sustainability

MDPI

Article

Flexible-Clustering Based on Application Priority to Improve IoMT Efficiency and Dependability

Amir Masoud Rahmani [1] and Seyedeh Yasaman Hosseini Mirmahaleh [2,*]

[1] Future Technology Research Center, National Yunlin University of Science and Technology, 123 University Road, Section 3, Douliou 64002, Taiwan
[2] Department of Electrical Engineering, Science and Technology, Lille University, 59000 Lille, France
* Correspondence: yasaman-hosseini@ieee.org

Abstract: The Internet of Medical Things (IoMT) has overcome the privacy challenges of E-healthcare-based Internet of Things (IoT) systems to protect the joined people's private records to IoMT infrastructures and support their information security in different layers. By deploying various medical applications, security and privacy are challenging for the IoMT via rising communications between its layers and nodes. Some case studies aimed to solve the issues and provided various methods and protocols to identify the malicious data and information, which had almost overlooked application and service priority to targeting the research and satisfying security. We addressed the dependability and privacy problems of IoMT-based applications by presenting an intelligent algorithm for node mapping and flexible clustering (NFC) via defining a graph and employing a neural network (NN). This work proposes a flexible clustering method to categorize the healthcare service providers for timely detecting faults and identifying the proper servers to join the cluster by considering service and application priority. We improve the application dependability and privacy by about 77.3–83.2% via pruning the defective nodes and employing the neighbor components to support faulty devices' role. By removing the failed or faulty nodes, the study reduces communication delay and energy consumption, approximately 19.3–21.7% and 10.3–11.8%, respectively.

Keywords: Internet of Medical Things (IoMT); Internet of Things (IoT); flexible clustering; graph; neural network (NN); dependability

check for updates

Citation: Rahmani, A.M.; Hosseini Mirmahaleh, S.Y. Flexible-Clustering Based on Application Priority to Improve IoMT Efficiency and Dependability. *Sustainability* **2022**, *14*, 10666. https://doi.org/10.3390/su141710666

Academic Editor: Changhyun Roh

Received: 24 July 2022
Accepted: 22 August 2022
Published: 26 August 2022

1. Introduction

Recently, medical studies addressed finding the solutions for timely detecting various diseases and providing correct healthcare services to patients. The different remote-controlling methods were presented by employing biosensors and other types. Researchers proposed E-healthcare systems to monitor people's health records and control their vital signs, supported by IoT technology. By increasing the number of medical-based applications and combining them with other demands, IoT technology faced satisfying privacy and security problems to protect the patients' private records and the doctor's treatment recommendations [1,2]. The IoMT technology was customized to support medical-based applications for protecting private records and recommendations from intruders and invalid users through specialized protocols and algorithms [3,4]. By deploying IoMT, the studies demonstrated rising the joined applications to the technology's infrastructures and creating security, availability, and dependability challenges because of growing up the complexity and communications between the different layers and nodes (Including physical, information, integration, and application layers) [5,6]. Authentication protocols aimed to identify the legitimate doctors and patients to access the shared information and data by defining the public and private keys for the joined users [7,8]. The protocols created an opportunity to protect the shared private records and recommendations for improving the security of IoMT applications. Researchers applied the cryptography techniques of symmetric and asymmetric algorithms to define different authentication keys, encryption, and decryption information to support the security of transferring data between the IoMT nodes and layers [9–13]. By providing trust-based services, the studies tried to solve the security and privacy problems via defining a hard limitation to provide or dedicate a server to a requester when correctness has a highlighter role than availability [14,15]. The trust-based approaches lead to the high limitations of providing a service. They can create dangerous situations in risky conditions, such as facing accidents, natural disasters, and limitations of service provider sources. Pelekoudas-Oikonomou et al. [16] focused on providing a method to utilize the blockchain approach in edge-networks of the IoMT applications to prevent sharing malicious healthcare information via the devices, which leads to improving the technology security. This work supported the security on the edge level's nodes by protecting the shared private records and treatment recommendations. Wazid et al. [17] employed a blockchain model to improve security by protecting the data transferring of the IoMT-based E-healthcare systems via defining a key to control and access the shared information via the wireless sensors. The blockchain-based methods increase the security of the IoMT applications by employing encryption, decryption, and the authentication key, which can lead to risky conditions for specific situations with availability as the service priority. Some researchers improved the security and privacy of the IoMT infrastructures by predicting the active and passive attacks in the application layer and identifying intruders before sharing malicious data, where the precision of prediction has an impressive effect on providing proper healthcare service to the patient [18,19].

To satisfy the IoMT privacy, case studies targeted a special application to provide a method for protecting the shared information between sensor nodes and healthcare service providers. Ibaida et al. [20] addressed the privacy problem to protect the output signals of Electrocardiogram (ECG) sensors from losing information by employing a machine learning algorithm (ML) and neural network. The study can only support the privacy-preserving of the related applications to ECG signals for providing proper service to a requester. Some case studies utilized the sparse learning method for the cryptography algorithm's decryption phase and improved the privacy of IoMT-based applications for increasing the accuracy of decrypted ECG signals, which only covered the related application to Electrocardiogram [21,22].

Some researchers focused on protecting the biosensors and other types of them to access the illegitimate doctors and vulnerabilities to avoid creating dangerous conditions by imposing malicious information on the connected sensors to the patients [23]. The studies attended to specific applications to detect cyber-attacks and intruders for preventing the different vulnerabilities by employing machine learning algorithms and Bayesian-based models to update the information [23]. The neural networks and their activation functions have an impressive effect on improving the efficiency of the IoMT applications and IoT-based E-healthcare systems by inferencing data and training them [24,25]. Nevertheless, the faint role of application and service priority creates limitations to the recent security and privacy approaches of the IoMT infrastructures.

The above-mentioned studies and concepts prove the reasons for attending the domain and providing our idea to improve the IoMT's efficiency and dependability, which include:

- Faint role of application and service priority to define and propose security methods for IoMT-based applications
- Lack of considering application priority to provide privacy methods for the IoMT.
- Customizing the security and privacy methods to the specific applications and sensors
- Lack of targeting and generalizing methods to support the applications in facing risky situations and unpredictable conditions
- Neglecting the imposing communication delay and energy consumption overheads to the performance of the IoMT applications

We addressed the mentioned issues to provide an idea for overcoming them and improving the IoMT applications' efficiency and dependability by analyzing availability, security, and confidence. This work pays attention to the service and application priority for presenting an intelligent four-phase algorithm to map the application nodes onto the vertices of a graph (including V vertices and E edges) and clustering them based on detecting faulty service providers. The main graph consists of the mapped application's service provider nodes and the relationship between them and physical and information layers nodes, which are defined by the edges. The initial cluster demonstrates the initial server nodes of an application where each node identifies the neighbor cluster to utilize them in facing the defective components. To determine the proper nodes in the neighbor cluster, we use the quasi-mapping method to transfer their role onto a neural network's neurons apart from the mapped service providers and other components of the main application [24]. NFC trains the neural network's weights and neurons based on the service and application priority and utilizes the backward chain to impose penalties onto the NN's weights. Due to decision making about the proper neighbor service provider and joining them to the cluster, we define a threshold value for each IoMT application by considering priority and the number of servers. After detecting faults and training the neural network, NFC prunes the weak weights and neurons identified as unfit service providers by imposing the estimated penalty on the weights. By removing the improper node and re-clustering the graph vertices and edges, we stop the defective nodes, eliminate their energy, and delay overheads for the application efficiency and improve dependability by avoiding sharing malicious data. Figure 1 indicates the steps of the case study to node mapping and flexible-clustering the graph vertices and edges, as well as quasi-mapping the neurons of a partial neural network.

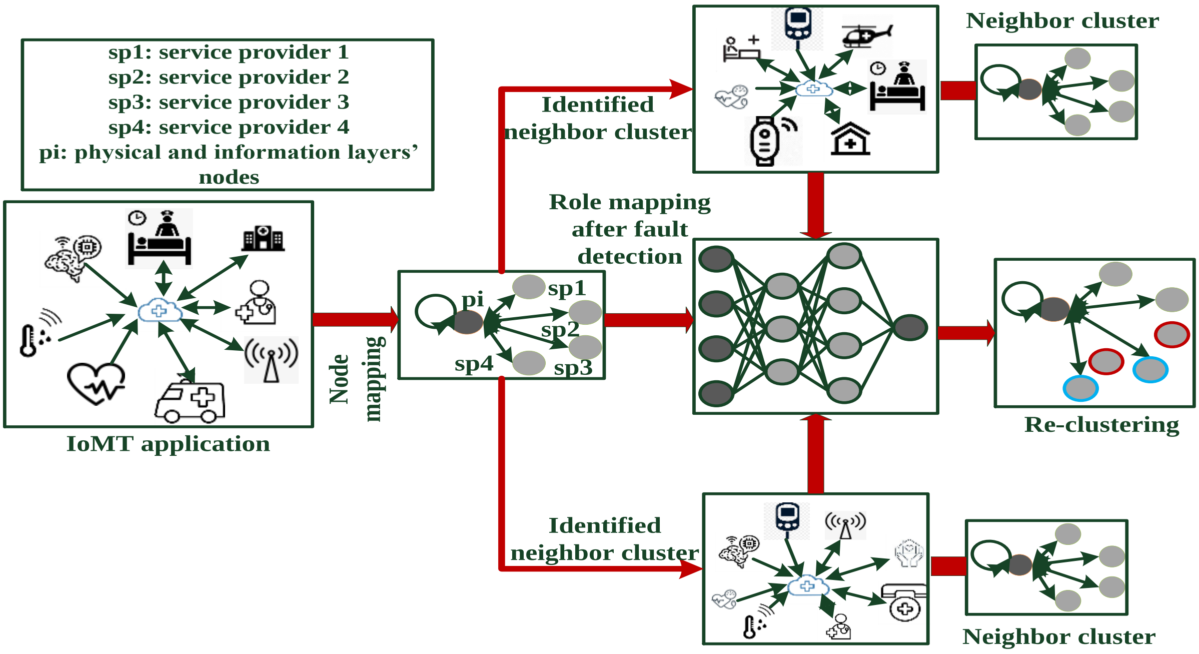


Figure 1. The steps of the study.

By presenting an overview of the proposed idea and pointing out the recent studies' challenges in the research domain, the main novelty and contribution of the study are as follows:

- Considering application priority to decision-making about dedicating a proper healthcare service to the requester
- Considering service priority to identify the appropriate neighbor nodes after detecting faults
- Analyzing the IoMT applications' efficiency and dependability based on the applications and services' priority

- Providing a method for flexible clustering a graph vertices and edges for improving security, availability, confidence, and privacy
- Providing the Equations and models to describe flexible clustering, node mapping, and estimating availability, security, privacy, and dependability
- Presenting node mapping method to transfer the application's service providers role onto a graph vertices
- Employing a quasi-mapping approach to transfer the role of graph's nodes onto a partial neural network's neurons to identify the appropriate service providers nodes for joining the cluster
- Pruning the unfit and defective nodes to improve dependability and performance (total delay and energy)

To assess the efficiency of the proposed algorithm and other contributions of the study, we define four healthcare-based applications with different priorities (availability, security, privacy, etc.) for implementing them by NFC. The study analyzes the IoMT applications, which have an impressive effect on medical science, and considers the risky conditions (such as facing the accident and pandemics a disease). The four applications consist of monitoring the vital signs of the climbers and investigating biosensors to detect the cancer COVID-19, about which we present more information in Section 3. This work reports the observations and simulation results after simulating the applications' node with the CupCarbon tool [26]. We estimate the applications' availability, security, privacy, and dependability by injecting the faults onto the simulated nodes with random and different rates using the CupCarbon tool.

The remainder of the manuscript is as follows. We review the recent case studies in the research domain in Section 2. The data collection methods, applications, Equations, and the proposed algorithm are defined and explained in Section 3. We then report the observations and simulation results in Section 4 and conclude about the presented concepts in Section 5.

2. Related Work

According to the rising number of IoT and IoMT applications, many studies addressed the technology's challenges and proposed approaches to tackle them and increase their efficiency. The section reviewed some research that applied different methods to support the security, trust, decision making, and other effective parameters to improve the quality of service in the IoT and Medical Things.

The security demands problems to IoMT-based infrastructure because of sharing the patients' private records between healthcare service providers, which some case studies addressed the issue to protect the information. Khan et al. [27] proposed a specific method to timely detect the attacks and decide about service providers, which had a higher potential for decision making than the machine learning-based algorithms. The study utilized the bidirectional simple recurrent units approach to speed up training the neural networks for timely detecting the complicated attacks on the IoMT. The restriction of IoMT devices' sources is challenging to protect privacy and security when researchers classify the different attacks and protocols to tackle the problem by considering the components' variants [28]. According to the role of IoMT application in detecting the various diseases, the researchers endeavored to satisfy the security to safety diagnosis in the medical-based applications, which the convolutional neural network (CNN) affected to support the purposes by speed up training and increasing precision [29].

The authentication protocols and methods helped to verify legitimate doctors and patients through sharing the private records and services, as the lack of identifying legitimacy endangers the people's health. Enamamu [30] designed a framework to authenticate the IoMT on different levels that utilized artificial intelligence to verify the output signals of the sensors as well as analyze the textual data. This work could identify the malicious and illegitimate signals and tackle them, whereas other authentication methods focused on detecting incorrect textual data. The researchers proposed different key agreements to support authentication in various levels of IoMT where the physical layer cannot utilize them because of its sensors' resources restriction. Lee et al. [31] presented an approach to verify the authentication in the physical layer and tackle the mentioned problem without defining the private key by employing the physically unclonable function.

Moreover, some case studies applied trust-based methods to decision making and provided dependable service to ensure receiving the correct HSs by the patients. With increasing the sensitivity of the shared data and information, defining a trust-based method is challenging; Jinila et al. [32] utilized an inverse method in facing the issue by providing a zero-trust security model. Blockchain technology has a highlighted role in protecting private records and satisfying the security of big data centers by detecting attacks and intruders and providing trust-based services [33,34]. Nevertheless, the trust-based methods create some limitations for the applications with specific situations, which lack of timely service endangers the patients' health.

Aside from providing security, authentication, and trust methods, service availability has a highlighted role in improving the IoMT's efficiency in supporting the requirements of the connected people to the technology. Some case studies utilized the blockchain to define the availability of the different IoMT applications that provided timely access to the data sources and services by predicting and managing them [35]. Nie et al. [36] focused on satisfying the availability of IoMT's physical layer and authenticating the private records via hash functions to profile matching. The study utilized the encryption algorithm and defined the private key to profile matching, which was allowed for users for secure-sharing of the records.

