computers

MDPI

Article

Enhanced Random Forest Classifier with K-Means Clustering (ERF-KMC) for Detecting and Preventing Distributed-Denial-of-Service and Man-in-the-Middle Attacks in Internet-of-Medical-Things Networks

Abdullah Ali Jawad Al-Abadi [1,*], Mbarka Belhaj Mohamed [2] and Ahmed Fakhfakh [3]

[1] Laboratory of Signals, Systems, Artificial Intelligence and Networks (SM@RTS), Digital Research Center of Sfax (CRNS), University of Sfax, National School of Engineers of Sfax (ENIS), Sfax 3038, Tunisia
[2] Laboratory of Signals, Systems, Artificial Intelligence and Networks (SM@RTS), Digital Research Center of Sfax (CRNS), University of Sfax, National School of Engineers of Gabes (ENIG), Gabes 6029, Tunisia; mbarka.belhajmohamed@gmail.com
[3] Laboratory of Signals, Systems, Artificial Intelligence and Networks (SM@RTS), Digital Research Center of Sfax (CRNS), University of Sfax, National School of Electronics and Telecommunications of Sfax (ENET'com), Sfax 1163, Tunisia; ahmed.fakhfakh@enetcom.usf.tn
* Correspondence: abdullah.jawad.1980@gmail.com

Abstract: In recent years, the combination of wireless body sensor networks (WBSNs) and the Internet ofc Medical Things (IoMT) marked a transformative era in healthcare technology. This combination allowed for the smooth communication between medical devices that enabled the real-time monitoring of patient's vital signs and health parameters. However, the increased connectivity also introduced security challenges, particularly as they related to the presence of attack nodes. This paper proposed a unique solution, an enhanced random forest classifier with a K-means clustering (ERF-KMC) algorithm, in response to these challenges. The proposed ERF-KMC algorithm combined the accuracy of the enhanced random forest classifier for achieving the best execution time (ERF-ABE) with the clustering capabilities of K-means. This model played a dual role. Initially, the security in IoMT networks was enhanced through the detection of attack messages using ERF-ABE, followed by the classification of attack types, specifically distinguishing between man-in-the-middle (MITM) and distributed denial of service (DDoS) using K-means. This approach facilitated the precise categorization of attacks, enabling the ERF-KMC algorithm to employ appropriate methods for blocking these attack messages effectively. Subsequently, this approach contributed to the improvement of network performance metrics that significantly deteriorated during the attack, including the packet loss rate (PLR), end-to-end delay (E2ED), and throughput. This was achieved through the detection of attack nodes and the subsequent prevention of their entry into the IoMT networks, thereby mitigating potential disruptions and enhancing the overall network efficiency. This study conducted simulations using the Python programming language to assess the performance of the ERF-KMC algorithm in the realm of IoMT, specifically focusing on network performance metrics. In comparison with other algorithms, the ERF-KMC algorithm demonstrated superior efficacy, showcasing its heightened capability in terms of optimizing IoMT network performance as compared to other common algorithms in network security, such as AdaBoost, CatBoost, and random forest. The importance of the ERF-KMC algorithm lies in its security for IoMT networks, as it provides a high-security approach for identifying and preventing MITM and DDoS attacks. Furthermore, improving the network performance metrics to ensure transmitted medical data are accurate and efficient is vital for real-time patient monitoring. This study takes the next step towards enhancing the reliability and security of IoMT systems and advancing the future of connected healthcare technologies.

Keywords: Internet of Medical Things; DDoS attacks; MITM attacks; machine learning; prevention; simulation

Citation: Al-Abadi, A.A.J.; Mohamed, M.B.; Fakhfakh, A. Enhanced Random Forest Classifier with K-Means Clustering (ERF-KMC) for Detecting and Preventing Distributed-Denial-of-Service and Man-in-the-Middle Attacks in Internet-of-Medical-Things Networks. *Computers* **2023**, *12*, 262. https://doi.org/10.3390/computers12120262

Academic Editor: Paolo Bellavista

Received: 19 November 2023
Revised: 12 December 2023
Accepted: 13 December 2023
Published: 17 December 2023

1. Introduction

WBSNs are made up of several small physiological sensor nodes (PSNs) in order to monitor vital health indicators, including blood pressure, blood-sugar levels, and chronic diseases; therefore, WBSNs have become a key advancement in the field of healthcare technology. These PSNs use wearable sensors to gather private information in real-time that is then painstakingly processed and sent to a central base station (BS) in an open area. The importance of WBSNs in healthcare applications is the capacity to identify medical problems at early stages, offering the invaluable opportunity for remote diagnosis. However, the deployment of a WBSN exposes it to internal and external dangers, leaving it vulnerable to different types of attacks. The security of the transmitted data is crucial due to the sensitive nature of the patient information handled by WBSNs. Protecting the privacy, usability, and integrity of the sensed data in the field of healthcare data management is a crucial requirement [1]. These types of services have effortlessly adapted to modern society, as characterized by the eradication of temporal and spatial limitations as a result of technological improvements. The healthcare industry is a leader in utilizing these technologies to provide all-around in-the-moment services. Entities such as people, machines, and various devices are intricately connected inside an information environment. They transcend geographical boundaries and time restrictions and are encompassed under the large umbrella of the Internet of Things (IoT).

The IoMT was created as a result of the growth and development of the IoT, which have transformed the healthcare sector. The IoMT comprise the global interconnection of medical equipment that enables accessibility at any time and anywhere for everyone. Within this environment, IoMT-based e-health apps have assumed a leading position in promoting a global push toward better living. As a result, healthcare services have undergone a paradigm shift and are now user-centric, accurate, widespread, and personalized, much like having a private healthcare provider on-call at all times. However, impending problems loom large and require prompt attention if the full potential of healthcare applications employing IoMT is to be realized. Particularly, IoMT devices that make up the core of the IoMT edge networks, including implantable sensors and medical wearables, have been susceptible to a variety of security risks. Patient privacy and safety have been seriously threatened by these vulnerabilities, which has necessitated strict mitigation and resolution techniques [2].

In the realm of security for IoMT networks, various techniques have been explored to enhance the detection and prevention of attacks. One approach, as outlined by Sami et al. [3], focused on identifying DDoS assaults by monitoring server usage patterns and setting specific thresholds based on metrics such as incoming requests, data-transfer velocity, and CPU utilization. By incorporating the time-to-live (TTL) component, this method scrutinized incoming requests, considering abnormal TTL values as potential indicators of malicious entities, particularly botnets. Countermeasures, such as blocking IP addresses and rate limiting, were activated upon detecting a probable DDoS attack. Another study by Hady et al. [4] introduced an intrusion detection system (IDS) for e-healthcare, integrating machine-learning algorithms, like random forest, K-nearest neighbors, support vector machines, and artificial neural networks, to improve the network security. Addressing false alarms, Iwendi et al. [5] employed a hybrid approach, combining a genetic algorithm with random forest in order to minimize false positives in intrusion detection systems. The research by Kamble and Gawade [6] emphasized the use of IoT and cryptographic encryption to safeguard healthcare automation against denial-of-service (DoS) and MITM attacks. Additional studies investigated issues such as adversarial attacks on machine-learning-based healthcare systems [7,8], the detection of multiple-initiators–multiple-attractors (MIMA) attacks in wireless sensor networks (WSNs) [9], and the application of deep learning for DDoS attack detection [10,11]. Additionally, research has explored the development of traffic generation tools [11,12] and secure machine-learning-based disease detection in medical wireless body sensor networks [13]. The integration of IoT and wireless body sensor networks was explored for continuous healthcare monitoring [13], and autonomous security

