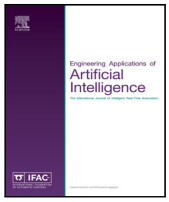
Engineering Applications of Artificial Intelligence 129 (2024) 107612

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai



Securing healthcare data in industrial cyber-physical systems using combining deep learning and blockchain technology

Mazin Abed Mohammed [a,h,i,*], Abdullah Lakhan [b,h,i], Dilovan Asaad Zebari [c], Mohd Khanapi Abd Ghani [d], Haydar Abdulameer Marhoon [e,f], Karrar Hameed Abdulkareem [g], Jan Nedoma [h], Radek Martinek [i]

[a] Department of Artificial Intelligence, College of Computer Science and Information Technology, University of Anbar, Anbar 31001, Iraq
[b] Department of Cybersecurity and Computer Science, Dawood University of Engineering and Technology, Karachi City 74800, Sindh, Pakistan
[c] Department of Computer Science, College of Science, Nawroz University, Duhok 42001, Kurdistan Region, Iraq
[d] Department of Software Engineering, Faculty of Information and Communication Technology, Universiti Teknikal Malaysia Melaka (UTeM), Malaysia
[e] Information and Communication Technology Research Group, Scientific Research Center, Al-Ayen University, Thi-Qar, Iraq
[f] College of Computer Sciences and Information Technology, University of Kerbala, Karbala, Iraq
[g] College of Agriculture, Al-Muthanna University, Samawah 66001, Iraq
[h] Department of Telecommunications, VSB-Technical University of Ostrava, 70800 Ostrava, Czech Republic
[i] Department of Cybernetics and Biomedical Engineering, VSB-Technical University of Ostrava, 70800 Ostrava, Czech Republic

ARTICLE INFO

Dataset link: https://github.com/ABDULLAH-RAZA/Assignment-

Keywords:
Blockchain
Deep learning
Industrial cyber-physical systems
Pattern proof malware validation
LSTM
Cyber-attacks
Healthcare data

ABSTRACT

Industrial cyber–physical systems (ICPS) are emerging platforms for various industrial applications. For instance, remote healthcare monitoring, real-time healthcare data generation, and many other applications have been integrated into the ICPS platform. These healthcare applications encompass workflow tasks, such as processing within hospitals, laboratory tests, and insurance companies for patient payments, which necessitate a sequential flow. The external wireless, fog, and cloud services within ICPS face security issues that impact end-users' healthcare applications. Blockchain technology offers an optimal solution for ICPS-enabled applications. However, blockchain technology for the ICPS platform is still vulnerable to cyberattacks, while microservices are essential for executing applications. This paper introduces the novel "Pattern-Proof Malware Validation" (PoPMV) algorithm designed for blockchain in ICPS. It exploits a deep learning model (LSTM) with reinforcement learning techniques to receive feedback and rewards in real-time. The primary objective is to mitigate security vulnerabilities, enhance processing speed, identify both familiar and unfamiliar attacks, and optimize the functionality of ICPS. Simulations demonstrate the superiority of the proposed approach compared to current blockchain frameworks, showcasing dynamic allocation of microservices and improved security with comprehensive attack detection by 30%.

1. Introduction

Industrial cyber–physical systems (ICPS) are a new paradigm for many Internet of Medical Things (IoMT) devices, nodes, and networks that are different from each other to work together. The ICPS supports many medical services based on Healthcare Industrial 4.0. These services use a network of biosensors to monitor patients' health in real-time. The main idea behind ICPS is to connect different industries and share their data (Kayan et al., 2022). Artificial intelligence is a broad domain into which various sub-domains, such as machine learning, deep learning, and others, have been subdivided. These schemes are optimal and make the optimal decision for the industrial applications in the ICPS network (Zhang et al., 2021). The workflow in healthcare is the combination of different services, which are scheduled from different nodes, such as patient login data, doctor counseling, live health records, payments, and other services processed in sequence on fog and cloud computing (Agrawal and Kumar, 2022). In ICPS, on the other hand, the nodes are homogeneous and share their data based on a trust contract about the data. For instance, all similar hospital industries can share the same patient data among all branches

* Corresponding author at: Department of Artificial Intelligence, College of Computer Science and Information Technology, University of Anbar, Anbar 31001, Iraq.
E-mail addresses: mazinalshujeary@uoanbar.edu.iq (M.A. Mohammed), abdullah.lakhan@duet.edu.pk (A. Lakhan), dilovan.majeed@nawroz.edu.krd (D.A. Zebari), khanapi@utem.edu.my (M.K.A. Ghani), haydar@alayen.edu.iq (H.A. Marhoon), khak9784@mu.edu.iq (K.H. Abdulkareem), jan.nedoma@vsb.cz (J. Nedoma), radek.martinek@vsb.cz (R. Martinek).

https://doi.org/10.1016/j.engappai.2023.107612
Received 6 September 2023; Received in revised form 17 November 2023; Accepted 23 November 2023
Available online 6 December 2023

and be easily accessible from anywhere. Furthermore, these hospitals allow different sectors, such as payment payers, insurance industries, and others, to embed their services with them and share trusted data for processing. Many security algorithms, like Advanced Encryption Standards (AES), are fully homomorphic and were made to handle the security and privacy of communication between these nodes. Based on these security methods, the three schemes enabled malware, and attack identification was suggested in these studies. Many studies used pattern-based malware detection, heuristic-based malware detection, and signature-based malware detection. However, these ICPS methods only worked for healthcare applications with known and fixed nodes that were all the same (Tanha et al., 2022).

Recently, blockchain-based decentralized and unknown heterogeneous nodes enabled data validation and immutable data sharing mechanisms introduced for healthcare applications. Blockchain technology is reliable and effective. It checks data and processes between known and unknown nodes using a method that cannot be changed. Three blockchain technologies are implemented in the public, private, and community schemes for healthcare applications. Different healthcare nodes offer additional services that are added to the workflow for complete processing in the network (Rosado et al., 2022).

The Blockchain systems have been widely implemented machine learning techniques in IoMT-enabled healthcare workflows and generate huge amounts of healthcare data from different bio-sensors that are offloaded to the system for processing. However, existing blockchain-based ICPS caused the IoMT to face research difficulties as the research questions in these studies (Almaiah et al., 2022; Rasool et al., 2022; Elhoseny et al., 2022; Chandra and Matuska, 2022; Yu et al., 2022). We are discussing the different research questions among existing studies. (I): The blockchain technologies we have now need to improve at stopping cyberattacks, and this has caused colossal performance problems in healthcare applications. (II): The blockchain schemes' consensus methods could not detect known and unknown cyber-attacks (e.g., malware and benign) in connected physical nodes for considered workflows. (III): The current machine learning-enabled blockchain technologies cannot handle any malware attacks at the runtime of execution. The main limitation of existing public technologies is that they cannot recognize the runtime malware, which is very new and does not show any patterns. Therefore, new solutions are required to deal with the runtime malware in ICPS for applications.