The researchers employed the IoMT infrastructure to improve the medical-based applications' efficiency in timely detecting diseases and providing the treatment methods by remote-controlling the patients' vital signs, such as diagnosing brain tumors and diabetes [37,38]. When the human's neural structures had a highlighted role in inspiring from their behavior in the neural networks, neuromorphic computing and spintronic detect the body language [39,40]. The studies evidenced the significant effect of machine learning algorithms and neural networks to improve the efficiency of IoMT-based applications with various properties. Table 1 presents an overview of the weaknesses and advantages of the mentioned works in comparing the proposed idea.

Table 1. Comparing the characteristics of the reviewed studies with the proposed idea.

Case Study	Method	Advantage	Weakness
Khan et al. [27]	- Machine learning algorithms - Recurrent units	- Timely decision-making - Detecting attacks	- Neglecting service priority - Fixed IoT nodes or applications
Rasool et al. [28]	- A contemporary review - Machine learning algorithm	- Supporting privacy in IoT - Supporting security in IoMT	- Neglecting service priority - Restrictions of the device's resources - Fixed IoT nodes or applications
Hossen et al. [29]	- Machine learning algorithms - Diagnosing diseases	- Timely detection of skin diseases - Joining the patient to IoMT	- Neglecting service priority - Fixed on the specific application - Algorithm complexity
Enamamu et al. [30]	- Artificial intelligent - Neural networks - Authentication protocol	- Detecting the legitimacy of the doctors in IoMT - Supporting security and privacy	- Focused on security - Neglecting service priority - Fixed IoT nodes or applications

Table 1. *Cont.*

Case Study	Method	Advantage	Weakness
Lee et al. [31]	- Key agreement - Neural networks - Authentication protocol	- Authentication on the physical level of IoMT	- Focused on security - Neglecting service priority - Fixed IoT nodes or applications
Bevish et al. [32]	- Zero trust method - Artificial intelligence	- Supporting trust in IoMT - Supporting dependability in IoMT	- Neglecting service priority - Fixed IoT nodes or applications
Samuel et al. [33]	- Blockchain - Artificial intelligent	- Supporting security in IoMT - Supporting privacy in IoMT	- Delay in timely decisionmaking - Neglecting service priority - Fixed IoT nodes or applications
Ali et al. [34]	- Blockchain - Artificial intelligent	- Supporting security in IoMT - Supporting privacy in IoMT	- Delay in timely decisionmaking - Neglecting service priority - Fixed IoT nodes or applications
Mubashar et al. [35]	- Blockchain - Artificial intelligent	- Timely access to data sources - Predicting and managing attacks - Supporting availability in IoMT	- Neglecting service priority - Fixed IoT nodes or applications
Nie et al. [36]	- Hash function - Profile matching - Encryption algorithm	- Supporting availability in the physical layer - Authenticating the private records	- Neglecting service priority - Fixed IoT nodes or applications
Kaur et al. [37]	- Artificial intelligent - Neural networks - Diagnosing diseases	- Joining the risky patients to the IoMT infrastructure - Timely detecting brain tumor - Timely providing treatment approach	- Neglecting service priority - Fixed on the specific application - Algorithm complexity
Reddy et al. [38]	- Artificial intelligent - Neural networks - Diagnosing diseases	- Joining the risky patients to the IoMT infrastructure - Timely diabetes detection - Timely providing treatment approach	- Neglecting service priority - Fixed on the specific application - Algorithm complexity
Sharifshazileh et al. [39]	- Neuromorphic systems - Artificial intelligence - Inspiring body languages	- Managing the people and risky condition	- Neglecting service priority - Fixed IoT nodes or applications - Dependability
Alon et al. [40]	- Artificial intelligence - Inspiring body languages	- Managing the people and risky condition	- Neglecting service priority - Fixed IoT nodes or applications - Dependability

Table 1. *Cont.*

Case Study	Method	Advantage	Weakness
This idea	- Flexible graph clustering - Neural network - Intelligent algorithm - Modeling the node clustering	- Timely decisionmaking - Service and application priorities - Supporting dependability, availability, security, and confidence in IoMT-based applications	- Timely identifying the stations from the vehicle health service providers in IoMT-based infrastructure

An overview of the related case studies to improve the IoMT's efficiency demonstrates targeting them to satisfy its security and dependability where the role of service priority and energy efficiency is invisible. Some researchers attended to manage the power supplies and dedicate them to the devices without considering the negative impact of the failed components on imposing energy consumption and delay overheads. Further, the studies focused on providing efficient protocols and methods for the cryptography of the shared information to support the IoMT-based application security and protect the private records, in which the application priority affects the improvement or deterioration of the approaches' efficiency. By utilizing the neural networks and clustering methods to categorize the proper service providers, the researchers can improve the prediction, detection, and decision-making methods in IoMT-based applications. This work aims to fill in the mentioned blanks of the related studies by presenting a model and algorithm to timely detect faulty service providers and cover their operations by employing the neighbor servers' nodes. We pay attention to the application and service priority for decision making about dedicating an appropriate service to the requester and clustering them to timely fault detection. This supports the flexibility of updating the graph vertices' initial statuses and joining new devices. The issue helps to tackle the challenges of the related energy consumption and delay to the failed or faulty nodes by pruning or stopping them. Apart from overcoming the performance problem, we improve the dependability and face the faulty nodes in risky and normal conditions based on the priorities of applications and services to increase the proposed idea's efficiency.

3. Flexible Cluster Modeling and NFC Algorithm

The study presents the Equation, model, and a four-phase intelligent algorithm to map the application's nodes onto graph vertices and cluster them after detecting defective service providers to identify the related neighbor servers to the application priority and join them to the cluster. To explain the proposed idea, we define four applications with different priorities (including application and service priority) and aim to support the risky conditions and recent research in medical science where collected a dataset for analyzing their efficiency. The section applies the mentioned concepts and issues of collecting datasets, describing Equations, node-mapping, and flexible-clustering algorithms.

3.1. Defining Applications and Collecting a Dataset

The subsection addresses to define the applications and explain them before describing the method for collecting a dataset in case of supporting the usages.

The defined application-1 includes covering the climbers in the mountains and tackling dangerous conditions (such as falling mountains and avalanches, occurring accidents, and disappearing the climbers) by monitoring their vital signs, locations, and weak signals, as shown in Figure 2a. We describe application-2 based on the biosensors' efficiency in diagnosing cancer and Alzheimer's by assessing the various types of proteins in people's blood, which Figure 2b demonstrates its nodes. Figure 2c shows application-3's nodes when describing detecting the infected people with COVID-19 and dedicating the emergency healthcare services to them. Due to the severe body sensitivities and allergies of the recovered people after infecting cancers, the study dedicates application-4 to monitoring their vital signs for timely identification of the risky conditions and providing the proper services. The application helps to prevent facing anaphylaxis shock by analyzing the Arterial Blood Gas Test (ABG) and coupled protein-G, which ABG consists of O_2, PH, and CO_2 [41–43].

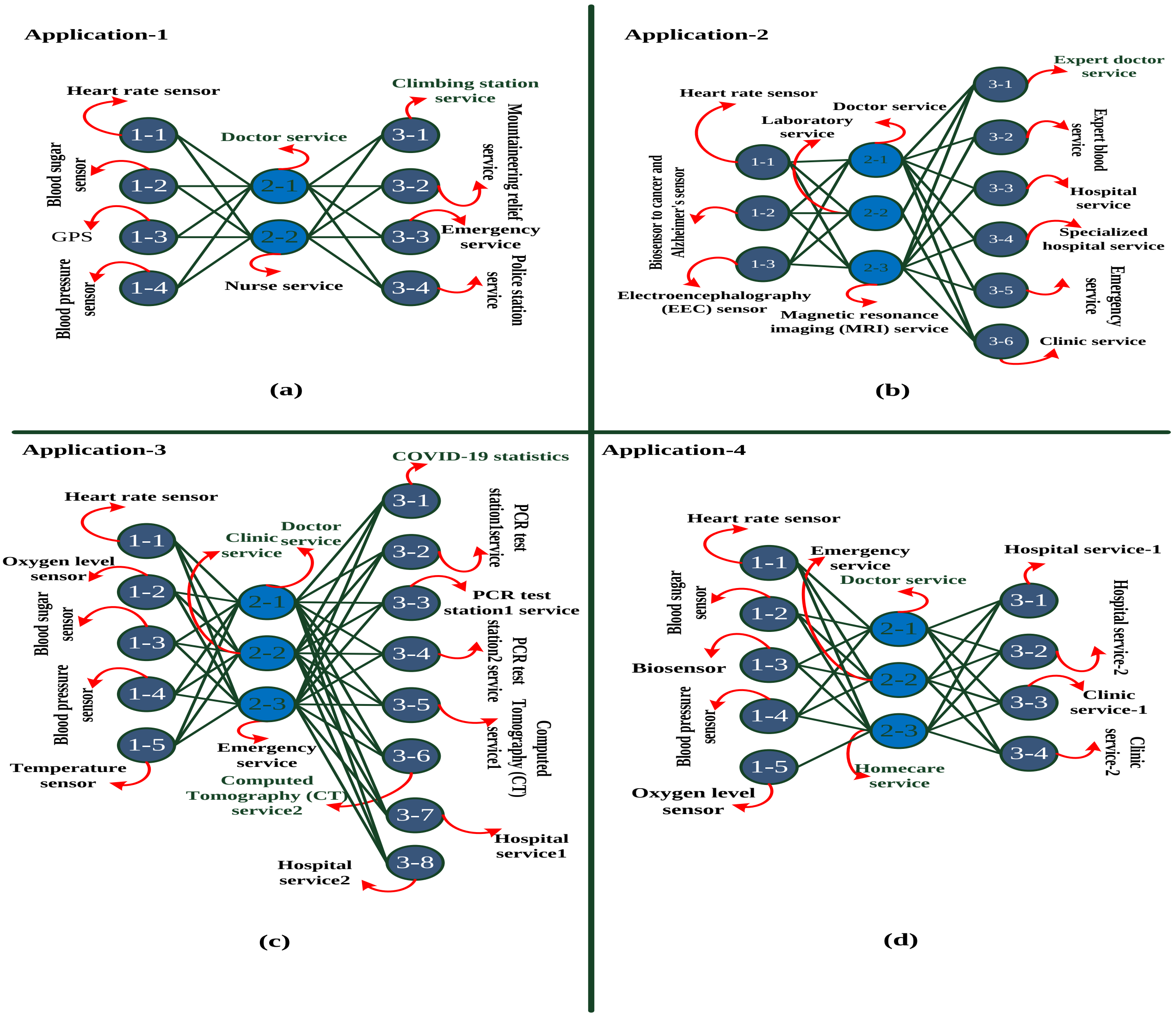


Figure 2. The defined IoMT-based applications: (**a**) Application-1 describes providing service to climbers; (**b**) Application-2 describes providing service to the identified people with cancer and Alzheimer's; (**c**) Application-3 describes providing service to patients with COVID-19; (**d**) Application-4 describes providing service to the identified persons with severe allergy.

According to the defined applications (1)–(4), we collect a dataset of the people's vital signs in normal and risky conditions and the healthcare services, which consists of the doctor's treatment recommendations, hospital, emergency, clinic services, etc. The study utilizes the shared valid statistics of the mountaineering stations and the people's vital signs (including the shared valid outputs of heart rate, blood pressure, sugar level, and location sensors) [44–46]. We monitor and analyze the outputs of biosensors and Electroencephalography (EEC) signals to collect a dataset of the physical and information layers for application-2 and register them, which were shared by valid references [47–51]. Due to application-2 and the proposed idea, the collected dataset also includes shared information about the related services to support the issue that covers the recommendations and diagnoses of the expert doctor, blood specialist, and hospital and clinic servers [52,53]. This work estimates the people's vital signs before and after being infected with COVID-19 and analyzes the appropriate healthcare services for them in dangerous and normal conditions. It utilizes the shared statistics of the disease by the shared credible medical references [54,55].

Moreover, we simulate the applications' nodes before and after performing NFC and pruning the defective components when analyzing their availability and security by injecting faults with random rates and registering them into the collected dataset (simulated by the CupCarbon tool). Figure 3 demonstrates the research method to collect a dataset, which we share on Github (https://github.com/yasamanhosseini/Flexible-clustering-IoMT-nodes, accessed on 20 July 2022) [56].

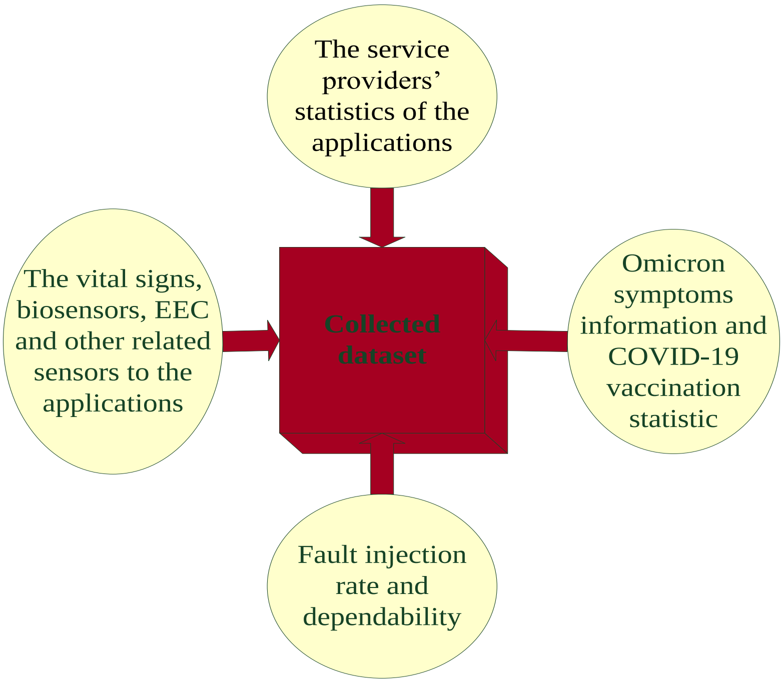
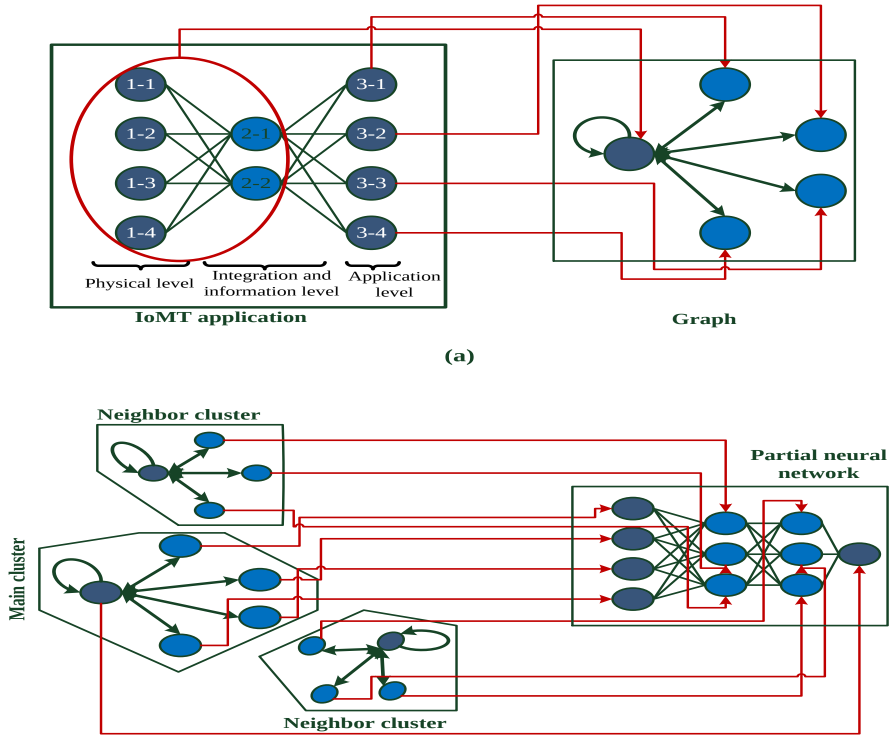

The service providers' statistics of the applications
The vital signs, biosensors, EEC and other related sensors to the applications
Collected dataset
Omicron symptoms information and COVID-19 vaccination statistic
Fault injection rate and dependability

Figure 3. Our method of data collection.