systems using edge computing and CNNs were proposed for DDoS detection in IoT [14]. A machine-learning technique combining clustering and graphing structural properties was introduced in order to anticipate DDoS attacks with industrial IoT devices [15]. These diverse approaches have collectively contributed to the advancement of cyber-security in IoMT networks. Allouzi and Khan [16] defined an attack vector for IoMT networks, employing a mix of a common vulnerability scoring system (CVSS) and Markov chain analysis to compute the probability distribution of security threats. Aljumaie et al. [17] conducted a comprehensive analysis of modern IoMT security approaches, emphasizing privacy, confidentiality, authentication, and detection methods. Si-Ahmed, Al-Garadi, and Boustia [18] focused on machine-learning-based intrusion-detection systems tailored to IoMT by analyzing the architecture layers and evaluating solutions. Kumar, Gupta, and Tripathi [19] proposed an ensemble-learning and fog-cloud-architecture-driven framework, demonstrating high accuracy and detection rates. Binbusayyis et al. [20] investigated machine-learning approaches for intrusion detection in IoMT networks, using the Bot-IoT benchmark dataset. Hernandez-Jaimes et al. [21] provided a comprehensive review of IoMT security, addressing ethical and legal considerations. Faruqui et al. [22] introduced SafetyMed, a CNN-LSTM-based intrusion-detection system for IoMT devices, showcasing robust defense. Salem et al. [23] proposed a framework to mitigate man-in-the-middle attacks in IoMT, ensuring the correct operation of health-monitoring systems. Collectively, these studies contributed to advancing security in IoMT networks, covering threat detection, vulnerability analysis, and security solutions.

This study included the following topics. In Section 2, some machine learning algorithms are explained, and Section 3 describes the work of ERF-ABE. In Section 4, the K-means algorithm is detailed, and in Section 5, the ERF-KMC algorithm is explained. In Section 6, the overall simulation is described, and in Section 7, the results are shown and discussed. In Section 8, our conclusion is presented.

2. Machine Learning

Machine-learning algorithms are a set of mathematical ideas and techniques that let computers learn from their experiences and grow without the need for explicit programming. These algorithms serve as the cornerstone of artificial intelligence (AI) and are used in a wide range of activities, such as in IoMT network security [24]. For example, the following are machine-learning algorithms:

- Logistic regression: a statistical method for assessing the associations between several predictor variables (either continuous or categorical) and a binary result.
- Decision tree (DT): a supervised learning approach for regression and classification problems. The internal nodes, leaf nodes, branches, and root nodes make up its hierarchical tree structure.
- Naïve Bayes (NB): a classification method with an independent assumption among predictors that is based on Bayes' theorem.
- Stochastic gradient descent (SGD): an iterative technique for maximizing an objective function with appropriate smoothness qualities.
- K-nearest neighbors (KNN): a non-parametric supervised-learning classifier that employs closeness to classify or anticipate how a single datum point will be grouped.
- Random forest (RF): a popular machine learning approach that aggregates the output of several decision trees to produce a single outcome.
- AdaBoost and CatBoost: boosting algorithms that create the final result by combining many simple models.

3. ERF-ABE for Detecting Attack Nodes

In our previous study [25], we focused on the network security needs for detecting DDoS and delay attacks, as well as enhancing the efficiency of network integrity and performance. The study employed various machine-learning algorithms, including logistic regression, DT, NB, SGD, KNN, and RF. These algorithms utilized data from attack simulations [25].

The setting parameters were a seed value of 42 and a learning rate of 0.009 at 100 iterations. The data were divided into 80% training and 20% testing sets. The performance of these models was evaluated based on accuracy, sensitivity, and execution time. Accuracy measured the model's ability to recognize the target class, sensitivity assessed its ability to identify positive data points, and the execution time was calculated in the testing phase. The research showed that despite the long execution time of random forest, as compared to other methods shown in Table 1, the random forest approach outperformed the others in terms of accuracy and sensitivity, as shown in Tables 2 and 3, respectively. To address the time issue, an ERF-ABE was proposed in [25]. This improved method utilized a principal component analysis (PCA) and parameter adjustments to maintain accuracy while speeding up performance. By fine-tuning settings and implementing PCA, the execution time was reduced by 3.116 s, as compared to the default random forest classifier while retaining accuracy and sensitivity at 99%, as shown in Table 4. The study emphasized the significance of parameter selection and dimensionality-reduction techniques, such as PCA, for enhancing the effectiveness of machine-learning models for detecting network-node attacks.

Table 1. Comparison of six algorithms in terms of execution time.

	Logistic Regression	Decision Tree	Naïve Bayes	Stochastic Gradient Descent	K-Nearest Neighbors	Random Forest
Execution time (s)	0.047	0.071	0.088	0.042	28.255	3.795

Table 2. Comparison of six algorithms in terms of accuracy.

	Logistic Regression	Decision Tree	Naïve Bayes	Stochastic Gradient Descent	K-Nearest Neighbors	Random Forest
Accuracy (%)	89.73	98.41	79.98	88.66	99.05	99.13

Table 3. Comparison of six algorithms in terms of sensitivity.

	Logistic Regression	Decision Tree	Naïve Bayes	Stochastic Gradient Descent	K-Nearest Neighbors	Random Forest
Sensitivity (%)	99.70	94.24	98.95	46.41	92.15	99.77

Table 4. Comparison between ERF-ABE and random forest.

	Random Forest	ERF-ABE
Execution time (s)	3.795	0.679
Accuracy (%)	99.126	99.053
Sensitivity (%)	99.772	99.701

The behavior of a random forest model can be significantly influenced by four important hyper-parameters, namely n_estimators, max_features, max_samples, and max_depth. The number of decision trees in the ensemble is determined by the parameter n_estimators. The default value of 100 produces a diverse and potent ensemble but can also result in lengthy training times and longer computational execution times, particularly for large datasets. ERF-ABE used a more restrained ensemble of the 20 decision trees. This tactical move greatly lessened the computational load during the training phase, which resulted in a faster procedure that would be especially helpful with tight deadlines. The number of features considered at each split was controlled by the parameter max_features. The square-root of the total number of features was set by max_features by default. This extensive

search could increase the computational load during the tree-building process and lengthen the execution time when the feature space was large.

ERF-ABE had been methodically set to a sensible value of 10, which effectively reduced the training times without sacrificing the detection power by quickly navigating the feature space and reducing the feature selection at each split. This more efficient feature exploration increased the speed of tree formation and the entire training process. The number of samples in the dataset was determined by the parameter max_samples. The default setting of the parameter max_samples was to use the entire dataset. Using the entire dataset could increase the computational difficulties and increase the execution time. Therefore, the max_samples value in ERF-ABE was calibrated to use a representative subset of the dataset at 80% (max_samples = 0.8). The effectiveness of the training phase was further increased by this strategy, which ensured that each decision tree would be trained on a more manageable subset of the data. The depth in each decision tree was controlled by the parameter max_depth. Trees would grow until the specific termination requirements were satisfied by default. Unrestrained growth could lead to the development of deep and intricate trees, which could result in lengthy calculation execution times, slow training, and slow data detection. The max_depth was set to 25 in ERF-ABE. The deliberate pruning increased the speed of training and the detection process because shallower trees were typically easier to navigate.

