applied sciences

MDPI

Article

An Efficient Data-Balancing Cyber-Physical System Paradigm for Quality-of-Service (QoS) Provision over Fog Computing

Muder Almiani [1,2], Abdul Razaque [3,*], Bandar Alotaibi [4,5,*], Munif Alotaibi [6,*], Saule Amanzholova [7] and Aziz Alotaibi [8]

1 Management Information System Department, Gulf University for Science and Technology, Hawally 32093, Kuwait; almiani.m@gust.edu.kw
2 Computer Information Systems Department, Al-Hussein Bin Talal University, Ma'an 71111, Jordan
3 Department of Computer Engineering and Cybersecurity, International Information Technology University, Almaty 050000, Kazakhstan
4 Department of Information Technology, University of Tabuk, Tabuk 47731, Saudi Arabia
5 Sensor Networks and Cellular Systems (SNCS) Research Center, University of Tabuk, Tabuk 47731, Saudi Arabia
6 Department of Computer Science, Shaqra University, Shaqra 11961, Saudi Arabia
7 Department of Cybersecurity, International Information Technology University, Almaty 050000, Kazakhstan; s.amanzholova@iitu.edu.kz
8 Department of Computer Science, College of Computers and Information Technology, Taif University, P.O. Box 11099, Taif 21944, Saudi Arabia; azotaibi@tu.edu.sa
* Correspondence: a.razaque@iitu.edu.kz (A.R.); b-alotaibi@ut.edu.sa (B.A.); munif@su.edu.sa (M.A.)

Abstract: Cyber-physical systems (CPSs) have greatly contributed to many applications. A CPS is capable of integrating physical and computational capabilities to interact with individuals through various new modalities. However, there is a need for such a paradigm to focus on the human central nervous system to provide faster data access. This paper introduces the CPS paradigm that consists of CPS enabled human brain monitoring (CPS-HBM) and efficient data-balancing for CPS (EDB-CPS). The CPS-HBM provides architectural support to make an efficient and secure transfer and storage of the sensed data over fog cloud computing. The CPS-HBM consists of four components: physical domain and data processing (PDDP), brain sensor network (BSN), Service-oriented architecture (SOA), and data management domain (DMD). The EDB-CPS module aims to balance data flow for obtaining better throughput and lower hop-to-hop delay. The EDB-CPS accomplishes the goal by employing three processes: A node advertisement (NA), A node selection and recruitment (NSR), and optimal distance determination with mid-point (ODDMP). The processes of the EDB-CPS are performed on the PDDP of the CPS-HBM module. Thus, to determine the validity of EDB-CPS, the paradigm was programmed with C++ and implemented on a network simulator-3 (NS3). Finally, the performance of the proposed EDB-CPS was compared with state-of-the-art methods in terms of hop-to-hop delay and throughput. The proposed EDB-CPS produced better throughput between 443.2–445.2 KB/s and 0.05–0.078 ms hop-to-hop delay.

Keywords: cyber-physical systems; brain sensor network; wireless sensor networks; cyberspace

Citation: Almiani, M.; Razaque, A.; Alotaibi, B.; Alotaibi, M.; Amanzholova, S.; Alotaibi, A. An Efficient Data-Balancing Cyber-Physical System Paradigm for Quality-of-Service (QoS) Provision over Fog Computing. *Appl. Sci.* **2022**, *12*, 246. https://doi.org/10.3390/app12010246

Academic Editor: Arcangelo Castiglione

Received: 1 December 2021
Accepted: 16 December 2021
Published: 27 December 2021

1. Introduction

The virtual world and the physical world are merging, and this is called cyberspace. When cyber and physical world are merging, and this is called Cyber-Physical systems [1]. Cyber-physical systems (CPSs) have emerged in recent years, and these systems are growing extremely fast, with almost 98% of microprocessors connected with the outside world through actuators and sensors [2]. CPSs are changing the way interactions take place in cyberspace. CPSs contribute to safety [1], efficiency [3], human health [4], networked navigation software [5], market structures [6], science and research [7], and heterogeneous networks [8]. CPSs combine strong network and real-time applications while focusing on

energy, mobility, health, and industry [9]. CPSs are real-time systems in which the data from real-time objects are computed and processed by the computational unit, resulting in acknowledgment effects. Even though this system can be very useful in real-time projects, its adoption has been delayed because of the mismatch between the abstraction and properties of the physical process [10–12]. It is currently being used for several applications, for example, handling the electric power transmission grid in which, by reviewing coordinate controls and system sensors on distributed energy resources and cyber coordinates, it detects and reacts to faults. Recently, there has been a need for these systems in natural resource awareness. By using massive networks of sensors and actuators, large environmental areas can be accessed, and, by using real-time data, it can revolutionize how science works [13,14]. A CPS can also be used to analyze the interconnection between power control applications and cyber systems [15,16]. Despite all of the applications in which cyber-physical systems are currently being used, there are many challenges that still arise, such as real-time system abstraction, security [17], robustness, systems engineering research, and trust in CPSs [18]. Most of the existing approaches for resolving the challenges presented by CPSs are based on static approaches. Static approaches reduce the chance of emergencies and interference from noise, which makes the model more stable and easier to analyze. In addition, static models tend to consume less power and financial resources, making it realistic for many situations. However, because more resources are required than are locally available, and because this model cannot give timely feedback, mobile cyber-physical systems need to be developed [19]. Although static approaches resolve problems, the mobility of a CPS has not been properly addressed. The development of the Internet, wireless communication, robots, vehicles, mobile phones, etc., can lead to ideal mobile nodes [20,21]. Mobile nodes can detect information over a large area and send it back to a base station for analysis. It solves the resource limitation problem in static models and improves efficiency. However, new problems are raised in the mobile model. Because of this mobility, mobile nodes are more likely to be attacked. Once some nodes fail to work, the performance of the whole system can be influenced, making the system unsafe [22]. In addition, it is expensive and time-consuming to develop practical applications for mobile nodes, which also discourages its development [23].

A constant growth and maturity in wireless sensor networks (WSNs) demand extended functionality that can be used to integrate it with other network systems using reliable and secure communication methods. A CPS bridges this gap by providing sensing applications as a platform to provide extended interactive functionality between real-time and virtual environments [24]. To this extent, CPSs have proven to be effective interconnection mechanisms for human-to-human, human-to-machine, and machine-to-machine interactions by seamless network connectivity and refined user control over the actuation side [25]. This also accords with the basic definition of a CPS, which is mainly responsible for providing a virtual environment to incorporate an interacting network of system elements with physical inputs at both ends. Figure 1 depicts the basic human central nervous system monitoring process with a CPS.