3.2. Presenting Equations and Models

We present Equations to describe node-mapping and flexible clustering and provide models to estimate availability, security, privacy, and dependability where the subsection addresses the mentioned issues and explains them.

Before addressing the Equations description, we indicate the process of mapping the application's nodes onto graph vertices and defining its edges based on the relationship between service providers and physical and information layers, as shown in Figure 4a. To identify the appropriate neighbor servers, the study employs a quasi-mapping method to transfer the role of graph vertices and neighbor service providers onto a partial neural network's neurons via training them with application priority; Figure 4b illustrates the mapping process. We utilize application-1 to indicate node and neural network mapping for clarifying the explanation of the mentioned processes and Equations, as shown in Figure 4a,b. The roles of physical and application layers' nodes are accumulated and mapped onto a vertex of the graph, which has a relationship with other vertices as the healthcare service providers at the application level. We utilize the role accumulation method to target the research on analyzing the service provider nodes and the relationship between them and other level components of the IoMT-based applications.



(a)

(b)

Figure 4. Node and role mapping methods: (**a**) Mapping the application nodes on the graph vertices; (**b**) Mapping the role of the graphs' vertices on a partial neural network's neurons (Quasi-mapping).

The study presents the parameters declaration and definitions of the employed variables in Equations (1)–(13) before their definition and explanation, which Table 2 demonstrates the description of the existing parameters.

Table 2. The existing parameters in equations.

Parameter	Description	Condition
A	Matrix of Graph A	—
U	Matrix of neural network's input layer	—
H	Matrix of neural network's hidden layer	—
G_A	Graph A based on the relationship between physical, information, and integration layers' nodes and healthcare service providers of an application	—
V	The vertices of Graph A	$0 \geq V$
E	The edges of Graph A	$0 \geq E$
i	The row number of matrix A	$0 \leq i \leq i_{max}$
j	Column number of matrix A	$0 \leq j \leq j_{max}$
p	The row number of matrix U	$0 \leq p \leq p_{max}$
q	Column number of matrix U	$0 \leq q \leq q_{max}$
y	The row number of matrix H	$0 \leq y \leq y_{max}$
z	Column number of matrix H	$0 \leq z \leq z_{max}$
pi	Physical, integration, and information layers' nodes of Graph A	—
sp	Healthcare service provider node of Graph A	—

Table 2. *Cont.*

Parameter	Description	Condition
n	The number of ahealthcare service providers	$0 \leq n \leq n_{max}$
s	The number of services of the neural network output	$0 \leq s \leq s_{max}$
$a_{i,j}$	Matrix A's element	($a_{i,j}$ = 0) OR ($a_{i,j}$ = 1)
$u_{p,q}$	Matrix U's element	($u_{p,q}$ = 0) OR ($u_{p,q}$ = 1)
$h_{y,z}$	Matrix H's element	($h_{y,z}$ = 0) OR ($h_{y,z}$ = 1)
i_{max}	The maximum number of matrix A's rows	$0 \geq i_{max}$
j_{max}	The maximum number of matrix A's columns	$0 \geq j_{max}$
p_{max}	The maximum number of matrix U's rows	$0 \geq p_{max}$
q_{max}	The maximum number of matrix U's columns	$0 \geq q_{max}$
y_{max}	The maximum number of matrix H's rows	$0 \geq y_{max}$
z_{max}	The maximum number of matrix H's columns	$0 \geq z_{max}$
n_{max}	The maximum number of a healthcare service provider	$0 \geq n_{max}$
s_{max}	The maximum number of services of the neural network output	$0 \geq s_{max}$
d	The total number of hidden layer's weights	$0 \geq d$
l	The total number of output layer's weights	$0 \geq 1$
GraphCondition	The relationship between service providers, physical and integration, and information layers nodes	$0 \geq$ *GraphCondition*
XNumber	The threshold number of defective nodes	$0 \geq$ *XNumber*
XjoinedNode	The number of newly joined nodes	$0 \geq$ *XjoinedNode*
Threshold	The threshold value of application validation based on priority	$0 \geq$ *Threshold*
FaultDetection	Detecting faults in the service providers	$0 \geq$ *FaultDetection*
ApplicationSituation	Investigating the IoMT application's situation in fault detection	(*ApplicationSituation* = 0) OR (*ApplicationSituation* = 1)
NeuronMapping	Mapping Graph's nodes onto a neural network neurons	—
OutputN	Output neuron	$0 \leq$ *OutputN* ≤ 1
Failedapplication	The situation of faulty service providers	(*Failedapplication* = 0) OR (*Failedapplication* = 1)
joinednode	The situation of new joined nodes	(*joinednode* = 0) OR (*joinednode* = 1)
Fault	The fault number	$0 \leq Fault \leq Fault_{max}$
$Fault_{max}$	The maximum number of fault	$0 \geq Fault_{max}$
$Penalty_{Fault}$	The penalty for a fault	$0 \geq Penalty_{Fault}$
ThresholdNN	The threshold value of the neural network's output	$0 \geq$ *ThresholdNN*
$Service_{s_t}$	The service with s number at t time	$0 \geq Service_{s_t}$
$WeightNN_{hidden_d}$	Hidden layer's weights' values with d index	—
$WeightHL_{intial_d}$	The initial value of the hidden layer's weights	—
$WeightNN_{output_l}$	Output layer's weights' values with l index	—
$WeightOL_{intial_l}$	The initial value of the output layer's weights	—
ServiceAvailability	Service availability of the IoMT application	$0 \leq$ *ServiceAvailability* ≤ 1
$OutputAvailability_t$	The availability of output neuron at t time	$0 \leq$ *OutputAvailability* ≤ 1
$ServiceSecurity_{pi}$	IoMT application security based on pi	$0 \leq ServiceSecurity_{pi} \leq 1$
$CorrectService_t$	The number of received correct services at t time	—
$TotalService_t$	The total number of received services at t time	—

Table 2. *Cont.*

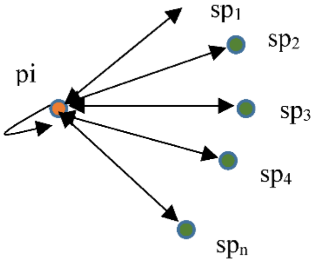
Parameter	Description	Condition
ServicePrivacy	A definition of application privacy is based on the relationship between its nodes and security	—
E_n	The related edge of Graph A to service provider n	—
α	Impact factor of availability on dependability	$0 \leq \alpha \leq 1$
β	Impact factor of security on dependability	$0 \leq \beta \leq 1$
μ	Impact factor of confidence on dependability	$0 \leq \mu \leq 1$
Dependability	IoMT application dependability	$0 \geq$ *Dependability*

To clarify the introduced parameters in Table 1 and the descriptions of Equations (1)–(13), we provide some definitions as follows:

1. $[A]_{i\times j}$ *demonstrates the matrix of the relationship between vertices of* G_A *= (V, E), where i and j are indexes of rows and columns number of the matrix.*
2. G_A *= (V, E) consists of V vertices and E edges based on the number of the application's nodes (Defining vertices) and the relationship between them (Defining edges) where (V = i) and (E = j), and (i = j = n) by describing n as the number service providers and pi node (as the accumulated node of the physical and information layers' components).*
3. *GraphCondition estimates the sum of* $[A]_{i\times j}$ *matrix's elements to investigate the relationship between service providers' vertices (sp) and pi for utilizing its value to detect fault where* $a_{i,j}$ *describes the located element of the matrix in I row and j column.*
4. *Threshold defines a value for each application based on service and their priority for deciding to employ the neighbor servers after detecting faults. Xnumber and* sp_n *demonstrate the threshold number of defective service providers and the total number of the initial servers of the application.*
5. *FaultDetection estimates a value to detect fault based on the sum of* $[A]_{i\times j}$ *matrix's elements (GraphCondition) and the expected value of the GraphCondition parameter, which is evaluated by* $(i_{max} + j_{max}) - 1$ *as the maximum number of rows and columns.*
6. *ApplicationSituation determines the situation of the application after detecting fault based on the estimated threshold's value by defining the FailedApplication parameter and quantifying it.*
7. *ClusteringVertices categorizes the vertices of* G_A *(A graph) to cluster them based on two conditions of normal and faulty situations when identifying the application is as correct (Initial clustering) and facing failed application or joining a new service provider (Re-clustering) to application, which the joined node describes the number of new joined servers to application.*
8. $[U]_{p\times q}$ *describes a matrix to define the neurons of a neural network input layer where p and q determine the number of rows and columns of the* $[U]_{p\times q}$ *matrix.*
9. $[H]_{y\times z}$ *is a matrix to describe the neurons of a NN's hidden layer with y rows and z columns.*
10. *NeuronMapping describes mapping the initial service provider, neighbor servers, and pi (Accumulated physical and information levels' nodes) nodes onto the input, hidden, and output layers' neurons, where* $u_{p,q}$ *and* $h_{y,z}$ *are the elements of* $[U]_{p\times q}$ *and* $[H]_{y\times z}$ *matrices.*
11. $Penalty_{Fault}$ *estimates the penalty value based on the threshold and output values and the total number of services at t time, where* $OutputN_t$ *and ThresholdNN indicate the output neuron's value at t time and the estimated threshold value of the neural network.*
12. $WeightNN_{hidden_d}$ *computes the value of the hidden layer's weight after imposing the estimated penalty value onto them by* $WeightHL_{intial_d}$ *(the initial value of the hidden layer's weight) and* $Penalty_{Fault_t}$ *values where d and Fault are the indexes for the weight and the related penalty to the detected fault.*
13. $WeightNN_{output_l}$ *demonstrates estimating the value of the output layer's weights at the time after imposing the computed penalty onto them where* $WeightOL_{intial_l}$ *and* $Penalty_{Fault_t}$ *illustrate the values of the initial output weight and penalty with d and Fault indexes, respectively.*

14. *ServiceAvailability estimates the application availability at t time based on analyzing the output neuron's value after pruning weak weights and neurons where OutputAvailable demonstrates the availability of the output neuron.*
15. *ServiceSecurity assesses the application security based on the number of correct services received ($CorrectService_t$) and the total number of services ($TotalService_t$) at t time.*
16. *ServicePrivacy analyzes the application's privacy by defining the edges between G_A vertices and removing the related connections to the faulty nodes where $ServiceSecurity_{pi}$ describes the security of the pi vertex.*
17. *Dependability estimates the application dependability where α, β, and μ are the impact factors of availability, security, and confidence.*

According to the above mentioned definitions, we describe analyzing the graph situation by presenting Equation (1). The service providers' vertices of the GA are updated, including the *sp* (1)–(*n*) when detecting the faults and joining the servers to the application by defining or removing the edges between the graph nodes. To clarify the Equations' descriptions, we use the presented application-1 to define them.



$$A = \begin{array}{cc} & \begin{array}{ccccccc} pi & sp_1 & sp_2 & sp_3 & sp_4 & \cdots & sp_n \end{array} \\ \begin{array}{c} pi \\ sp_1 \\ sp_2 \\ sp_3 \\ sp_4 \\ \vdots \\ sp_n \end{array} & \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & \dots & 1 \\ 1 & 0 & 0 & 0 & 0 & \dots & 0 \\ 1 & 0 & 0 & 0 & 0 & \dots & 0 \\ 1 & 0 & 0 & 0 & 0 & \dots & 0 \\ 1 & 0 & 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \dots & \vdots \\ 1 & 0 & 0 & 0 & 0 & \dots & 0 \end{bmatrix} \end{array} \quad (1)$$

$$G = (V, E)$$

$$A = \begin{bmatrix} a_{0,0} & \cdots & a_{0,j} \\ \vdots & \vdots & \vdots \\ a_{i,0} & \cdots & a_{i,j} \end{bmatrix} \qquad [A]_{i \times j} = \begin{cases} i = n \\ i = j \end{cases} \qquad G_A = (n, n)$$

$$GraphCondition = \sum_{i=0}^{i=i_{max}} \sum_{j=0}^{j=j_{max}} a_{i,j}$$

Equation (2) demonstrates estimating a threshold value for each application, which is determined by analyzing the application and service priority, and considering the total number of servers. Due to the equation, we define the specific threshold values for the IoMT applications based on the threshold number of defective nodes when facing more leads to a failure.

$$Threshold = \frac{XNumber}{\sum_{n=0}^{n=n_{max}} sp_n} \quad (2)$$

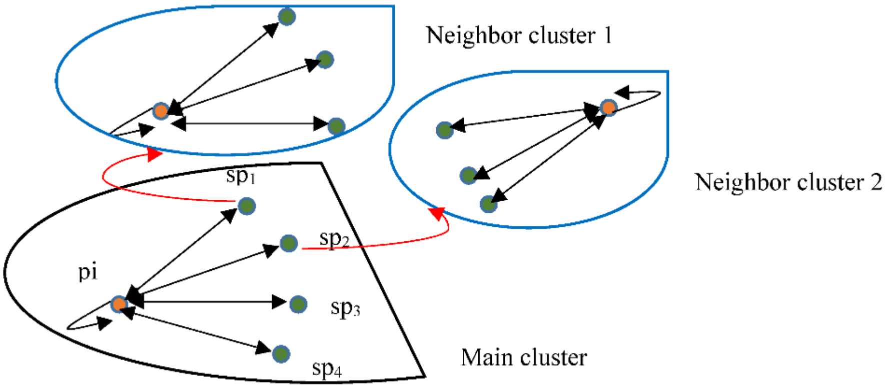
By presenting Equation (3), the study detects the faults on the related graph vertices to the mapped service providers' nodes where $(i_{max} + j_{max}) - 1$ demonstrates the expected edges (connections between *sp* and *pi* nodes).