The ERF-ABE form of the random forest algorithm had been painstakingly honed in [25] to produce quick model training and short detection execution times. ERF-ABE is a valuable tool in applications where computational effectiveness and quick execution are crucial because it frequently fulfills time-sensitive requirements. Figure 1 shows one of the decision trees for the ERF-ABE model.

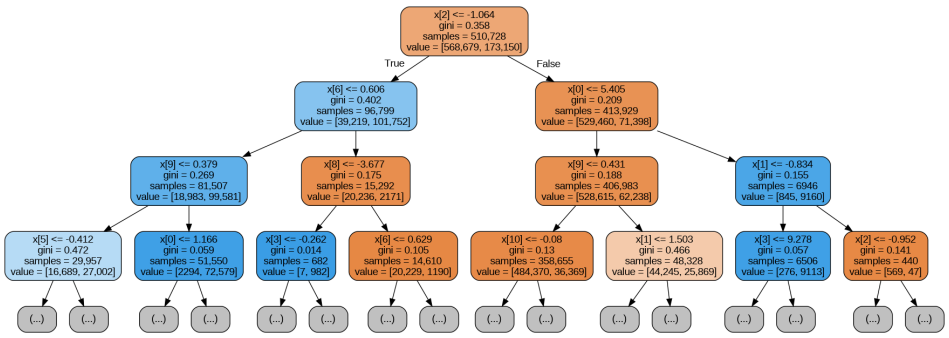

Figure 1. The decision tree of ERF−ABE.

In addition to the ERF-ABE evaluation in [25], this study validated the advantages of ERF-ABE in the field of IoMT network security by comparing ERF-ABE with AdaBoost and CatBoost, which are boosting algorithms that have proven their ability in the field of network security. Accuracy and execution time were measured as part of the evaluation process, as shown in Table 5. After the performance measures had been examined, it was found that ERF-ABE outperformed the two algorithms with an accuracy rate of 99.053%. By comparison, the accuracy rates of AdaBoost and CatBoost were marginally lower, at 92.654% and 98.855%, respectively. The execution time of ERF-ABE was 0.679 s, which was superior to AdaBoost and CatBoost at 1.533 and 0.851 s, respectively. This indicated that ERF-ABE was better than AdaBoost and CatBoost in terms of security. ERF-ABE was adopted in this study for detecting attack nodes.

Table 5. Comparison between ERF-ABE, AdaBoost, and CatBoost.

	ERF-ABE	**AdaBoost**	**CatBoost**
Accuracy (%)	99.053	92.654	98.855
Execution time (s)	0.679	1.533	0.851

ERF-ABE showed an advantage in terms of execution time with a remarkably low processing time of 0.679 s. Even though AdaBoost and CatBoost had greater execution times, they nevertheless performed competitively at 1.533 s and 0.851 s, respectively. These results show that, in this particular analysis, ERF-ABE stood out as a good option because of its remarkable accuracy and quick execution.

To further evaluate its performance, the ERF-ABE algorithm was evaluated by training it on another dataset, BoTNeTIoT-L01. The BoTNeTIoT-L01 dataset encompassed data from IoT devices involved in the detection of DDoS. The dataset was created based on four extracted features: packet count, jitter, size of outbound packets, and the combined size of outbound and inbound packets. Statistical measures such as mean, variance, count, magnitude, radius, covariance, and correlation coefficient were computed for each feature, generating a total of 23 features. The dataset contained 2,426,574 rows and 25 columns. The target column was called label and contains two categories: 1 (DDoS) and 0 (normal). The dataset was pre-processed, and then PCA was applied to reduce the number of features to 12 before they were divided into two sets: 80% for training and 20% for testing. ERF-ABE and the previously mentioned algorithms were trained on training data in order to evaluate all of the algorithms on the testing data. Table 6 shows the comparison of the algorithms according to accuracy, sensitivity, execution time, and precision.

Table 6. Comparison of the algorithms based on the BoTNeTIoT-L01 dataset.

	Logistic Regression	Decision Tree	Naïve Bayes	Stochastic Gradient Descent	K-Nearest Neighbors	Random Forest	AdaBoost	CatBoost	ERF-ABE
Accuracy (%)	91.357	99.178	88.57	93.142	99.568	99.894	95.891	99.304	99.845
Execution time (s)	0.105	0.108	0.12	0.09	25.78	2.458	1.547	0.754	0.551
Sensitivity (%)	99.265	95.784	97.87	66.48	93.54	99.359	97.152	99.125	99.282
Precision (%)	99.325	69.45	98.154	88.51	98.48	99.651	98.654	99.452	99.564

4. K-Means Clustering Algorithm for Classifying Attack Type

The K-means algorithm is a widely used, unsupervised, machine-learning clustering algorithm that is known for its effectiveness and simplicity. It divides a dataset into K-clusters, with "K" being a user-defined or random value. The process involves iteratively assigning data points to the closest centroid, then initializing and updating the centroids until convergence has occurred or a specified number of iterations had been reached. The algorithm's efficiency, especially for large datasets, is a notable advantage. K-means is suitable for situations with roughly spherical clusters of comparable sizes, and it is favored in exploratory data analysis for its simplicity.

K-means can identify and discover hidden patterns in a dataset. The algorithm recognizes the structure and organization of data by default because of its centroid update mechanism and iterative assignment. K-means efficiently identifies inherent patterns by clustering data points according to how close they are to the centroids. This allows for the visualization of differences and similarities in the dataset. K-means is a useful tool for exploratory data analysis because of its capacity for pattern recognition. It may be used to group together similar data points and enable a more thorough comprehension of the underlying structure of the available data.

5. ERF-KMC Algorithm

Designing an algorithm to achieve high security in IoMT networks is a major challenge. The ERF-KMC algorithm combined the ERF-ABE algorithm to detect attacking nodes and the K-means algorithm to classify the attacking nodes as MITM or DDoS attacks. Moreover, the ERF-KMC algorithm effectively provided high security in IoMT networks by detecting, classifying, and preventing node attacks. Figure 2 shows the workflow of the ERF-KMC algorithm. Firstly, the server received messages from the nodes and performed specific operations, as follows:

1. The server collects the message's crucial features, like duration, FlowBytesSent, FlowSentRate, FlowBytesReceived, FlowReceivedRate, etc., with a total of 27 features.
2. PCA is performed to reduce 27 features to 12.
3. The ERF-ABE algorithm determines whether the message represents an attack.
4. If the ERF-ABE detection indicates that the message is normal, the message is permitted into the server processes.
5. If ERF-ABE detects an attack, the message information and the node's IP are stored in the ERF-KMC memory.
6. K-means then identifies the attack type, distinguishing between DDoS and MITM attacks.
7. Based on the attack type, the ERF-KMC algorithm prevents messages by blocking the node's IP, in the case of MITM attacks, or limits the broadcasting for DDoS attacks by sending the node's IP to the cloud.
8. The ERF-KMC algorithm prevents future messages from attack nodes by comparing the node's IP to the IP addresses of attacking nodes stored by the ERF-KMC algorithm, thereby effectively preventing any more intrusion attempts.

