

In recent years, numerous approaches have been proposed for building health monitoring systems, and the following are some examples of them.

Fotouhi *et al.* propose a general framework for a healthcare monitoring system [1]. The system consists of three components: a coordinator, access points, and a gateway. The coordinator is a node that lies on the human body to gather information from the sensors. The Access Points (APs) are static nodes attached to the walls in the room that use the same communication protocol as the one used by the sensors (i.e., ZigBee, 6LoWPAN, or BLE). These APs forward the data to a gateway, which forwards the data to the cloud through the Internet. In this system, some general approaches have been proposed for securing data but without a concrete description and testing. Also, the authors have not proposed a solution for discovering successful attack scenarios.

ML has been used in healthcare as a tool for many purposes, such as managing and controlling false alerts while reporting serious health threats, as explained by Clifton *et al.*, where a wearable health monitoring system has been described [10]. In their approach, the generated data is collaborated with the clinical observations of a specific patient

to give early alerts of any expected emergencies. The experimental work has been tested at Oxford University Hospital. This approach has not tackled security problems in such a system.

In [11], a cloud-based healthcare system has been proposed by Rani *et al.*, where data is accessed only by authorized users. The system uses the SVM method to predict patients' conditions and expected diseases. This system uses an ML approach for data mining and not to attack discoveries in data like our system.

Chakraborty *et al.* [12] propose a healthcare system design framework using blockchain technology. The blockchain technology is known to assure security, but the authors have not investigated the framework or tested it to present any benchmark results.

Alabdulatif *et al.* implement a system that provides a privacy-preserving cloud-based real-time change detection and abnormality prediction framework for multiple vital signs of a patient in [13]. The system is composed of three main blocks; the Smart Community Resident, where data is collected and aggregated to be sent to the Cloud Storage, where data is stored in an encrypted format. The last and main block is the Smart Prediction model, which uses mathematical models of the data without decryption to detect any abnormal changes and thus detects attacks. This approach focuses on conventional methods for securing data but does not consider new methods as ML for predicting security violations.

A hardware approach is proposed by Tao *et al.* in [14], where KATAN Hardware approaches for the security of IoT based healthcare monitoring systems have been introduced. A secret cipher algorithm is implemented and optimized on the FPGA hardware platform for data collection with security. This approach has the complications of hardware approaches in addition to the problems in [13].

Zhang *et al.* propose a security framework that detects anomaly traffic using the RF method on the KDD 1999 dataset [15]. The accuracy of the RF method as an anomaly detector is 95%, with a 1% false-positive rate. Note that the KDD dataset a generic "Knowledge Discovery and Data mining" dataset used in many competitions since 1999 [16]. It is not specific to healthcare and is very old. Although one of the methods in our system uses the same ML method, we have implemented a testbed to collect a dataset that closely resembles real healthcare monitoring system applications. Furthermore, our proposed system uses network flow metrics along with biometrics as features for anomaly detection.

The authors of [17], [18] use the KNN method as a basis for their cybersecurity methods. In [17], Rao and Swathi use Indexed Partial Distance Search k-Nearest Neighbor (IKPDS) to test different types of attacks, and it results in an accuracy of 99.6%. Shapoorifard and Shamsinejad in [18] focus on reducing the false alarm rate and show an accuracy of 85.2% [18]. These two approaches use an enhanced version of the KDD dataset but still suffer from the same problems and differences we mentioned earlier with the original KDD dataset.

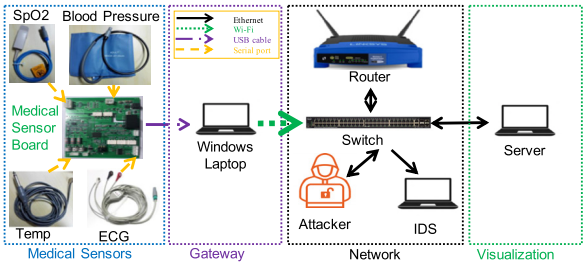


FIGURE 1. EHMS testbed.

III. ENHANCED HEALTHCARE MONITORING SYSTEM (EHMS) TESTBED

Our testbed, as shown in Figure 1, has been built using a health monitoring sensor board that collects data from several healthcare sensors placed on the patient's body. The board is attached to a Windows-based computer using a USB port. C++ based software has been developed to capture the sensed data. The computer acts as the gateway from which data is transferred to a server through Wi-Fi using TCP/IP protocol. All the machines are connected to a switch using Ethernet cables except the gateway computer. The switch is connected to the Internet through a router that the gateway is connected via Wi-Fi. Securing transferred data in the testbed mainly relies on the use of ML to help the healthcare monitoring system detecting any tampering in the transmitted data between the nodes in the network in real-time. If detected, the system reports a threat alert to the system managers. In addition to these flow packets, the sensed data from the sensors attached to the patient's body are collected to help train the model. We have assumed that the data is being transmitted in plain text since the other methods like Transport Layer Security (TLS) certificates require more processing power, which is generally not feasible with low-cost sensors.

