Article

Resilient Security Framework Using TNN and Blockchain for IoMT

Rayan A. Alsemmeari[1], Mohamed Yehia Dahab[2], Abdulaziz A. Alsulami[3,*], Badraddin Alturki[1] and Sultan Algarni[3]

[1] Department of Information Technology, Faculty of Computing and Information Technology, KinAbdulaziz University, Jeddah 21589, Saudi Arabia; ralsemmeari@kau.edu.sa (R.A.A.); baalturki@kau.edu.sa (B.A.)
[2] Department of Computer Science, Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah 21589, Saudi Arabia; mdahab@kau.edu.sa (M.Y.D.)
[3] Department of Information Systems, Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah 21589, Saudi Arabia; aaalsulami10@kau.edu.sa (A.A.A.); saalgarni@kau.edu.sa (S.A.)
* Correspondence: aaalsulami10@kau.edu.sa

Abstract: The growth in the Internet of Things (IoT) devices in the healthcare sector enables the new era of the Internet of Medical Things (IoMT). However, IoT devices are susceptible to diverse cybersecurity attacks and threats that lead to negative consequences. Cyberattacks can harm not just the IoMT devices being used but also human life. Currently, several security solutions are proposed to enhance the security of the IoMT, which uses machine learning (ML) and blockchain. ML can be used to develop detection and classification methods to identify cyberattacks targeting IoMT devices in the healthcare sector. In addition, blockchain technology enables a decentralized approach to the healthcare system and eliminates some disadvantages of a centralized system, such as a single point of failure. This paper proposes a resilient security framework integrating a Tri-layered Neural Network (TNN) and blockchain technology in the healthcare domain. The TNN detects anomalies in data measured by medical sensors to find fraudulent data. Therefore, cyberattacks are detected and discarded from the IoMT system before data is processed at the fog layer. In addition, a blockchain network is used in the fog layer to ensure that the data is not altered, boosting the integrity and privacy of the medical data. The experimental results show that the TNN and blockchain models produce the expected result. Furthermore, the accuracy of the TNN model reached 99.99% on the F1-score accuracy metric.

Keywords: machine learning; blockchain; IoMT; IoT; cybersecurity; security framework

1. Introduction

The rapid advancement in the technology of the IoT plays a vital role in the healthcare sectors; therefore, the term IoMT has become more prevalent [1]. In addition, the development of high-speed network systems and the growing use of portable monitors, smartphones, wearable devices, and electronic health records in healthcare aid this tremendous growth of the IoT devised in the healthcare industry [2]. Integrating IoT devices in healthcare systems increases system interconnectivity and interoperability because it allows collaboration between different isolated systems within the healthcare domain [3]. Furthermore, it pushes toward transferring to the decentralized infrastructure since the computation and services can be processed through IoMT devices. However, IoT devices are susceptible and vulnerable to various security risks and attacks; since they do not have the capability of self-protection, they can be easily compromised. Recent research found that over 90% of IoT devices transmit data insecurely, meaning that 57% are vulnerable to attacks that leak sensitive data [4]. Cyberattacks can endanger not only the operated IoMT devices but also could put human lives in danger [5]. Figure 1 depicts a general scenario in which the architecture of IoMT could be threatened by a cyberattack. As shown in the figure, a threat actor can establish a cyberattack at various points of the architecture. For example, a threat actor could inject malicious data into the sensor measurements, transmitting inaccurate data to the receiver devices. In addition, a cyberattack can be launched on IoMT devices in the fog layer. Thus, there is a need for a security mechanism to detect cyberattacks and ensure the integrity of information transmitted or stored in the fog or cloud layer [6]. Several cyberattacks can target IoMT systems, such

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 18 April 2023 doi:10.20944/preprints202304.0500.v1

as Denial of Service (DDoS) attacks, Injection attacks, Data leakage, and sensor attacks. Therefore, the security of IoMT considers a considerable challenge, and more research is needed in this field.

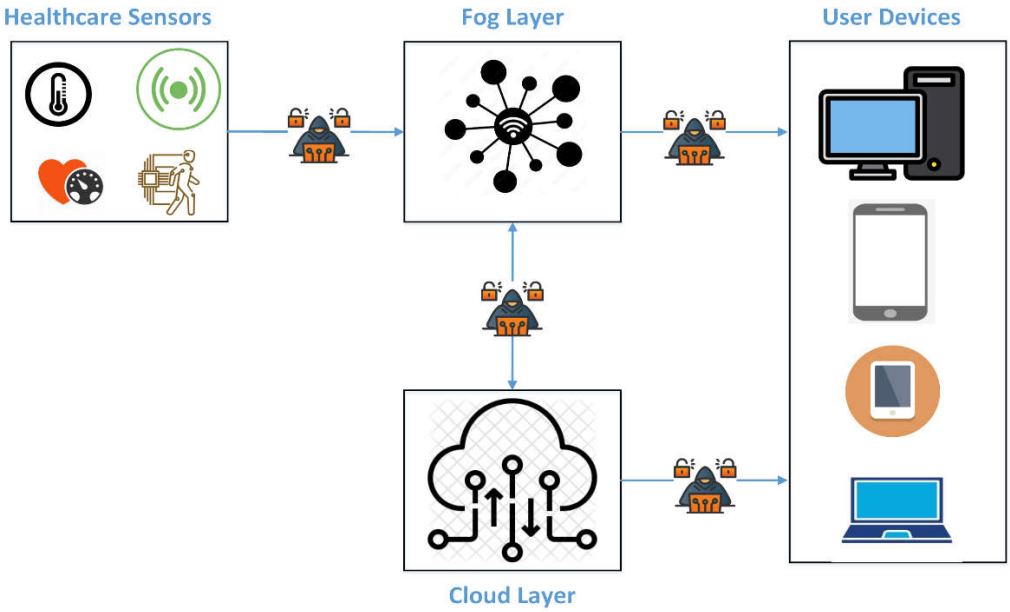

Figure 1. The Architecture of IoMT and Cyberattack.

Several studies have attempted to enhance the security of IoMT, including ML and blockchain [7]. Researchers generally used three ML taxonomies: supervised, unsupervised, and reinforcement learning. Supervised ML learning is used when the model's output is known and labeled; however, the relations of the input data are not always known. Unsupervised learning is the opposite approach of supervised learning since the model's output is unknown. In addition, during the training time, the output class of unsupervised learning is not labeled. Finally, reinforcement learning is used when the model can gain experience through its operational environment and learn based on trial and error [8]. ML can be used to develop intrusion detection systems (IDS) and intrusion classification systems (ICS) to tackle cyberattacks [9]. IDS is usually used as a binary classification to scan through data to distinguish normal data and malicious data. Despite this, ICS is used to detect multi-class data; therefore, it is useful to identify the types of cyberattacks.