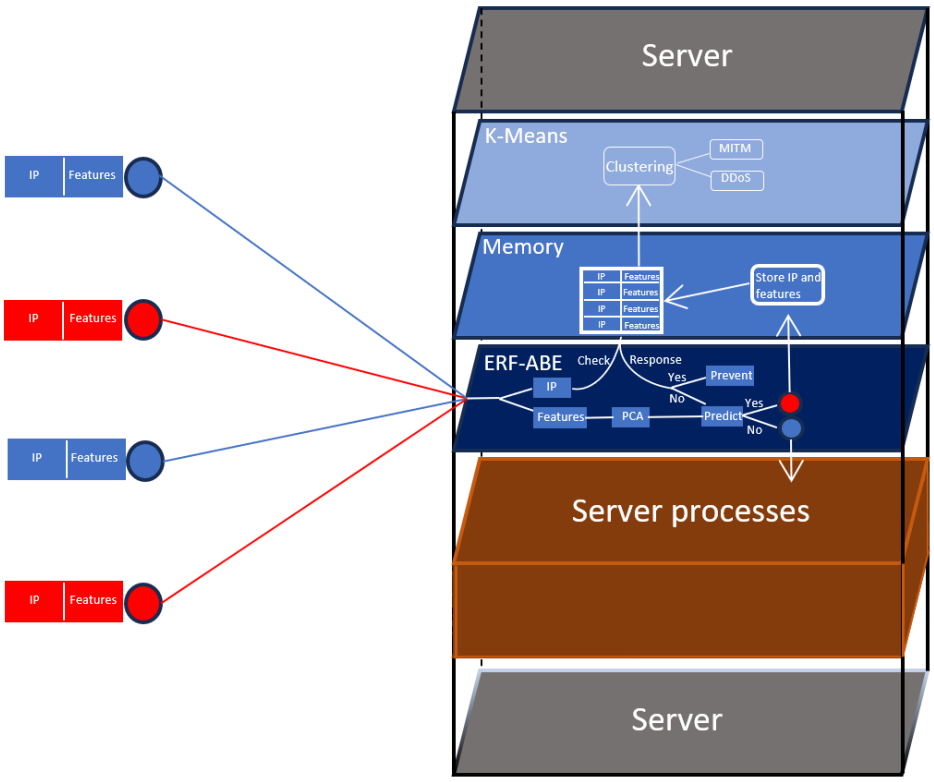

Figure 2. The workflow of the ERF−KMC algorithm.

The ERF-ABE algorithm was trained and evaluated for use in ERF-KMC and was able to accurately detect the attack. ERF-ABE constituted a key component of the ERF-KMC algorithm because attack detection is the basis for achieving IoMT network security. K-means clustering assumed a central role in the functioning of the ERF-KMC algorithm, especially considering its specialized training on the same dataset employed by ERF-ABE. During this training, K-means exclusively processed samples associated with confirmed attacks, sharpening its capacity to differentiate between the intricate patterns characterizing DDoS and MITM attacks. Focused on this selective dataset, K-means refined its ability to group incoming messages into distinct clusters based on the learned attack patterns. Its prowess in identifying similarities and anomalies in the data effectively aided ERF-KMC by categorizing network traffic into clusters that indicated malicious behavior linked to either a DDoS or MITM attack. The resulting clustering information became critical input for the ERF-KMC algorithm, enabling it to make well-informed decisions in order

to prevent potentially harmful messages from compromising the server. The synergistic interplay of K-means clustering and ERF-ABE within the ERF-KMC algorithm significantly heightened the accuracy and the efficiency of attack detection and prevention in IoMT networks, thereby enhancing the overall security of the system.

Detecting the attack using ERF-ABE and then determining the type of attack using the K-means algorithm constituted the working approach of the ERF-KMC algorithm. The ERF-KMC algorithm was able to prevent attacks through the combination of ERF-ABE for attack detection and K-means for identifying the attack type.

6. Simulation Approach

A thorough and complex technique was developed to improve the security and functionality of IoMT. Three key network performance characteristics comprised the complex interplay that supported this methodology, which were throughput, PLR, and E2ED. Each parameter became of paramount significance when assessing the resilience and the effectiveness of IoMT in the presence of disruptive factors. The experimental basis of this methodology was based on the introduction of attack nodes into the IoMT network, where a dynamic and changing environment was reproduced that could mimic real-world conditions, enabling a thorough assessment of how attacks would affect the network performance. This research expanded its scope to address the vulnerabilities introduced by DDoS and MITM attacks in IoMT. The next sections of this paper provide significant details concerning how traffic was generated, how the experiments were designed, and how the machine-learning models were combined. The ERF-KMC algorithm provided a better understanding of the crucial role that machine-learning techniques play in protecting IoMT networks from external threats. The real-time simulation using Python version 3.11.4 enabled us to evaluate the practical effects of the ERF-KMC algorithm on throughput, PLR, and E2ED.

Nodes, routers, servers, cloud components, and IoMT components were used in the simulated IoMT network. The study investigated three different scenarios with 25, 50, and 100 nodes. Each scenario was a mixture of normal and attack nodes in order to evaluate the ERF-KMC algorithm.

6.1. Simulation Components

In our previous study [26], we used OMNET++ 4.3 as the simulation software and the Neta 1.1 framework, in addition to the INET 3.3 framework. This simulation was carried out to evaluate the effectiveness of DDoS and delay attacks on WBSNs. The system received 100 nodes, and those nodes shared the message among themselves during a 300-second simulation. Then, two, four, six, and then eight attack nodes were introduced to the system to investigate the impact of attacker tactics. As a strategy, attackers either dropped or delayed the data that were being transmitted. A transmission control protocol (TCP) was selected to broadcast the packets using several sources (Nodes [1...99]), where only Node [0] could receive data. The results in [26] confirmed that increasing the number of attack nodes led to an increase in throughput, PLR, and E2ED, and this harmed the operation of WBSNs. It was a challenging, yet crucial, undertaking to create a simulation for an IoMT network that could simulate the dynamics of actual healthcare systems in their complexity and variety while paying close attention to every detail.

The foundation for frictionless communication between the nodes and the server involved the use of a TCP, which is a robust and widely used protocol. Interconnected devices on a network can transmit information reliably and systematically via TCP. The experimental network was organized around nodes that were divided into two primary types: attack nodes and normal nodes. These nodes were thoughtfully dispersed across five different geographic regions to reflect the regional diversity frequently found in healthcare environments. Each region had a specific base station that acted as a key communication hub. The data and messages produced by the nodes within each region were collected by these base stations. The geographical distribution of the base stations reproduced the decentralization observed

in many IoMT networks, thereby ensuring the effective collection of information and transmission while catering to the unique needs of every region. Two central routers received the data gathered by the base stations. These routers were essential for receiving and transferring data, as well as for maintaining the network's redundancy and dependability.

In IoMT networks, this redundancy was essential to provide ongoing connectivity despite network outages and attacks. In the proposed network simulation, as shown in Figure 3, the data were transmitted to a cloud server by the routers once they had received it from the base stations. The cloud replicated the cloud-based technology that was frequently employed in IoMT networks. The real-time access and analysis of the gathered data were made feasible by the centralized data storage and management. The data were then sent from the cloud server to a separate analytical server. The incoming data could be processed in the server using the ERF-KMC algorithm to detect attack nodes in order to assist in healthcare decision making, diagnosis, and patient care.