Our EHMS testbed system works as shown in Figure 2, data flows across the system from sensors attached to the patient's body through the sensor board to the gateway to the switch and finally, to the display screen of the server. On the journey of the data from the switch to the server, an attacker may intrude to spoof or alter data before its arrival at the server. Meanwhile, network and patient data metrics are captured at the IDS computer. Data is processed at the IDS for training and testing the machine learning methods as well as real-time detection of any abnormalities.

Our system uses Argus to collect all network traffic flows and patient data between the gateway and the server. Argus is open-source software that is used to monitor the network flow traffic in real-time [19].

A. MODEL ARCHITECTURE

The system consists of six building blocks: a multi-sensor board, a gateway, a server, an IDS, an attacker, and a network. The functionality of each block is summarized below:

1. **PM4100 Six Pe Multi-Sensor Board** A product of Medical Expo that is used for sensing the patient's

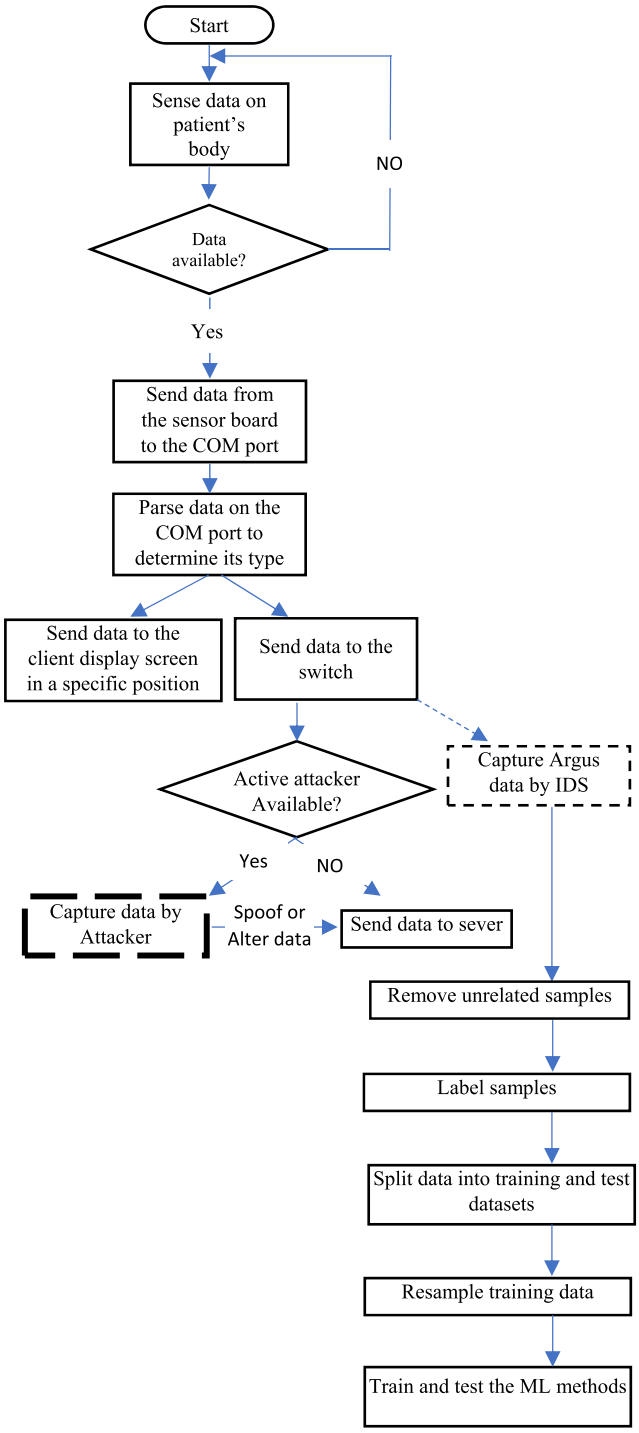


FIGURE 2. EHMS flowchart.

biometric data using a set of sensors attached to the patient's body [20]. The board has four sensors, as shown in Figure 3:

i Electrocardiogram (ECG or EKG) sensor consists of three-electrode pads attached to the patient's body to measure the patient's heart electricity.

ii Blood Oxygen Saturation (SpO2) sensor is used to measure the oxygen level in the patient's blood and the heart rate. A value of 95-100 percent is considered normal. While a level below 90 percent results in hypoxemia, levels below 80 percent may compromise brain and heart functions and may lead to respiratory or cardiac arrest.

iii The temperature sensor is used to measure the patient's body temperature.

iv The blood pressure sensor is a step-wise gassing method adopted to measure the patient's systolic and diastolic arterial pressure.