Blockchain is a groundbreaking technology first introduced by Satoshi Nakamoto in 2008 [10]. It consists of a chain of blocks that holds information. Blockchain is a decentralized and distributed infostructure that enables secure and transparent transactions without the use of intermediaries [11]. It is not limited to being used only with cryptocurrencies but conceivably employs other applications such as IoT, healthcare, and energy sectors [12]. In the scope of IoT security, blockchain may be utilized to construct a decentralized devices network capable of securely exchanging information as well as transactions [13]. It becomes considerably more challenging for hackers to breach the security of a system that is built based on the blockchain's decentralized structure [14]. Furthermore, blockchain enables the construction of smart contracts that may be used to autonomously enforce security standards in an IoT network [15].

Furthermore, using a decentralized system overcomes some drawbacks of the centralized system, such as a single point of failure, since centralized infrastructure relies on the client and server approach, which means all devices need to be authorized by a server [16]. Blockchain is a peer-to-peer network system that allows communication between untrusted devices without a third party [1]. Therefore, data exchanged between the devices can be maintained and tracked without a centralized server. In order to add new data to the blockchain, users are required to solve a cryptographic puzzle (proof of work). Blockchain consists of N blocks; the first is called the Genesis block. Basically, blockchain data is stored in chronological order, and the information is held in the

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 18 April 2023 doi:10.20944/preprints202304.0500.v1

blocks and linked to each other through the chain. Participants are allowed to view the transaction; however, each user's identity is kept secret.

This research proposes a resilient security framework using a TNN model and blockchain technology. TNN is an artificial neural network with three layers of hidden nodes used for prediction purposes. TNN may be utilized to recognize and avoid cyber assaults in the area of IoMT security by scanning device activity and interactions. TNN may also be utilized to construct a predictive security model to detect and classify cyberattacks, which can assist in improving the security of the IoMT network. Furthermore, the TNN performs anomaly detection to identify malicious data collected from medical sensors. Therefore, cyberattacks are captured and dropped from the IoMT system before the data is processed in the fog layer. The blockchain model is used to distribute the data after being cleared from a cyberattack to IoMT devices in the fog layer. This ensures the security of the transactions since they cannot be altered, increasing the data's trust and integrity.

The contribution of this paper can be summarised as the following:

- Reviewing the recent state-of-the-art methods used to enhance the security of the IoMT.
- Using a TNN to perform anomaly detection procedures for identifying normal data (true data) and anomalous data (cyberattacks) collected from medical sensors.
- Using a blockchain-based scheme for non-financial applications to simulate blockchain activity in fog nodes of the IoMT to enhance the data's integrity and privacy.
- Proposing a security framework for IoMT that combines the power of TNN and blockchain, the TNN is utilized for anomaly detection to capture data injected with a cyberattack. Blockchain maintains the integrity and privacy of the data to ensure that stored and transmitted data can not be altered.

The dataset called ICUDatasetProcessed [17] was used to test and validate the performance of the proposed framework. The results show that the TNN model recorded 99.99% on the F1-score accuracy metric. In addition, the blockchain-based scheme offers the expected results.

This paper is structured as follows: section 2 represents the most current security methods used to leverage the security of the IoMT systems. Section 3 gives information about the dataset used to assess this research's proposed framework. Section 4 discusses and explains the research methodology, TNN, and blockchain technology. Section 5 investigates the result and discussion found in this research. Finally, section 6 drives a conclusion to this research work.

2. Related Work

This section represents the up-to-date security methods and strategies utilized to enhance the security of IoT devices. Also, it reviews the recent technology used to mitigate malicious activities conducted through IoT devices on healthcare systems.

Artificial Intelligence (AI) is a common approach used by researchers for protecting IoT devices from threat actors, which provides detection techniques to scan for unusual activities. Several AI models exist, such as neural networks (NN), linear regression, and support vector machines (SVM) [18]. In this research [19], authors used deep learning models to detect the distributed DDoS targeting IoT systems using CICIDS2017 datasets. The authors implemented four deep-learning approaches: long short-term memory (LSTM), convolutional neural network (CNN), and CNN + LSTM. Among these deep learning approaches, CNN + LSTM achieved the highest accuracy, 97.16%.

Authors in [9] utilized features engineering techniques to improve ML's detection and classification accuracy. The authors employed several ML models to identify cyberattacks such as DoS, Mirai, Scan, and man-in-the-middle (MITM) attacks in IoT systems. The authors used Support Vector Machine (SVM), Shallow Neural Networks (SNNs), K-Nearest Neighbor (KNN), Decision Trees (DT), and Bagged Trees (BT). To evaluate their models, they used the IoTID20 dataset. As a result, the accuracy of their ML models ranges between 99.40% to 100%.

Authors in [15] proposed a data security paradigm in an IoT platform incorporating a deep neural network and blockchain. As a result, the platform improves performance regarding latency and accuracy. Also, the work in [20] presents the security architecture of the IoT platform to give

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 18 April 2023 doi:10.20944/preprints202304.0500.v1

scalable and safe IoT data to the IoT platform in a decentralized manner. This technology addresses the problem of centralized data in an IoT network.

The study [21] explored the vulnerability of the Internet of Things (IoT) in three layers: the terminal, network, and application layers. In the terminal layer, different types of devices are connected through the network layer, which transmits data to the cloud. To tackle that issues, authors utilized blockchain technology to provide decentralized security for IoT devices without needing a third party. The study also uses verification and machine learning techniques to detect unusual network activity.

Authors in [22] proposed a security mechanism based on InterPlanetary File System (IPFS) and a blockchain network. IPFS is a cluster node primarily used to authenticate patients and their medical devices. The blockchain network is responsible for securing the transferred data between agents such as patients and doctors. Patients and medical devices are initially registered and authenticated before being submitted to the blockchain network through IPFS. Then, the authority and authenticity of the registered patients and medical devices are synchronized. Finally, the information is disseminated into the blockchain to allow the secure transmission of data with different users.

In this research [23], the authors developed an IoMT security assessment framework (IoMT-SAF) based on web applications. Therefore the stakeholders such as system administrators, patients, and medical professionals through IoMT can examine the degree of security of the IoMT solutions. IoMT solution refers to medical devices services and platforms. Medical devices come in different types, for example, wearable devices such as heart monitors, implantable devices such as cardiac function monitors, ambient such as door sensors, and stationery such as computerized tomography scanners. Services refer to web or mobile applications and can be used to analyze data. Platforms used to facilitate and manage smart devices and applications. Then based on the scenario that was selected by the stakeholders, the system will show the possible security risks and recommended countermeasures.