We present the LSTM and reinforcement learning-empowered blockchain-enabled ICPS for healthcare services The study considered heterogeneous IoMT bio-sensors, mobile, fog, and cloud nodes and implemented blockchain schemes for microservices workflow applications. The study divided the scheduling problem into different states and formulated it as a Markovian decision problem. We have following contributions in manuscript.

- We present ICPS that consists of different layers, such as the IoMT biosensor layer, the fog layer, and the cloud layer, where different industries offer their microservices and run them under the implementation of blockchain technology, aware of known and unknown attacks in the network.
- The study formulates the problem as a partition Markov decision problem and suggests model-free reinforcement learning schemes to run the healthcare microservices-based workflow on different nodes. Each node is considered a state with attributes (e.g., state, action, timestamp, hashing, malware, benign, and deadline) and transacts processing data to another autonomous state for processing.
- The work presents the reinforcement learning and deep learning-enabled LSTM-empowered malware and benign schemes for each node during the processing and offloading of data between nodes.
- The study devises the Pattern Proof of Malware Validation (PoPMV) algorithm scheme that adds the microservice workflow goal method, dynamic programming-enabled scheduling, and blockchain malware validation scheme. The main goal is to identify the malware at the runtime of applications in ICPS.

The paper has following parts. Section 2 is about related work. Section 3 is about proposed method. Section 4 is about methodology. Section 5 is about performance evaluation. Section 6 is about conclusion and future work.

2. Existing work

Many studies suggested that, in practice, malware detection enabled schemes for healthcare applications. The study presented an industrial cyber–physical system (ICPS) based on Kayan et al. (2022) static heuristic malware detection scheme for healthcare applications. These ICPS frameworks designed based on industry 4.0. The study designed the list of attacks at the design stage and identified them in advance of healthcare applications. The compiled-time malware detection scheme in ICPS for the workflows of healthcare-integrated applications designed in these studies (Zhang et al., 2021; Agrawal and Kumar, 2022; Tanha et al., 2022). These studies investigated malware avoidance schemes to restrict attacks against the system. However, these malware detection schemes are annotated and static for the malware applications.

The dynamic malware detection schemes and frameworks suggested in these studies (Rosado et al., 2022; Almaiah et al., 2022; Rasool et al., 2022; Elhoseny et al., 2022; Chandra and Matuska, 2022; Yu et al., 2022). The goal is to identify the malware based on pattern, behavior, and heuristic schemes. In the distributed mobile fog cloud networks, the algorithms Advanced Standard Encryption (AES), MD5, CRC32, and RSA-based hashing are made for each workload of healthcare applications. For further analysis, these applications transfer hashed data between computing nodes, such as fog and clouds. However, these mechanisms worked on the system's single and node-to-node security attacks for healthcare applications. However, adaptive known and unknown malware identification was still present in the system.

Adaptive malware and unknown attacks like denial of service, Trojan horses, ransomware, backdoors, and passive attacks are hazardous to distributed healthcare IoT applications ICPS netowrks. These studies considered IoT ICPS assisted applications (Li et al., 2022; Zhu et al., 2022; Jayanetti et al., 2022; Kaur et al., 2022; Mastoi et al., 2021; McGibney and Bharti, 2022; Qu et al., 2022) suggested the adaptive known malware-enabled behavioral heuristic schemes based on machine learning and meta-heuristics in the distributed network. The goal is to handle malware attacks in the system. However, it is at the network level and is difficult to manage in the system. These applications suggested blockchain-enabled security frameworks (Abdelmoneem et al., 2020; Erol et al., 2023; Kumar et al., 2023; Aujla and Jindal, 2020) for healthcare applications. The reinforcement learning and long-term short memory (LSTM) enabled deep learning models widely implemented inside blockchain in the ICPS network for the healthcare workflows in these studies (Wang, 2020; Zheng et al., 2022; Yang et al., 2022; Baysal et al., 2023; Lakhan and Li, 2019; Rana et al., 2022). These studies considered the hashing validation based on smart contracts among ICPS heterogeneous nodes in the network. These studies (Manthiramoorthy et al., 2024; Arsyad et al., 2022; Utku, 2023) suggested different cloud security, blockchain and Covid-19 for healthcare microservices. The main goal is to reduce the security risks on microservice data in fog cloud networks. These studies (Lakhan et al., 2023b,a; Mohammed et al., 2023; Lakhan et al., 2022) suggested the blockchain technologies for heterogeneous healthcare nodes for service providing. The objective was to processed data on different nodes with the valid form and avoid from any security issues.

However, existing blockchain hashing techniques based on Sha-256 are inefficient for the malware and benign attacks for the healthcare workflows in ICPS heterogeneous. Even though deep learning models in blockchain technology based on Sha-256 are not malware efficient optimal for workflows. In this paper, the study presents the LSTM and reinforcement learning-empowered blockchain-enabled ICPS for healthcare services in networks of heterogeneous fog clouds. The study introduced the new pattern hashing inside blockchain technology that is most efficient against malware and cyber-attacks. The study divided the scheduling problem into different states and formulated it as a Markovian decision problem (Ren et al., 2022).

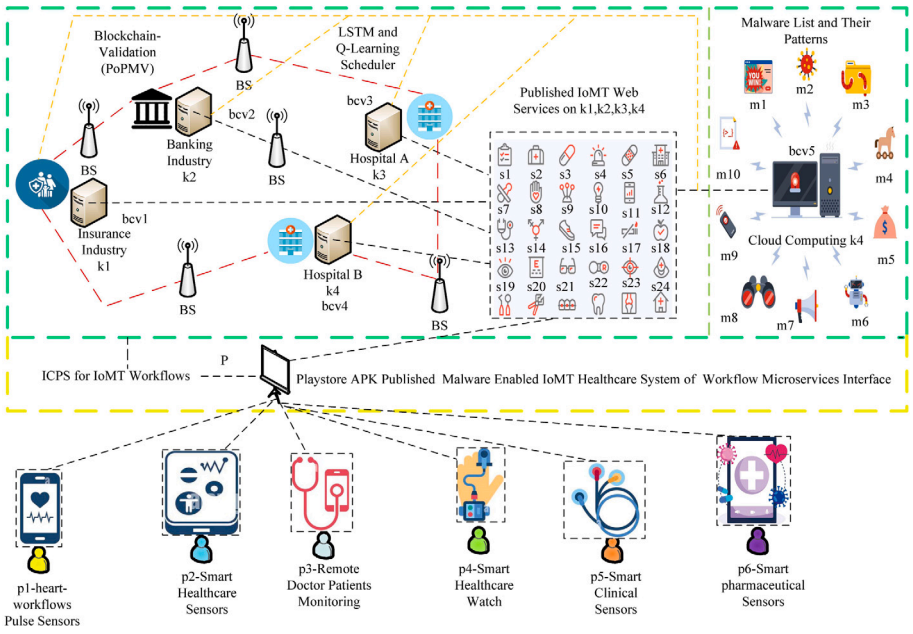


Fig. 1. An Improved Healthcare Industrial System Based on Deep-Learning Enabled Blockchain Technology.