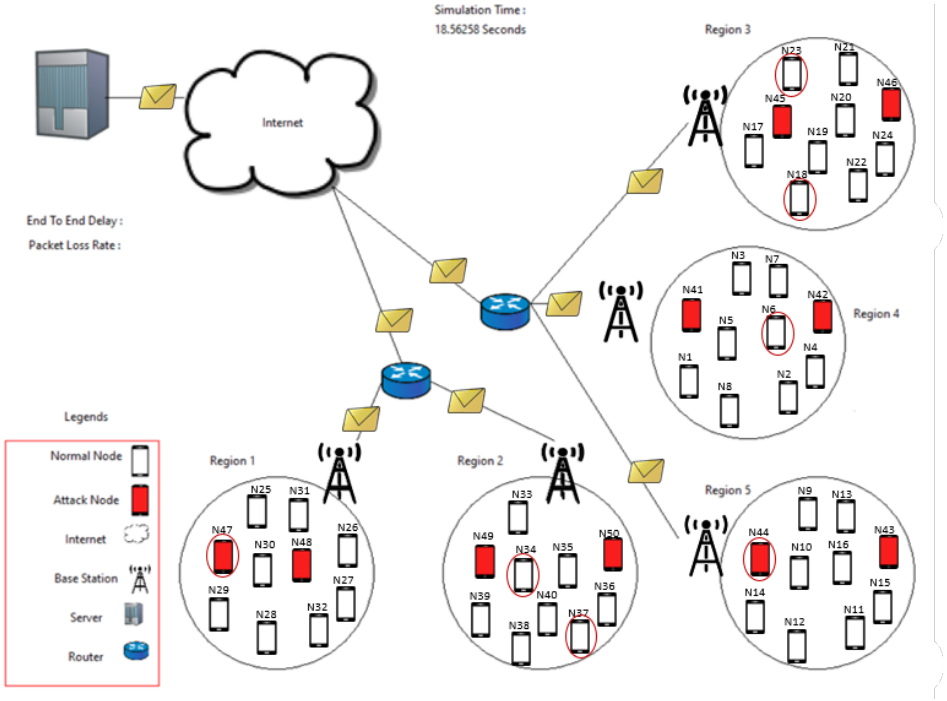

Figure 3. Network simulation.

This thorough and accurate simulation offered a useful platform for researchers, practitioners, and developers to evaluate and improve the performance, security, and scalability of IoMT networks. In addition, it improved the dependability and safety of healthcare services while assisting stakeholders in understanding and addressing the opportunities and difficulties prevalent in real-world IoMT situations. In Table 7, the network parameters are presented.

Table 7. Network parameters.

Parameter	Value
Network simulator	Python version 3.11.4
Network area	2000 m × 2000 m
Network components	Five base stations, two routers, cloud, server
Simulation time	300 s
Number of nodes	25, 50, 100 nodes
Number of attacks	20% of total nodes
Packet size	512 B
Node mobility	Random
Transmission protocol	TCP protocol
Attack types	DDoS and MITM
Machine learning	ERF-KMC

6.2. Simulation Experiments

A comprehensive analysis of an IoMT network using ERF-KMC in its server was conducted. The analysis included three scenarios based on the number of nodes in the network: 25, 50, and 100. A balanced distributed approach was followed to simulate a real IoMT network environment. In particular, 20% of the nodes were designated as attack nodes while the remaining 80% performed as normal nodes. These nodes were evenly dispersed across several regions to maintain geographic variety. Each region consisted of a mixture of normal and attack nodes to imitate the diverse character of real-world IoMT networks. Normal nodes were simulated to send messages by standard patterns that reflected typical data-transfer properties in medical environments.

The attack nodes were more assertive and persistent, simulating the increased activity signaling prospective security risks. The network became more complicated due to this dichotomy in node behavior, enabling the evaluation of how these various behaviors could affect network performance. The server was at the center of the scenario and acted as the main hub for message reception and analysis. The server received messages from both normal and attack nodes, without identifying the source. Every message in the simulation contained all the necessary data, including metadata and its IP address. A deeper comprehension of the communication origins and content was made possible by this extensive data in order to assess security and performance indicators. The server carried out specific operations once the message had arrived. The operations involved the calculation of critical features, including (duration, FlowBytesSent, FlowSentRate, FlowBytesReceived, FlowReceivedRate, etc.).

These features were fundamental to identifying the intent of the messages and whether they were sent by attack or normal nodes. The server had a predetermined amount of storage, and incoming messages were regarded as lost if this limit was reached, thereby simulating the typical limitations on servers in the real world. This restriction emphasized how important effective data management is in IoMT networks. The network performance metrics were carefully tracked after each simulation, including E2ED, PLR, and throughput.

The distinctive features of the server, augmented with the ERF-KMC algorithm, were related to its unique strategy that enhanced network security and streamlined the execution time. Figure 4 shows the work of the ERF-KMC algorithm by flowchart diagram. The server configuration combined the ERF-ABE and K-means algorithms to strengthen the defenses against attack nodes. Firstly, the server received the messages and then extracted 27 features. After that, PCA was performed to reduce the 27 features to 12. Then, the ERF-ABE algorithm employed its accurate detection capability to determine whether the message represented an attack. The message was only permitted into the server for processing if ERF-ABE's detection matched the message's actual content.

If ERF-ABE detected an attack, the message information and the node's IP were stored in the ERF-KMC memory. K-means then identified the attack type, distinguishing between DDoS and MITM. Then, the ERF-KMC algorithm would prevent future messages by blocking the node's IP, in case of an MITM attack, or limiting broadcasting, in the case of a DDoS attack, by sending the node's IP to the cloud.

The ERF-KMC algorithm could prevent messages from MITM attack nodes from entering the server by comparing the IP address against those that had already been stored in memory by the ERF-KMC algorithm.

DDoS attack nodes send a high volume of messages over a short period of time, overwhelming servers, which could cause errors in ERF-KMC's attack detection. By sending the IP address of the DDoS attack nodes to the cloud, it prevented messages from entering the server, and one could compare the IP address coming from ERF-KMC with messages coming from the DDoS attack nodes. Therefore, the ERF-KMC algorithm could effectively prevent additional intrusion attempts.

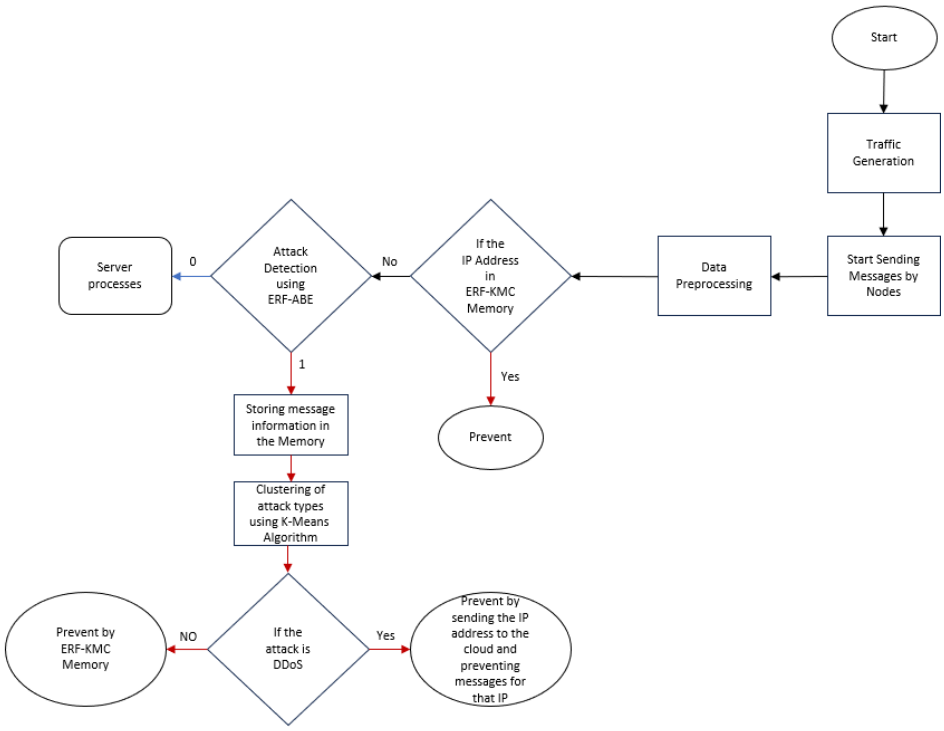

Figure 4. Flowchart of ERF−KMC.