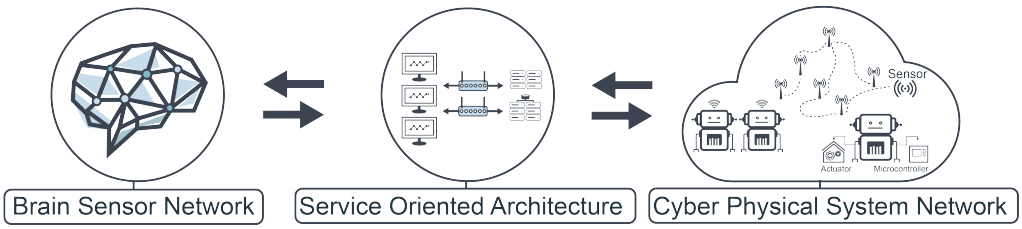

Figure 1. Basic human central nervous system monitoring process with a CPS.

1.1. Research Contributions

The main contributions of the paper are as follows:

- Efficient data balancing has been maintained in a CPS using three exciting processes: a node advertisement, a node selection and recruitment, and an optimal distance determination with mid-point. These modules enable the physical domain to efficiently maintain a data balance when collecting data from the human central nervous system. Furthermore, using EDB-CPS, hop-to-hope delay is reduced to obtain better throughput.
- An optimal distance determination with mid-point algorithm is introduced to allow the sensor node to identify the nearest actuators to share data efficiently and avoid any potential data loss or data delivery.

1.2. Paper Organization

Section 2 discusses the salient features of existing state-of-the-art methods. Section 3 presents a system model of the proposed CPS model for human brain monitoring. Section 4 presents efficient data balancing for the CPS. Section 5 discusses test results. Section 6 discusses the significance of the results and the limitations. Section 7 concludes the paper.

2. Related Work

In this section, the salient features of the state-of-the-art methods are discussed. The intelligent architecture for cyber-physical systems management (IA-CPS) was introduced in Reference [26]. The IA-CPS is supported with a service-oriented architecture (SOA) that provides the functionalities of an event-driven system. This model permits the connection of different types of devices with the system; however, there is a possibility of transmission failure. A CPS based on a blockchain-enabled smart modular integrated construction (BSMIC) model was introduced in Reference [27]. A practical roadmap was used for the design, development, placement, and application of BSMIC with new guidelines and opportunities. However, an increased number of nodes in a system caused a rise in power use, which led to untimely defeat.

The accepted requirements for the physical components and the unified invariant approach for cyber components are the reason for the improvement of such functions as the stability and security of CPSs [28]. Ensuring the autonomy of the system is a fundamental idea of the procedure, while invariants are used to determine the sequences of stable and unstable switching systems. As the authors assume, switched systems are controlled by a complex and distributed cyber process. However, in cases when the system does not respond or an analytical requirement arises, the system is not able to detect errors, and this method is restricted.

The probability of the cluster-based method uses a large number of wireless sensors installed in one area to provide a solution to the problem of removing sensors from CPSs [29]. It is used when checking intruders, and, when detected, it is based on unreliable sensor data and draws a graph of the relationship between sensors and intruders, thereby analyzing the unstable attributes of the attacker. However, this system is not effective in areas, such as electrical conditions, vehicle monitoring, and public observation systems.

An intelligent transportation system based on CPSs was introduced in Reference [30]. This system provided safety features for a transportation system and used the verbal warning utility scale for support. It was limited in safety and failed to provide warnings to vehicles in the case of an emergency. An industrial automation system introduced in Reference [31] used the concept of cyber-physical system mobility. It was composed of physical plants that perform physical processes and a network of embedded systems. This embedded system used programmable logic controllers (PLCs). However, it did not address the complexity of today's systems. The StreamLAB framework was introduced for computations with less memory consumption and guaranteed run-time. The applicability of StreamLAB employed typical monitoring tasks for a CPS. However, the authors failed to define and implement the idea properly. The cyber-physical system mobility concept was also used in vehicle applications introduced in Reference [32]. In this approach, the information provision service was provided to drivers to utilize the data communication

network between vehicles and infrastructure. However, the combination of physical data analysis and cyber data analysis in real time was a problem.

In this paper, we implement the concept of a human-in-the-loop cyber-physical system, in which brain signals can be converted into robot signals by a signal processing unit, which will then provide a physical component and work accordingly with mind signals.

3. Proposed CPS-HBM

The CPS model is responsible for an efficient flow of information from patient to practitioner, depicted in Figure 2.

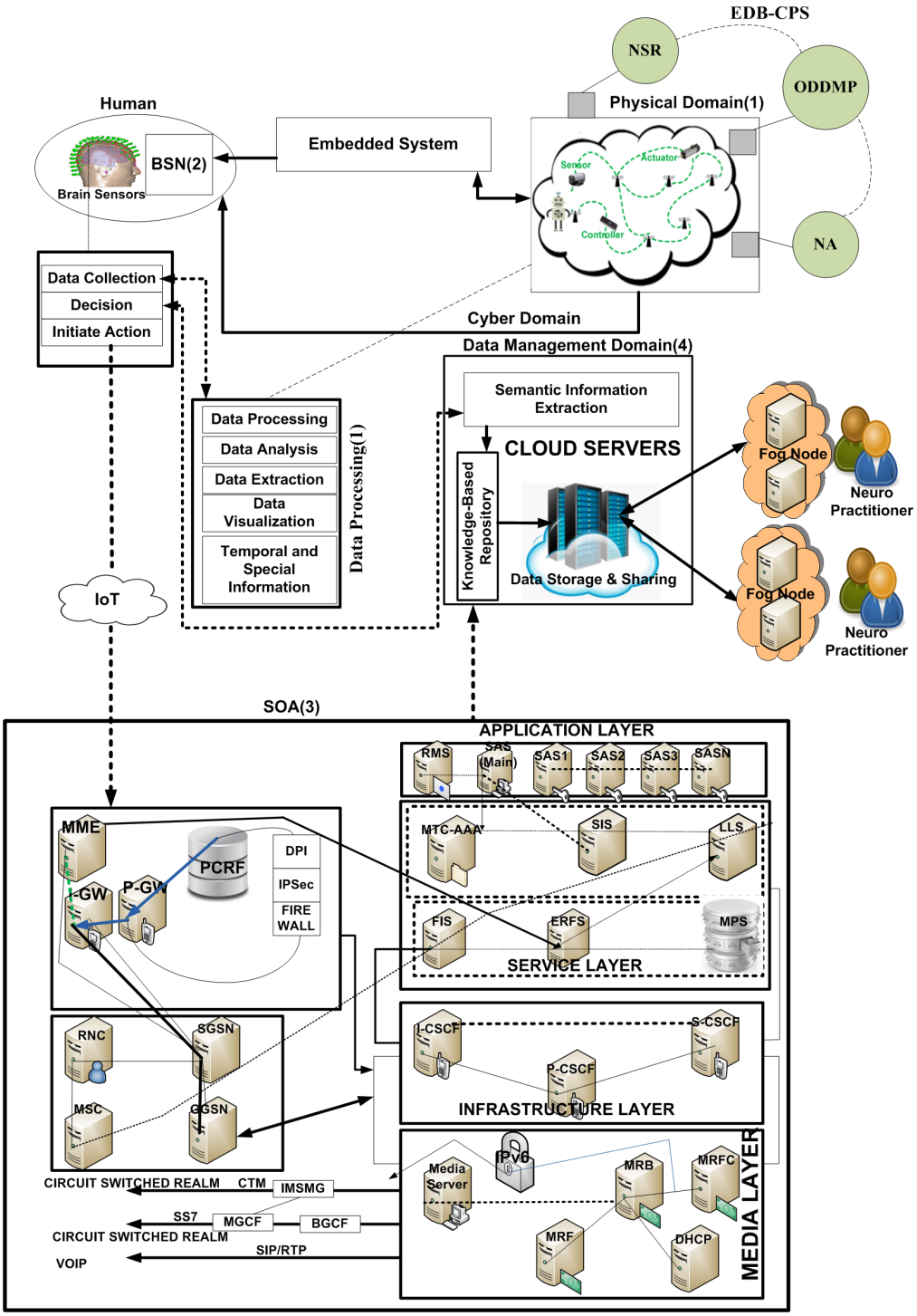

Figure 2. Proposed CPS-HBM to support efficient data balancing.