3. Proposed blockchain based distributed healthcare system

The study presents the pattern attack efficient hashing scheme of Blockchain for the Internet of Medical Things enabled runtime healthcare microservice workflow as shown in Fig. 1. The main objective of the proposed system is to offer healthcare microservices with additional services. For instance, banking payment and insurance services for the patients and designing the sequence of microservices at runtime when different patients select them. The study implements different computing nodes, such as hospital servers, insurance servers, bank servers, and power cloud servers. These servers are heterogeneous and have a decentralized environment, sharing all services. For instance, the hospital server has the accessibility of all microservices as the rest of the servers have and can create the workflow according to users' choice for particular goals. All the nodes are connected via different communication channels called base stations, with fixed upload and download bandwidths.

Blockchain technology is the key component of the proposed system, where each transaction among different workflow microservices is to be done securely and validly. Prior blockchain technologies (Kayan et al., 2022; Rosado et al., 2022; Elhoseny et al., 2022; Chandra and Matuska, 2022) exploited SHA-256 algorithms, which need to be secured more against cyber-attacks for workflow transactions. The study presents novel hashing schemes that work optimally against cyber-attacks. Each transaction is to be done based on the secure hashing algorithm, where the hashing of each transaction is divided into parts. All the nodes are decentralized. Therefore, no centralized decision-making node decides for the entire situation when there is a security issue at work. The study devises a pattern-efficient blockchain scheme that validates each node transaction and verifies the malware attacks trained based on higher cloud processing speed servers. There are known and unknown malware types that exist in the system. Hence, the study initially implements blockchain technology to create a decentralized transaction. After, the pattern-efficient hashing scheme generates the hash based on an asymmetric mechanism with the annotated pattern for each microservice in the system. The reinforcement Q-Learning scheduler executes all microservices-based workflows based on their quality of service for all healthcare applications. Table 1 represents the mathematical symbols of the problem in detail.

Table 1
Notations.

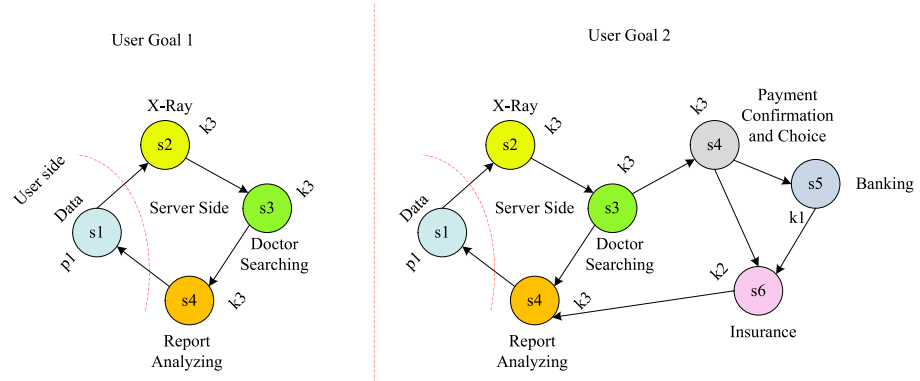
Problem notations	Notation definitions
P	Number of patient devices
p	Particular of patient device
ζ_p	Speed of patient device
ϵ_p	Resource of device
K	Computing processing machines
k	Particular of computing machine
ζ_k	Speed of computing node
ϵ_k	Resource of distinct machine
M	Number of known malware
m	Particular known malware
U	Number of unknown malware
u	particular unknown malware
S	Set of microservices
W	Total workloads of microservices
s	Particular microservices
w_s	Workload of microservice
s_d	Deadline of microservice
s_{status}	Status of the microservice
BS	Set of homogeneous base-stations

3.1. IoMT microservices problem formulation

In this work, the healthcare application packages (APK) are published by the healthcare microservices represented by $S = \{s = 1, \ldots, sS\}$. Each microservice has a workload w_s, deadline d_s, status $status_s$, and b_s requirements for offloading between computing nodes. The study considers the distinct heterogeneous computing nodes, $K = 1\{k = 1, \ldots, kK\}$. Each node has distinct speeds, resources, and blockchain policies, $\{k, bcv, \epsilon_k, \zeta_k\}$. The number of patients or users is $P = \{p = 1, \ldots, pP\}$.

3.1.1. Healthcare microservices workflow

In the study, the microservices consist of different services that aim to achieve patients' healthcare objectives through hospitals' other banking and insurance nodes. Let us suppose the study discusses the two cases as shown in Fig. 2(a) and (b) parts for the patients and services scenarios. One user can use more than one application at a time. Therefore, designing the microservice workflows decides on the runtime of the microservice combination for execution. The user goal 1,



(a) Microservice Workflow Doctor Searching Job

(b) Microservice Workflow Payment for Doctor Fees

Fig. 2. User Defined Microservice Workflows.

as shown in Fig. 2(a), designs the microservices-based workflow for the doctor searching in which many services are combined among different nodes. For instance, if a patient device has bio-data at $p1$, search for the doctor at the particular hospital for the report analysis. The patient microservice $p1 \leftarrow s1$ exploits the X-ray microservice (s2) at node $k3$ and seeks the doctor. The report analyzes and combines the microservices $s3$ and $s4$. The microservices workflow is designed so that $s1, s2, s3, s4 \leftarrow p1, k3$ for the single goal. The main objective of the first user goal is to search for a particular doctor that belongs to one specific hospital among different doctors and ask him/her to analyze the X-ray for further assistance in the flow. The designed microservice workflow shown in Fig. 2(a) will be changed as shown in Fig. 2(b) when the patient is satisfied and the doctor's information and assistance satisfy him/her. Therefore, the patient is ready to pay the doctor's attendant fees via bank and insurance companies in the selected workflows. The new workflow microservices are $s1, s2, s3, s4, s5, s6 \leftarrow p1, k1, k2, k3$, data, X-ray, doctor searching, a report analyzing, payment confirmation, banking for balance validation, and insurance company for payment submission. The red-dotted line divides the microservice workflow into the patient and server sides. It shows that the patient data offloads to the servers for processing. However, this process of both Fig. 2(a) and (b) was done on the server $k3$, because the main subject was the doctor's searching and analyzing the report. Therefore, all the execution is done on server 3 and updates the transactions on all connected servers in the system.

3.1.2. A pattern efficient hashing of blockchain in microservice workflows

The study devises the new hashing pattern based on eight bits following four computations: one character, one special character, one symbol, and one whole number. At the same time, each data point in hashing is divided into two parts, such as each data point being divided into two hashing parts, index [0] hash, and [1] hash, with the proposed pattern. In proposed work, each blockchain scheme considered the computing nodes as blocks with the different sizes and resources.