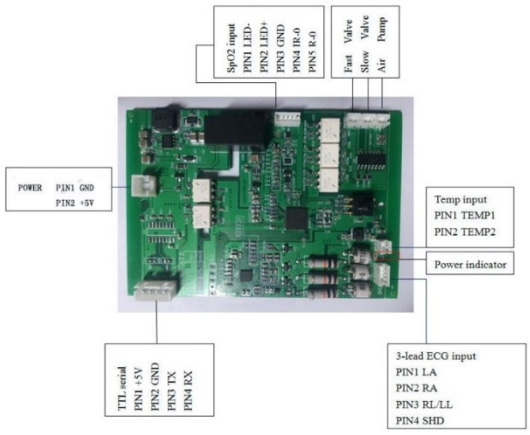


FIGURE 3. PM4100 six pe multi-sensor board.

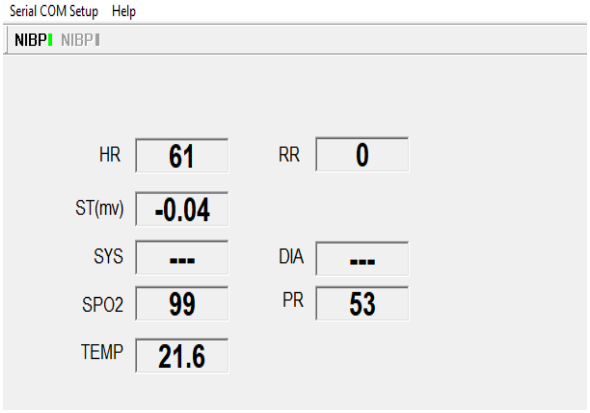


FIGURE 4. Gateway graphical user interface.

2. **The Gateway:** A Windows-based laptop to which the multi-sensor board is connected via a USB port. The data received from the board is presented on the Graphical User Interface (GUI) to monitor the patient's biometric data. The gateway sends this real-time data to the server for processing. All this process is done via a C++ program. This gateway is connected to the switch with an Ethernet cable. The GUI, as shown in Figure 4, shows the following:
HR: Heart Rate in Beats Per Minute (BPM)
RR: Respiration Rate in BPM
ST: Electrically neutral area between ventricular depolarization (QRS complex) and repolarization (T wave) in millivolts (mv).
SYS: SYStolic blood pressure.

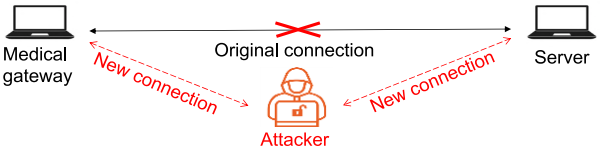


FIGURE 5. MITM attack.

DIA: DIAstolic blood pressure.
SPO2: Blood oxygen.
PR: Pulse Rate in BPM.
TEMP: Temperature in degrees Celsius.

3. **Server:** An Ubuntu-based laptop to which the data is transmitted from the gateway for further saving and analysis to make suitable medical decisions. The data is collected using a C++ program.
4. **Network:** A regular Ethernet switch to connect the server, the IDS, and the attacker computer in one network is used. A router has been connected to this switch to assign IP addresses for all computers dynamically. The gateway is attached to this router via Wi-Fi.
5. **IDS:** The switch makes a copy of (i.e., mirrors) all packets going to the server and sends it to IDS computer. This computer runs Argus network flow monitoring software and collects network flow metrics as well as the patient's biometric data. This computer also makes an online decision for any new traffic packet with any of the four methods.
6. **Attacker:** A Kali-Linux-based computer is used to initiate attacks on the system and mimic a dangerous scenario in healthcare monitoring systems. These attacks include spoofing and altering a patient's biometric data during its transmission over the network. A python script with the use of a Scapy library has been used to initiate these attacks [21]. This library features sniffing of live connections, spoofing packets, and packet alteration on the fly. It supports active and passive dissection of many protocols and includes many features for network and host insecurity analysis.

B. TYPES OF ATTACKS

The system uses a MITM attack where the attacker pretends to be a router and gets the packets first. It spoofs/alters the packets and redirects them to the server, as shown in Figure 5 and discussed below:

1. **Spoofing attacks:** In this attack, the attacker gets a copy of each packet in the network. This violates the confidentiality and privacy that is legally required in healthcare systems.
2. **Data alteration:** In this attack, the attacker alters some parts of the data that have been redirected to the attacker computer from the gateway computer. The alterations may be random or according to a rule. It then redirects the packet back to the server. This may cause severe harm to the patients as they may get the wrong treatment based on the false diagnostics resulting from the modifications made by the attacker.

TABLE 1. Machine Learning Features.