It involves four components that perform conjointly:

- a BSN,
- physical domain and data processing,
- an SOA, and
- a data management domain.

3.1. Brain Sensor Network

A BSN is a network of wireless sensors that detect the brain's activity. BSN devices can be embedded in the brain, implanted or attached to the surface of the brain, or combined with devices that people carry in various places. Initial applications of the brain's computer network are primarily expected in the field of healthcare, especially for the continuous monitoring and recording of important data about patients suffering from chronic diseases, such as diabetes, asthma, and heart attacks. Detectors in the brain can provide warnings that assist diagnosis, arrangements, and therapy.

3.2. Physical Domain and Data Processing

The physical domain of a system includes actuators, sensors, and controllers. These sensors report data to a chosen base station (BS) that can play the role of the head node. The BS transports it to the actuator or controller. Finally, the robot decides which action should be performed. Through this process, the brain signal is converted into a robot signal, which can later be easily stored and analyzed. After the human signal has been converted, the data processor works to evaluate these data. This process is further divided down into three parts: data collection, decision, and action initiation. In the first part (i.e., the data collection), the tasks of the data processing, data analysis, data extraction, data visualization, and temporal processing with special information are conducted. Based on the obtained results, a decision is made that stores the information in cloud servers by using semantic information extraction. Furthermore, the cloud servers provide the access to medical practitioners through fog nodes.

3.3. SOA

There are four layers in the SOA model:

- an application layer,
- a service layer,
- an infrastructure layer, and
- a media layer.

3.3.1. The Application Layer

The application layer is based on Secure Authentication Servers (SASs) and is divided into two types: a secure authentication server and a role-based management server. A secure authentication server is a kind of verification process that ensures users' confirmation. For occurrence, if the full framework has one verification server, others will access the information effortlessly. When the sensors distinguish a user's thought of something imperative, the enemy will capture this thought effectively as their own thoughts. As a result, the security of users' information cannot be guaranteed. In any case, a secure confirmation server can separate users' information into distinctive sorts of servers, which will ensure the security of the complete framework. A role-based management server makes the framework secure and steady. It can create an open key to decrease malicious harm. Role tasks and administration can offer assistance in disseminating secret keys to clients. To guarantee the security of the SOA, these two servers complement each other.

3.3.2. The Service Layer

The service layer consists of six different servers: a Machine Type Communication, Authentication, Authorization and Accounting (MTC-AAA), Subscriber Information Server

(SIS), Location Locater Server (LLS), Feature Integrating Server (FIS), Efficient Route Finding Server (ERFS), and an MPS. The International Mobile Subscriber Identity (IMSI), with the outside identifier of the Equipment Identifiers (UEs), is mapped by the MTC-AAA server, which makes a difference in Subscriber Information Retrieval (SIR), and then sent to the SIS. It also receives a request from the MTC-AAA server, and the SIS begins to check for substantial membership for authentic mobile cloud clients. On the off chance that the user's personality is affirmed (as of now stored in the SIS), the SIS sends an agreed reaction to the MTC-AAA server and the LLS; otherwise, the SIS denies it and sends a negative message to the MTC-AAA server. In the previous circumstance, the LLS is capable of deciding the area of the mobile cloud client. The ERFS receives an upgraded area from the LLS to perform legitimate steering. In the expansion, the MPS possesses profiles of enrolled clients, undergoes a verification process at the benefit layer, and stores the QoS data of the particular service and supporter.

3.3.3. The Infrastructure Layer

The infrastructure layer comprises a call session control function (CSCF), which works to create a boundary between a mobile cloud user's IP address and their open identity. A proxy-CSCF (P-CSCF), a serving-CSCF (S-CSCF), and an interrogating-CSCF (I-CSCF) constitute a CSCF. This layer underpins diverse sorts of services, such as web, video conferencing, mail, and communication.

3.3.4. The Media Layer

To bring the best multimedia experience, the media layer combines a media resource broker (MRB) and a media resource function controller (MRFC). The quick, consistent handoff portable IPv6 (FSHIPv6) is proposed to unravel handoff bundle misfortune and idleness. The MRB and MRFC are both associated with the IPv6 to form beyond any doubt the handoff handle. Moreover, the MRF and DHCP are associated with the MRB. This makes the full handle easier to achieve.

3.4. Data Management Domain

The data management domain comprises three parts: semantic data extraction, knowledge-based storage, and cloud servers. Semantic data extraction may be a framework that can look, analyze, and conclude. A robot makes choices based on data that sensors distinguish, at which point these choices are transmitted for semantic data extraction. Semantic data extraction analyzes these choices and makes a judgment of which choices can be put away in knowledge-based storage. For illustration, the robot makes the choices agreeing with the patient's movement, at which point semantic data extraction investigates the choice to make a conclusion, so the specialist can effortlessly grant feedback.

This model can visually allow clients to induce crucial data rapidly and effectively. The data of choice (extricated by semantic data extraction) and the start activity (from the SOA) are stored in the knowledge-based repository, whose structure makes the framework intelligent. Information in the knowledge-based storage is various leveled. Information within the lowest level is "fact"; that within the middle level is "rules and processes" (activity); that within the most noteworthy level is "strategy" (choice). The structure is chosen by the characteristic of the information itself. The initial information can be both organized and unstructured, and knowledge-based storage can make them modular. In expansion, completely different information layers are all stamped with validity, which suggests that questionable data do not exist. At that point, a knowledge-based repository transfers information to cloud servers. Cloud servers are where information is stored and shared. Cloud servers can have either physical or virtual frameworks, and it ensures beyond any doubt that information-accessing dependable personnel (such as specialists) can access information remotely through the web. Cloud servers provide quick and continuous communication; in the meantime, they are successful because users have to pay for what they require, helping to avoid additional costs. At that point, the framework

achieves human-to-machine interaction. This is actualized within the genuine world, where it can be used to prevent illnesses, such as heart disease, cancer, and hypertension. The utilized SOA does not guarantee security, whereas performed errands are organized from the choice. It progresses in QoS and vitality. Our model also provides secure strategies that avoid secrecy issues.