The study applied the blockchain hashing pattern and proof of validation to the patients' designed microservice workflows, as shown in Fig. 3. All the suggested algorithms are processed in different schemes, as shown in Fig. 3. It can be observed that many microservices of workflow can be executed on a single computing node, such as $s2, s3, s4$ on the node $k3$. Microservice $s1$ takes input at the patient device, $s1 \leftarrow p1$. However, $s4, s5, s6 \leftarrow k1, k2, k3$. Each microservice has data, deadline, and status attributes and applies the pattern hashing with the 8 bits on the microservice before transferring data to another microservice for execution. For instance, each microservice data has the following hashing at the user node: The patient data $s1 \leftarrow p1$: data converted into a hash $[0]1\$a \sim [1]\&0b? \leftarrow s1$. There are 9 bits, the index [0] and index [1], and one space character among all the hashing in the blockchain node. The patient device node is the mobile device and applies the hashing at the local device to offload data to the next microservice based on the validation method. The blockchain validation method, e.g., $bcv1$. All the nodes have different attributes such as previous hash, current hash, and validation method to ensure both indexes have the valid block in the blockchain network. As the study discusses the hashing mechanism, the patient data hash is offloaded to another microservice, such as data and X-ray, and microservice $s2$ has the following validation. The previous hash: $[0]1\$a \sim [1]\&0b? \leftarrow s1$ must be validated all pattern sequences, and after matching it will be converted into new hashing based on data and plus X-ray data into a new hash: $[0]8\$\#a \;\; [1]\%z?0 \leftarrow s1 \sim s2$. Furthermore, for the microservice $s3$ the previous hash must be validated $[0]8\$\#a \;\; [1]\%z?0 \leftarrow s2 \sim s3$, and the new hash is generated in the following way. $[0] * \$2b \;\; [1]\%z@0 \leftarrow s3$. Similarly, for the rest of microservices: Previous hash $[0] * \$2b \;\; [1]\%z@0 \leftarrow s3 \sim s4$, and new the hash is: $[0]\#2b* \;\; 0cm \;\; [1]\%x@1 \leftarrow s3 \sim s4$. Microservice in banking and insurance has the following validation. For instance, $[0]\#1g* \;\; [1]\%f!6s4 \sim s5$ and $[0] * a^* \;\; [1]\#f!1s5 \sim s6$.

The study considers the M number of known malware and U number of unknown malware in the system. Whereas, each known malware, e.g., $\{m = 1, \ldots, M\}$ and unknown malware $\{u = 1, \ldots, U\}$ has different attributes and properties in the system.

There are two types of computing mechanisms for the healthcare workflow microservices execution at the local patient devices: local computing nodes in the system. The binary variable designated in the devices' application package (APK) either which workflow microservice executes on the local devices and the rest of the microservices are offloaded to the other computing nodes for execution.

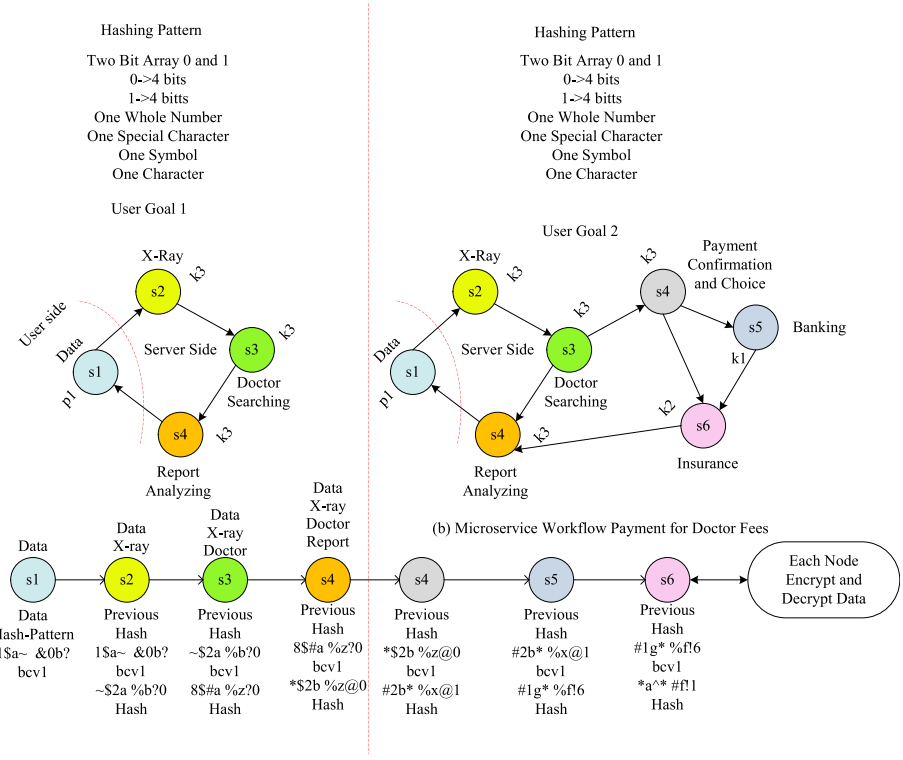
$$x_{s,p,k} = \begin{cases} x_{s,p} & = 1, Local \\ x_{s,k} & = 2, Offloaded \end{cases} \tag{1}$$

Eq. (1) demonstrates either the workflow's starting microservice executes at the patient device or offloaded to the other computing node for the execution. The local execution time of the microservices for the particular workflow is determined in the following way.

$$L_s^e = \sum_{s=1}^{S} \sum_{p=1}^{P} x_{s,p} \times \frac{w_s}{\zeta_p} \tag{2}$$
$$+Hash + Validation + Pattern + C_s si \sim sj.$$

Eq. (2) determines the local execution time of microservices at the patient devices.

$$R_s^e = \sum_{s=1}^{S} \sum_{k=1}^{K} x_{s,k} \times \frac{w_s}{\zeta_k} \tag{3}$$
$$+Hash + Validation + Pattern + C_s si \sim sj.$$



(a) Pattern Efficient Blockchain Hashing and Validation for Microservice Workflow

Fig. 3. Blockchain-Pattern Validation Among Healthcare Microservices.

Eq. (3) determines the remote execution time of microservices at the different nodes.

$$C_s si \sim sj = \sum_{s=1}^{S} \sum_{BS=1}^{BS} (\frac{w_s}{Upload - bandwidth}) + (\frac{w_s}{Download - bandwidth}). \tag{4}$$

Eq. (4) determines the communication between microservice s_i and microservice s_j.

$$Hash = \sum_{s=1}^{S} Encryption(Publickey, w_s) + Decryption(Privatekey, w_s). \tag{5}$$

Eq. (5) determines the encryption and decryption time based on multi-dimensions proposed pattern two indexes array.

$$validation = \sum_{s=1}^{S} \sum_{p=1}^{P} \sum_{k=1}^{K} Previous - Hash[Enc(PK, w_s) \leftarrow p1 \in P] == Previous - Hash[Encryption(PK, w_s) \leftarrow k1 \in K]. \tag{6}$$

Eq. (6) determines the blockchain validation time and processing secure time.

$$Pattern = \sum_{m=1}^{M} \sum_{u=1}^{U} Encryption(PK, w_s) \neq Encryption (PK, w_s) \in m, u. \tag{7}$$

Eq. (7) determines the pattern matching of the hashing in the system. The total validation time (TVT) is determined in Eq. (8).

$$TVT = L_s^e + R_s^e + C_s si \sim sj. \tag{8}$$

4. Proposed pattern proof of malware validation (PoPMV) algorithm schemes flowchart

The study devises algorithmatic flowchart that consists of different schemes. For instance, flowchart consists of the Pattern Proof of Malware Validation (PoPMV) algorithm scheme that adds the microservice workflow goal method, dynamic programming enabled scheduling, and blockchain malware validation scheme as shown in Algorithm 1.