Azeem et al. [24] proposed an Efficient and Secure Data Transmission and Aggregation (ESDTA) model to enhance the security of the IoMT a the remote healthcare system. Their methods employ Secure Message Aggregation (SMA) and Security Message Decryption (SMD) algorithms to ensure the security of aggregated and transmitted data. Data aggregation decreases redundant communications while enabling efficient bandwidth and energy consumption. However, data aggregation needs a security mechanism hence a symmetric key is used to encrypt the data in the sensor node, which is then encrypted in the fog node. In the beginning, sensor measurements of wearable devices such as blood pressure, body temperature, and oxygen level are collected and aggregated in a data collector node, a mobile node. Then, data is encrypted and transmitted to the fog node. After that, the data is decrypted and sent to the cloud node for storage and analytical purposes. Finally, doctors can access patient data to diagnose it.

According to several research papers, most IoMT approaches rely on a centralized cloud server [25]. However, this will not cope with the tremendous growth of IoT devices in the healthcare domain. Therefore, blockchain technology can be exploited to move toward a decentralized architecture. Hence transmitted data is safely stored in peers rather than a centralized server, which is exposed by a singular point of failure.

In this paper [26], the authors introduced a blockchain model with IoT devices to increase the security and privacy of patient data collected by medical sensors, which doctors can access remotely. In the scenario, authors considered that the patients use wearable devices to monitor health data such as heartbeats, sleeping conditions, and walking distance. The collected sensor data is sent to the smart contracts responsible for analyzing the received data. Smart contracts are automation algorithms that exist on the blockchain to increase trust instead of relying on a third party. If a specific condition is met, an alert is created and sent back from the smart contracts to the patient. Simultaneously, the alert is transmitted to the could server, which confirms the digital signature of the node, and if the node is not verified, then the alert is ignored. The hash function is calculated at the cloud server, then data is transferred to the overlay, a peer-to-peer network consisting of cluster nodes. There are two types of

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 18 April 2023 doi:10.20944/preprints202304.0500.v1

nodes in the overlay network patient devices and healthcare providers. Table 1 contains a list of security methods discussed in the related work section.

Table 1. Summary of Related Work Security Methods

Research	Security Methods
Roopak et al. [19]	CNN + LSTM
Alsulami et al. [9]	KNN, SNNsNN,SVM,DT,BT
Wang et al. [15]	deep neural network and blockchain
Qian et al. [21]	Blockchain
Randhir et al. [22]	IPFS
Faisal et al. [23]	IoMT-SAF
Muhammad et al. [24]	ESDTA
Dwivedi et al. [26]	blockchain

Overall, the related work focused on the security methods used to enhance the security of IoMT that were implemented based on an ML model and blockchain technology. However, this research proposes a resilient security framework that integrates ML and blockchain technology to ensure the security of IoMT.

3. Dataset

A dataset named ICUDatasetProcessed [17] was used to test and validate the proposed security framework of this research. The dataset contains 42 features and 187,643 records. In addition, it consists of three labels: patient monitoring, environment monitoring labeled as normal data, and cyberattack labeled as anomalous. Table 2 lists a statistical summary of the dataset records. There are 108,568 records labeled as normal data and 79,075 as anomalous data.

Table 2. Labels and Number of Records

Labels	Type	Number of Records
Patient monitoring	Normal	76,810
Environment monitoring	Normal	31,758
Cyberattack	Anomalous	79,075
Total Records	**187,643**	

The IoMT dataset was generated using IoT-Flock, an open-source tool that can generate IoT traffic with many scenarios depending on a user's choice [17]. The scenario created by the authors of the ICUDatasetProcessed dataset is based on Intensive Care Unit (ICU).

The following medical devices were used in the ICU scenario:

- ECG monitoring: this device is used to monitor the heart rate.
- Infusion Pump: this device is used to deliver medications or nutrients.
- Pulse Oximeter (SPO2): this device can be placed on the finger of the patient to measure the Oxygen level.
- AirFlow Sensor: This device is used to measure the breathing level of the patient.
- Blood pressure monitor: this device is used to monitor blood pressure.
- Glucometer: this device is used to calculate the glucose in the blood.
- Body Temperature Sensor: this device is used to sense the patient's body temperature.
- Electromyography (EMG) Sensor: this device monitors the electrical signal generated by the muscles.
- Galvanic Skin Response (GSR) Sensor. This device is used to measure the electric signal generated by the skin.

 doi:10.20944/preprints202304.0500.v1

The following environmental sensors were also involved in the ICU scenario:

- Air humidity sensor: this device is used to measure the humidity of the air.
- Air temperature sensor: this device is used to calculate the temperature of the air.
- CO sensor: this device is used to sense the carbon monoxide level in the ICU room.
- Fire sensor: this device is used to detect fire or flame in the ICU room.
- Smoke sensor: this device is used to detect the level of smoke in the ICU room.
- Barometer: this device is used to measure the air pressure in the ICU room.
- Solar radiation sensor: this device is used to measure the power of the heat of the light or the sun.

There are four cyberattacks generated at the application layer: MQTT DDoS, MQTT publishes flood attack, brute force attack, and SlowITE attack. MQTT is an acronym for Message Queuing Telemetry Transport. MQTT is a publisher and subscriber protocol used for message queuing at the application layer in IoT systems. [27].

4. Methodology

This section explores and discusses the topology of this study's proposed method, which combines TNN and blockchain models. The TNN is responsible for detecting anomalies in the data collected from medical sensors. Therefore, abnormal data is discarded from the IoMT traffic; however, normal data is transmitted to the blockchain model to ensure that data cannot be tempered after filtered by the TNN model. In addition, this section provides a detailed explanation of the implementation of TNN and blockchain models.

4.1 Proposed Security Framework of IoMT

It is well known that IoMT devices are not immune from cybersecurity threats. Therefore this paper proposed a security framework based on the TNN model and blockchain technology. Figure 2 illustrates a block diagram of the top architecture of the security framework. Firstly, two types of data are sent to the TNN model. The first is that normal data is measured by medical sensors, and the second is anomalous data which was injected by an attacker. Next, the TNN performs anomaly detection to distinguish normal data (patient monitoring, environment monitoring) and anomalous data (MQTT DDoS, MQTT publishes flood, brute force, and SlowITE attacks). As a result, anomalous data is excluded from the IoMT system, while normal data is transmitted to the fog layer. Then, the blockchain is established in the fog layer to ensure data integrity and immutability, which is accomplished by validating each transaction in the blockchain [28]. Finally, because of the limited capacity of the fog layer, data can be sent to the cloud layer for further analysis. Also, it can be noticed that data can be viewed through user devices from fog or cloud layers.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 18 April 2023 doi:10.20944/preprints202304.0500.v1