$$FaultDetection = \frac{GraphCondition}{(i_{max} + j_{max}) - 1} \times Threshold \quad (3)$$

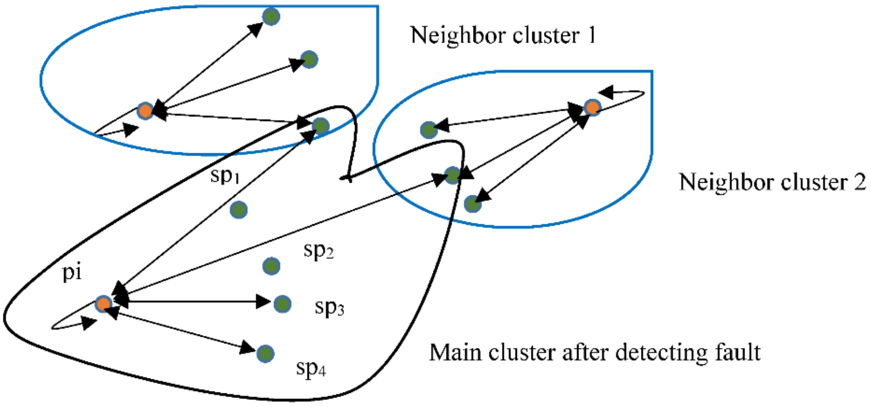
We present Equations (4) and (5) to determine the application situations and cluster the graph vertices based on them, which consists of correct ($Failedapplication = 0$) and failed ($Failedapplication = 1$) statuses by meeting $(FaultDetection \simeq Threshold)$ $\vee$

$(FaultDetection > Threshold)$ and $(FaultDetection < Threshold)$, respectively. After determining the situation of the application, this work categorizes the graph vertices for the initial and renewed clustering them where the conditions of $(Failedapplication = 0) \wedge (Joinednode = 0)$ and $(Failedapplication = 1) \vee (Joinednode = 1)$ are met, respectively.

$$ApplicationSituation = \begin{cases} Failedapplication = 0\ , & (FaultDetection \simeq Threshold) \vee (FaultDetection > Threshold) \\ Failedapplication = 1\ , & (FaultDetection < Threshold) \end{cases} \tag{4}$$



$$ClusteringVertices = \begin{cases} \sum_{t=0}^{t=t_{max}} \sum_{n=0}^{n=n_{max}} (V_{n_t} + 1), & (Failedapplication = 0) \wedge (Joinednode = 0) \\ \sum_{t=0}^{t=t_{max}} \left(\sum_{n=0}^{n=n_{max}} (V_{n_t} + 1) + (XNumber - 1) + (XjoinedNode) \right), & (Failedapplication = 1) \vee (Joinednode = 1) \end{cases} \tag{5}$$



To identify the proper service providers in the neighbor clusters after detecting faults on the application nodes, we utilize quasi-mapping methods to transfer the role of graph vertices onto a partial neural network's neurons by presenting Equation (6). The main and neighbor service providers are mapped onto input and hidden layers' neurons via meting $(0 \leq i \leq i_{max}) \wedge (1 \leq j \leq j_{max})$ and $(i_{max} < i \leq i_{max+XNumber}) \wedge (j_{max} < j \leq j_{max+XNumber})$ conditions.

$$U = \begin{bmatrix} u_{0,0} & \cdots & u_{0,q} \\ \vdots & \vdots & \vdots \\ u_{p,0} & \cdots & u_{p,q} \end{bmatrix} \qquad H = \begin{bmatrix} h_{0,0} & \cdots & h_{0,z} \\ \vdots & \vdots & \vdots \\ h_{y,0} & \cdots & h_{y,z} \end{bmatrix}$$

$$NeuronMapping = \begin{cases} u_{p,q} = a_{i,j}, \ (0 \leq i \leq i_{max}) \wedge (1 \leq j \leq j_{max}) \\ h_{y,z} = a_{i,j}, \ (i_{max} < i \leq i_{max+XNumber}) \wedge (j_{max} < j \leq i_{max+XNumber}) \\ OutputN = a_{0,0} \end{cases} \tag{6}$$

Equation (7) illustrates computing a threshold value for analyzing the situation of the output neuron and deciding to impose a penalty on the trained initial weights after mapping the partial neural network.

$$Penalty_{Fault} = \frac{\sum_{t=0}^{t=t_{max}} (OutputN_t - ThresholdNN)}{\sum_{s=0}^{s=s_{max}} (Service_{s_t} + 1)} \quad , \qquad 0 \leq t \leq t_{max} \tag{7}$$

We impose the estimated value of the related penalty to the detected fault into the initial weights of the hidden and output layers by presenting Equations (8) and (9).

$$\begin{aligned} &d = p_{max} \times q_{max} \\ &l = y_{max} \times z_{max} \\ &WeightNN_{hidden_d} = WeightHL_{intial_d} - \textstyle\sum_{Fault=0}^{Fault=Fault_{max}} Penalty_{Fault_t} \, , \; 0 \leq t \leq t_{max} \end{aligned} \tag{8}$$

$$WeightNN_{output_l} = WeightOL_{intial_l} - \textstyle\sum_{Fault=0}^{Fault=Fault_{max}} Penalty_{Fault_t}, \; 0 \leq t \leq t_{max} \tag{9}$$

To assess the application availability, the study presents Equation (10) to appraise the parameter for service providers based on analyzing the output neurons' values and the number of unavailable servers.

$$ServiceAvailability = \frac{1}{\sum_{t=0}^{t=t_{max}} (1 - OutputAvailable)_t} \tag{10}$$

Equations (11) and (12) are provided to compute and evaluate services security and privacy to estimate the mentioned characteristics of the application by investigating the output neuron's values, and the number of the received correct services at t time. We define and estimate the application privacy by removing the related edges to the detected defective service providers' nodes.

$$ServiceSecurity_{pi} = \frac{\sum_{t=0}^{t=t_{max}} CorrectService_t}{TotalService_t} \tag{11}$$

$$ServicePrivacy = \begin{cases} E_n : pi \leftarrow sp_n, & ServiceSecurity_{pi} \simeq 1 \\ E_n : pi \leftarrow 0, & ServiceSecurity_{pi} \neq 1 \end{cases} \tag{12}$$

Application dependability parameter depends on availability, security, and confidence with the different impact factors (α, β, and μ), which directly relate to service priority, as shown in Equation (13). To determine the mentioned impact factors, we define the weights that are demonstrated the role of service availability, security, and confidence based on application priorities. Equation (13) evaluates the application dependability after applying the values of α, β, and μ.

$$\begin{gathered} (\alpha + \beta + \mu) = 1 \\ Dependability = \alpha(ServiceAvailability) + \beta(ServiceSecurity) + \mu\left(\frac{\sum_{t=0}^{t=t_{max}} ConfidenceService}{TotalService_t}\right) \end{gathered} \tag{13}$$

3.3. Node Mapping and Flexible Clustering Algorithm

The case study presents a four-phase intelligent algorithm to map the application's nodes onto graph vertices and transfer their role onto the neural network's neurons for identifying the appropriate service providers in the neighbor clusters by applying the service priority to the NN's weights. After mapping the graph nodes and defining their edges based on the relationship between service providers and other application components, NFC categorizes the vertices for clustering them to timely fault detection and tackle them. The algorithm's four phases cover node mapping onto the graph vertices and initial clustering them, role transferring onto a partial NN's neurons, detecting the defective service providers, and training neural network and re-clustering. To clarify explaining

NFC's operations, we utilize application-1 and declare the algorithm inputs, outputs, and parameters where Algorithm 1 demonstrates the mentioned descriptions. Application-1 consists of four service providers (*sp*) and five physical and information (*pi*) layers' nodes, which are employed the characteristics to define the index number of the graph's vertices, as shown in Algorithm 1.

```
Algorithm 1 Node mapping and flexible clustering (NFC)
Inputs:      1-PIS: The node's values of the Physical and information and integration layers
             (HeartRateSensor, SugarLevelSensor, BloodPressureSensor, GPSSensor,
             DoctorService, NurseService) whereas PIS $\subset$ Natural number
             2-HSP: The nodes' value of healthcare service providers (ClimbingStationService,
             MauntaieeringReliefService, EmergencyService, PoliceStationService)
Output:      ApplicationDependability: The output's value for analyzing the application
             dependability
             whereas $0 \leq$ ApplicationDependability $\leq 1$
Parameter declaration:
$PI$: Physical sensor and integration service = {HeartRate, SugarLevel, BloodPressure, GPS, Doctor, Nurse}
$SP$: ={ClimbingStation, MountaineeringRelief, Emergency, PoliceStation}
$pi$: ={0, 1, . . . , 4, 5}
$s$: ={1, 2, 3, 4}
$i$: Row index = {0, 1, 2, 3}
$j$: Column index = {0, 1, 2, 3}
$nc$: Neighbor cluste = {0, 1, 2}
$p$: ={0, 1, . . . , 4, 5}
$q$: ={0, 1, . . . , 4, 5}
$x$: Neighbor node whereas $x \subset$ Natural number and $x \geq 0$
$y$: Joined node whereas $y \subset$ Natural number and $y \geq 0$
$t$: Available time whereas $t \subset$ Natural number and $t \geq 0$
$\alpha$: The impact factor of IoMT application availability whereas $\alpha \subset$ Binary number
$\beta$: The impact factor of IoMT application security whereas $\beta \subset$ Binary number
$\mu$: The impact factor of IoMT application confidence whereas $\mu \subset$ Binary number
```

In phase 1, NFC first maps and accumulates the role of physical and information layers' nodes to a graph's vertex for addressing the relationship between it and service providers to timely identify the defective servers. After mapping the application's nodes onto graph vertices and defining its edges, we categorize the primary nodes for initial clustering to simplify analyzing the efficiency and detecting faults. The algorithm identifies the new joined nodes to the application and relevant service providers of the neighbor clusters to the faulty devices where (NewJoinedNode = True) and (CreatConnetionBetweenGraphNode0AndGraphNodes = True) are met, as shown in Algorithm 2.

Phase 2 supports the neural network mapping operations to transfer the role of graph vertices onto the NN neurons to identify the appropriate neighbor service providers and decide about dedicating or joining them to the cluster after detecting faulty components. Algorithm 3 demonstrates the quasi-mapping operations of phase 2 of the proposed algorithm. We map the new joined components to the application and neighbor service providers onto the hidden layer neurons, which the role of physical and information levels' nodes (*pi* vertex) for transferring to the output neuron. NFC transfers the role of the essential service providers' vertices of the graph onto the input layer's neurons of a partial neural network where (InputLayerNeuron) $_{i,j} \leftarrow$ (GraphNode)$_{((3\times j)+i)+1}$ operation is performed.

In phase 3, the algorithm addresses detecting faulty service providers and identifying the proper neighbor components to re-clustering the graph vertices and cover the operations of failed nodes, as shown in Algorithm 4. After mapping the neurons and detecting faulty nodes, NFC starts training the neural network based on the application and service priority by applying them and estimated penalty values into the NN's weights when (FaultDetection < Threshold) is met. By imposing the penalty values into the weights and re-training the neural network, we prune the weak neurons and weights, which re-cluster the graph vertices based on the joined or removed nodes by fulfilling the (NewJoinedNode = True) condition.

Algorithm 2 NFC's phase 1

```
Start
    Procedure RPD (PIS, HSP)
        Start phase 1 // Node mapping onto Graph vertices and main clustering
            do in parallel
                for each pi in PI index do // Physical and integration layers' nodes mapping on a Graph vertices
                    (GraphNode) $_0$ ← (PIS) $pi$
                    (GraphNode) $_0$ ← Accumulated (Graph node) $_0$// Creating main cluster of Graph nodes
                    (MatrixGraphNode) $_{0,0}$ ← 1
                end for;
                (ClusteredNode) $_0$ ← (GraphNode) $_0$
                for each s in SP index do // Healthcare service providers' nodes mapping onto a Graph vertices
                    (GraphNode) $_s$ ← (HSP) $_{s-1}$
                    (ClusteredNode) $_s$ ← (GraphNode) $_s$ // Creating main cluster of Graph nodes
                    If (CreatConnetionBetweenGraphNode0AndGraphNodes = True) do
                        (MatrixGraphNode) $_{s,s}$ ← 1
                        else
                        (MatrixGraphNode) $_{s,s}$ ← 0
                    end if;
                end for;
            end parallel;
            do in parallel
                for each nc in Neighbor cluster do // Identifying neighbor healthcare service provider nodes
                    Identifying new neighbor node in the neighbor clusters
                    (NeighborAndJoinedNode) $_x$ ← (IdentifiedNode) $_x$
                    $x \leftarrow x + 1$
                end for;
                if (NewJoinedNode = True) do // Identifying new joined healthcare service provider nodes
                    (NeighborAndJoinedNode) $_{x+1}$ ← (JoinedNode) $_y$
                    $y \leftarrow y + 1$
                end if;
            end parallel;
        end phase 1
```

Algorithm 3 NFC's phase 2

```
Start phase 2 // Graph's nodes mapping onto a neural network (NN)'s neurons
    do in parallel
        for each i in Row index do // Mapping the main service provider nodes of the application onto NN input layer's nodes
            for each j in Column index do
                (InputLayerNeuron) $_{i,j}$ ← (GraphNode)$_{((3\times j)+i)+1}$
            end for;
        end for;
        while ((NeighborAndJoinedNode) ≠ 0) do
            for each p in Hidden row index do // Mapping the neighbor and new joined service provider nodes onto NN hidden layer's nodes
                for each q in Hidden column index do
                    (HiddentLayerNeuron) $_{p,q}$ ← (NeighborAndJoinedNode)$_{((5\times q)+p)}$
                end for;
            end for;
        end while;
        (OutputNeuron) ← (GraphNode) $_{0,0}$
    end parallel;
end phase 2
```

Due to the phase 4 operations, we estimate the application dependability by evaluating availability, security, and confidence, which depend on service priority. NFC investigates the application and service priority to assess the availability, security, and confidence by defining the different impact factors of α, β, and μ when Algorithm 5 indicates the mentioned operations of phase 4. The algorithm starts computing the application's availability and security based on Equations (10) and (11) where ((ApplicationPriority = ServiceAvailability), (Timely receiving service degree $\leq$ Threshold value)), ((ApplicationPriority = ServiceSecurity), and (Timely receiving service degree $\leq$ Threshold value)) are fulfilled. By meeting the (ApplicationPriority $\neq$ Confidence) condition, NFC resets the related impact factor to the confidence for deactivating the parameter in estimating the application dependability.