Algorithm 1: PoPMV Schemes

Input : $\{S, P, K\}$

1 **begin**
2 Call Add Microservices Workflow Scheme;
3 Call Local Knapsack Local Scheduling Scheme;
4 Call Blockchain Validation Pattern Scheme;
5 Call Remote Knapsack Scheduling Scheme;

In this paper, Algorithm 1 shows the flowchart of schemes that is consisted different methods in terms of steps to process the microservices in workflows. The S, P, K parameters are processed in steps 1 to 5 with the different schemes. In the PoPMV Algorithm 1 consisted of different blockchain schemes in ICPS to handle the security issues for applications. The schemes are workflow scheme, local knapsack, blockchain validation and remote scheduling.

4.1. Add microservices workflow scheme

The study devises the add-microservices workflow scheme in which users can create their own workflow, in which sequences of microservices are connected to perform a single goal. For instance, a user $p1$ wants to upload an X-ray report to the service and search for a doctor for the analysis. Once the doctor is analyzed, it will ask the insurance

Algorithm 2: Adding Healthcare Microservice Workflow Scheme

Input : $\{S, W, P\}$
begin
  Goal[1];
  Goal[2];
  foreach *($s = 1$ to S)* **do**
    Add Goal[1]=$s1, s2, s3, s4 \sim S \leftarrow W \leftarrow p1 \in P$;
    Add Goal[2]=$s1, s2, s3, s4, s5, s6 \sim S \leftarrow W \leftarrow p1 \in P$;

company to pay their bill, and an insurance company will pay the bill via different bank services in a single workflow.

As shown in Fig. 2, the study showed two microservices workflows of different healthcare services in which the first goal has four microservices and the goal has six microservices. However, we can add more workflows till all available services are available for the different healthcare processes in the system. The study devises healthcare services' added microservice workflow goals as shown in Algorithm 2. The goal[1] is the first goal which has Goal [1]=$s1, s2, s3, s4$ and Goal [2]=$s1, s2, s3, s4, s5, s6$. However, these goals of the microservice workflow of the healthcare services can only be added to the user devices during the initial states in the system.

4.2. Reinforcement learning and LSTM malware detection approaches in ICPS

The PoPMV algorithm of blockchain scheme in ICPS used the deep learning to identify the runtime security aspects regarding malware in mobile fog cloud networks. The healthcare workflows combine microservices from different nodes located at other places. To complete remote healthcare processing, workflow $w1$, for example, combines microservices from the hospital, insurance, and banking nodes. Therefore, each node is considered a "state". Related microservices are run in different states, and each state is regarded as one of the processing nodes in use. Each state has ways to get rewards based on time series analysis, blockchain validation, and long-term memory. Q-learning is the key scheme that tracks each state's performance while the workflow is done on different nodes. However memory and gradient vanishing are big problems for Q-learning when each autonomous state loses information for various reasons. The study combined short-term memory (LSTM) with Q-learning, with state sequences and time series evaluated for each node in the system. The study suggests patterns, signatures, behaviors, and heuristics that can be used inside the blockchain to keep security, data validation, and immutable performance. This keeps attacks that are common in open network nodes from happening.

4.3. Local knapsack local scheduling scheme

The study demonstrates the dynamic programming-enabled knapsack local scheduling scheme in which all the microservice local execution and applied encryption and decryption are performed in the system. The study devises the local scheduler as shown in Algorithm 3.

Algorithm 3: Adding Healthcare Microservice Workflow Scheme

Input : Goal[1], Goal[2]
begin
  foreach *(S as Goal[1,2])* **do**
    Determined execution time of selected microservices based on Equation (2);
    Goal[1]=$s1 \leftarrow p1 \in P \leq s_d \& \epsilon_p$;
    Goal[2]=$s1 \leftarrow p1 \in P \leq s_d \& \epsilon_p$;
  Call Blockchain Malware and Hash Validation;
  Offload Microservices workload to the remote;

The study executes the local devices with the pre-set goal of microservices. The Algorithm 3 determines the execution of microservices under their deadlines and the limitation of their resources. All the microservices malware and hash validation are done based on the blockchain scheme in the system.

4.4. Blockchain PoPMV validation pattern scheme for known and unknown malwares

The study applied the blockchain hashing pattern and proof of validation for security analysis to the patients' designed microservice workflows, as shown in Fig. 3. It can be observed that many microservices of workflow can be executed on a single computing node, such as $s2, s3, s4$ on the node $k3$. As, microservice $s1$ takes input at the patient device, $s1 \leftarrow p1$. However, $s4, s5, s6 \leftarrow k1, k2, k3$. Each microservice has data, deadline, and status attributes and applies the pattern hashing with the 8 bits on the microservice before transferring data to another microservice for execution. For instance, each microservice data has the following hashing at the user node: The patient data $s1 \leftarrow p1$: data converted into a hash $[0]1\$a \sim [1]\&0b? \leftarrow s1$. There are 9 bits, the index [0] and index [1] one space character among all the hashing in the blockchain node. The patient device node is the mobile device and applies the hashing at the local device to offload data to the next microservice based on the validation method. The blockchain validation method, e.g., $bcv1$. All the nodes have different attributes such as previous hash, current hash, and validation method to ensure both indexes have the valid block in the blockchain network. As the study discusses the hashing mechanism, the patient data hash is offloaded to another microservice, such as data and X-ray, and microservice $s2$ has the following validation. The previous hash: $[0]1\$a \sim [1]\&0b? \leftarrow s1$ must be validated all pattern sequences, and after matching, it will be converted into new hashing based on data and plus X-ray data into a new hash: $[0]8\$\#a \quad [1]\%z?0 \leftarrow s1 \sim s2$. Furthermore, for the microservice $s3$ the previous hash must be validated $[0]8\$\#a \quad [1]\%z?0 \leftarrow s2 \sim s3$, and the new hash is generated in the following way. $[0] * \$2b \quad [1]\%z@0 \leftarrow s3$. Similarly, for the rest of microservices: Previous hash $[0] * \$2b \quad [1]\%z@0 \leftarrow s3 \sim s4$, and new the hash is: $[0]\#2b*$ 0cm $[1]\%x@1 \leftarrow s3 \sim s4$. Microservice in banking and insurance has the following validation. For instance, $[0]\#1g* \quad [1]\%f!6s4 \sim s5$ and $[0] * a^* \quad [1]\#f!1s5 \sim s6$. The study considers the M number of known malware and U number of unknown malware in the system. Whereas, each known malware, e.g., $\{m = 1, \ldots, M\}$ and unknown malware $\{u = 1, \ldots, U\}$ has different attributes and properties in the system.