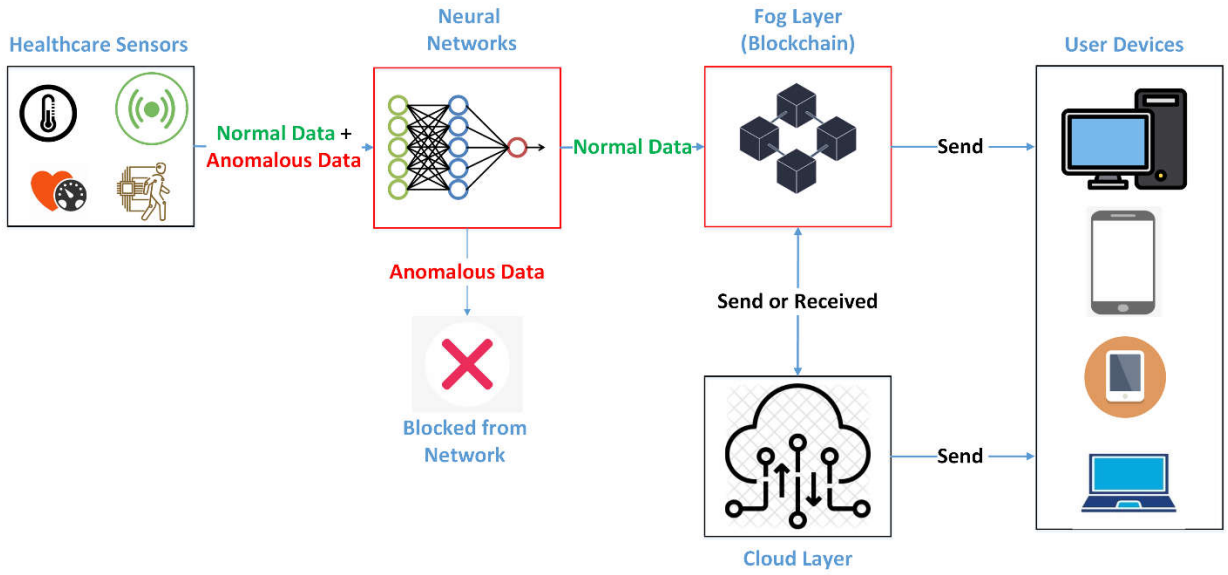

Figure 2. The Proposed Security Architecture of IoMT

4.2 Tri-layered Neural Network Classifiers

This research employs a TNN model to detect the medical sensors data, which has two categories normal data (patient monitoring and environment monitoring) and attack. The dataset used to train and test the TNN model is ICUDatasetProcessed [17] since it was collected from the IoMT domain; therefore, it is relevant for this research. The architecture of the TNN model is depicted in Figure 3. It is a feedforward neural network following a supervised learning approach, consisting of three fully connected hidden layers and one output layer. Each hidden layer comprises ten neurons, and the output layer has only one neuron. The input vector, in our case, the 42 features, is fed to the first hidden layer. Each layer supposes to multiply the input vector by the weight (w), and the result is added with the bias (b) vector. Then the activation function at each layer is used to allow the use of non-linearity and prevent linearity [29]. The activation function technique used is Rectified Linear Units (ReLUs) [30]. The output of the TNN is patient monitoring, environment monitoring, and attack.

Figure 4 shows the preprocessing flow of the TNN detection model. The first stage is to import the dataset Comma-Separated Values (CSV) file into a Matlab project. The second step is to prepare the data in which any record with an empty or inappropriate value is deleted. There are two numerical labels in the data, 0 refers to normal, and 1 refers to attack. The third step is to train and test the implemented TNN model. The size of the dataset used for training and testing is (187,643 x 42). It contains 187,643 records and 42 features. 70% of the data was used for training, and 30% was used for testing. In order to validate the training procedure, the k-fold cross-validation technique was used, and the k value equals 10 [31]. This means the entire data is divided randomly into 10 folds with equal sizes to eliminate overfitting. Technically, one fold is used for testing, and the remaining folds are used for training. In the fourth step, the output of the TNN detection model is obtained, which is a normal record or anomaly (attack) record. The fifth step is to evaluate the TNN model, which will be covered in the Result and Discussion section.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 18 April 2023 doi:10.20944/preprints202304.0500.v1

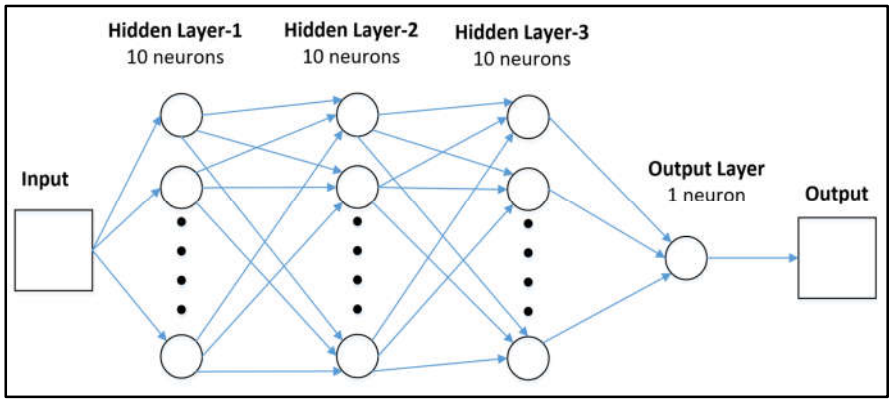

Figure 3. TNN Topology.

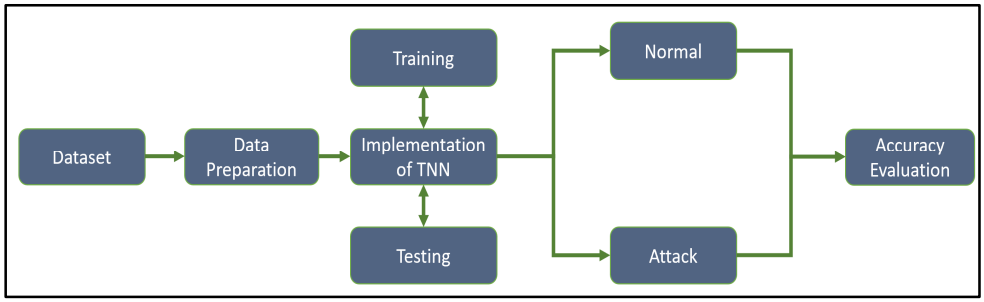

Figure 4. The preprocessing Flow of the TNN Detection Model.