To clarify the NFC operations and explanations, we try to illustrate pruning the faulty nodes and re-clustering the graph vertices based on the joined and removed service providers by defining the four hypotheses and implementing them in the applications (1)–(4). Figures 5–8 show the four hypotheses that are implemented in the applications (1)–(4) by considering their priority in analyzing their dependability. According to the algorithm operations, we first determine the failure situation for application-1, prune the defective service providers, and re-clustering the graph vertices by analyzing the threshold value and identifying the appropriate neighbor node, as shown in Figure 5a,b. Its application priority is availability that NFC identifies the available neighbor service provider to join the cluster and cover the defective components by training the neural network and pruning the irrelevant server.

Algorithm 4 NFC's phase 3

```
Start phase 3 // Fault detection, neural network training, and re-clustering Graph nodes
    (GraphSituation) ← $\sum_{i=0}^{i=4}\sum_{j=0}^{j=4}(MatrixGraph)_{i,j}$
    Estimating Threshold value based on Equation (2)
    (FaultDetection) ← $\left(\frac{\text{GraphSituation}}{(i_{max}+j_{max})-1}\right)$ × Threshold // Fault detection
    if ((FaultDetection < Thershold)) do // Determining application situation
        Training the mapped neural network based on application and service priority
        Estimating penalty values based on Equation (7)
        do in parallel
            for each i in Row index do // Imposing penalty into neural network weights based on backwarding chain
                for each j in Column index do
                    (WeighHiddenLayer) $_{i\times j}$ ← $(\text{WeighHiddenLayer})_{\text{initial}} - \sum_{\substack{\text{Fault}=0\\ t=o}}^{\substack{\text{Fault}=\text{Fault}_{\max}\\ t=t_{\max}}} \text{Penalty}_{\text{Fault}_t}$
                end for;
            end for;
            for each p in Hidden row index do // Imposing penalty into neural network weights based on backwarding chain
                for each q in Hidden column index do
                    (WeighOutputLayer) $_{p\times q}$ ← $(\text{WeighOutputLayer})_{\text{initial}} - \sum_{\substack{\text{Fault}=0\\ t=o}}^{\substack{\text{Fault}=\text{Fault}_{\max}\\ t=t_{\max}}} \text{Penalty}_{\text{Fault}_t}$
                end for;
            end for;
        end parallel;
        Pruning weak neurons and weights
        if ((NewJoinedNode = True)) do // Re-clustering based on joining neighbor and new nodes
            Cluster ← $\sum_{t=0}^{t=t_{\max}}\left(\sum_{s=1}^{s=4}\left(V_{s_t}+1\right)+(\text{NeighborAndJoinedNode})-1)+(\text{NewJoinedNode})\right)$
            else
            Cluster ← $\sum_{t=0}^{t=t_{\max}}\left(\sum_{s=1}^{s=4}\left(V_{s_t}+1\right)+(\text{NeighborAndJoinedNode})-1)\right)$ // Re-clustering based on neighbor nodes
        end if;
    end if;
end phase 3
```

Algorithm 5 NFC's phase 4

```
Start phase 4 // Estimating dependability of IoMT application based on service and application priority
    Analyzing the priority of IoMT application and service
    Estimating the threshold value of timely receiving service by OutputNeuron
    if ((ApplicationPriority = ServiceAvailability) and (Timely receiving service degree ≤ Threshold value)) do
        $\alpha \leftarrow 1$
        (ServiceAvailability) ← $\frac{1}{\sum_{t=0}^{t=t_{\max}}(1 - \text{OutputAvailable})_t}$ // Estimating service availability based on Equation (10)
    end if;
    if ((ApplicationPriority = ServiceSecurity) and (Timely receiving service degree ≤ Threshold value)) do
        $\beta \leftarrow 1$
        (ServiceSecurity) ← $\frac{\sum_{t=0}^{t=t_{\max}} \text{CorrectService}_t}{\text{TotalService}_t}$ // Estimating service security based on Equation (11)
    end if;
    if (ApplicationPriority ≠ Confidence) do
        $\mu \leftarrow 0$
    end if;
    $ApplicationDependability \leftarrow \alpha(ServiceAvailability) + \beta(ServiceSecurity) + \mu\left(\frac{\sum_{t=0}^{t=t_{max}} ConfidenceService}{TotalService_t}\right)$
    Normalizing ApplicationDependability value whereas 0 ≤ ServiceDependability ≤ 1
    Updating the nodes and edges of main cluster
end phase 4
end
```

By evaluating application-2 priority, the algorithm starts applying service priority, imposing the penalty value onto the neural network weights, and re-training it to decide the appropriate neighbor servers after detecting faulty nodes and failure situations. Security is defined as the application-2 priority because of the risky conditions of the patients with Alzheimer's and cancer to receive a secure healthcare service. We prune the weak neurons and weights to identify the secure neighbor servers and re-cluster the graph vertices after passing the mentioned operations when Figure 6a,b illustrate the fault hypothesis for application-2.

Due to the estimated threshold value to application-3 and its priority, NFC identifies the neighbor service providers, covering the security and availability by imposing penalties on the weights and re-training the neural network. The algorithm starts the re-clustering phase due to the fault hypothesis and passes the condition of detecting the failure situation, as shown in Figure 7a,b. Application-4 supports the patients with COVID-19, in which the priority consists of the security and availability because of facing the risky conditions of the infected people and the high prevalence rate of the disease.

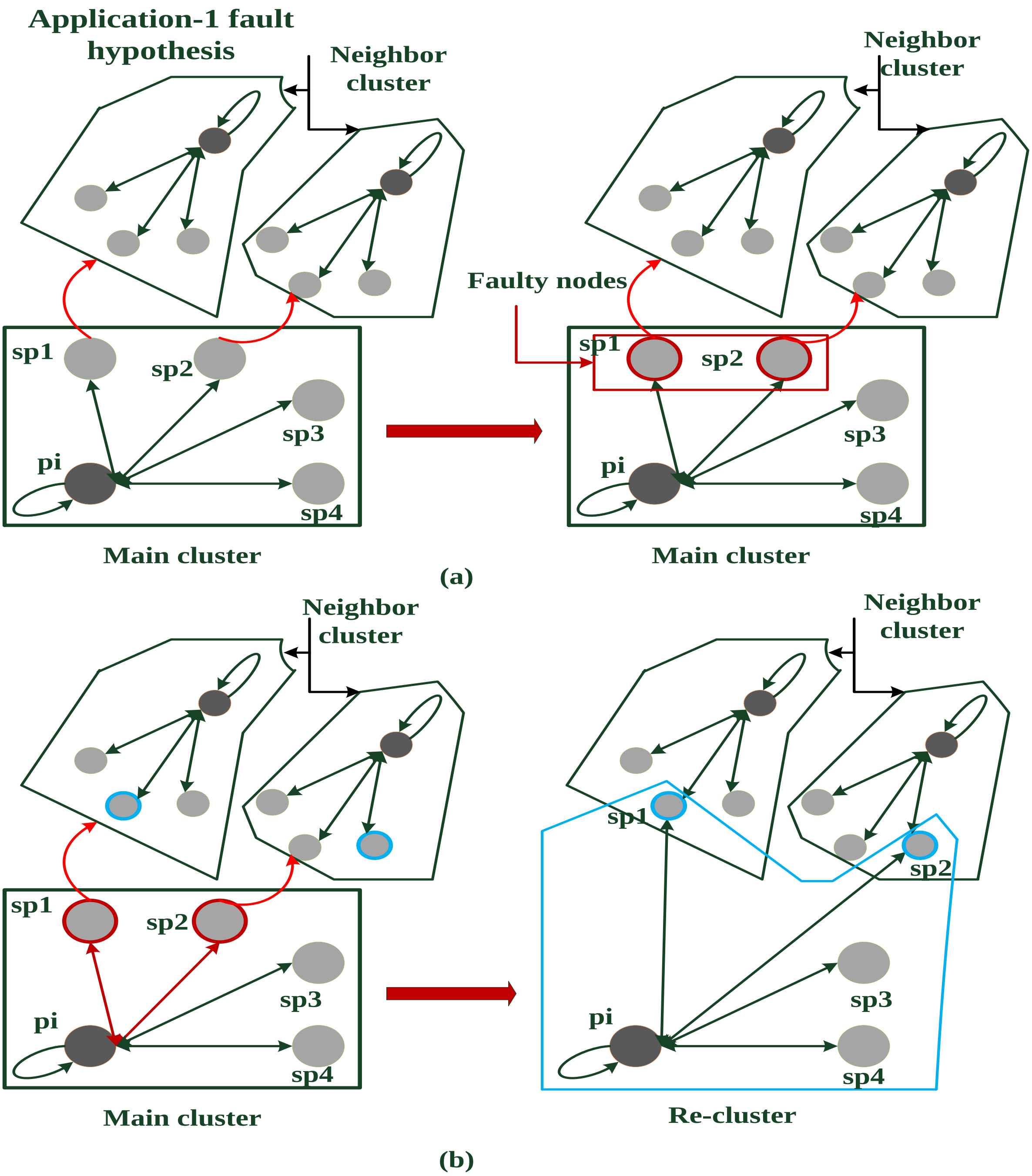


Figure 5. Fault hypothesis for application-1: (**a**) Detecting faulty nodes and identifying the failure situation based on threshold value; (**b**) Identifying the proper neighbor service providers based on application priority and re-clustering vertices.

Figure 8a,b indicate the fault hypothesis for application-4 and the phases-passing of NFC after detecting the faulty service providers and re-clustering the graph vertices. The availability characteristic is the application-4 priority by considering the dangerous and specific conditions of the recovered patients with cancer and their severe allergies and weak immune systems.

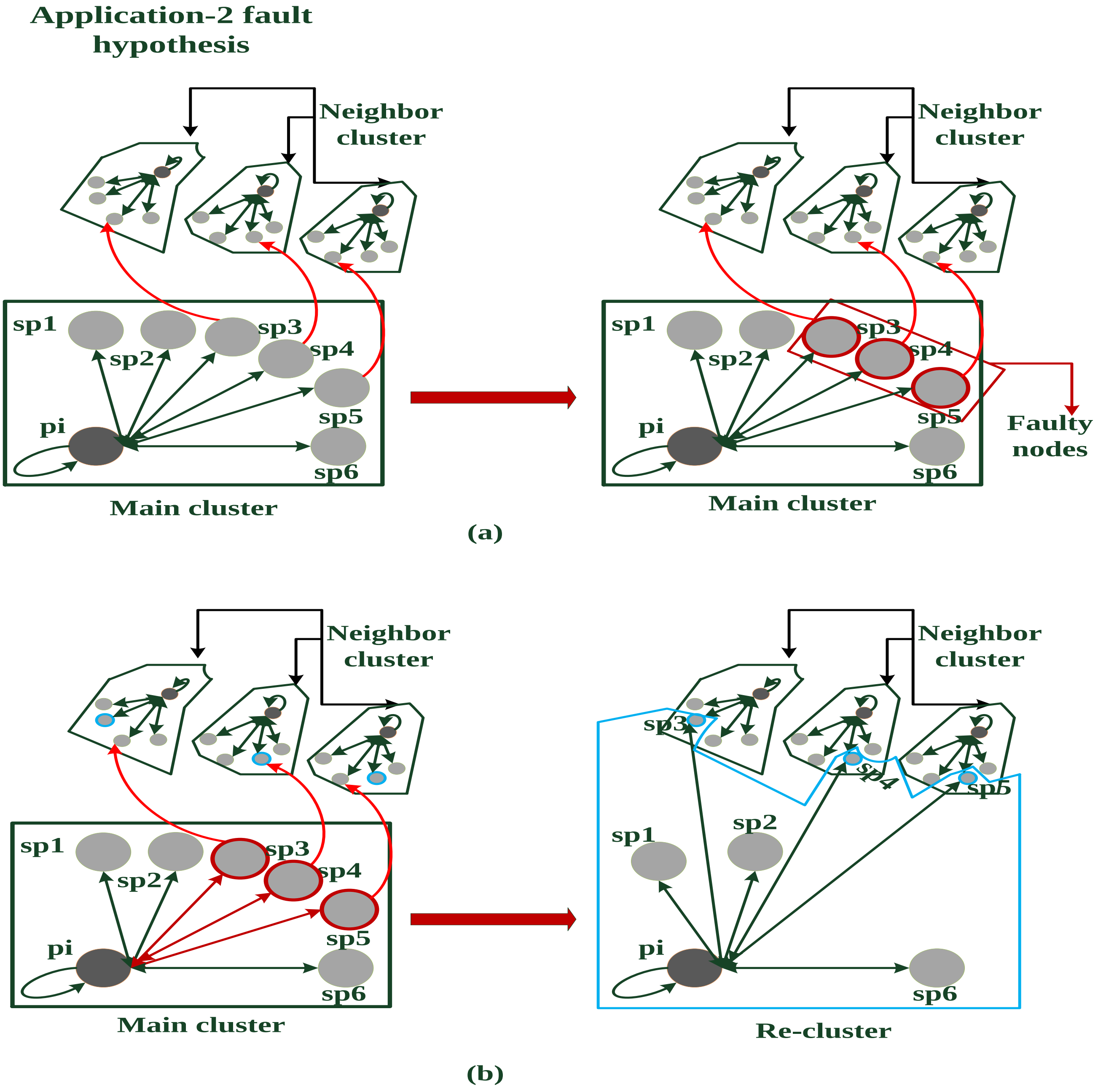


Figure 6. Fault hypothesis for application-2: (**a**) Detecting faulty nodes and identifying the failure situation based on threshold value; (**b**) Identifying the proper neighbor service providers based on application priority and re-clustering vertices.

An overview of the existing explanations of NFC and additional information on its operations proves the significant effect of the algorithm on improving the efficiency of the IoMT applications by employing the appropriate neighbor servers and stopping the defective components. The algorithm improves the privacy and dependability of the different IoMT-based applications by applying the service priority to decision making about dedicating the appropriate healthcare servers and joining them to a cluster to cover the operations of the defective nodes. Moreover, we remove the imposed energy consumption and communication delay overheads of the failed or faulty devices to the performance of the applications and improve it by pruning or stopping the defective service providers.