The proposed LSTM-based blockchain scheme always exploits the backpropagation feature inside the scheme, where the failure of running tasks from the end will be rescheduled from the start based on the given deadline. The main reason is that all the tasks and microservices must be executed without malware attacks inside the system based on the LSTM blockchain scheme. The study presented the proposed scheme's security model and security analysis based on blockchain technology and suggested new pattern partial schemes. The study considers the on-device, network, and distributed server types of malware. The study handles all malware based on the system's behavior, patterns, and heuristics. The study builds a blockchain malware detection method based on the convolutional neural network on a cloud computing machine. All the malware attack tasks in LSTM must be rescheduled based on their deadlines using the scheme's backpropagation features. Initially, the method trains and tests based on the following known malware in the system. The algorithm partially trains on the available data based on existing malware behavior, pattern, and heuristics: for instance, botnet, Keylogger, Cryptocurrency miner, Ransomware, Rootkit, Spyware, zero-day attacks, polymorphic viruses, and Trojan. The malware detection raw data the study used was obtained from the security industries such as Meraz-18, Techno-Cultural Festival, and their legitimate files for research purposes. The signature-based attacks are expected to be easily identified based on static analysis.

Algorithm 4: Pattern Enabled and Hashing Validation Blockchain Scheme

Input : {Goal[1], Goal[2], M, U}

begin
  Hash[indexes]=null;
  foreach *(s=1 to S← Goal[1,2])* **do**
    Determined the hashing and validation based on equation (5) and equation (6);
    if *(*$s1 \leftarrow s_w \sim hash1 == s_{status} \leftarrow 1$*)* **then**
      $p1{=}s1 \leftarrow [0]8\$\#a \;\; [1]\%z?0 \leftarrow s1 \sim s2$;
      if *(*$p1{=}s1 \leftarrow [0]8\$\#a$ *Matched* $s1 \sim s2$*)* **then**
        $s_{status}.validation == true$;
      else
        Searching Pattern;
        $p1{=}s1 \leftarrow [0]8\$\#a \leftarrow M$;
        $p1{=}s1 \leftarrow [0]8\$\#a \leftarrow U$;
      if *(*$p1{=}[1]\%z?0 \leftarrow s1 \sim s2$*)* **then**
        $s_{status}.validation == true$;
      else
        Searching Pattern;
        Apply backpropagation and reschedule tasks ;
        $M[]p1 = [1]\%z?0 \leftarrow s1 \sim s2 \leftarrow M$;
        $U[]p1 = [1]\%z?0 \leftarrow s1 \sim s2 \leftarrow U$;
      End Inner Searching;
    End Outer Searching;
    if *(*$p1 = [1]\%z?0 \leftarrow s1 \sim s2 \leftarrow M\&U$*;*
) **then**
      Determined the Malware based on Convolutional Neural Network Scheme;
    else
      return the $s1, \ldots S$status=true;
  End Validation of all blockchain nodes;
End Main;

However, unknown attacks with complex polymorphic and zero-day attacks with random extensions are difficult to locate in the system. The study devises Algorithm 4 determines the blockchain validation and identifies the malware based on their known and unknown behavior in the system. The main goal is to execute the patients' goals with the proper validation. Line 1 to 4 of Algorithm 4 initializes the blockchain hashing and the patient's goals in the system. From 5 to 8 steps, the workload is converted into hashing, and workload matching is based on index 0 and 1 between nodes in the system. If any malware exists, the study will call the malware detection technique based on the CNN method as shown in Fig. 4.

The malware data sampling must need pre-processing to remove the null values and noise from the data sample in the processing. The study exploited the DNN pre-processing scheme, where all null values and noise from data will be eliminated based on the given pre-threshold. The proposed scheme phases, as shown in Fig. 4 presented the deep neural network and LSTM-enabled blockchain process for the microservice workflow healthcare in the distributed fog cloud network. The study implemented LSTM inside blockchain network-based data transaction memory and long-term validity. The proposed LSTM inside blockchain has backpropagation where malware attacked the executed workloads will be rescheduled from input initial request again under their given deadlines. The LSTM-based malware detection has five steps, as it will take input data validation from the blockchain workflow microservices based on their hashing values. After that, the algorithm applies the pre-processing to validate the data based on 0 and 1 indexes for each microservices during the known malware layer and unknown layer by using the LSTM method. After the classification and feature extraction, the training and testing phase will send the validated data to the predication method where each hash is compared to the original one, either status true or false as defined in Algorithm 4. The known malware will be detected based on the given list of malware patterns and unknown malware and put into the list. Algorithm 4 will validate all data between microservices on the different nodes until and unless valid data is reached and executed on the different nodes. Fig. 4 shows that each blockchain validation is to be done on different states based on LSTM where four different methods are implemented: behavioral, signature, heuristic, and pattern for detection of malware and benign. Each state is assumed to be distinct with different features such as processing, security, validation, and immutable in the network. The LSTM has a back prorogation feedback mechanism to monitor the performance of each workflow in the system.

4.5. Reinforcement learning Q-learning PoPMV based scheduling scheme

We suggested the rewards enabled q-learning scheme with the PoPMV scheme to improve the functionality of ICPS at the runtime. The study demonstrates the reinforcement learning Q-Learning PoPMV based adaptive scheduling scheme in which all the microservices design the workflows and apply encryption and decryption at the users and server nodes in the network. The Q-Learning scheduler is model-free, where the initial microservice workflow does not depend on the current state. It is a model and depends upon the current state in the designated states. In the considered problem, the study derives q-learning as the function that maximizes the reward of the designed workflows in the network based on different state actions. The malware detection and security are determined based on different rewards. The study devises the local scheduler as shown in Algorithm 5.

Algorithm 5: Q-Learning-Enabled Microservice Workflows Scheme

Input : Goal[1], Goal[2], Reward[],states, actions

begin
  foreach *(S as Goal[1,2])* **do**
    Determined execution time of selected microservices based on Equation (3);
    states[]← $s1 \leftarrow k1 \in K \leq s_d \& \epsilon_k$;
    actions[]← $s1 \leftarrow k1 \in K \leq s_d \& \epsilon_k$;
    Goal[1]=$s1 \leftarrow k1 \in K \leq s_d \& \epsilon_k$+actions[]+states[];
    Goal[2]=$s1 \leftarrow k1 \in K \leq s_d \& \epsilon_k$+actions[]+states[];
    rewards=Goal[1]+Goal[2];
  Call Blockchain Malware and Hash Validation;

The states are distributed fog cloud environments where microservices are scheduled based on their deadlines. The actions are decisions of the scheduler for allocations of microservices to the fog and cloud nodes for executions. The rewards are the criteria based on given deadlines. They will be rewarded if the workflow microservices are executed under their deadline. Otherwise, it will gain negative results. The states, actions, rewards, and goals are constraints of the Q-learning-enabled scheduler during the execution of workflow microservices in the fog cloud networks. The study executes the local devices with the pre-set goal of microservices. The Algorithm 5 determines the execution of microservices under their deadlines and the limitation of their resources. The q-learning function schemes add the states and actions and achieve different rewards if the scheduler meets the quality of service in the scheduler network.