4. Efficient Data-Balancing for CPS

Our CPS is designed to support brain monitoring. This model was proposed and tested to analyze the physical domain and the human domain. In this model, data are stored and shared to be accessed by medical practitioners in the data management domain. The distributed system is deployed with the support of fog nodes. This module includes three processes:

- A node advertisement process.
- A node selection and recruitment process.
- Optimal distance determination with mid-point.

4.1. Node Advertising Module

This module functions differently than an IP network because an IP network is used to make an agent discovery phase for a foreign agent and a home agent. The sensors use advertisements to confirm whether it is coupled to its respective home network or the foreign network. This advertisement process helps sensors advertise their lifetime within the network. The lifetime of the sensors in the WSNs is associated with time constraints so that it is more important to determine the remaining lifetime of the sensors (RLS). Let us assume that the sensors are homogeneous and possess the same physical capability as the communication range and the sensing power. The location of the sensor node is stationary. The location of the sensor is stationary or mobile. The stationary location of the sensors and actuators is only used for monitoring the static objects (patients). The sensors can communicate within the communication range using a multi-hop process. The remaining energy of the sensors (RES) defines the RLS. Furthermore, we believe that the remaining lifetime of each sensor is advertised when competing for some particular cycles for receiving and sending messages. The packets can also be retransmitted if the WSN is unstable. Thus, we also focus on the loss rate and link quality prior to advertising the lifetime of each sensor. Therefore, we can define the RLSs after determining the consumed energy for message transmission. Therefore, the RLS is the ratio of the remaining energy to the set initial energy for each sensor, which can be calculated as follows.

$$R_l = \frac{E_i - \sum_{i=0}^{T_C} \times i(E_p) \times N(E_p) \times \beta(E_p) \times E_{\Delta s} \times \varpi \times R}{E_i},$$

$$R_l = 1 - \frac{E_i - \sum_{i=0}^{T_C} \times i(E_p) \times N(E_p) \times \beta(E_p) \times E_{\Delta s} \times \varpi \times R}{E_i}, \quad (1)$$

where R_l is the remaining lifetime of the k sensor, E_i is the initial energy of the sensor, T_C is the transmission cycles for monitoring the events, $N(E_p)$ is the number of the packets received by each sensor device during the communication, $\beta(E_p)$ is number of the retransmitted packets,$E_{\Delta s}$ is the amount of energy consumed by each sensor device for a single received packet, ϖ is the number of reply messages sent to each sensor device, and R is number of the retransmissions experienced by each sensor device.

4.2. Node Selection and Recruitment Module

The objective of recruiting the sensor node and selecting legitimate actuators helps to improve throughput and decrease latency. The recruitment process is employed if the actuator hub (the cluster head node) does not discover sufficient sensor nodes in its cluster space. As a result, the actuator node starts the recruitment request from another actuator

node. First, the actuator node checks its zone regions by sending a recruitment request. If the actuator does not discover the required nodes in its neighborhood, it broadcasts the multicasting message to recruit the nodes. When the actuator node comes to the nodes from its non-adjacent cluster, the pipelining-based method (which allows different practical units of a system to function synchronously) is applied to reduce the latency that could be caused by a long distance. Furthermore, the recruiting actuator first recruits the sensors from its neighbor and then recruits them from nonadjacent neighbors. Let us assume that the actuator node recruits the sensor nodes from other cluster regions. The actuator is static and gathers data about the person; thus, every monitoring point M_p requires sensor nodes via recruitment to gather the data from a human. The probability pr of a sensing requirement (N_r, P_r) can be calculated as

$$pr = \int_{N_r}^{n} fx(N \times \gamma, \omega)\partial N, \tag{2}$$

where pr is the probability of the recruited sensor, and N_r is the recruited sensor.

The probability of checking point M_p can be decided when the activity is completed at another checking point M_{p1}, which is $\partial_{p1,p}$. Thus, with the probability of having to observe point p, information can be detected utilizing recruitment node N_r, decided by

$$N_{r(d)} = 1 - \sum_{p}^{p=\infty} p\left(1 - \left(\partial_{p1,p} \int_{N_r}^{n} fx(N \times \gamma, \omega)\partial N\left(1 - \sum_{t=0}^{n} t(1 - \partial_{p0,p1})\right)\right)\right), \tag{3}$$

where $N(r(d))$ is the recruited sensor sensing data, $\partial_{p1,p}$ is the distance of the recruited sensor from its domain to the recruiting sensor's domain, and t is recruitment time.

Once the recruitment sensor node begins to sense the data, if the sum of the data is more than the detecting capability of the deployed and recruitment node, the actuator starts the extra sensor recruitment process from neighboring and non-neighboring cluster spaces, given by

$$A_{rec} = 1 - \prod_{N_{r\in R}} N \times 1 - \sum_{p}^{p=\infty} p\left(1 - \left(\partial_{p1,p} \int_{N_r}^{n} fx(N \times \gamma, \omega)\partial N\left(1 - \sum_{t=0}^{n} t(1 - \partial_{p0,p1})\right)\right)\right), \tag{4}$$

where A_{rec} is the recruiting actuator, and R is the cluster domain that gives the sensor as a recruitment sensor.

After an additional sensor recruitment process, we obtain a new vector P_r' that illustrates the probability of monitoring point M_p, which needs to be covered by recruiting the additional sensor nodes.

$$P_r' = \begin{cases} \frac{P_r - A_{rec}}{1 - A_{rec}}, & \text{if } A_{rec} \leq P, \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

where P_r' is the new vector that indicates the probability of a grid station.

4.3. Optimal Distance Determination with Mid-Point Module

Moreover, the arrangement of the actuators is vital and might influence the execution performance and coverage. Therefore, the actuators should cover the entire placed sensor nodes. As a result, a mid-point calculation is connected to adjust the correct position of the sensor nodes explained in Algorithm 1. Thus, the optimal number of actuators A_{opt} can be obtained as

$$A_{opt} = \left\{ \sqrt{\frac{S_n}{2\pi}} \times \sqrt{\frac{FS_e}{MP_e}} \times \frac{A_{net}}{D_{avg}} \right\}, \tag{6}$$

where S_n is the number of the sensor nodes, FS_e is the amplifier energy of free space, MP_e is the multipath energy, A_{net} is the network area, and D_{avg} is the mean distance from actor to base station.

Algorithm 1 Optimized distance determination from sensor node to actuator.