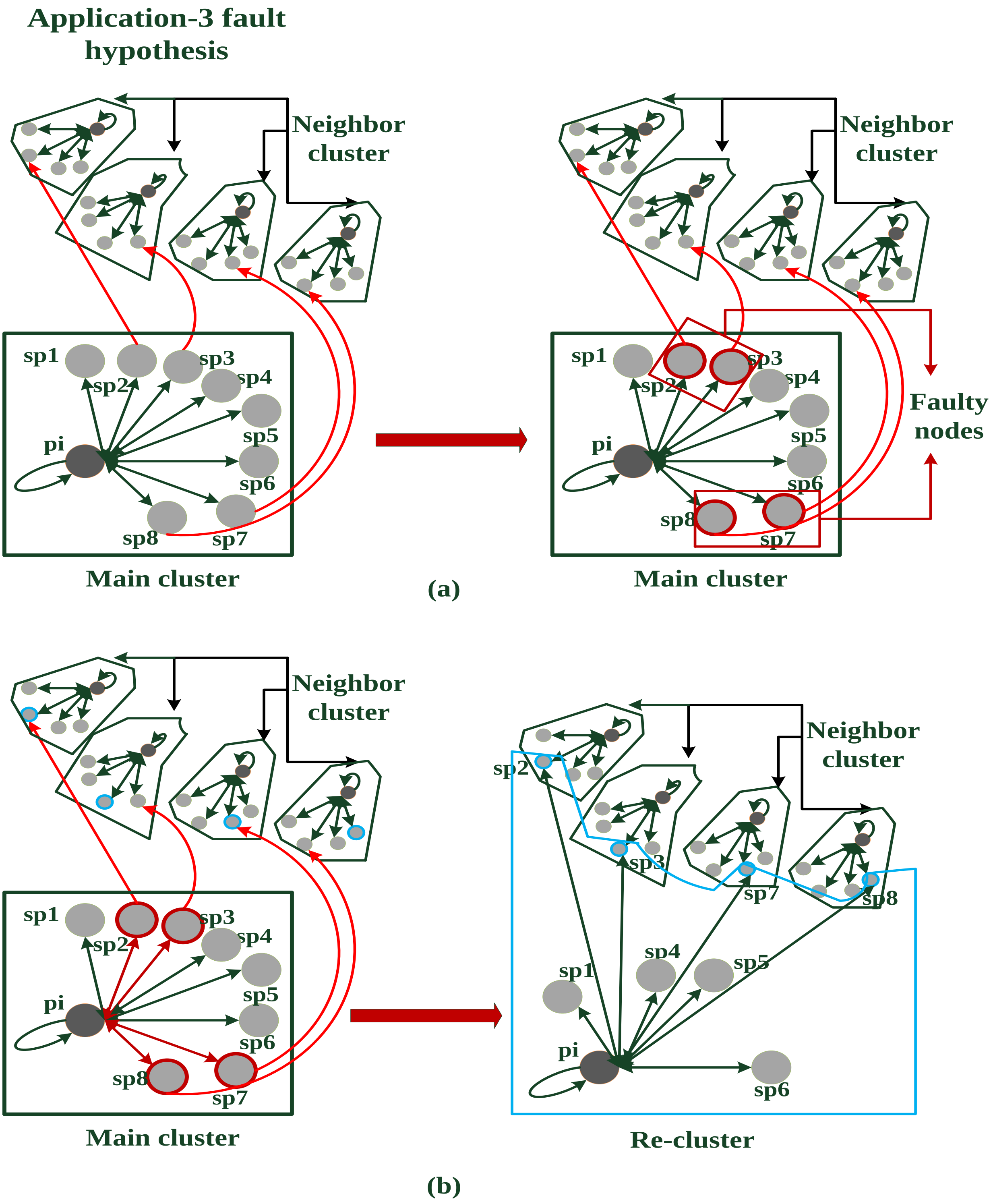


Figure 7. Fault hypothesis for application-3: (**a**) Detecting faulty nodes and identifying the failure situation based on threshold value; (**b**) Identifying the proper neighbor service providers based on application priority and re-clustering vertices.

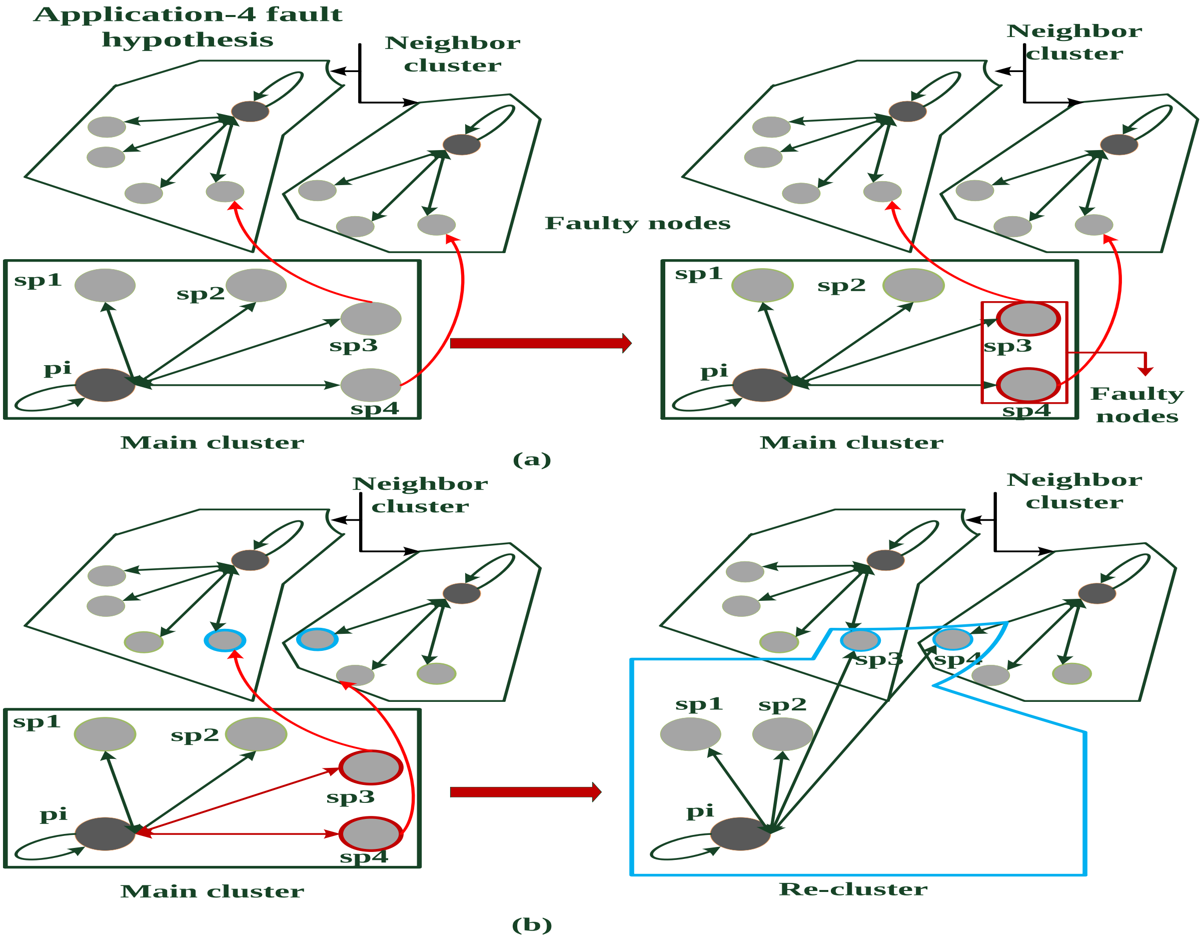


Figure 8. Fault hypothesis for application-4: (**a**) Detecting faulty nodes and identifying the failure situation based on threshold value; (**b**) Identifying the proper neighbor service providers based on application priority and re-clustering vertices.

4. Simulation Results and Observations

The section analyzes the proposed algorithm and Equation efficiency in satisfying the mentioned purposes of the study and improving the performance and dependability of the IoMT applications. To follow the issue, we report the availability, dependability, and total delay and energy of applications (1)–(4) before and after performing NFC and pruning the faulty service providers and replacing them with the neighbor servers. The study utilizes the CupCarbon tool to simulate the applications for analyzing the efficiency of NFC and flexible-clustering in improving the performance of the IoMT via describing their nodes by Java and Python co-designs [26,57,58]. We evaluate the dependability, security, and availability of the applications' nodes via injecting fault with random rates by the simulator, which runs on the Ubuntu-LTS 16.04 and MAC operational systems [59,60]. The study reports the estimated availability, dependability, and performance of the applications by exerting the mentioned hypotheses (Section 3) on their simulations.

This work employs a partial neural network to map the server nodes onto the NN's neurons and train them by considering the application and service priority to identify the proper service providers in the neighbor clusters. We train the neural network based on the Gradient Descent method and utilize the rectified linear unit (ReLU) activation function to generate the outputs where NN is described by the Python co-design and TensorFlow library on the Jupyter notebook [58,61,62]. The neural network is learned and generates outputs before and after imposing the estimated penalty values into the NN's weights by the supervised learning-based methods. This work determines the number of NN's hidden layers based on the total number of neighbor clusters to map their service providers on their nodes for deciding on the appropriate servers by pruning the weak neurons and weights.

Due to the relationship between fault injection rate and application availability, we compare the characteristics to verify the reported simulation results after detecting fault service providers and pruning or stopping the servers' nodes, as shown in Figure 9a–d.

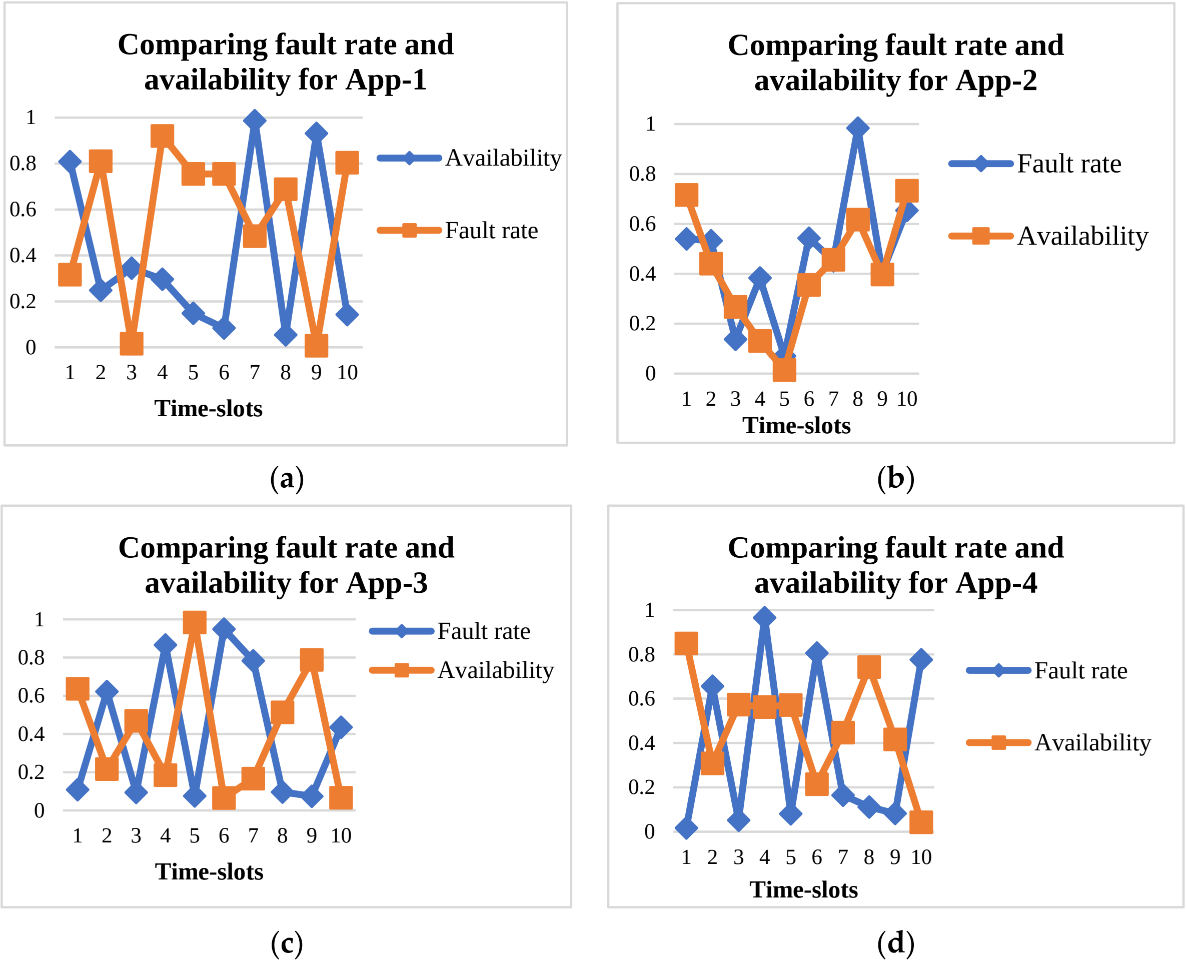
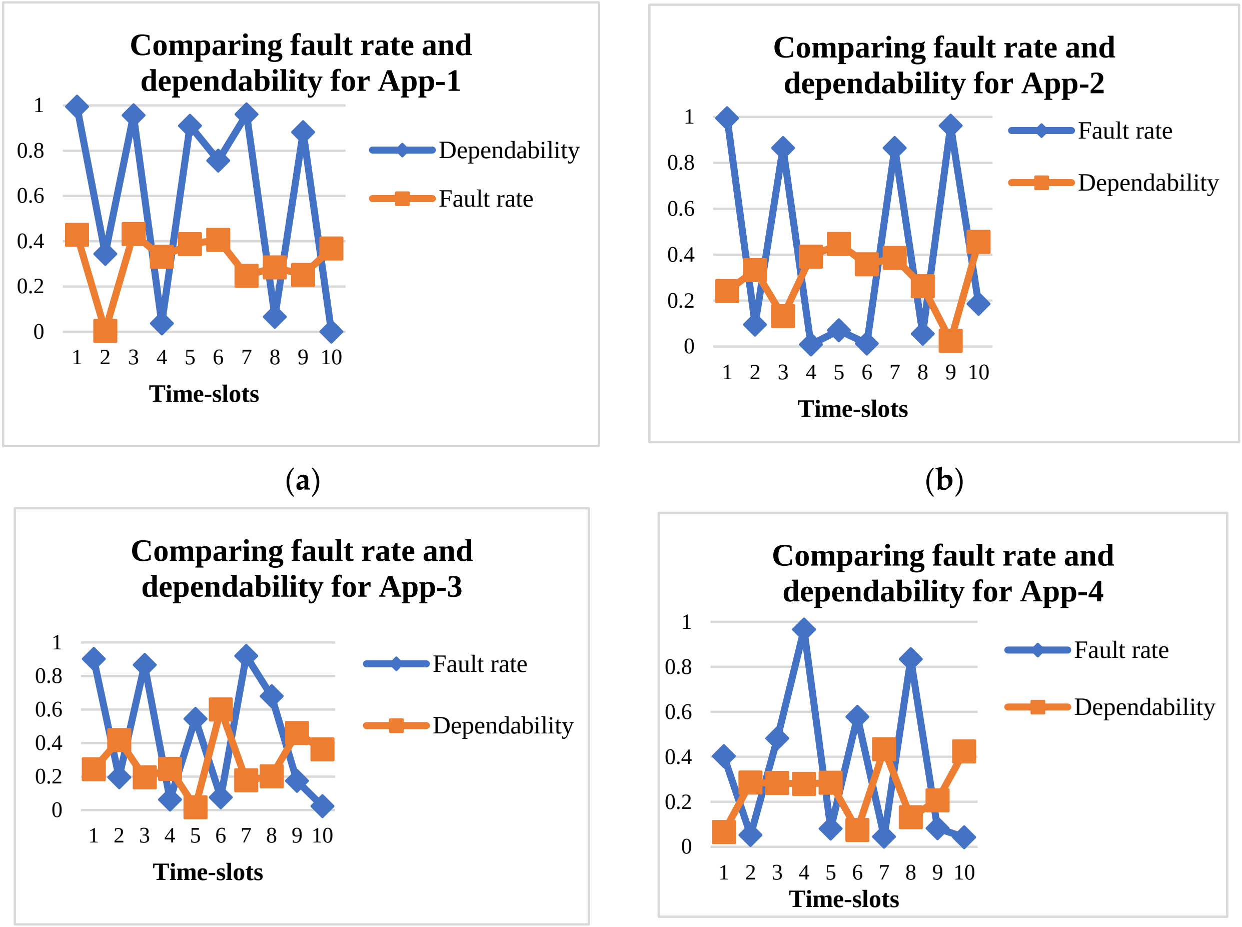


Figure 9. The relationship between fault rate and availability for applications (1)–(4): (**a**) The relationship between fault rate and availability for application-1 (App-1); (**b**) The relationship between fault rate and availability for application-2 (App-2); (**c**) The relationship between fault rate and availability for application-3 (App-3); (**d**) The relationship between fault rate and availability for application-4 (App-4).