Metric	Description	Type
SrcBytes	Source Bytes	Flow metric
DstBytes	Destination Bytes	Flow metric
SrcLoad	Source Load	Flow metric
DstLoad	Destination Load	Flow metric
SrcGap	Source missing bytes	Flow metric
DstGap	Destination missing bytes	Flow metric
SIntPkt	Source Inter Packet	Flow metric
DIntPkt	Destination Inter Packet	Flow metric
SIntPktAct	Source Active Inter Packet	Flow metric
DIntPktAct	Destination Active Inter Packet	Flow metric
SrcJitter	Source Jitter	Flow metric
DstJitter	Destination Jitter	Flow metric
sMaxPktSz	Source Maximum Transmitted Packet size	Flow metric
dMaxPktSz	Destination Maximum Transmitted Packet size	Flow metric
sMinPktSz	Source Minimum Transmitted Packet size	Flow metric
dMinPktSz	Destination Minimum Transmitted Packet size	Flow metric
Dur	Duration	Flow metric
Trans	Aggregated Packets Count	Flow metric
TotPkts	Total Packets Count	Flow metric
TotBytes	Total Packets Bytes	Flow metric
Loss	Retransmitted or Dropped Packets	Flow metric
pLoss	Percentage of Retransmitted or Dropped Packets	Flow metric
pSrcLoss	Percentage of Source Retransmitted or Dropped Packets	Flow metric
pDstLoss	Percentage of Destination Retransmitted or Dropped Packets	Flow metric
Rate	Number of Packets per Second	Flow metric
Load	Load	Flow metric
Temp	Temperature	Biometric
SpO2	Peripheral Oxygen Saturation	Biometric
Pulse_Rate	Pulse Rate	Biometric
SYS	Systolic Blood Pressure	Biometric
DIA	Diastolic Blood Pressure	Biometric
Heart_Rate	Heart Rate	Biometric
Resp_Rate	Respiration rate	Biometric
ST	ECG ST segment	Biometric

C. DATASET COLLECTION

The data features used for training and testing are presented in Table 1. Sixteen thousand data samples were collected and labeled as 0 for normal (non-attack) traffic, and 1 for the attack traffic. Source MAC address is used to label the

data where the samples with the attacker computer MAC addresses are labeled as 1 while the rest as 0. In addition, unrelated samples to the gateway, attacker, and server MAC addresses are removed.

D. ML MODELS

We used four ML methods for training and testing the system against attacks. RF, KNN, SVM, and ANN are used to build the attack detection models. The following will highlight these methods to give the reader a brief overview of their concepts, but extensive details can be found in [6]–[9]:

1. **RF:** a set of decision trees from a random subset of the dataset. It then collects all the votes from these decision trees to determine the suitable class for the test objects. In this method, the maximum number of features for the best split in the trees can be assigned. We set the maximum number of features at 18 features for the network-only and combined set of features since it achieves the highest performance for both of them. Since only eight biometric features are involved in the bio-related features, we set the maximum number of biometric features to three.
2. **KNN:** a non-parametric method that classifies the test object by a plurality vote of its neighbors with the object being assigned to the class most common among its k-nearest neighbors. The hyperparameters used for all types of features (Net-only, Bio-only, combined) are as follows:
    a. The number of neighbors equals to 2 where it is the best out of a range from 1 to 100.
    b. Power parameter equals to 4 where it is the best out of a range from 1 to 100.
3. **SVM:** The SVM method used in this paper is linear-SVM, which is a parametric method. It classifies the test object by separating the objects using a hyperplane.
4. **ANN:** a multi-layer network that is fully connected, which is a brain-like system used to find patterns in data with input, hidden, and output layers. We have set the layers as follows: 40, 40, 20, 10, 10, 10, 10, 1 where 40 is the dimension of the input layer, 1 is the dimension of the output layer, and the rest are for hidden layers. The initial settings of this setup have been taken from [22].

Our dataset consists of 14k normal samples and 2k attack samples making a total of 16k samples. We used 80% of these for training and the rest for testing.

IV. RESULTS

In this section, we present our analysis and results using the dataset and ML methods discussed above. First, we discuss the dataset preprocessing stage, including the cleaning and resampling techniques. Then we evaluate the ML methods using the Accuracy and Area-under the ROC Curve (AUC) metrics.

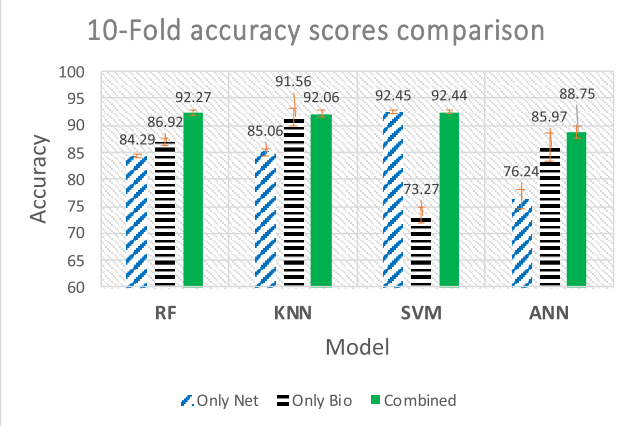


FIGURE 6. 10-Fold accuracy scores comparison.