4.3 Blockchain

This research utilized blockchain methodology because it is a modern technology that solves the challenge of centralization, privacy, and trust. Blockchain is a formal technique used in the cryptocurrency industry, such as Bitcoin [32]. One can think of blockchain as a database storing transactions inside blocks using a peer-to-peer network with a decentralization approach that eliminates the existence of the third party. The decentralized nature of the blockchain improves the system's availability and reliability and reduces the possibility of a single point of failure. Figure 5 depicts the blockchain model implemented in this study. The first block in the blockchain is called the genesis block, and there are N numbers of blocks. Each block contains two hashes (except Genesis block), difficulty, nonce, timestamp, and transactions. A hash is a mathematical function used to encrypt data with variable length into encrypted data with fixed length [33]. The hash is used to replace the trust in the traditional functional system. Each block except the Genesis block has two hashes, the previous hash, and its own hash; therefore, the block only has information about the previous block. Thus, blockchain technology maintains the immutability of data stored in the fog nodes by utilizing the blockchain hashing feature, which forms secure interconnections between the blocks. When a computer wants to participate in blockchain networks, it typically chooses the longest valid blockchain to join. It is clear that controlling the growth of blocks becomes important since attackers might increase the number of fabricated blocks to attract the new joiner. This is why mining difficulty is used, which refers to delaying the creation of a new block by increasing the mining difficulty level using a Proof-of-Work (PoW) algorithm [34]. Therefore, difficulty is important to control how fast the block can be mined. PoW is the most popular consensus protocol used in blockchain industries, such as Bitcoin. A nonce is a random value miners use to compute the hash [32]. It also verifies the hash because a nonce number is used once. Transactions are the activities produced by system participants. In addition, each block has many transactions from 1 to N, depending on the block size. When a transaction is initiated, then this transaction is broadcasted to every node in the network. All nodes that receive the new transaction verify it and hold it in their ledger.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 18 April 2023 doi:10.20944/preprints202304.0500.v1

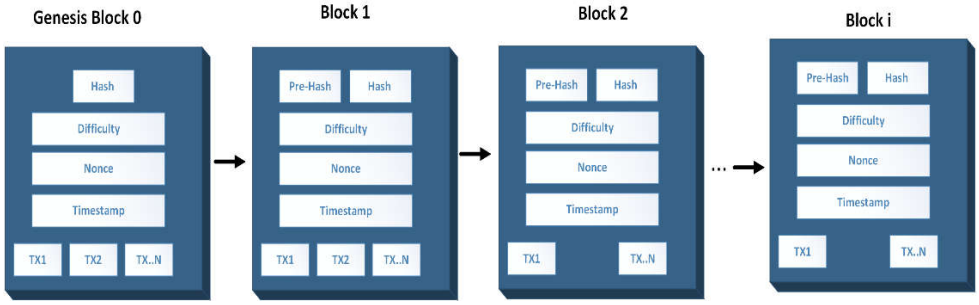

Figure 5. Blockchain.

4.3.1 Blockchain Scheme

This section presents the implementation of the blockchain scheme of this research. Four algorithms are implemented using Python: BlockchainMain(), BlockchainClass, and BlockClass. We follow the basic model of designing a public blockchain scheme, which was first introduced in [35]

Algorithm 1 demonstrates the steps of the main function of the blockchain model implemented in this research. The two lists, *rows*, and *chunk*, are declared in the beginning. The *rows* list stores every record of the dataset, and the *chunk* list is used to store a piece of records depending on the number of IoMT devices used in the blockchain network. The variable chunk_size is assigned a value of N depending on the number of simulated IoMT devices. Then the CSV file, which has the dataset, is read. After that, data is split into chunks. Then the loop from lines 7 to 14 shows the main procedures of the blockchain technique. In line 8, the blockchain class is called, which will be explained in Algorithm 2. In line 10, the device class is called, which will be discussed in Algorithm 3. In line 12, a new thread of a device object is created. Finally, line 13 stops calling new threads until the currently processed threads are terminated.

Algorithm 1: BlockchainMain()	
1:	*Initialize the following variables:*
2:	*rows ← [ ]*
3:	*chunk ← [ ]*
4:	*chunk_size ← N*
5:	*Read the CSV file and store the rows in the list "rows"*
6:	*Split the list "rows" into chunks based on the chunk_size*
7:	***for** each chunk in chunks:*
8:	*call BlockchainClass*
9:	*blockchain = BlockchainClass()*
10:	*call device class*
11:	*deviceObject = DeviceClass(blockchain, chunk)*
12:	*Start a new thread of device object*
13:	*Wait for all device threads to finish*
14:	***End for***

Algorithm 2 shows the implementation of the blockchain class. Initially, an object lock is created to serve as a lock for the thread, so only one thread can access the critical code section at a time. Then the difficulty is set to one as a proof-of-concept prototype of our blockchain-based solution to reduce the computing power needed to mine a block. After that, the block list is created, which stores the values of the block. The first block stored in the list is the Genesis block, with the previous hash value equal to zero because it is the first block. Next, the transactions list is initialized to store the transactions of each block. Then, in line 5, the add_block function is created and used to create a new block to be stored in the blocks list. The if clause is used to check whether the previous block's hash is equal to the last block's hash to confirm the validity of the new block. Next, the

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 18 April 2023 doi:10.20944/preprints202304.0500.v1

broadcast_transaction function is used to add a new transaction to the transactions list while locking the list to prevent concurrent access. Then the mine_block function creates a new_block object with a list of transactions, the last block hash, and difficulty. Next, the mine function is called and will be explained in Algorithm 3. Finally, the object new_bock is returned.

Algorithm 3: BlockClass	
1:	*Initialize the objects:*
2:	*data,*
3:	*previous_hash,*
4:	*hash,*
5:	*difficulty,*
6:	*nonce,*
7:	*timestamp*
8:	***mine()***
9:	*Calculate hash using sha256*

Algorithm 2: BlockchainClass	
1:	*Create an object "lock" with a threading lock*
2:	*Set the difficulty to 1*
3:	*Initialize the blocks list and assign the Genesis_block as the first item*
4:	*Initialize the transactions list =[ ]*
5:	***add_block(block)***
6:	***if** block.previous_hash == blocks[last].hash*
7:	*add the block to the blocks list*
8:	***return True***
9:	***else***
10:	***return False***
11:	***broadcast_transaction(transaction)***
12:	*Use thread lock object*
13:	*transactions=[transaction]*
14:	***mine_block()***
15:	*new_block = BlockClass(transactions, blocks[last].hash, difficulty)*
16:	*new_block = mine()*
17:	*return new_block*

Algorithm 3 demonstrates the block class used to create a block for a blockchain network. The class has the following attributes:

- data: the data stored in the block.
- previous_hash: the hash value of the previous block in the blockchain.
- hash: the hash value of the current block.
- difficulty: the difficulty level for mining the block.
- nonce: the arbitrary number used to change the block's hash value during ming.
- timestamp: the time at which the block was mined.

The mine function is used to add a block to the blockchain network. In this process, the previous hash is needed to link the new block with the previous block.