4.6. Time complexity

The time complexity of the modified PoPMV and LSTM schemes determined in different phases. PoPMV consisted of different N steps that performed the hashing, validation, and malware detection and recovery and represented by $log(N \times N)$. The log determined the validation of microservices in different steps. The LSTM has many gates such as L, therefore all gates are represented by $log(l \times l)$.

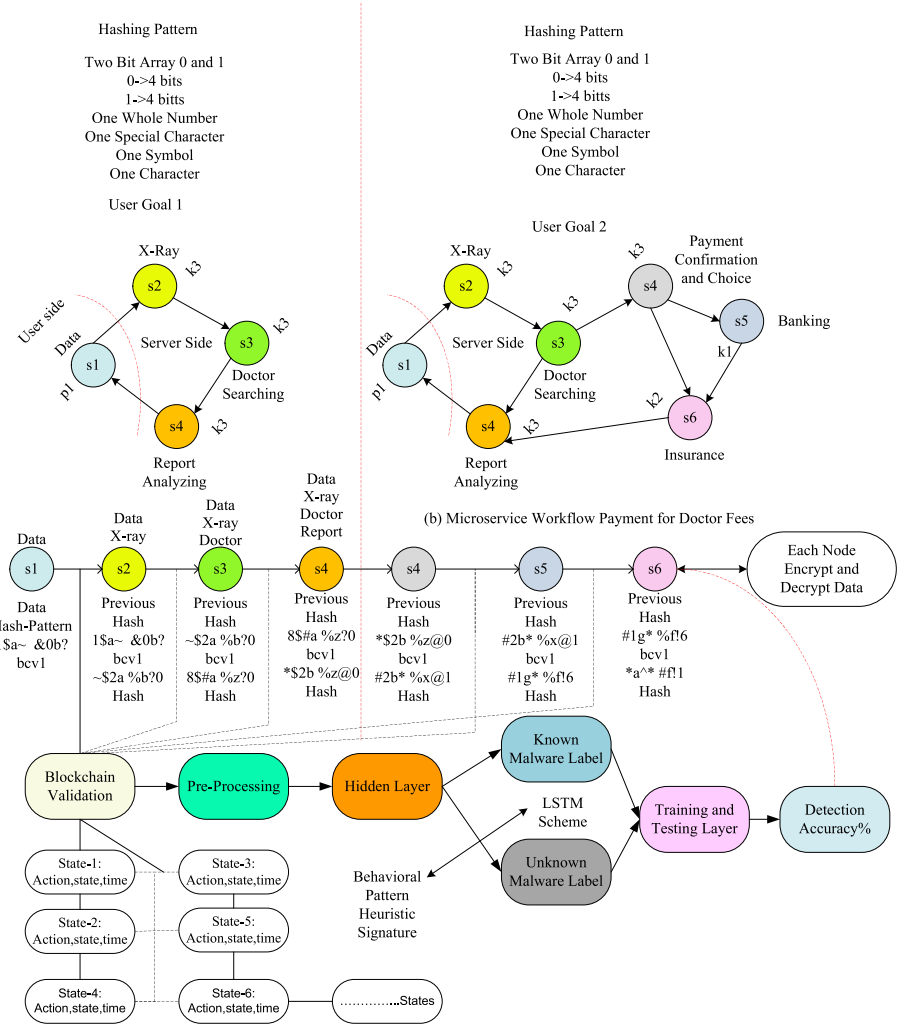


Fig. 4. Blockchain-Pattern Enabled Security Based on Q-Learning with LSTM in Networks.

5. Performance evaluation

In this part, the study conducted different experiments based on given workloads and configured parameter the setting for healthcare applications. The study implemented blockchain technology based on the fundamental rules in the experiment part. The blockchain consensus methods include proof of work (PoW) and proof of stake (PoS) with the proposed method PoPMV for blockchain validation and malware detection. The simulation parameters for the system design and experimental parameter setting are defined in Table 2. The experiment was conducted on the docker container and implemented on the Windows operating system. The workload and system were designed in the JAVA, XML, and Python languages for the blockchain development and scheduling for the run of the experiment for the workflow microservices healthcare workloads in the network. The simulator was designed based on cloud computing and Amazon machines and implemented with the edges-foundry for the microservices workflow healthcare applications.

The total simulation time takes 6 times and repeats 50 times, the machines are 11 generations, 1000 GB-ROM, 32 RAM, M=1500 malware samples, patient devices P=100, number of computing nodes K=4, and number of B=10 base-stations. In this study, we designed a malware dataset with known and unknown malware as shown in Table 3. The different kinds of malware are considered in work in the form of known and unknown malware with the characteristics, size of code, malware comes from, attack on which data, and current status workload after being attacked by the malware.

Table 2
Experimental setting parameters.

Simulation parameters	Values
Environment	Windows Docker Container
Languages	JAVA, Python, and XML
Libraries	EdgeX Foundary
Experiment Time	6 h
Experiment Repetition	50 times
Deep learning values	**Parameters**
W=	6
S	=25
M	=1200
P	=100
K	=4, 1000 GB, 32 RAM, 11 generations
BS	=10
Epoch	2000
Workflows	1000

In this study, the proposed blockchain-enabled healthcare works are implemented based on a real-time research tool, and the source and dataset are publicly available on the following github. The malware dataset is available on the following link: https://github.com/ABDULLAH-RAZA/Assignment-. The given URL is designed data which is shown in Table 3 with their properties and characteristics during the experimental part in the network. Table 4 enlisted the attacks on

Table 3
Malware dataset for training and testing purposes 1500 records.

Name	M	Patterns	Size	Attacked	Labeled
memtest.exe	Worms	Numeric	511 984	w=1	Known
ose.exe	Worms	Image	461 986	w=2	Unknown
DW20.exe	Worms	Audio	661 986	w=3	Unknown
dwtrig20.exe	Worms	Encrypted	021 987	w=1	Known
Horse.bat	Worms	Encrypted	031 985	w=1	Unknown
memtest.exe	Worms	Numeric	231 983	w=1	Unknown
memt.vbs	Worms	Image	641 966	w=3	Unknown
test.cmd	Worms	Audio	671 966	w=1	Unknown
st.hta	Worms	Numeric	987 985	w=2	Known
memtest.html	Worms	Numeric	981 123	w=1	Unknown
dot.scr	Worms	Audio	921 125	w=1	Known
file.msi	Worms	Numeric	311 983	w=1	Unknown
image.msp	Worms	Text	161 982	w=1	Known
video.pif	Worms	Text	661 984	w=1	Unknown
audio.bot	Worms	Numeric	461 982	w=1	Unknown
MD5.png	Worms	Text	771 981	w=3	Known
SHA.cp	Worms	Numeric	111 984	w=1	Known
CRC.jav	Worms	Numeric	221 932	w=2	Unknown
AES.mp	Worms	Numeric	561 933	w=2	Known
res.cls	Worms	Encrypted	761 932	w=3	Unknown
gif.ncl	Worms	Numeric	141 966	w=3	Known
jar.cpn	Worms	Encrypted	141 944	w=1	Known

Table 4
Unknown malware attacks on workflow microservices.