A. DATA PREPROCESSING

In any ML application, preprocessing the data is an essential step since the ML method results are as good as the data used. Hence, the traffic flow metrics and biometrics are first preprocessed using the following steps:

1. **Splitting data into train and test datasets:** To correctly measure the performance of the ML models, we split the dataset into training and testing datasets with a distribution of 80% and 20%, respectively.
2. **K-Fold:** The K-fold method with ten folds was applied only on the training dataset to show the variety of the performance among the folds [23].
3. **Resampling:** The collected dataset was unbalanced, where normal samples constituted about 88% of the data. This can result in bad models that are unable to classify attacks [24]. Therefore, we used an over-sampling technique, SMOTE, to balance the dataset at the training stage [25].

B. MODELS' EVALUATION

To check the validity of using ML to differentiate between normal and attack biometric data, we used four ML methods and compared them based on their performances using accuracy and AUC metrics. Accuracy is the ratio of the number of samples that are correctly predicted to the total number of samples, while AUC summarizes the area under the ROC curve into a float number ranging from 0 to 1. ROC is an excellent evaluation metric to measure the trade-off between sensitivity and specificity [26]. K-Fold Cross-Validation with 10-folds is used for the statistical validation of the results on the training dataset. For this, the dataset is divided into ten subsets; in each fold, nine subsets are used for training and one for testing [23].

Figure 6 shows the accuracy results for all four models built with only biometrics features, only network features, and combined features. As can be seen, all models perform better with combined features compared to only biometrics features. Compared to only network features, RF, KNN, and ANN

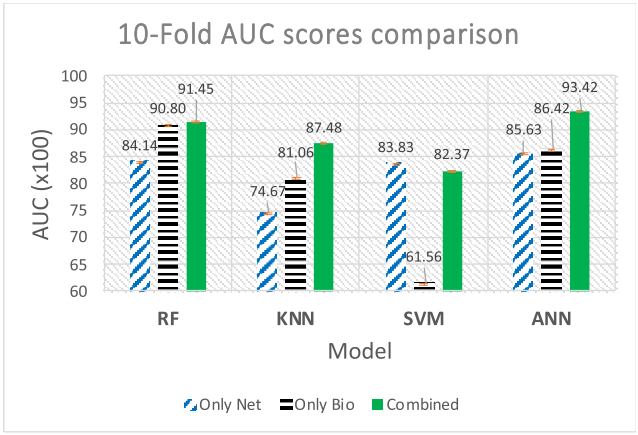
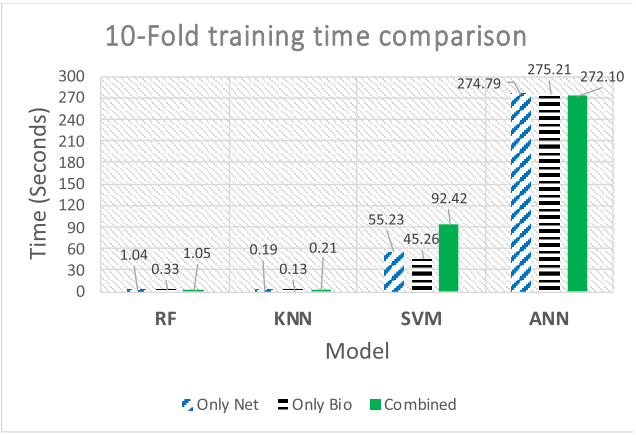
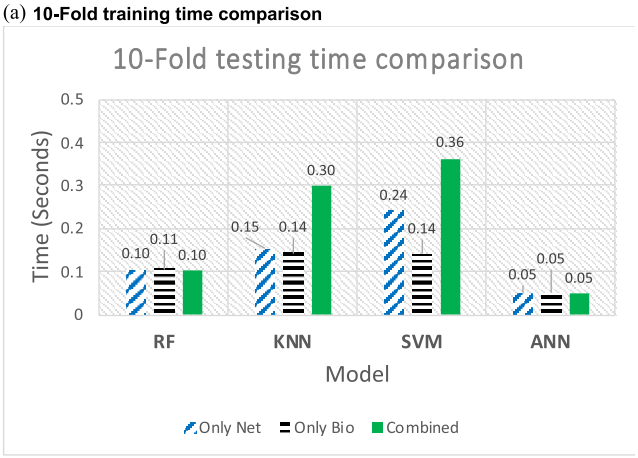


FIGURE 7. 10-Fold AUC scores comparison.



(a) 10-Fold training time comparison



(b) 10-Fold testing time comparison

FIGURE 8. Time comparison for all the models.