Input: r in
Output: r_{iin} out
1: **Initialization:** $\{\gamma o$: *Origin;* γe: *Each point; r: Distance;* r_{iin}: *Initial centroid distance;* r_s: *Sorting distance*$\}$
2: **Determine** r between γo & γe
3: **Set** r_s in ascending order
4: **Separate** the r into A_{opt} equal sets
5: **if** $midpoint == r_{iin}$ **then**
6: **Set** r_{iin}
7: **end if**

Algorithm 1 separates the optimal number of actuators with respect to the clusters. Each cluster is headed by an actuator. The actuator broadcasts the packet to the sensors to form the cluster. The packet comprises the actuator's location and identity. On receipt of the packet, the sensor device acknowledges with its identity and residual energy. In existing approaches, when a sensor device receives a cluster formation message from more than one actuator, it chooses to join only the nearest actuator based on the location inserted in the packet. However, this cluster formation association increases the path length because it could be that the actuator is located far from the base station. Thus, avoiding back transmissions, an average midpoint of the optimal actuator algorithm is useful. If the sensors receive a higher received signal strength indicator (RSSI) from the base station compared to the actuator, then the sensor device should send data to the base station rather than the actuator.

Similarly, the sensor can calculate the distance between itself and the base station, and then determine its midpoint. Based on the midpoint, the sensor decides to send the data either to the actuator or the base station, as depicted in Figure 3.

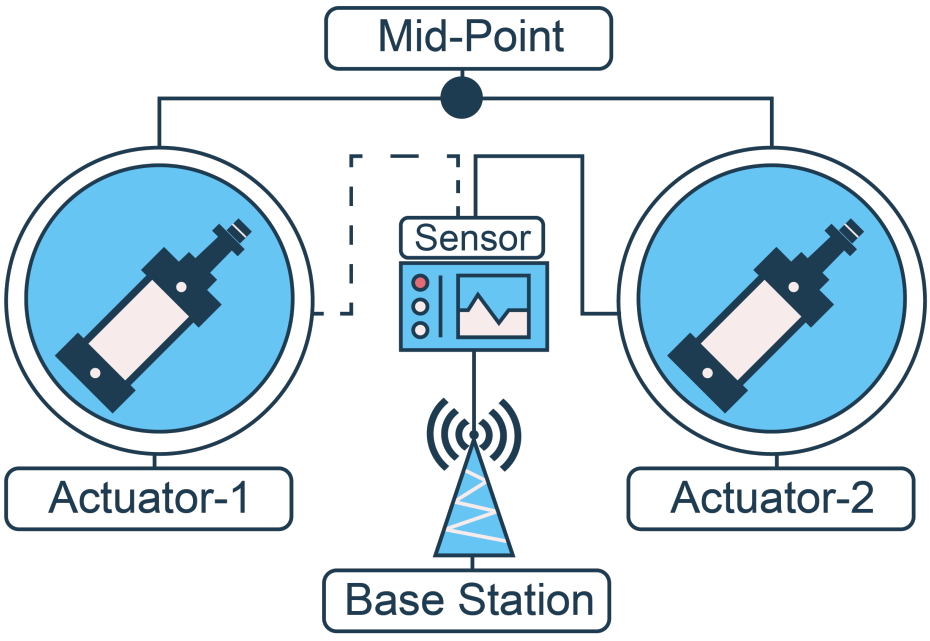

Figure 3. Average midpoint of the optimal actuator.

The sensor node relates to Actuator-1 due to the receipt of a higher signal strength, but Actuator-1 is far from the base station compared to Actuator-2. As a result, additional energy is consumed, and the delay is extended. Thus, the proposed algorithm is applied to determine the midpoint to reduce the delay and improve the energy efficiency. Table 1 provides a description of used notations for the node selection and recruitment module.

Table 1. Variables used in the recruitment process.

Notation	Description
A_{net}	Network area
A_{rec}	Recruiting actuator
D_avg	Mean distance from actor to base station
d_N	Distance of the recruited sensor from the recruiting cluster head sensor
$\partial_{p1,p}$	Distance of the recruited sensor from its domain to the recruiting sensor's domain
FS_e	Amplifier energy of free space
M_p	Monitoring point
MP_e	Multipath energy
$N_{r(d)}$	Recruited sensor sensing data
N_r	Recruited sensor
P'_r	New vector that indicates the probability of a grid station
P_r	Recruited sensor sensing data
R	Probability of the recruited sensor
S_n	Number of sensor nodes
t	Recruitment time
C_{adj}	Adjacent cluster domain or Nonadjacent cluster domain
N_{rec}	Number of recruited sensors

5. Testing Results

Upon receipt of Δd messages, the actuator regenerates each $b_{k1,}i + b_{k2,}i$, which can be calculated by an equation.

To validate the effectiveness of the proposed system, we demonstrated a framework with C++ and tested it on NS3. The tests were conducted on a tablet PC with a 3.0 GHz Intel inside Core i3 CPU and 4 GB of Smash. The test machine used a 64-bit adaptation of Windows 8. The network includes the parameters listed in Table 2. EDB-CPS was compared with the state-of-the-art methods: A-CPS [26], BSMIC [27], and StreamLAB [33]. Based on the testing process, the following results were calculated:

- throughput;
- hop-to-hop delay.

Table 2. Parameters in the simulation setup.

Used Parameters	Detailed Parameters
Transmission range	30 m
Sensing range of the node	25 m
Initial energy of the node	5 Joules
Bandwidth of the node	45 Kb/s
Simulation time	36 min
Number of sensors	360
Network size	600 × 600 m^2
Number of hops in the network	18 Maximum
Number of clusters	06
Buffering capacity	50 Packets buffering capacity at each node
Mobility model	Lattice mobility model [34].
Mobility (Speed of the nodes)	0 m/s to 15 m/s
Data packet size	128 bytes
Initial pause time	30 Seconds
R_x energy	14 Mw
T_x energy	18 Mw
Power intensity	−18 dBm to 12 dBm
Sink location in each region	(0, 230)
Contending paradigm	IA-CPS [26], BSMIC [27], and StreamLAB [33]
Mobility (speed of the nodes)	0 m/s to 20 m/s

5.1. Average Throughput Performance

The throughput can be utilized to record the sum of organized information transmission in a specific period of time. A much higher throughput provides higher proficiency, which suggests little delay within the information transmission, a quicker transmission speed, and greater sensitivity to outside impacts. Figure 4a,b demonstrate the trade-off between an average throughput and the allotted time. Two different scenarios have been created to test and validate the performance of the proposed EDB-CPS and contending methods: IA-CPS, BS-MIC, and StreamLAB. In the first scenario, no malicious nodes are generated; in the second scenario, 5% of the nodes are malicious have been created. Figure 4a demonstrates the result of the first scenario, and Figure 4b shows the result of the second scenario. Figure 4a shows that the proposed EDB-CPS was able to receive a 445.2 kb/s throughput during the maximum 30 min, while the contending modes demonstrate much less throughput as compared to the proposed model. The BS-MIC, StreamLAB, and IA-CPS produced 441.3 kb/s, 438.3 kb/s, and 437.9 kb/s of throughput, respectively.