7. Results and Discussion

In the IoMT network, the evaluation of the network performance was crucial for ensuring the efficient and reliable transfer of data. This study focused on the end-to-end delay, packet loss rate, and throughput, and the number of attacking nodes greatly affected these metrics, which had been confirmed in [26]. Therefore, these metrics were important for evaluating the ERF-KMC algorithm. The results in this section were based on comparisons of the ERF-KMC algorithm with other algorithms, such as AdaBoost, CatBoost, and random forest algorithms, which are commonly used in network security.

7.1. Performance Evaluation and Security Analysis of Wireless Body

The evaluation of network performance is crucial in the dynamic environment of computer networks. Both end-users and network administrators must have access to network performance indicators because they offer important information about the effectiveness and dependability of the data transfer. Three important network performance metrics stand out among the rest: throughput, PLR, and E2ED.

- Throughput: The amount of data that can be sent from one node to another within a given period. Data packets per time slot, data packets per second, and bits per second are the most common units of measurement. Several variables have an impact on the throughput of a network, such as a network's node count, traffic load, and communication-channel bandwidth.
- Packet loss rate (PLR): A measurement of the proportion of data packets that are missed or lost during transmission. It is frequently stated as a percentage. The effectiveness of the communication channel, the volume of traffic on the network, and the routing algorithm all have an impact on the PLR.
- End-to-end delay (E2ED): The total amount of time a packet takes to travel from the time it is created by a source node to the time it is received by a destination node. Usually, it is expressed in seconds (s). The distance between the source and destination nodes, the number of hops in the path, and the amount of network traffic all have an impact on the E2ED.

7.2. Comparative Analysis of End-to-End Delay

The investigation into end-to-end delay within diverse network configurations yielded significant insights, particularly when considering the results presented in Figure 5. The ERF-KMC algorithm consistently exhibited superior performance in all evaluations, with remarkably low delays of 0.753, 0.753, and 0.774 s for 25, 50, and 100 nodes, respectively. This pattern underscored the algorithm's robustness, as it could maintain minimal delays regardless of the network scale.

Conversely, AdaBoost revealed a notable increase in the end-to-end delay, as compared to the ERF-KMC algorithm, with values of 2.041, 1.624, and 1.459 s for 25, 50, and 100 nodes, respectively. This trend suggested the potential sensitivity of AdaBoost to network size variations.

In contrast, the rest of the algorithms were stable despite network size variations, but ERF-KMC's results were superior when compared to CatBoost, which had delays of 1.299, 1.129, and 1.178 s; random forest with delays of 1.34, 1.182, and 1.19; and K-nearest neighbors with delays of 0.944, 1.154, and 1.354 s, for 25, 50, and 100 nodes, respectively.

The proposed model continuously outperformed the other algorithms across all the simulations of varied node densities. ERF-KMC's ability to detect and prevent the attack nodes resulted in faster and more efficient data transmission and processing.

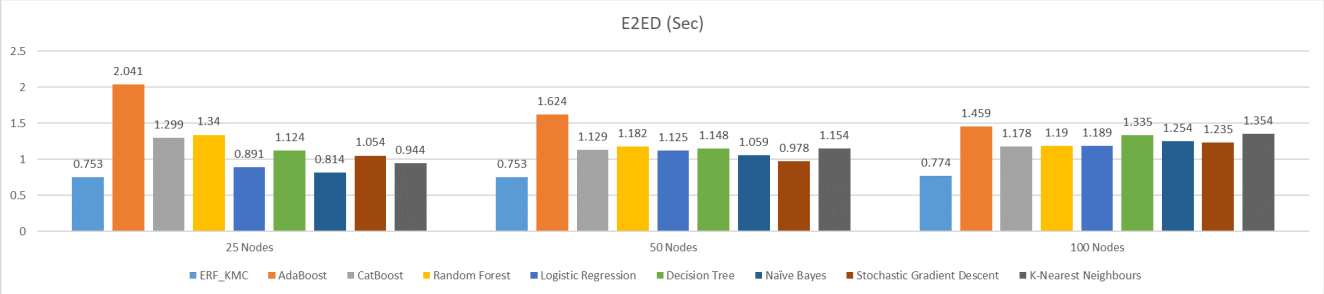

Figure 5. End-to-end delay in scenarios with 25, 50, and 100 nodes.

7.3. Comparative Analysis of Packet Loss Rate

The investigation into the packet loss rate across varying network scales, as shown by the results presented in Figure 6, offered insightful perspectives on the performance of the machine-learning algorithms. The ERF-KMC algorithm emerged as particularly robust, exhibiting consistently low packet loss rates of 0.046, 0.023, and 0.138 for 25, 50, and 100 nodes, respectively. This indicated a robust ability to maintain data integrity across diverse network configurations. Conversely, AdaBoost demonstrated a significant variability in packet loss rates, with values of 0.394, 0.135, and 0.151 for 25, 50, and 100 nodes, respectively, suggesting its potential sensitivity to the network scale.

The ERF-KMC algorithm outperformed the other algorithms, as CatBoost had a PLR of 0.527, 0.459, and 0.504 for 25, 50, and 100 nodes, respectively, while random forest had a PLR of 0.526, 0.509, and 0.515 for 25, 50, and 100 nodes, respectively. K-nearest neighbors had a PLR of 0.389, 0.548, and 0.356 for 25, 50, and 100 nodes, respectively.

The consistent reduction in PLRs across all the simulations with various node densities was based on the significant enhancements introduced by the ERF-KMC algorithm for the detection and prevention of attack node messaging.

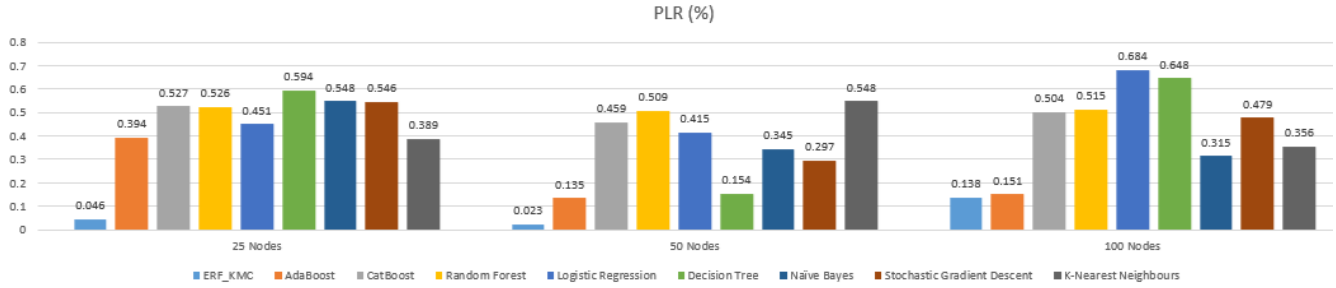

Figure 6. Packet loss rate in the case of 25, 50, and 100 nodes.