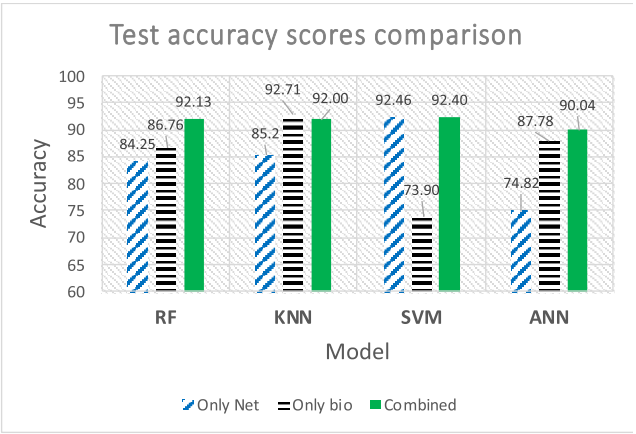


FIGURE 9. Test Accuracy scores comparison.

show significantly better results while SVM performance is similar. This indicates that using combined features provides better results than using only one of the two types of features. However, some of the confidence intervals of the accuracy results over the ten K-fold runs overlap. This indicates that accuracy is invariant in these overlapping cases, or the performance is not statistically different.

Giving the previous invariant results and the fact that accuracy is not a good measure for security application [27], we also used the AUC metric to show the validity of the accuracy results. As shown in Figure 7, the AUC scores confirm the advantage of using combined features, with no overlap.

Finding the optimal model is essential in healthcare systems, but the time spent in training and predicting the samples is as important. As a result, the average training time and prediction time using the K-fold method for all the four ML methods have been shown in Figure 8.a and 8.b, respectively.

As shown in Figure 8, the training times for RF, KNN, and SVM are less than 1.5 minutes across different types of features, compared to ANN, which is around 5 minutes. Also, it is clear that the training time increases as the number of features increases in the first three methods. However, the training time is during offline mode. On the other hand, prediction time is crucial since it is during the online mode, and every second is essential for these systems. The time all the models have taken is 300 milliseconds in the worst-case scenario. However, in such systems, this time is still high, considering the real-time requirements of the system. ANN shows the lowest in prediction time and the highest in AUC compared to the other three models. Thus, this model is the best for these systems.

As shown in Figures 9 and 10, applying the same models to the test dataset, we can see that all the models perform similarly or better using the combined features. These results are similar to the K-fold results where AUC distinguishes their performance better than the accuracy. The improvement in AUC scores reaches up to 25% (in the SVM model.) In addition, ANN shows the highest performance compared to other methods with an AUC score of 92.98%. Because the training and prediction time for all the models are similar to the average timing in the K-fold experiment, we do not show their figures.

These results lead to the conclusion that using network flow metrics with patients' biometrics enhanced the ML

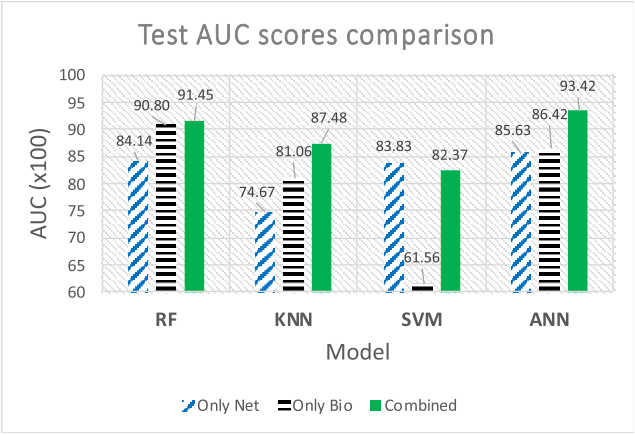


FIGURE 10. Test AUC scores comparison.

methods for securing health monitoring systems. Also, these results have shown that not all ML methods are suitable for health monitoring systems, especially in terms of prediction time. ANN requires the lowest time for prediction compared to the other methods.

V. CONCLUSIONS

Due to the high demand for remote healthcare monitoring systems nowadays, a secure system that guarantees the integrity and confidentiality of the data is required. Several small sensors are attached to a patient's body to record the biometric data to keep track of the patient's health. To achieve the full advantages of these sensors, their ability to communicate with remote servers is essential. However, their physical constraints, such as low processing power and limited battery power, may prevent them from providing required security and privacy for the patient's data. One of the solutions to such constraints is using IDSs to ensure the security requirements of such systems.

Nevertheless, most of the available healthcare IDSs either use network flow metrics or patients' biometric data to build their datasets. In this paper, we presented the design of an EHMS testbed, where several small sensors were attached to a patient's body. We created a realistic healthcare dataset of more than 16 thousand records of normal and MITM attack packets. To build an efficient IDS, we proposed to combine the network flow metrics along with the patient's biometrics as features to enhance the system performance. We used four different ML methods, RF, KNN, SVM, and ANN. Then, we compared their performance using three different types of features to train them. Results showed that the AUC could be enhanced by up to 25% by combining the flow metrics and biometrics data. Furthermore, these features had minimal effect on the testing prediction time for the best performing model.