Algorithm 4 is used to demonstrate the device class. The class device is inherited from the thread function to simulate the concurrent operation of multi-devices. In our case, each block created in a device should contain 35 transactions (this number can be customized). However, if the number of transactions reaches 35, then the block is mined.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 18 April 2023 doi:10.20944/preprints202304.0500.v1

Algorithm 4: DeviceClass	
1:	*Define a class named "device" that inherits from the thread function*
2:	*Use the following object:*
3:	*blockchain*
4:	*rows*
5:	*pending_transactions= 0*
6:	*create a dictionary called transaction[]*
7:	***run()***
8:	***for** each row in rows*
9:	***if** pending_transactions < 35*
10:	*transaction = [data]*
11:	*broadcast_transition (transaction)*
12:	*pending_transactions += 1*
13:	***else***
14:	*block = mine_block()*
15:	*pending_transactions= 0*

5. Result and Discussion

The TNN model was evaluated using a confutation matrix to calculate the accuracy of the TNN. The confutation matrix general form is shown in Figure 6. True positive (TP) means the TNN successfully classified normal sensor data as normal. Also, true negative means the TNN was able to classify anomalous data as anomalous. However, false positive (FP) means the TNN mistakenly classified normal sensor data as anomalous. Finally, a false negative (FN) means the TNN mistakenly classified anomalous data as normal [36].

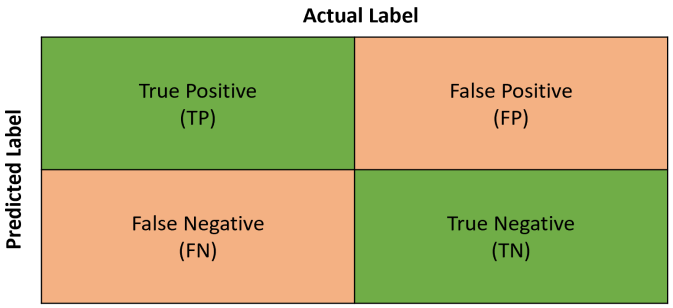

Figure 6. Confusion Matrix.

The detection result of the TNN model is represented in Figure 7. TNN was able to classify 32,568 records of the tested data as normal and only missed classifying 2 records as anomalous. However, the model successfully classified all malicious data as anomalous records. Therefore, information obtained from Figures 6 and 7 is used to compute the accuracy metrics. The classification accuracy was calculated using Equation 1, and the TNN model scored 99.99% [37].

$$Classification\ Accuracy \frac{TP + TN}{TP + FP + TN + FN}\ x\ 100 \quad (1)$$

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 18 April 2023 doi:10.20944/preprints202304.0500.v1

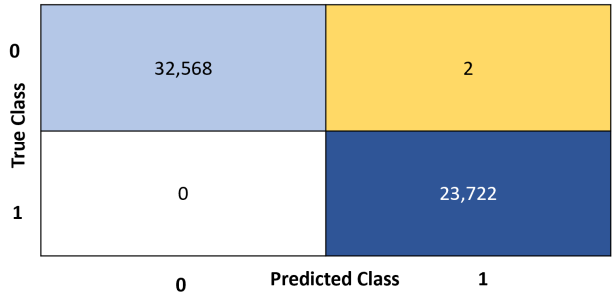

Figure 7. Confusion Matrix Detection Result.

Researchers usually do not rely only on classification accuracy. For that reason, F1-score is calculated using Equation 4. One way to easily calculate F1-score is by first calculating precision and recall, as shown in equations 2 and 3. Then, the calculation result for the precision is 99.99%, and 100% for the recall. Finally, the F1 score for the model is 99.99% [36].

$$Precision = \frac{TP}{TP + FP}\ x\ 100 \quad (2)$$

$$Recall = \frac{TP}{TP + FN}\ x\ 100 \quad (3)$$

$$F1 - score = \frac{Precision\ x\ Recall}{Precision + Recall}\ x\ 100 \quad (4)$$

It can be observed that because the TNN model achieved high accuracy based on the accuracy metrics discussed above, TNN can serve this research purpose as a detection model. To validate the performance of the TNN model, we compared our finding results with Hussian et al. [17]. We compared our findings with [17] because the authors used the same dataset we used in this research, resulting in fairness. Table 3 lists various detection models implemented based on ML algorithms to detect cyberattacks in the dataset ICUDatasetProcessed. Regarding the classification accuracy metric, our proposed TNN scored the highest results compared with the rest of the detection models. The Naive Bayes (NB) scored the lowest classification accuracy, precision, and F-1score results. Nevertheless, the TNN achieved the highest result in each metric. K-Nearest Neighbors (KNN) Random Forest (RF), Adaboost (AB), Logistic Regression (LogR), and Decision Tree (DT) scored very close results to the TNN. Overall, by looking at the table, we can observe that TNN outperforms other ML models based on accuracy, precision, recall, and F1-score metrics. The closest model in terms of performance to our model is RF.

Figures 8 and 9 show samples of the JSON output of two sequential blocks in the blockchain network. JASON stands for JavaScript Object Notation, a textual representation of transporting and storing data [38]. JSON can be used to transfer data between computers using any programming language. Figure 8 represents a block sample with the following parameters: hash, previous_hash, difficulty, nonce, timestamp, and transactions. The transactions parameter has several transactions based on the size of the block. In our case, we assumed that each block has up to 35 transactions, but users can customize this number based on the capacity of the IoMT devices. Figure 9 represents another block sample; we noticed that the parameter previous_hash has the same value as the hash in Figure 8; because the two blocks sample is consecutive.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 18 April 2023 doi:10.20944/preprints202304.0500.v1

Table 3. Accuracy of Various Security Detection Techniques.

Detection Model	Classification Accuracy	Precision	Recall	F1-score
NB	52.18%	79.67%	99.71%	68.51%
KNN	99.49%	99.65%	99.69%	99.59%
RF	99.51%	99.71%	99.80%	99.65%
AB	99.50%	99.55%	99.45%	99.47%
LogR	99.50%	95.29%	90.35%	94.71%
DT	99.48%	99.69%	99.80%	99.64%
Proposed TNN	**99.99%**	**99.99%**	**100%**	**99.99%**

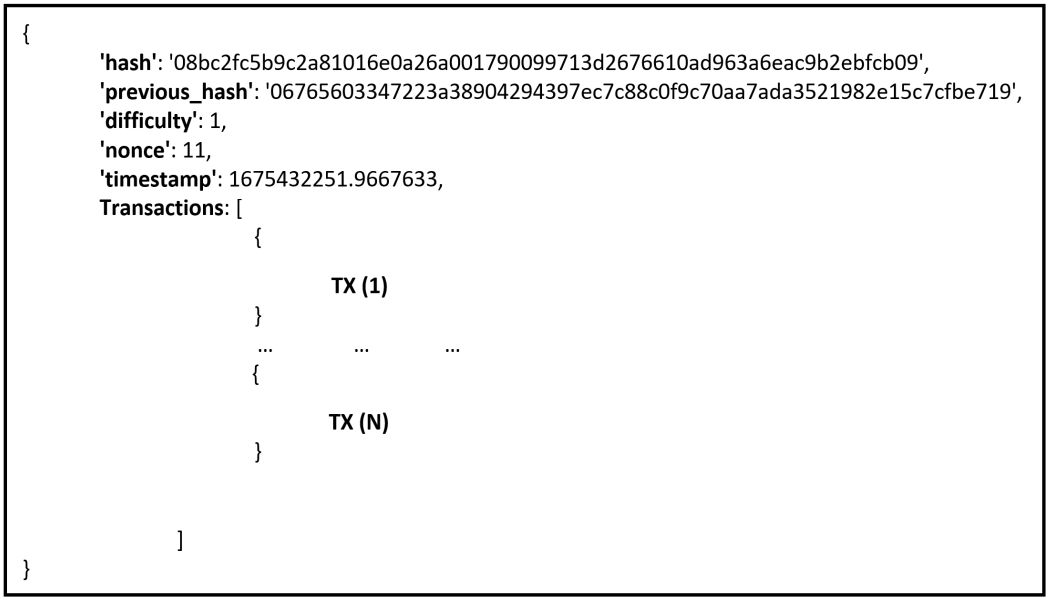

Figure 8. Sample (1) information of block in the blockchain.