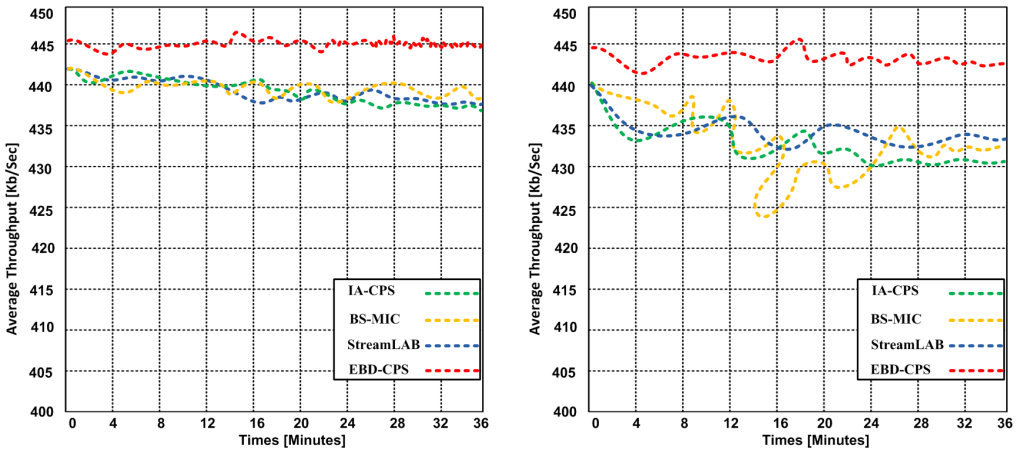

Figure 4. (**a**) Average throughput performance without malicious nodes of the proposed EDB-CPS and the contending models: IA-CPS, BS-MIC, and StreamLAB. (**b**) Average throughput performance with 5% malicious nodes for the EDB-CPS and contending models: IA-CPS, BS-MIC, and StreamLAB.

When the malicious nodes were generated, the throughput of the proposed model marginally reduced, while other contending modes reduced more throughput as compared to our proposed system (EBS-CPS).

The results demonstrate that the proposed system yields an approximately 443.2 kb/s throughput, while StreamLAB, IA-CPS, and BS-MIC obtained throughputs of 434.2 kb/s, 432.3 kb/s 429.3 kb/s, respectively. Thus, the proposed system shows better throughput in both scenarios: without and with malicious nodes.

Throughput of the single node T_p can be obtained as

$$T_p = \sum_{i=1}^{m} \frac{\eta_d^k}{\delta_\rho^{z,\mu}}, \tag{7}$$

where m is the total number of participating devices in this system, ρ is the density of nodes, $\delta_\rho^{z,\mu}$ is the single-hop delay of each node in many cases, and η_d^k represents the amount of data sent from the node to the adjacent actuator. The above equation only shows the amount of data sent during the allotted time because the data sending and receiving rates are the same in our case. Due to the distribution randomness in the WSN, the throughput can be approximately signified by the average throughput of a single sensor device and the

number of sensors participating in the entire network. Thus, the throughput of the entire network can be obtained as

$$T_p = S \cdot T_s = 2\pi Z\rho \cdot \frac{1}{Z}\int_0^Z \frac{\omega_d^k}{\delta_\rho^{z,\mu}} dk, \tag{8}$$

where S represents the total number of sensors in the network (the product of the sensor device density and the area of the whole network), while T_s calculates the average throughput of a single sensor. This formula can estimate the throughput of the whole network.

5.2. Hop-by-Hop Delay

The trade-off between the number of hops and the hop-by-hop delay is shown in Figure 5a,b. As the number of hops increases, the hop-by-hop delay also increases. We tested two scenarios. In the first scenario, the total number of hops is 27, and the second scenario uses a maximum of 54 nodes. Based on the results, we compared the hop-to-hop delay of our proposed EDB-CPS and that of the contending models: IA-CPS, BS-MIC, and StreamLAB. We observed that the proposed model shows a lower hop-to-hop delay compared with the contending models. The standard average delay D_a for a single hop can be obtained by

$$D_a = \sum_{i=0}^{n} \frac{S_{ts}}{d_{cy}} \times \frac{S_{ts}}{2}. \tag{9}$$

The time for the first $n-1$ slot can be obtained as $S_{ts} = A_{wt} = 1, 2, \cdots, n-1$.

$$D_a = (n-1)P(S_{ts}) \times \frac{S_{ts}}{2} + P_t(S_{ts}) \times \frac{S_{ts}}{2}. \tag{10}$$

When a sensor node sends the number of beacons b_n per duty cycle, the one-hop delay is calculated as

$$D_a = \left[\frac{(n-1)A_{wt^2}}{2d_{cy}} + \frac{\left\{(A_{wt}) + (1-b_n) \times 2d_{cy}\right\}^2}{2d_{cy}} \right], \tag{11}$$

where $t_n = \left(\frac{b_n \times d_{cy}}{P + A_{wt}} \right)$.

Hence, the delay for multiple-hop $t(D_a)$ can be obtained as

$$t(D_a) = \left\{ D_a \cdot t_{hops} \right\}, \tag{12}$$

where S_{ts} is a short time slot either for listening or sleep, d_{cy} is a duty cycle time length for the nodes, P is permeable, P_t is the total number of permeable, t_n is the total number of slots, t_{hops} is the total number of hops, and A_{wt} is the sensor wait time.

Based on the result depicted in Figure 5a, it is observed that the proposed EDB-CPS shows a hop-to-hop delay of 0.05 ms with 27 hops, whereas the contending models BS-MIC, StreamLAB, and IA-CPS show a hop-to-hop delay of 0.609, 0.642, and 0.667 ms, respectively. When the number of hops increases to 54, the hop-to-hop delay increases at a similar rate. Figure 5b shows that the proposed EDB-CPS has a hop-to-hop delay of 0.0781 ms, whereas the contending models achieve a hop-to-hop delay of 0.094, 0.96, and 0.097 ms for StreamLAB, IA-CPS, and BS-MIC, respectively. Thus, the proposed EDB-CPS shows a better performance than the contending models.

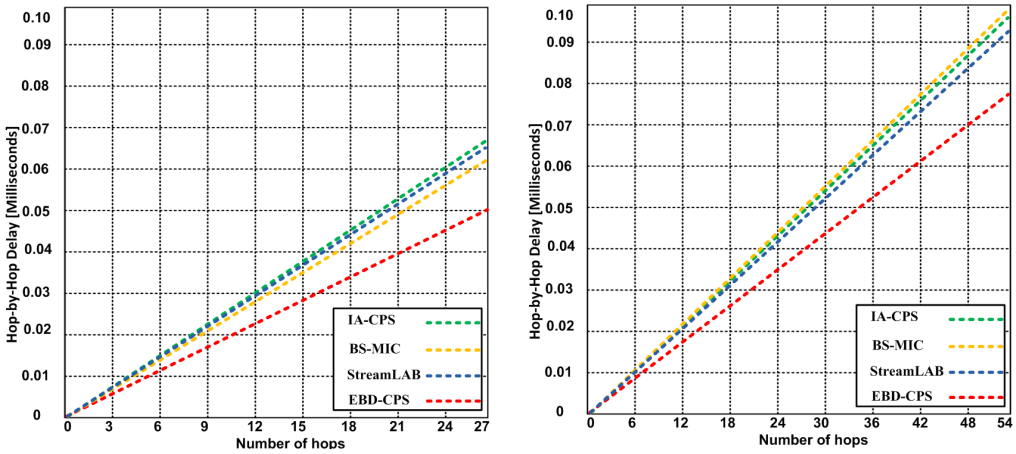

Figure 5. (**a**) Number of hops versus the required amount of time for the proposed EDB-CPS and the contending models, IA-CPS, BS-MIC, and StreamLAB, with a maximum of 27 hops. (**b**) Number of hops versus the required amount of time for the proposed EDB-CPS and the contending models, IA-CPS, BS-MIC, and StreamLAB, with a maximum of 54 hops.