However, the results show that the system performance is not optimal, which requires further investigation. For future work, we plan to enhance the methods' performance by choosing optimal hyperparameters, reducing feature space, and launching more sophisticated attacks.

REFERENCES

[1] H. Fotouhi, A. Causevic, K. Lundqvist, and M. Bjorkman, "Communication and security in health monitoring systems—A review," in *Proc. IEEE 40th Annu. Comput. Softw. Appl. Conf. (COMPSAC)*, Jun. 2016, pp. 545–554.

[2] Y. Xin, L. Kong, Z. Liu, Y. Chen, Y. Li, H. Zhu, M. Gao, H. Hou, and C. Wang, "Machine learning and deep learning methods for cybersecurity," *IEEE Access*, vol. 6, pp. 35365–35381, 2018.

[3] A. L. Buczak and E. Guven, "A survey of data mining and machine learning methods for cyber security intrusion detection," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 2, pp. 1153–1176, 2nd Quart., 2016.

[4] Q. Zhang, P. Wang, D. S. Reeves, and P. Ning, "Defending against sybil attacks in sensor networks," in *Proc. 25th IEEE Int. Conf. Distrib. Comput. Syst. Workshops*, 2005, pp. 185–191.

[5] A. Mathews, "What can machine learning do for information security?" *Netw. Secur.*, vol. 2019, no. 4, pp. 15–17, Apr. 2019.

[6] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.

[7] Wikipedia. *K-Nearest Neighbors Algorithm*. Accessed: Mar. 7, 2020. [Online]. Available: https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm

[8] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, 1995.

[9] Wikipedia. *Artificial Neural Network*. Accessed: Mar. 7, 2020. [Online]. Available: https://en.wikipedia.org/wiki/Artificial_neural_network

[10] L. Clifton, D. A. Clifton, M. A. F. Pimentel, P. J. Watkinson, and L. Tarassenko, "Predictive monitoring of mobile patients by combining clinical observations with data from wearable sensors," *IEEE J. Biomed. Health Informat.*, vol. 18, no. 3, pp. 722–730, May 2014.

[11] A. Rani, A. Viswasa, and E. Baburaj, "Secure and intelligent architecture for cloud-based healthcare applications in wireless body sensor networks," *Int. J. Biomed. Eng. Technol.*, vol. 29, no. 2, pp. 186–199, 2019.

[12] S. Chakraborty, S. Aich, and H.-C. Kim, "A secure healthcare system design framework using blockchain technology," in *Proc. 21st Int. Conf. Adv. Commun. Technol. (ICACT)*, Feb. 2019, pp. 260–264.

[13] A. Alabdulatif, I. Khalil, A. R. M. Forkan, and M. Atiquzzaman, "Real-time secure health surveillance for smarter health communities," *IEEE Commun. Mag.*, vol. 57, no. 1, pp. 122–129, Jan. 2019.

[14] H. Tao, M. Z. A. Bhuiyan, A. N. Abdalla, M. M. Hassan, J. M. Zain, and T. Hayajneh, "Secured data collection with hardware-based ciphers for IoT-based healthcare," *IEEE Internet Things J.*, vol. 6, no. 1, pp. 410–420, Feb. 2019.

[15] J. Zhang, M. Zulkernine, and A. Haque, "Random-Forests-Based network intrusion detection systems," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 38, no. 5, pp. 649–659, Sep. 2008.

[16] UCI KDD Archive. (Oct. 2007). *KDD Cup 1999 Data*. Accessed: Mar. 7, 2020. [Online]. Available: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html

[17] B. B. Rao and K. Swathi, "Fast kNN classifiers for network intrusion detection system," *Indian J. Sci. Technol.*, vol. 10, no. 14, pp. 1–10, Apr. 2017.

[18] H. Shapoorifard and P. Shamsinejad, "Intrusion detection using a novel hybrid method incorporating an improved KNN," *Int. J. Comput. Appl.*, vol. 173, no. 1, pp. 5–9, Sep. 2017.

[19] *Argus Project*. Accessed: Mar. 7, 2020. [Online]. Available: https://qosient.com/argus

[20] Medical Expo. *ECG Module For Multi-Parameter Monitor/SPO2/Blood Pressure*. Accessed: Mar. 7, 2020. [Online]. Available: https://www.medicalexpo.com/prod/shanghai-berry-electronic-tech-co-ltd/product-122578-866837.html

[21] Scapy. *Scapy Project*. Accessed: Mar. 7, 2020. [Online]. Available: https://scapy.net

[22] N. Koroniotis, N. Moustafa, E. Sitnikova, and B. Turnbull, "Towards the development of realistic botnet dataset in the Internet of Things for network forensic analytics: Bot-IoT dataset," *Future Gener. Comput. Syst.*, vol. 100, pp. 779–796, Nov. 2019.