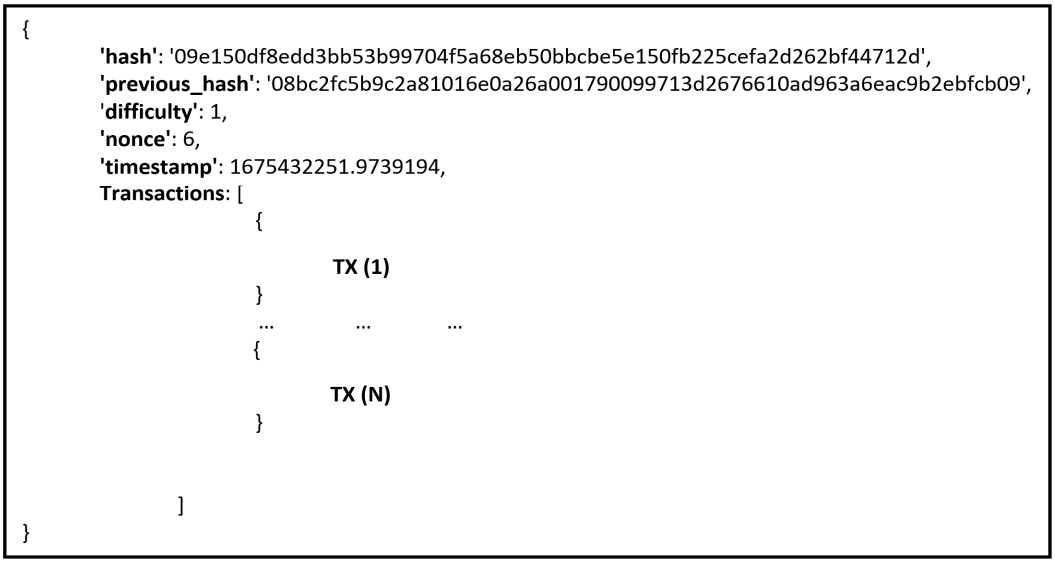

Figure 9. Sample (2) information of block in the Blockchain.

6. Conclusions

The security of IoMT is a critical concern due to the system's complexity and the security limitation of IoT devices which are vulnerable and susceptible to cyberattacks. This paper proposed a robust security framework leveraging Tri-layered Neural Network (TNN) and Blockchain models. Specifically, the TNN model was utilized to detect cyberattacks from medical sensors by identifying anomalous data, which was then blocked from the IoMT system. At the fog layer, a Blockchain model was employed to ensure the integrity of the data stored in the IoMT devices after being vetted by TNN. The results demonstrate that the TNN model achieved high accuracy, precision, and F1-score metrics, as well as perfect recall performance, outperforming existing methods such as NB, KNN, RF, AB, and LogR. Furthermore, the proposed blockchain-based simulation scheme yielded the expected outcomes, and its parameters can be customized to cater to different purposes. Given its versatility, this framework can be easily adapted to other applications and datasets by adjusting the settings to the new environment. Due to its trade-off, the proposed blockchain-based solution concentrates on security, not performance. The system performance will be examined and compared with a private blockchain in future work.

Supplementary Materials: Not applicable.

Author Contributions: Conceptualization, A.A.A. and R.A.A.; methodology, A.A.A., B.A., and R.A.A.; software, A.A.A. validation, M.Y.D., B.A., and S.A.; formal analysis, R.A.A.; investigation, B.A.; resources, S.A.; data curation, A.A.A.; writing—original draft preparation, A.A.A. and R.A.A.; writing—review and editing, A.A.A, R.A.A., B.A., M.Y.D., and S.A.; visualization, R.A.A.; supervision, M.Y.D., and A.A.A.; project administration, A.A.A.; funding acquisition, A.A.A. and R.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: For more information about the data used in this study, we refer the readers to the following link: https://github.com/ThingzDefense/Malicious-Traffic-Detection-in-IoT-Healthcare-Environment

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

References

1. Dilawar, N.; Rizwan, M.; Ahmad, F.; Akram, S., Blockchain: securing internet of medical things (IoMT). *International Journal of Advanced Computer Science and Applications* **2019,** 10, (1).
2. Mavrogiorgou, A.; Kiourtis, A.; Perakis, K.; Pitsios, S.; Kyriazis, D., IoT in healthcare: achieving interoperability of high-quality data acquired by IoT medical devices. *Sensors* **2019,** 19, (9), 1978.
3. Koutras, D.; Stergiopoulos, G.; Dasaklis, T.; Kotzanikolaou, P.; Glynos, D.; Douligeris, C., Security in IoMT communications: A survey. *Sensors* **2020,** 20, (17), 4828.
4. Aman, A. H. M.; Hassan, W. H.; Sameen, S.; Attarbashi, Z. S.; Alizadeh, M.; Latiff, L. A., IoMT amid COVID-19 pandemic: Application, architecture, technology, and security. *Journal of Network and Computer Applications* **2021,** 174, 102886.
5. Yaacoub, J.-P. A.; Noura, M.; Noura, H. N.; Salman, O.; Yaacoub, E.; Couturier, R.; Chehab, A., Securing internet of medical things systems: Limitations, issues and recommendations. *Future Generation Computer Systems* **2020,** 105, 581-606.
6. Almalki, J.; Al Shehri, W.; Mehmood, R.; Alsaif, K.; Alshahrani, S. M.; Jannah, N.; Khan, N. A., Enabling Blockchain with IoMT Devices for Healthcare. *Information* **2022,** 13, (10), 448.
7. Ghubaish, A.; Salman, T.; Zolanvari, M.; Unal, D.; Al-Ali, A.; Jain, R., Recent advances in the internet-of-medical-things (IoMT) systems security. *IEEE Internet of Things Journal* **2020,** 8, (11), 8707-8718.
8. Bagaa, M.; Taleb, T.; Bernabe, J. B.; Skarmeta, A., A machine learning security framework for iot systems. *IEEE Access* **2020,** 8, 114066-114077.