Availability is defined as the application and service priority for the defined applications (other than application-2) because of the highlighted role of timely providing services to the requesters. The simulation results indicate the indirect relationship between the parameter and fault injection rate. The security characteristic has an impressive effect on improving the efficiency of application-2 when the observations demonstrate the faint role of increasing fault injection rate on deteriorating the application availability (as shown in Figure 9b).

Figure 10a–d illustrate the dependency between rising fault rate and falling the dependability of applications (1)–(4) to verify the presented simulation results and Equation (13) efficiency in estimating the addressed parameter.



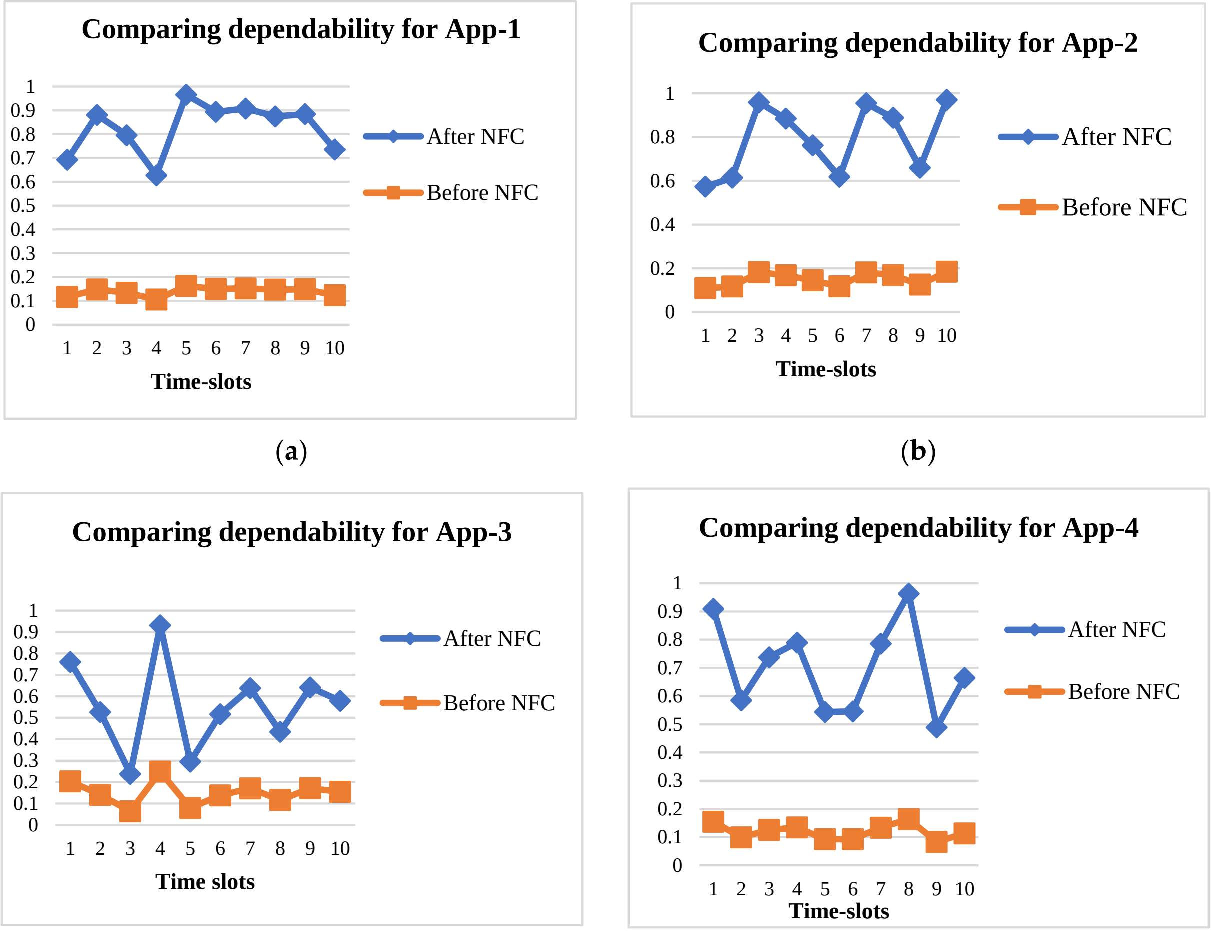
(**a**) (**b**)

(**c**) (**d**)

Figure 10. The relationship between fault rate and dependability for applications (1)–(4): (**a**) Comparing fault rate and dependability for application-1 (App-1); (**b**) Comparing fault rate and dependability for application-2 (App-2); (**c**) Comparing fault rate and dependability for application-3 (App-3); (**d**) Comparing fault rate and dependability for application-4 (App-4).

Figure 11a–d illustrate the role of applying NFC and the proposed idea in improving the dependability of applications (1)–(4) by approximately 77.3–83.2% compared with other approaches (Before NFC). We consider the application and service priority to analyze the efficiency of the applications and define a specific threshold value to decide their situation for re-clustering the graph vertices. After detecting the faulty service providers and determining the failed situations, NFC stops the connections between the servers and other nodes of the applications by identifying the proper neighbor devices and pruning their inappropriate cases. By applying the mentioned operations, the study improves the dependability of the different IoMT applications compared with other recent methods because of their lack of support from the priority feature.

This work pays attention to improving the efficiency of the IoMT application by providing an intelligent algorithm to raise the performance and dependability. To increase the performance of the applications, we prune the faulty nodes and remove their delay and energy consumption overheads, where Figure 12,b demonstrate improving the parameters about 19.3–21.7% and 10.3–11.8% for applications (1)–(4) compared with lack of employing NFC (Without NFC). The study simulates the applications' nodes and reports the total delay and energy before and after performing the algorithm and pruning the faulty components with the CupCarbon tool. The recent case studies ignored imposing latency and energy consumption overheads of the failed devices on the performance of IoMT-based applications. This work aims to remove the destructive parameters and improve their efficiency. The simulation results prove the highlighter role of the pruning method in reducing the total delay than the energy consumption because of replacing the failed nodes with the neighbor components.

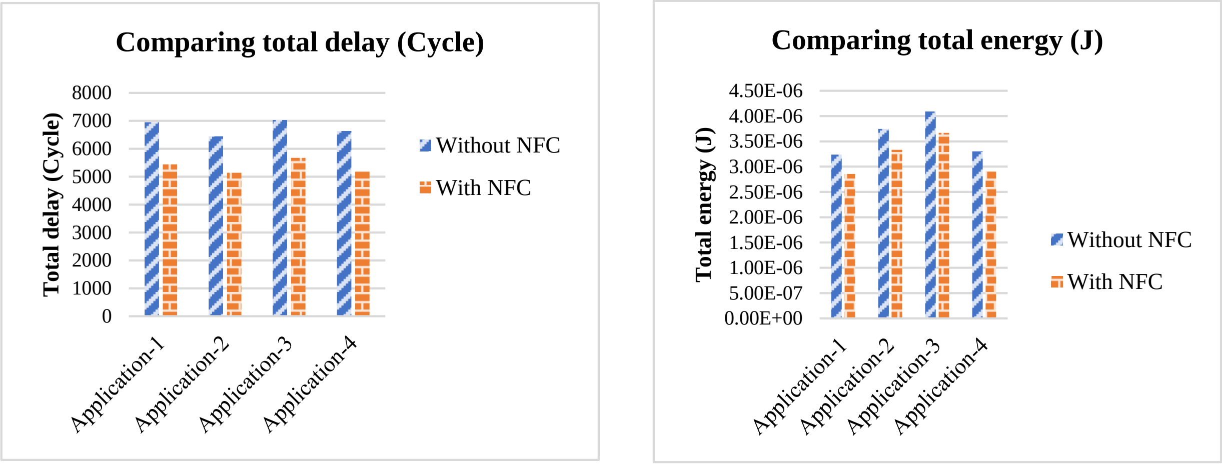


(**a**) (**b**)

(**c**) (**d**)

Figure 11. Comparing the dependability of applications (1)–(4) before and after performing NFC: (**a**) Comparing the dependability of application-1 (App-1) before and after performing NFC; (**b**) Comparing the dependability of application-2 (App-2) before and after performing NFC; (**c**) Comparing the dependability of application-3 (App-3) before and after performing NFC; (**d**) Comparing the dependability of application-4 (App-4) before and after performing NFC.

The above reported simulation results and their analysis evidence the impact of NFC and other research steps on improving the dependability, privacy, and performance of the IoMT applications. The study targeted the simulation results to analyze the performance and dependability of applications (1)–(4) with different service priorities and properties to prove supporting various IoMT applications by employing the algorithm. NFC covered the risky and normal conditions for providing dependable service to the requesters and patients to prevent a dangerous situation by failing the servers and applications. We supported the application's privacy by stopping forwarding and transferring malicious data between the server and sensor nodes to remove the related graph's edges to the defective vertices. NFC created the opportunity to improve the application dependability by re-clustering and updating the graph vertices to timely detect faulty service providers and failed statuses, which were replaced with the appropriate neighbor components. Moreover, the algorithm affected improving the IoMT applications' performance by removing the defective nodes' energy consumption and latency overheads via employing the neural network and pruning its weak weights and neurons.



(**a**) (**b**)

Figure 12. Comparing the total delay and energy of applications (1)–(4) before and after performing NFC: (**a**) Comparing the total delay of applications (1)–(4) between NFC (With NFC) and other approaches (Without NFC); (**b**) Comparing the total energy of applications (1)–(4) between NFC (With NFC) and other approaches (Without NFC).

Apart from proving the idea's efficiency in improving the IoMT-based applications' performance, we analyze its role when facing vehicle service providers in a specific scenario. NFC can detect the proper vehicle HSs to join a cluster after exiting the verified domain, which leads to challenges of losing services and risky situations by assigning a task. Due to the addressed problem scenario, we define the conditions for the joined devices to overcome the challenges by considering the stations to cluster them and time limitation (Time availability margin) for vehicle healthcare service providers.

We obtain the logs of the CupCarbon simulations and short movies of running the applications and injecting faults with random rates that shared them on Github (https://github.com/yasamanhosseini/Flexible-clustering-IoMT-nodes, accessed on 20 July 2022) to verify the reported results [56].

5. Conclusions

The IoMT technology appeared to solve the problem of supporting the security and private records on the IoT-based E-healthcare systems by raising the variates of the joined applications to the IoT infrastructures. To grow up the IoMT-based applications, protecting the patients' private records and doctors' treatment recommendations is challenging for the technology by increasing the complexity and communication between its layers and nodes. The recent case studies addressed solving the issue and presented the different approaches and algorithms to satisfy the IoMT security. The inconspicuous role of application and service priority creates some limitations for providing their ideas. Aside from the negative impact of the faulty nodes on the IoMT's dependability, their total energy and delay overheads hurt the performance of the technology. This work tried to overcome the mentioned problems and improve the efficiency of the IoMT-based applications by considering the application and service priority in providing the idea. We presented an intelligent algorithm, Equation, and model to detect the defective service providers and utilize the proper neighbor servers to support the operations of the failed nodes. The study improved the dependability of the IoMT applications by timely detecting defective components via providing a flexible clustering method to categorize the appropriate service providers. By pruning or stopping the faulty servers, we removed the energy consumption and communication delay overheads of the failed nodes from the performance of the applications.

Flexible clustering can create an opportunity to identify the domain of high, low, and empty demands and classify them toward decision-making about dedicating the fixed healthcare service providers. The scope can be followed to timely predict and detect the faulty devices in the clusters and fail them based on a threshold value, which helps to define flawless sub-clusters and improve the dependability of IoMT applications. Due to the energy efficiency and latency, the approach can also affect detecting the highly cost-based clusters, balancing load onto their devices, and accelerating the task and flow-mapping onto the network nodes by re-clustering IoMT devices.

Author Contributions: A.M.R.: Formal analysis, Methodology, Writing—original draft preparation, Software, Writing—review and editing. S.Y.H.M.: Formal analysis, Methodology, Writing—original draft preparation, Software, Writing—review and editing. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Informed Consent Statement: We confirm that there have been no human participants in this work. The manuscript has been read and approved by all named authors.

Data Availability Statement: Availability of supporting data: https://github.com/yasamanhosseini/Flexible-clustering-IoMT-nodes, accessed on 20 July 2022.