6. Discussion of Results

Our proposed EDB-CPS model consists of five phases that constitute an effective and secure cyber-physical system. Although the IA-CPS presented in Reference [26] introduced a clustering strategy in a WSN to minimize energy consumption, it failed to account for circumstances of a limited number of sensors in a cluster, which likely occurs in practical use. The sensor recruitment and node selection modules provide a solution for clusters with a minimum number of sensors, where the actuator is responsible for sending a recruitment request to the other clusters to determine the expected number of sensors. The mobile robot sensors monitor the moving patients (there is a certain region where any mobile robot sensor that enters may become a head static head sensor or a mobile robot sensor), and the actuator (head sensor) is identified in this model. The latency mobile model is deployed to handle the moment of patients [34]. Therefore, it improves the system's reliability but sometimes reduces efficiency because of the actuator, as it is located farther, which causes a back-transmission penalty. To overcome this limitation, we introduced an average midpoint in the optimal actuator algorithm (Algorithm 1). In the future, we can specify a certain range in which sensors can directly send data to the selected BS without redundant calculations to improve work performance. Furthermore, the sensor advertisement is quite efficient; it showed a better performance compared to other state-of-the-art methods (A-CPS, BSMIC, and StreamLAB) in terms of throughput and hop-to-hop delay. The results in Table 3 confirm the effectiveness of our proposed model (EDB-CPS). The model proposed in Reference [27] only supports static sensors, whereas our proposed model supports static and mobile robot sensors. To handle mobile sensors, a lattice mobility model can be deployed. In Table 3, the proposed EDB-CPS produces a better throughput of 445.2 kb/s without malicious nodes. When the 5% malicious nodes are generated, then the proposed EDB-CPS also produces a better throughput as compared to contending methods. The proposed EDB-CPS gets a better hop-to-hop delay (it means lower hop-to-hop delay). The lower hop-to-hop delay causes increasing the throughput. The reason for getting the better performance of the EDB-CPS is to use NSR, NA, and ODDMP processes.

Table 3. Comparative analysis of the proposed EDB-CPS and the contending models—IA-CPS, BS-MIC, and StreamLAB.

Model	Throughput without Malicious Nodes	Throughput with 5% Malicious Nodes	Hop-to-Hop Delay with 27 Hops	Hop-to-Hop Delay with 54 Hops
LA-CPS	437.9 kb/s	432.3 kb/s	0.667 ms	0.667 ms
BS-MIC	441.3 kb/s	429.3 kb/s	0.609 ms	0.097 ms
StreamLAB	438.3 kb/s	434.2 kb/s	0.642 ms	0.094 ms
EDB-CPS	445.2 kb/s	443.2 kb/s	0.05 ms	0.078 ms

7. Conclusions and Future Work

This section concludes the paper and demonstrates the performance of the proposed EDB-CPS and the contending models, IA-CPS, BS-MIC, and StreamLAB.

7.1. Conclusions

Efficient data-balancing cyber-physical systems over fog computing is introduced in this paper for monitoring of the human central nervous system. The EDB-CPS aims to provide faster data access to improve throughput and reduce hop-to-hop delay. The EDB-CPS comprises five components: node selection and recruitment, a BSN, physical domain and data processing, an SOA, and a data management domain. These components have successfully collected data from humans and have stored data on via fog-based cloud computing. Furthermore, the EDB-CPS involves two modules: node advertisement, and node selection with recruitment. These modules enable a data-balancing load in the WSNs. The proposed EDB-CPS model was programmed with C++ and changed C++ into an object tool command language (OTCL) that supports NS3. Based on the results, the proposed EDB-CPS outperforms the contending models—IA-CPS, BS-MIC, and StreamLAB—in terms of throughput and hop-to-hop delay.

7.2. Future Work

The proposed EDB-CPS will be tested on real devices, and experiments will be conducted in hospitals. A new mobility model will be included to support mobile robots in monitoring moving patients in hospitals. Additionally, additional parameters (e.g., energy efficiency, security, and reliability) will be measured. Finally, a privacy-preserving model will be included to protect the confidentiality of data from unauthorized users.

Author Contributions: Conceptualization and methodology, M.A. (Muder Almiani) and A.R.; writing—original draft preparation and visualization, B.A. and M.A. (Munif Alotaibi); writing—review and editing, S.A. and A.A.; funding acquisition, B.A. and A.A. All authors have read and agreed to the published version of the manuscript.

Funding: This work was partially supported by the Sensors Networks and Cellular Systems (SNCS) Research Center under Grant 1442-002. This work was also supported by Taif University Supporting Project number (TURSP-2020/302), Taif University, Taif, Saudi Arabia.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare that there are no conflict of interest.

References

1. Sood, S.K.; Rawat, K.S. A fog assisted intelligent framework based on cyber-physical system for safe evacuation in panic situations. *Comput. Commun.* **2021**, *178*, 297–306. [CrossRef] [PubMed]
2. Marwedel, P. *Embedded System Design: Embedded Systems Foundations of Cyber-Physical Systems, and the Internet of Things*; Springer Nature: Basingstoke, UK, 2021.
3. Törngren, M.; Grogan, P.T. How to deal with the complexity of future cyber-physical systems? *Designs* **2018**, *2*, 40. [CrossRef]
4. Sadiku, M.N.O.; Wang, Y.; Cui, S.; Musa, S.M. Cyber-physical systems: A literature review. *Eur. Sci. J.* **2017**, *13*, 52–58. [CrossRef]