9. Alsulami, A. A.; Abu Al-Haija, Q.; Tayeb, A.; Alqahtani, A., An Intrusion Detection and Classification System for IoT Traffic with Improved Data Engineering. *Applied Sciences* **2022,** 12, (23), 12336.
10. Hariguna, T.; Durachman, Y.; Yusup, M.; Millah, S., Blockchain technology transformation in advancing future change. *Blockchain Frontier Technology* **2021,** 1, (01), 13-20.
11. Laroiya, C.; Saxena, D.; Komalavalli, C., Applications of blockchain technology. In *Handbook of research on blockchain technology*, Elsevier: 2020; pp 213-243.
12. Bao, J.; He, D.; Luo, M.; Choo, K.-K. R., A survey of blockchain applications in the energy sector. *IEEE Systems Journal* **2020,** 15, (3), 3370-3381.
13. Yazdinejad, A.; Srivastava, G.; Parizi, R. M.; Dehghantanha, A.; Choo, K.-K. R.; Aledhari, M., Decentralized authentication of distributed patients in hospital networks using blockchain. *IEEE journal of biomedical and health informatics* **2020,** 24, (8), 2146-2156.
14. Kumar, R.; Sharma, R., Leveraging blockchain for ensuring trust in IoT: A survey. *Journal of King Saud University-Computer and Information Sciences* **2022,** 34, (10), 8599-8622.
15. Wang, S.; Ouyang, L.; Yuan, Y.; Ni, X.; Han, X.; Wang, F.-Y., Blockchain-enabled smart contracts: architecture, applications, and future trends. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* **2019,** 49, (11), 2266-2277.
16. Atlam, H. F.; Alenezi, A.; Alassafi, M. O.; Wills, G., Blockchain with internet of things: Benefits, challenges, and future directions. *International Journal of Intelligent Systems and Applications* **2018,** 10, (6), 40-48.
17. Hussain, F.; Abbas, S. G.; Shah, G. A.; Pires, I. M.; Fayyaz, U. U.; Shahzad, F.; Garcia, N. M.; Zdravevski, E., A framework for malicious traffic detection in IoT healthcare environment. *Sensors* **2021,** 21, (9), 3025.
18. Kuzlu, M.; Fair, C.; Guler, O., Role of artificial intelligence in the Internet of Things (IoT) cybersecurity. *Discover Internet of things* **2021,** 1, 1-14.
19. Roopak, M.; Tian, G. Y.; Chambers, J. In *Deep learning models for cyber security in IoT networks*, 2019 IEEE 9th annual computing and communication workshop and conference (CCWC), 2019; IEEE: 2019; pp 0452-0457.
20. Rathore, S.; Kwon, B. W.; Park, J. H., BlockSecIoTNet: Blockchain-based decentralized security architecture for IoT network. *Journal of Network and Computer Applications* **2019,** 143, 167-177.
21. Qian, Y.; Jiang, Y.; Chen, J.; Zhang, Y.; Song, J.; Zhou, M.; Pustišek, M., Towards decentralized IoT security enhancement: A blockchain approach. *Computers & Electrical Engineering* **2018,** 72, 266-273.
22. Kumar, R.; Tripathi, R., Towards design and implementation of security and privacy framework for internet of medical things (iomt) by leveraging blockchain and ipfs technology. *the Journal of Supercomputing* **2021**, 1-40.
23. Alsubaei, F.; Abuhussein, A.; Shandilya, V.; Shiva, S., IoMT-SAF: Internet of medical things security assessment framework. *Internet of Things* **2019,** 8, 100123.
24. Azeem, M.; Ullah, A.; Ashraf, H.; Jhanjhi, N.; Humayun, M.; Aljahdali, S.; Tabbakh, T. A., Fog-oriented secure and lightweight data aggregation in iomt. *IEEE Access* **2021,** 9, 111072-111082.
25. Jolfaei, A. A.; Aghili, S. F.; Singelee, D., A survey on blockchain-based IoMT systems: Towards scalability. *Ieee Access* **2021,** 9, 148948-148975.
26. Dwivedi, A. D.; Malina, L.; Dzurenda, P.; Srivastava, G. In *Optimized blockchain model for internet of things based healthcare applications*, 2019 42nd international conference on telecommunications and signal processing (TSP), 2019; IEEE: 2019; pp 135-139.

27. Vaccari, I.; Aiello, M.; Cambiaso, E., SlowITe, a novel denial of service attack affecting MQTT. *Sensors* **2020,** 20, (10), 2932.
28. Omar, I. A.; Jayaraman, R.; Salah, K.; Yaqoob, I.; Ellahham, S., Applications of blockchain technology in clinical trials: review and open challenges. *Arabian Journal for Science and Engineering* **2021,** 46, 3001-3015.
29. Chen, Y.; Dai, X.; Liu, M.; Chen, D.; Yuan, L.; Liu, Z. In *Dynamic relu*, Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIX 16, 2020; Springer: 2020; pp 351-367.
30. Wang, G.; Giannakis, G. B.; Chen, J., Learning ReLU networks on linearly separable data: Algorithm, optimality, and generalization. *IEEE Transactions on Signal Processing* **2019,** 67, (9), 2357-2370.
31. Maray, M.; Alghamdi, M.; Alazzam, M. B., Diagnosing cancer using IOT and machine learning methods. *Computational Intelligence and Neuroscience* **2022,** 2022.
32. Nofer, M.; Gomber, P.; Hinz, O.; Schiereck, D., Blockchain. *Business & Information Systems Engineering* **2017,** 59, 183-187.
33. How Block Hashes Work in Blockchain? https://www.geeksforgeeks.org/how-block-hashes-work-in-blockchain/
34. Yang, X.; Chen, Y.; Chen, X. In *Effective scheme against 51% attack on proof-of-work blockchain with history weighted information*, 2019 IEEE International Conference on Blockchain (Blockchain), 2019; IEEE: 2019; pp 261-265.
35. Nakamoto, S., Bitcoin: A peer-to-peer electronic cash system. *Decentralized business review* **2008**, 21260.
36. Al-Haija, Q. A.; Alsulami, A. A., High performance classification model to identify ransomware payments for heterogeneous bitcoin networks. *Electronics* **2021,** 10, (17), 2113.
37. Alsulami, A. A.; Abu Al-Haija, Q.; Alqahtani, A.; Alsini, R., Symmetrical Simulation Scheme for Anomaly Detection in Autonomous Vehicles Based on LSTM Model. *Symmetry* **2022,** 14, (7), 1450.
38. Pezoa, F.; Reutter, J. L.; Suarez, F.; Ugarte, M.; Vrgoč, D. In *Foundations of JSON schema*, Proceedings of the 25th international conference on World Wide Web, 2016; 2016; pp 263-273.