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

References

1. Jha, A.; Athanerey, A.; Kumar, A. Role and challenges of internet of things and informatics in Healthcare research. *Health Technol.* **2022**, *12*, 701–712. [CrossRef]
2. Gupta, S.; Yadav, B.; Gupta, B. Security oIoT-based e-healthcare applications using blockchain. In *Advances in Blockchain Technology for Cyber Physical Systems*; Springer: Cham, Switzerland, 2022; pp. 79–107.
3. Almalki, F.A.; Othman, S.B.; Sakli, H.; Angelides, M. Revolutionizing healthcare by coupling Unmanned Aerial Vehicles (UAVs) to Internet of Medical Things (IoMT). In *Digital Health Transformation with Blockchain and Artificial Intelligence*; CRC Press: Boca Raton, FL, USA, 2022; pp. 47–59.
4. Mohapatra, S.; Sahoo, P.K. Internet of medical things: Applications and research issues in healthcare monitoring. In *InIoT Applications for Healthcare Systems*; Springer: Cham, Switzerland, 2022; pp. 1–31.
5. Verma, G.; Shahi, A.P.; Prakash, S. A study towards recent trends, issues and research challenges of intelligent IoT healthcare techniques: IoMT and CIoMT. In *Proceedings of the Trends in Electronics and Health Informatics*; Springer: Singapore, 2022; pp. 177–190.
6. Yasmeen, G.; Javed, N.; Ahmed, T. Interoperability: A Challenge for IoMT. *ECS Trans.* **2022**, *107*, 4459–4467. [CrossRef]
7. Adil, M.; Khan, M.K.; Jadoon, M.M.; Attique, M.; Song, H.; Farouk, A. An AI-enabled Hybrid lightweight Authentication Scheme for Intelligent IoMT based Cyber-Physical Systems. *IEEE Trans. Netw. Sci. Eng.* **2022**. [CrossRef]
8. Amintoosi, H.; Nikooghadam, M.; Shojafar, M.; Kumari, S.; Alazab, M. Slight: A lightweight authentication scheme for smart healthcare services. *Comput. Electr. Eng.* **2022**, *99*, 107803. [CrossRef]
9. Haque, R.U.; Hasan, A.S. Overview of blockchain-based privacy preserving machine learning for IoMT. In *Big Data Intelligence for Smart Applications*; Springer: Cham, Switzerland, 2022; pp. 265–278.
10. Haque, R.U.; Hasan, A.S.; Nishat, T.; Adnan, M.A. Privacy-preserving-means clustering over blockchain-based encrypted IoMT Data. In *Advances in Blockchain Technology for Cyber Physical Systems*; Springer: Cham, Switzerland, 2022; pp. 109–123.
11. Lakhan, A.; Mohammed, M.A.; Nedoma, J.; Martinek, R.; Tiwari, P.; Vidyarthi, A.; Alkhayyat, A.; Wang, W. Federated-learning based privacy preservation and fraud-enabled blockchain IoMT system for healthcare. *IEEE J. Biomed. Health Inform.* **2022**, 1–11. [CrossRef]
12. Gaba, G.S.; Hedabou, M.; Kumar, P.; Braeken, A.; Liyanage, M.; Alazab, M. Zero knowledge proofs based authenticated key agreement protocol for sustainable healthcare. *Sustain. Cities Soc.* **2022**, *80*, 103766. [CrossRef]
13. Rana, A.; Chakraborty, C.; Sharma, S.; Dhawan, S.; Pani, S.K.; Ashraf, I. Internet of medical things-based secure and energy-efficient framework for health care. *Big Data* **2022**, *10*, 18–33. [CrossRef]
14. Ghazal, T.M.; Hasan, M.K.; Abdallah, S.N.; Abubakkar, K.A. Secure IoMT pattern recognition and exploitation for multimedia information processing using private blockchain and fuzzy logic. *Trans. Asian Low-Resour. Lang. Inf. Process.* **2022**. [CrossRef]
15. Fuke, R.P.; Mahajan, R.P. Pragmatic Analysis of IoMT Network Modelling Techniques from a Statistical Perspective. In Proceedings of the 2022 International Conference on Electronics and Renewable Systems (ICEARS), Tuticorin, India, 16–18 March 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 531–542.

16. Pelekoudas-Oikonomou, F.; Zachos, G.; Papaioannou, M.; de Ree, M.; Ribeiro, J.C.; Mantas, G.; Rodriguez, J. Blockchain-based security mechanisms for IoMT Edge networks in IoMT-based healthcare monitoring systems. *Sensors* **2022**, *22*, 2449. [CrossRef]
17. Wazid, M.; Gope, P. BACKM-EHA: A novel blockchain-enabled security solution for IoMT-based e-healthcare applications. *ACM Trans. Internet Technol. (TOIT)* **2022**. [CrossRef]
18. Sampathkumar, A.; Tesfayohani, M.; Shandilya, S.K.; Goyal, S.B.; Shaukat Jamal, S.; Shukla, P.K.; Bedi, P.; Albeedan, M. Internet of Medical Things (IoMT) and reflective belief design-based big data analytics with Convolution Neural Network-Metaheuristic Optimization Procedure (CNN-MOP). *Comput. Intell. Neurosci.* **2022**, *2022*, 2898061. [CrossRef] [PubMed]
19. Garg, N.; Petwal, R.; Wazid, M.; Singh, D.P.; Das, A.K.; Rodrigues, J.J. On the design of an AI-driven secure communication scheme for Internet of medical things environment. *Digit. Commun. Netw.* **2022**. [CrossRef]
20. Ibaida, A.; Abuadbba, A.; Chilamkurti, N. Privacy-preserving compression model for efficient IoMT ECG sharing. *Comput. Commun.* **2021**, *166*, 1–8. [CrossRef]
21. Wei, T.; Liu, S.; Du, X. Learning-based efficient sparse sensing and recovery for privacy-aware IoMT. *IEEE Internet Things J.* **2022**, *9*, 9948–9959. [CrossRef]
22. Si-Ahmed, A.; Al-Garadi, M.A.; Boustia, N. Survey of machine learning based intrusion detection methods for internet of medical things. *arXiv preprint* **2022**, arXiv:2202.09657.
23. Nayak, J.; Meher, S.K.; Souri, A.; Naik, B.; Vimal, S. Extreme learning machine and bayesian optimization-driven intelligent framework for IoMT cyber-attack detection. *J. Supercomput.* **2022**, *78*, 14866–14891. [CrossRef]
24. Rahmani, A.M.; Ali Naqvi, R.; Ali, S.; Hosseini Mirmahaleh, S.Y.; Hosseinzadeh, M. Quasi-Mapping and Satisfying IoT Availability with a Penalty-Based Algorithm. *Mathematics* **2021**, *9*, 3286. [CrossRef]
25. Pustokhina, I.V.; Pustokhin, D.A.; Gupta, D.; Khanna, A.; Shankar, K.; Nguyen, G.N. An effective training scheme for deep neural network in edge computing enabled Internet of medical things (IoMT) systems. *IEEE Access* **2020**, *8*, 107112–107123. [CrossRef]
26. CupCarbon Simulator. Available online: http://cupcarbon.com/ (accessed on 1 June 2022).
27. Khan, I.A.; Moustafa, N.; Razzak, I.; Tanveer, M.; Pi, D.; Pan, Y.; Ali, B.S. XSRU-IoMT: Explainable simple recurrent units for threat detection in Internet of Medical Things networks. *Future Gener. Comput. Syst.* **2022**, *127*, 181–193. [CrossRef]
28. Rasool, R.U.; Ahmad, H.F.; Rafique, W.; Qayyum, A.; Qadir, J. Security and privacy of Internet of medical things: A contemporary review in the age of surveillance, botnets, and adversarial ML. *J. Netw. Comput. Appl.* **2022**, *201*, 103332. [CrossRef]
29. Hossen, M.N.; Panneerselvam, V.; Koundal, D.; Ahmed, K.; Bui, F.M.; Ibrahim, S.M. Federated Machine Learning for Detection of Skin Diseases and Enhancement of Internet of Medical Things (IoMT) Security. *IEEE J. Biomed. Health Inform.* **2022**. [CrossRef] [PubMed]
30. Enamamu, T.S. Intelligent authentication framework for Internet of Medical Things (IoMT). In *Illumination of Artificial Intelligence in Cybersecurity and Forensics*; Springer: Cham, Switzerland, 2022; pp. 97–121.
31. Lee, T.F.; Ye, X.; Lin, S.H. Anonymous Dynamic Group Authenticated Key Agreements Using Physical Unclonable Functions for Internet of Medical Things. *IEEE Internet Things J.* **2022**, *6*, 15336–15348. [CrossRef]
32. Bevish Jinila, Y.; Prayla Shyry, S.; Christy, A. A Multi-component-based zero trust model to mitigate the threats in internet of medical things. In *Data Engineering for Smart Systems*; Springer: Singapore, 2022; pp. 605–613.
33. Samuel, O.; Omojo, A.B.; Onuja, A.M.; Sunday, Y.; Tiwari, P.; Gupta, D.; Hafeez, G.; Yahaya, A.S.; Fatoba, O.J.; Shamshirband, S. IoMT: A COVID-19 Healthcare System driven by Federated Learning and Blockchain. *IEEE J. Biomed. Health Inform.* **2022**. [CrossRef]
34. Ali, A.; Almaiah, M.A.; Hajjej, F.; Pasha, M.F.; Fang, O.H.; Khan, R.; Teo, J.; Zakarya, M. An industrial IoT-based blockchain-enabled secure searchable encryption approach for healthcare systems using neural network. *Sensors* **2022**, *22*, 572. [CrossRef] [PubMed]
35. Mubashar, A.; Asghar, K.; Javed, A.R.; Rizwan, M.; Srivastava, G.; Gadekallu, T.R.; Wang, D.; Shabbir, M. Storage and proximity management for centralized personal health records using an ipfs-based optimization algorithm. *J. Circuits Syst. Comput.* **2022**, *31*, 2250010. [CrossRef]
36. Nie, X.; Zhang, A.; Chen, J.; Qu, Y.; Yu, S. Blockchain-empowered secure and privacy-preserving health data sharing in edge-based IoMT. *Secur. Commun. Netw.* **2022**, *2022*, 8293716. [CrossRef]
37. Kaur, D.; Singh, S.; Mansoor, W.; Kumar, Y.; Verma, S.; Dash, S.; Koul, A. Computational intelligence and metaheuristic techniques for brain tumor detection through IoMT-Enabled MRI Devices. *Wirel. Commun. Mob. Comput.* **2022**, *2022*, 1519198. [CrossRef]
38. Reddy, D.K.; Behera, H.S.; Nayak, J.; Routray, A.R.; Kumar, P.S.; Ghosh, U. A Fog-Based Intelligent Secured IoMT Framework for Early Diabetes Prediction. In *Intelligent Internet of Things for Healthcare and Industry*; Springer: Cham, Switzerland, 2022; pp. 199–218.
39. Sharifshazileh, M.; Burelo, K.; Sarnthein, J.; Indiveri, G. An electronic neuromorphic system for real-time detection of high frequency oscillations (HFO) in intracranial EEG. *Nat. Commun.* **2021**, *12*, 3095. [CrossRef]
40. Alon, H.D.; Ligayo, M.A.D.; Melegrito, M.P.; Cunanan, C.F.; Uy, I.I.E. Deep-Hand: A deep inference vision approach of recognizing a hand sign language using american alphabet. In Proceedings of the 2021 International Conference on Computational Intelligence and Knowledge Economy (ICCIKE), Dubai, United Arab Emirates, 17–18 March 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 373–377.
41. Sellaturay, P.; Nasser, S.; Ewan, P. Polyethylene glycol–induced systemic allergic reactions (anaphylaxis). *J. Allergy Clin. Immunol. Pract.* **2021**, *9*, 670–675. [CrossRef]

42. Dick, K.; Pattang, A.; Hooker, J.; Nissan, N.; Sadowski, M.; Barnes, B.; Tan, L.H.; Burnside, D.; Phanse, S.; Aoki, H.; et al. Human–Soybean allergies: Elucidation of the seed proteome and comprehensive protein–protein interaction prediction. *J. Proteome Res.* **2021**, *20*, 4925–4947. [CrossRef]
43. Muramatsu, K.; Imamura, H.; Tokutsu, K.; Fujimoto, K.; Fushimi, K.; Matsuda, S. Epidemiological study of hospital admissions for food-induced anaphylaxis using the Japanese Diagnosis Procedure Combination Database. *J. Epidemiol.* **2022**, *32*, 163–167. [CrossRef] [PubMed]
44. Rauch, S.; Wallner, B.; Ströhle, M.; Dal Cappello, T.; Brodmann Maeder, M. Climbing accidents—Prospective data analysis from the international alpine trauma registry and systematic review of the literature. *Int. J. Environ. Res. Public Health* **2020**, *17*, 203. [CrossRef] [PubMed]
45. Satava, R.; Angood, P.B.; Harnett, B.; Macedonia, C.; Merrell, R. The physiologic cipher at altitude: Telemedicine and real-time monitoring of climbers on Mount Everest. *Telemed. J. E-Health* **2000**, *6*, 303–313. [CrossRef] [PubMed]
46. Dataset. Available online: https://books.google.com/books/about/The_Himalaya_by_the_Numbers.html?id=9kiiuAAACAAJ&source=kp_book_description (accessed on 10 May 2022).
47. Dataset. Available online: https://medium.com/ai-techsystems/analyzing-eeg-brainwave-data-to-detect-emotions-on-cainvas-48833f3f0811 (accessed on 10 May 2022).
48. Dataset. Available online: https://www.kaggle.com/datasets/piotrgrabo/breastcancerproteomes (accessed on 10 May 2022).
49. Dataset. Available online: https://www.kaggle.com/datasets/andrewgao/alzheimers-gene-expression-profiles (accessed on 10 May 2022).
50. Dataset. Available online: https://ncdrisc.org/data-downloads-blood-pressure.html (accessed on 10 May 2022).
51. Dataset. Available online: https://www.kaggle.com/datasets/johnsmith88/heart-diseasedataset?resource=download&select=heart.csv (accessed on 10 May 2022).
52. Dataset. Available online: https://www.ons.gov.uk/peoplepopulationandcommunity/birthsdeathsandmarriages/deaths/datasets/climaterelatedmortalityandhospitaladmissionsenglandandwales (accessed on 10 May 2022).
53. Dataset. Available online: https://www.ons.gov.uk/peoplepopulationandcommunity/healthandsocialcare/healthcaresystem/datasets/nationalsurveyofbereavedpeoplevoices (accessed on 10 May 2022).
54. Dataset. Available online: https://www.ons.gov.uk/peoplepopulationandcommunity/healthandsocialcare/conditionsanddiseases (accessed on 10 May 2022).
55. Dataset. Available online: https://ourworldindata.org/covid-vaccinations (accessed on 10 May 2022).
56. Collected Dataset. Available online: https://github.com/yasamanhosseini/Flexible-clustering-IoMT-nodes (accessed on 20 July 2022).
57. Dataset. Available online: https://www.javatpoint.com/design-patterns-in-java (accessed on 10 May 2022).
58. Dataset. Available online: https://www.geeksforgeeks.org/python-design-patterns/ (accessed on 10 May 2022).
59. Dataset. Available online: https://releases.ubuntu.com/16.04/ (accessed on 10 May 2022).
60. Dataset. Available online: https://support.apple.com/downloads/macos (accessed on 10 May 2022).
61. Dataset. Available online: https://jupyter.org/ (accessed on 10 May 2022).
62. Dataset. Available online: https://www.tensorflow.org/ (accessed on 10 May 2022).