5. Ali, S.; Qaisar, S.B.; Saeed, H.; Khan, M.F.; Naeem, M.; Anpalagan, A. Network challenges for cyber physical systems with tiny wireless devices: A case study on reliable pipeline condition monitoring. *Sensors* **2015**, *4* , 7172–7205. [CrossRef]
6. Letmathe, P.; Schinner, M. Competence management in the age of cyber physical systems. In *Industrial Internet of Things*; Springer: Cham, Switzerland, 2017; pp. 595–614.
7. Alguliyev, R.; Imamverdiyev, Y.; Sukhostat, L. Cyber-physical systems and their security issues. *Comput. Ind.* **2018**, *100*, 212–223. [CrossRef]
8. Humayed, A.; Lin, J.; Li, F.; Luo, B. Cyber-physical systems security—A survey. *IEEE Internet Things J.* **2017**, *4*, 1802–1831. [CrossRef]
9. Cernian, A.; Vasile, N.; Sacala, I.S. Fostering Cyber-Physical Social Systems through an Ontological Approach to Personality Classification Based on Social Media Posts. *Sensors* **2021**, *21*, 6611. [CrossRef]
10. Razaque, A.; Amsaad, F.; Khan, M.J.; Hariri, S.; Chen, S.; Siting, C.; Ji, X. Survey: Cybersecurity vulnerabilities, attacks and solutions in the medical domain. *IEEE Access* **2019**, *7*, 168774–168797. [CrossRef]
11. Aciti, C.; Cayssials, R.; Ferro, E.; Urriza, J.; Orozco, J. Embedded real-time systems in cyber-physical applications: A frequency domain analysis methodology. *Int. J. Gen. Syst.* **2020**, *49*, 201–221. [CrossRef]
12. Jamal, A.A.; Majid, A.A.M.; Konev, A.; Kosachenko, T.; Shelupanov, A. A review on security analysis of cyber physical systems using Machine learning. *Mater. Today Proc.* 2021, *in press*.
13. Yilma, B.A.; Panetto, H.; Naudet, Y. Systemic formalisation of Cyber-Physical-Social System (CPSS): A systematic literature review. *Comput. Ind.* **2021**, *129*, 103458. [CrossRef]
14. Tyagi, A.K.; Aswathy, S.U.; Aghila, G.; Sreenath, N. AARIN: Affordable, accurate, reliable and innovative mechanism to protect a medical cyber-physical system using blockchain technology. *Int. J. Intell. Netw.* **2021**, *2*, 175–183. [CrossRef]
15. Ravikumar, G.; Hyder, B.; Govindarasu, M. Hardware-in-the-loop cps security architecture for der monitoring and control applications. In Proceedings of the 2020 IEEE Texas Power and Energy Conference (TPEC), College Station, TX, USA, 6–7 February 2020; pp. 1–5.
16. Gatouillat, A.; Badr, Y.; Massot, B.; Sejdic, E. Internet of medical things: A review of recent contributions dealing with cyber-physical systems in medicine. *IEEE Internet Things J.* **2018**, *5*, 3810–3822. [CrossRef]
17. Razaque, A.; Frej, M.B.H.; Alotaibi, B.; Alotaibi, M. Privacy Preservation Models for Third-Party Auditor over Cloud Computing: A Survey. *Electronics* **2021**, *10*, 2721. [CrossRef]
18. Lv, Z.; Han, Y.; Singh, A.K.; Manogaran, G.; Lv, H. Trustworthiness in industrial IoT systems based on artificial intelligence. *IEEE Trans. Ind. Inform.* **2020**, *17*, 1496–1504. [CrossRef]
19. Wang, H.; Fan, K.; Zhang, K.; Wang, Z.; Li, H.; Yang, Y. Secure and Efficient Data Privacy-preserving Scheme for Mobile Cyber Physical Systems. *IEEE Internet Things J.* **2021**, 1. [CrossRef]
20. Huang, H.; Savkin, A.V.; Ding, M.; Huang, C. Mobile robots in wireless sensor networks: A survey on tasks. *Comput. Netw.* **2019**, *148*, 1–19. [CrossRef]
21. Weichhart, G.; Panetto, H.; Molina, A. Interoperability in the cyber-physical manufacturing enterprise. *Annu. Rev. Control.* **2021**, *51*, 346–356. [CrossRef]
22. Zheng, Y.; Li, Z.; Xu, X.; Zhao, Q. Dynamic defenses in cyber security: Techniques, methods and challenges. *Digit. Commun. Netw.* 2021, *in press*. [CrossRef]
23. Razaque, A.; Al Ajlan, A.; Melaoune, N.; Alotaibi, M.; Alotaibi, B.; Dias, I.; Oad, A.; Hariri, S.; Zhao, C. Avoidance of Cybersecurity Threats with the Deployment of a Web-Based Blockchain-Enabled Cybersecurity Awareness System. *Appl. Sci.* **2021**, *11*, 7880. [CrossRef]
24. Bordel, B.; Alcarria, R.; Robles, T.; Martín, D. Cyber–physical systems: Extending pervasive sensing from control theory to the Internet of Things. *Pervasive Mob. Comput.* **2017**, *40*, 156–184. [CrossRef]
25. Chang, K.C.; Chu, K.C.; Wang, H.C.; Lin, Y.C.; Pan, J.S. Agent-based middleware framework using distributed CPS for improving resource utilization in smart city. *Future Gener. Comput. Syst.* **2020**, *108*, 445–453. [CrossRef]
26. Gomez, H.D.; Garcia-Rodriguez, J.; Azorin-Lopez, J.; Tomas, D.; Fuster-Guillo, A.; Mora-Mora, H. IA-CPS: Intelligent architecture for cyber-physical systems management. *J. Comput. Sci.* **2021**, *53*, 101409. [CrossRef]
27. Jiang, Y.; Liu, X.; Kang, K.; Wang, Z.; Zhong, R.Y.; Huang, G.Q. Blockchain-enabled cyber-physical smart modular integrated construction. *Comput. Ind.* **2021**, *133*, 103553. [CrossRef]
28. Latif, S.A.; Wen, F.B.X.; Iwendi, C.; Li-li, F.W.; Mohsin, S.M.; Han, Z.; Band, S.S. AI-empowered, blockchain and SDN integrated security architecture for IoT network of cyber physical systems. *Comput. Commun.* **2021**, *181*, PP.274–283. [CrossRef]
29. Kabashkin, I. Reliability of cluster-based nodes in wireless sensor networks of cyber physical systems. *Procedia Comput. Sci.* **2019**, *151*, 313–320. [CrossRef]
30. Li, S.; Zhao, P. Big data driven vehicle battery management method: A novel cyber-physical system perspective. *J. Energy Storage* **2021**, *33*, 102064. [CrossRef]
31. Tripathy, A.K.; Tripathy, P.K.; Mohapatra, A.G.; Ray, N.K.; Mohanty, S.P. WeDoShare: A ridesharing framework in transportation cyber-physical system for sustainable mobility in smart cities. *IEEE Consum. Electron. Mag.* **2020**, *9*, 41–48. [CrossRef]
32. Deka, L.; Khan, S.M.; Chowdhury, M.; Ayres, N. Transportation cyber-physical system and its importance for future mobility. In *Transportation Cyber-Physical Systems*; Elsevier: Amsterdam, The Netherlands, 2018; pp. 1–20.

33. Faymonville, P.; Finkbeiner, B.; Schledjewski, M.; Schwenger, M.; Stenger, M.; Tentrup, L.; Torfah, H. StreamLAB: stream-based monitoring of cyber-physical systems. In Proceedings of the International Conference on Computer Aided Verification, New York, NY, USA, 15–18 July 2019; Springer: Cham, Switzerland; pp. 421–431.
34. Al-Rahayfeh, A.; Razaque, A.; Jararweh, Y.; Almiani, M. Location-Based Lattice Mobility Model for Wireless Sensor Networks. *Sensors* **2018**, *18*, 4096. [CrossRef]