7.4. Comparative Analysis of Throughput

The examination of throughput, as shown in Figure 7, offered valuable insights into the efficacy of machine-learning algorithms across diverse network scales. The ERF-KMC algorithm demonstrated consistent and competitive throughput values, with rates of 4.536, 9.041, and 15.96 for 25, 50, and 100 nodes, respectively.

AdaBoost had throughput values of 4.665, 9.562, and 16.054 for 25, 50, and 100 nodes, respectively. The CatBoost had throughput values of 4.835, 9.832, and 16.152; the random forest algorithm had throughput values of 4.902, 10.152, and 16.088; and K-nearest neighbors had throughput values of 4.761, 9.945, and 16.164, for 25, 50, and 100 nodes, respectively.

These results demonstrated that the ERF-KMC algorithm outperformed the other algorithms, even at different node densities. The ERF-KMC algorithm improved the network throughput due to the accuracy and efficiency of its attack detection and prevention processes.

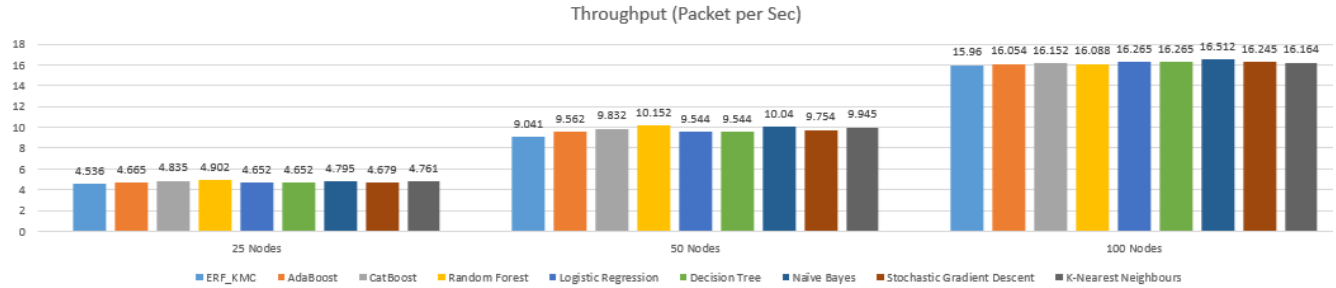

Figure 7. Throughput in case of 25, 50, and 100 nodes.

7.5. The Significance of Systematic Data Management and Comprehensive Documentation Practices

Saving information is necessary for robust analysis and system evaluation. The ERF-KMC algorithm carefully documented crucial information in an Excel spreadsheet for further study and reference. This information included a full list of message attributes. These attributes contained the time of the message arrival to the server (Time), the identification of the node that sent the message (Node_ID), the IP address (IP_Address), and the features (FlowBytesSent, FlowSentRate, etc.). In addition, the ERF-ABE algorithm's detection (flag) indicated whether the message was considered an attack (flag = 1) or normal (flag = 0). When the flag was equal to one, the server added an extra entry that specified the type of attack, DDoS or MITM, according to the K-means algorithm.

This detailed record permitted a thorough post-analysis and provided a structured information-keeping system. Security administrators could perform more research and gain important insights by reviewing the stored data in the Excel spreadsheet, which ensured the continued improvement of the ERF-KMC algorithm and could advance network security in IoMT situations. Additionally, this data-storage strategy was a useful resource for upcoming research projects that require a deeper understanding of online threats in order to enhance IoMT networks' robustness to dynamic security threats. Table 8 shows the Excel spreadsheet and the information it recorded.

Table 8. Excel spreadsheet information.

Time	Node_ID	IP_Address	Flow-BytesSent	Flow-SentRate	...	Flag	Type
2.95939	90	192.168.1.92	9132	356.45742	...	1	MITM
2.99429	93	192.168.1.95	1579	85.150962	...	1	MITM
1.47790	14	192.168.1.16	110	2.4418721	...	0	
3.02530	81	192.168.1.83	1153	260.62075	...	1	MITM
3.03020	82	192.168.1.84	166736	1382.3347	...	1	DDoS
3.03320	97	192.168.1.99	52721	439.27005	...	1	DDoS
6.98640	60	192.168.1.62	342	5.7868757	...	0	

7.6. Discussion

This study investigated the performance of machine-learning algorithms for optimizing network efficiency and security within IoMT networks by examining E2ED, PLR, and throughput, across diverse network scales. The analysis revealed that the ERF-KMC algorithm consistently outperformed the other algorithms with shorter E2E delays and lower PLRs while demonstrating competitive throughput values. Notably, ERF-KMC's ability to detect and prevent attack nodes contributed to its superior performance. The meticulous documentation of message attributes in an Excel spreadsheet supported the post-analysis and system evaluation, providing valuable insights for security administrators. This structured information-keeping system is crucial for advancing the IoMT network security and contributes to a deeper understanding of evolving online threats. The presented findings underscored the efficacy of the ERF-KMC algorithm for enhancing both the network performance and security in IoMT environments, and it is a practical resource for ongoing research and improving proactive security measures.

8. Conclusions

In conclusion, the ERF-KMC algorithm represents a groundbreaking solution to the escalating security challenges in IoMT networks, which have ushered in a transformative era in healthcare technology. The integration of WBSNs and the IoMT has enabled real-time monitoring and communication among medical devices. By harnessing the clustering capabilities of K-means and the detection accuracy of ERF-ABE, the ERF-KMC algorithm uniquely identified and categorized attack nodes to prevent MITM and DDoS attacks. Future research will explore expanding its applicability for a broader spectrum of security threats. Additionally, it is recommended to augment the ERF-KMC algorithm with deep-learning techniques, so it can adapt to complex and dynamic attack patterns, further fortifying its robustness. This research significantly enhanced the IoMT network security, ensuring the accurate and swift transmission of medical data for real-time patient monitoring. The model's proficiency in detecting and preventing attack nodes averted unauthorized access attempts, safeguarding patient privacy and maintaining the healthcare data integrity. The simulation results confirmed ERF-KMC's effectiveness when compared with other algorithms. The ERF-KMC algorithm marks a substantial stride toward the enhanced development of interconnected healthcare technologies and promises heightened effectiveness in security, precision, and patient safety.

Author Contributions: Conceptualization, A.A.J.A.-A.; methodology, A.A.J.A.-A.; software, A.A.J.A.-A.; validation, A.A.J.A.-A., M.B.M. and A.F.; formal analysis, A.A.J.A.-A.; investigation, A.A.J.A.-A.; resources, A.A.J.A.-A.; data curation, A.A.J.A.-A.; writing—original draft preparation, A.A.J.A.-A.; writing—review and editing, M.B.M. and A.F.; visualization, A.A.J.A.-A.; supervision, M.B.M. and A.F.; project administration, A.A.J.A.-A., M.B.M. and A.F.; funding acquisition, A.A.J.A.-A. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The BoTNeTIoT-L01 dataset can be found at kaggle (accessed on 10 December 2023).