[23] Wikipedia. *Cross-Validation (Statistics)*. Accessed: Mar. 7, 2020. [Online]. Available: https://en.wikipedia.org/wiki/Cross-validation_(statistics)#k-fold_cross-validation

[24] S. Liu, Y. Wang, J. Zhang, C. Chen, and Y. Xiang, "Addressing the class imbalance problem in Twitter spam detection using ensemble learning," *Comput. Secur.*, vol. 69, pp. 35–49, Aug. 2017.
[25] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *J. Artif. Intell. Res.*, vol. 16, pp. 321–357, Jun. 2002.
[26] Wikipedia. *Receiver Operating Characteristic*. Accessed: Mar. 7, 2020. [Online]. Available: https://en.wikipedia.org/wiki/Receiver_operating_characteristic
[27] Jason Brownlee. *Failure of Classification Accuracy for Imbalanced Class Distributions*. Accessed: Mar. 7, 2020. [Online]. Available: https://machinelearningmastery.com/failure-of-accuracy-for-imbalanced-class-distributions/

ANAR A. HADY (Member, IEEE) received the B.S. degree in computer engineering from the Computer Engineering and Systems Department, Faculty of Engineering, Ain Shams University, in 2002, the M.Sc. degree from the Faculty of Engineering, Cairo University, in 2007, and the Ph.D. degree from the Faculty of Engineering, Ain Shams University, in 2014.

She was a Postdoctoral Scholar of computer science and engineering at the Washington University in St. Louis, Missouri, MO, USA, from 2018 to 2019. She is currently a Researcher with the Electronics Research Institute (ERI), Egypt. She is an author of many articles in reputable journals, conferences, and book chapters. She was a Co-PI of a finished project for developing a prototype of a sensor network for precision agriculture. Her research interests include wireless sensor networks, network security, and the Internet of Things.

ALI GHUBAISH (Graduate Student Member, IEEE) received the B.S. degree in computer engineering (minor in networking) from Prince Sattam Bin Abdulaziz University, AlKharj, Saudi Arabia, in 2013, and the M.S. degree in computer engineering (minor in networking) from Washington University in St. Louis, Missouri, MO, USA, in 2017, where he is currently pursuing the Ph.D. degree in computer science and engineering.

From 2013 to 2014, he worked as a Teaching Assistant at Prince Sattam Bin Abdulaziz University. Since 2018, he has been working as a Graduate Research Assistant with Washington University in St. Louis. His research interests include network and system security, the Internet of Things, healthcare systems, and unmanned aerial vehicles (UAVs) communications.

TARA SALMAN (Graduate Student Member, IEEE) received the B.S. degree in computer engineering and the M.S. degree in computing (networking minor) from Qatar University, Doha, Qatar, in 2012 and 2015, respectively. She is currently pursuing the Ph.D. degree in computer science and engineering with Washington University in St. Louis, Missouri, MO, USA.

From 2012 to 2015, she worked as a Research Assistant with Qatar University on an National Priorities Research Program (NPRP) funded project, targeting physical layer security. She has been working as a Graduate Research Assistant with Washington University in St. Louis, since 2015. She is an author of one book chapter and many international conferences and journals. Her research interests include network security, distributed systems, the Internet of Things, and financial technologies.

DEVRIM UNAL (Member, IEEE) received the M.Sc. degree in telematics from Sheffield University, U.K., and the Ph.D. degree in computer engineering from Bogazici University, Turkey, in 1998 and 2011, respectively. He is currently a Research Assistant Professor of Cyber Security with the KINDI Center for Computing Research, College of Engineering, Qatar University. He is an active participant in international research projects, and was a member of the NATO SPS ISEG Committee from 2015 to 2018. His research interests include cyberphysical systems and IoT security, wireless security, artificial intelligence, and next generation networks.

RAJ JAIN (Life Fellow, IEEE) is currently the Barbara J. and Jerome R. Cox, Jr., Professor of computer science and engineering with Washington University in St Louis. He is a Fellow of the ACM and AAAS. He was one of the Co-founders of Nayna Networks, Inc., a next-generation telecommunications systems company in San Jose, CA, USA. He was a Senior Consulting Engineer at Digital Equipment Corporation in Littleton, Mass, and then a Professor of computer and information sciences, The Ohio State University in Columbus, Columbus, OH, USA. He is a recipient of the 2018 James B. Eads Award from St. Louis Academy of Science, the 2017 ACM SIGCOMM Life-Time Achievement Award, the 2015 A. A. Michelson Award from Computer Measurement Group and ranks among the Most Cited Authors in Computer Science. He is the author of the *Art of Computer Systems Performance Analysis*, which won the 1991 "Best-Advanced How-to Book, Systems" Award from the Computer Press Association.