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

ERF-KMC	Enhanced Random Forest Classifier with K-Means Clustering
ERF-ABE	Enhanced Random Forest classifier for Achieving the Best Execution Time
WBSNs	Wireless Body Sensor Networks
IoMT	Internet of Medical Things
MITM	Man-in-the-Middle
PLR	Packet Loss Rate
E2ED	End-to-End Delay

References

1. Kumar, A.; Singh, K.; Khan, T. L-RTAM: Logarithm based reliable trust assessment model for WBSNs. *J. Discret. Math. Sci. Cryptogr.* **2021**, *24*, 1701–1716. [CrossRef]
2. Papaioannou, M.; Karageorgou, M.; Mantas, G.; Sucasas, V.; Essop, I.; Rodriguez, J.; Lymberopoulos, D. A survey on security threats and countermeasures in internet of medical things (IoMT). *Trans. Emerg. Telecommun. Technol.* **2022**, *33*, e4049. [CrossRef]
3. Sami, I.; Ahmad, M.B.; Asif, M.; Ullah, R. DoS/DDoS Detection for E-Healthcare in Internet of Things. *Int. J. Adv. Comput. Sci. Appl.* **2018**, *9*, 297–300.
4. Hady, A.A.; Ghubaish, A.; Salman, T.; Unal, D.; Jain, R. Intrusion detection system for healthcare systems using medical and network data: A comparison study. *IEEE Access* **2020**, *8*, 106576–106584. [CrossRef]
5. Iwendi, C.; Anajemba, J.H.; Biamba, C.; Ngabo, D. Security of things intrusion detection system for smart healthcare. *Electronics* **2021**, *10*, 1375. [CrossRef]
6. Kamble, P.; Gawade, A. Automation in Healthcare Using IoT and Cryptographic Encryption against DOS and MIM Attacks. In *Advanced Computing Technologies and Applications, Proceedings of the 2nd International Conference on Advanced Computing Technologies and Applications—ICACTA, Mumbai, India, 28–29 February 2020*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 97–105. [CrossRef]
7. Hussain, F.; Abbas, S.G.; Shah, G.A.; Pires, I.M.; Fayyaz, U.U.; Shahzad, F.; Zdravevski, E. A framework for malicious traffic detection in IoT healthcare environment. *Sensors* **2021**, *21*, 3025. [CrossRef] [PubMed]
8. Newaz, A.I.; Haque, N.I.; Sikder, A.K.; Rahman, M.A.; Uluagac, A.S. Adversarial attacks to machine-learning-based smart healthcare systems. In Proceedings of the GLOBECOM 2020—2020 IEEE Global Communications Conference, Taipei, Taiwan, 7–11 December 2020; pp. 1–6. [CrossRef]
9. Kore, A.; Patil, S. IC-MADS: IoT enabled cross layer man-in-middle attack detection system for smart healthcare application. *Wirel. Pers. Commun.* **2020**, *113*, 727–746. [CrossRef]
10. Yaser, A.L.; Mousa, H.M.; Hussein, M. Improved DDoS Detection Utilizing Deep Neural Networks and Feedforward Neural Networks as Autoencoder. *Future Internet* **2022**, *14*, 240. [CrossRef]
11. Wang, Y.; Li, Y.; Wang, X.; Zhao, X. A novel traffic generator for switch testing. In Proceedings of the 2015 International Conference on Environmental Engineering and Remote Sensing, Phuket, Thailand, 23–24 August 2015; pp. 66–69. [CrossRef]
12. Megyesi, P.; Szabo, G.; Molnár, S. User behavior based traffic emulator: A framework for generating test data for DPI tools. *Comput. Netw.* **2015**, *92*, 41–54. [CrossRef]
13. Mohamed, M.B.; Meddeb-Makhlouf, A.; Fakhfakh, A.; Kanoun, O. Secure and Reliable ML-based Disease Detection for a Medical Wireless Body Sensor Networks. *Int. J. Biol. Biomed. Eng.* **2022**, *16*, 196–206. [CrossRef]
14. Lee, S.-H.; Shiue, Y.-L.; Cheng, C.-H.; Li, Y.-H.; Huang, Y.-F. Detection and Prevention of DDoS Attacks on the IoT. *Appl. Sci.* **2022**, *12*, 12407. [CrossRef]
15. Jing, H.; Wang, J. Detection of DDoS Attack within Industrial IoT Devices Based on Clustering and Graph Structure Features. *Secur. Commun. Netw.* **2022**, *2022*, 1401683. [CrossRef]
16. Allouzi, M.A.; Khan, J.I. Identifying and modeling security threats for IoMT edge network using markov chain and common vulnerability scoring system (CVSS). *arXiv* **2021**, arXiv:2104.11580.
17. Aljumaie, G.S.; Alzeer, G.H.; Alghamdi, R.K.; Alsuwat, H.; Alsuwat, E. Modern study on internet of medical things (IOMT) security. *Int. J. Comput. Sci. Netw. Secur.* **2022**, *21*, 254–266. [CrossRef]
18. Si-Ahmed, A.; Al-Garadi, M.A.; Boustia, N. Survey of Machine Learning based intrusion detection methods for Internet of Medical Things. *Appl. Soft Comput.* **2023**, *140*, 110227. [CrossRef]
19. Kumar, P.; Gupta, G.P.; Tripathi, R. An ensemble learning and fog-cloud architecture-driven cyber-attack detection framework for IoMT networks. *Comput. Commun.* **2021**, *166*, 110–124. [CrossRef]
20. Binbusayyis, A.; Alaskar, H.; Vaiyapuri, T.; Dinesh, M. An investigation and comparison of machine-learning approaches for intrusion detection in IoMT network. *J. Supercomput.* **2022**, *78*, 17403–17422. [CrossRef]
21. Hernandez-Jaimes, M.L.; Martinez-Cruz, A.; Ramírez-Gutiérrez, K.A.; Feregrino-Uribe, C. Artificial intelligence for IoMT security: A review of intrusion detection systems, attacks, datasets and Cloud-Fog-Edge architectures. *Internet Things* **2023**, *23*, 100887. [CrossRef]
22. Faruqui, N.; Yousuf, M.A.; Whaiduzzaman, M.; Azad, A.; Alyami, S.A.; Liò, P.; Kabir, M.A.; Moni, M.A. SafetyMed: A Novel IoMT Intrusion Detection System Using CNN-LSTM Hybridization. *Electronics* **2023**, *12*, 3541. [CrossRef]

23. Salem, O.; Alsubhi, K.; Shaafi, A.; Gheryani, M.; Mehaoua, A.; Boutaba, R. Man-in-the-Middle attack mitigation in internet of medical things. *IEEE Trans. Ind. Inform.* **2021**, *18*, 2053–2062. [CrossRef]
24. Janiesch, C.; Zschech, P.; Heinrich, K. Machine learning and deep learning. *Electron. Mark.* **2021**, *31*, 685–695. [CrossRef]
25. Al-Abadi, A.A.J.; Mohamed, M.B.; Fakhfakh, A. Robust and Reliable Security Approach for IoMT: Detection of DoS and Delay Attacks through a High-Accuracy Machine Learning Model. *Int. J. Recent Innov. Trends Comput. Commun.* **2023**, *11*, 239–247. [CrossRef]
26. Al-Abadi, A.A.J.; Mohamed, M.B.; Fakhfakh, A. Impact Of Availability Attacks On Enabling IoT Based Healthcare Applications. In Proceedings of the 2023 International Wireless Communications and Mobile Computing (IWCMC), Marrakesh, Morocco, 19–23 June 2023; pp. 1666–1671. [CrossRef]