Microservices	Detection	Node	Method
onService	Connected	k1	PoW
API call	signature	k1	PoW
bindService	Malware	k1	PoW
API call	signature	k1	PoW
Doctor	Interface	k2	PoW
API call	signature	k2	PoS
Service	Connection	k3	PoW
API call	signature	k1	PoS
android.os	Binder	k2	PoW
API call	signature	k1	PoW
SEND	_SMS	k4	PoW
Manifest	Permission	k4	PoW
Ljava.lang	malware	m1	PoW
Storage	signature	m1	PoW
Ljava.lang	Benign	m2	PoW
API call	signature	m1	PoW
Ljava	cast	m1	PoW
API call	signature	m1	PoW
Ljava.net	URLDecoder	m1	PoW
API call	signature	m1	PoW
READ	_PHONE_STATE	m1	PoW
Manifest	Permission	m1	PoW
API call	signature	m1	PoW
Payment	Malware	k2	PoW
API call	signature	k2	PoW
Insurance	Malware	k3	PoW
Patients	_ACCOUNTS	m1	PoW
Manifest	Permission	m1	PoW

the different microservices and trained them before execution in the system.

5.1. Real scenario of blockchain infrastructure for healthcare system and result discussion of proof of validation algorithms

For the experiment, we have exploited the Anova statistical method to compare the objective constraints of the study. We discussed the practical and managerial implications extracted according to the obtained results and conducted analyses after experiments. We discussed the realtime scenario of blockchain technology for healthcare applications. The study shows the performances of secure healthcare microservices in the distributed cloud environment. The study discussed the real-time scenario, where users can design their healthcare microservices and securely process them based on blockchain technology.

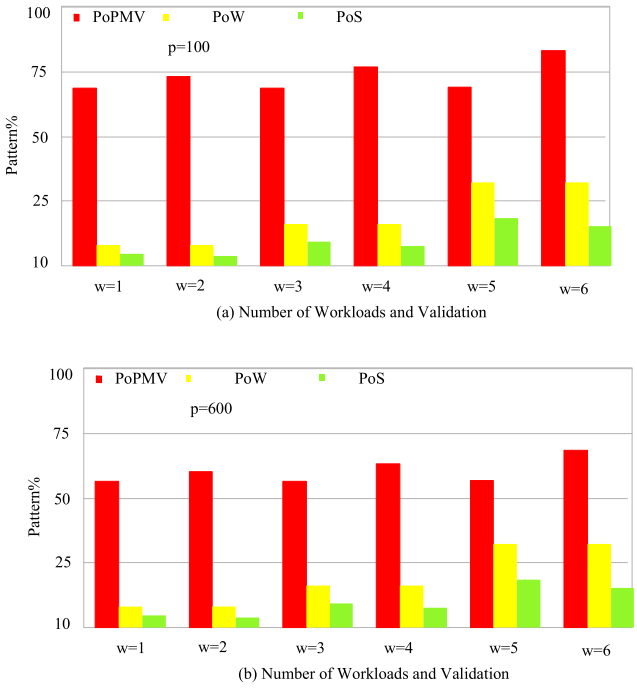


Fig. 5. Performance of Healthcare Systems Based on Blockchain Validation Schemes in Terms of Data Transactions.

The study presented a secure and valid healthcare system in which heterogeneous and autonomous workflows are processed securely. The study presented a novel pattern-enabled proof of validation method of blockchain technology. All the LSTM-enabled blockchain parameters as defined in Table 2 are implemented in the configuration simulation files in the system. The study shows the performance of proof of validation of blockchain consensus methods with their characteristics and properties. The study implemented different healthcare microservices and added them to the new workflow applications and validated them with the proof of validation methods. The study implemented PoW, PoS, and PoPMV schemes for the 6 workflows execution during experiment execution. This parameter setting has p=100 patient devices that are requesting microservices workflows, e.g., W=6. The pattern% is the accuracy of the method which detects both unknown attacks during the execution of microservices workflows on the different fog and cloud networks. Fig. 5(a) shows the performance microservices enabled workflows with the different known and unknown and their performances during the experiment. In Fig. 5(a), the results showed that PoPMV outperformed as compared to existing proof of validation regarding malware detection, identifying hash, and validation of the microservices workflow applications.

In this part, the study shows the performance of proof of validation of blockchain consensus methods with their characteristics and properties. The study implemented different healthcare microservices and added them to the new workflow applications and validated them with the proof of validation methods. The study implemented PoW, PoS and PoPMV schemes for the 6 workflows execution during experiment execution. This parameter setting has $p = 600$ patient devices that are requesting microservices workflows, e.g., W=6. The pattern% is the accuracy of the method which detects both unknown attacks during the execution of microservices workflows on the different fog and cloud networks. Fig. 5(b) shows the performance microservices enabled workflows with the different known and unknown and their performances during the experiment. In Fig. 5(b), the results showed that PoPMV outperformed as compared to existing proof of validation in terms of malware detection, identifying hash, and validation of the microservices workflow applications.

Table 5
Algorithm comparisons.

$W \sim S$	M	Algorithm	TVT (seconds)
$w1 \sim s1, s2, s3, s4$	10	cloud← k5	1000
$w1 \sim s1, s2, s3, s4$	10	fog1← k4	10 000
$w1 \sim s1, s2, s3, s4$	10	fog2← k1,2,3	100 000
$w2 \sim s1, s2, s3, s4, s5, s6$	20	cloud← k5	12 000
$w2 \sim s1, s2, s3, s4, s5, s6$	20	fog1← k4	130 000
$w2 \sim s1, s2, s3, s4, s5, s6$	20	fog2← k1,1,2,3	1 500 000
$w3 \sim s1, s2, s3, s4, s5, s6, s7$	100	cloud← k5	15 000
$w3 \sim s1, s2, s3, s4, s5, s6, s7$	100	fog1← k4	180 000
$w3 \sim s1, s2, s3, s4, s5, s6, s7$	100	fog2← k1,1,2,3	2 300 000
$w4 \sim s1, s2, s3, s4, s5, s6, s7$	400	cloud← k5	15 000
$w4 \sim s1, s2, s3, s4, s5, s6, s7$	400	fog1← k4	190 000
$w4 \sim s1, s2, s3, s4, s5, s6, s7$	400	fog2← k1,1,2,3	2 500 000
$w5 \sim s1, s2, s3, s4, s5, s6, s7$	800	cloud← k5	19 000
$w5 \sim s1, s2, s3, s4, s5, s6, s7$	800	fog1← k4	220 000
$w5 \sim s1, s2, s3, s4, s5, s6, s7$	800	fog2← k1,1,2,3	3 000 000
$w6 \sim s1, s2, s3, s4, s5, s6, s7$	1500	cloud← k5	23 000
$w6 \sim s1, s2, s3, s4, s5, s6, s7$	1500	fog1← k4	300 000
$w6 \sim s1, s2, s3, s4, s5, s6, s7$	1500	fog2← k1,1,2,3	3 300 000

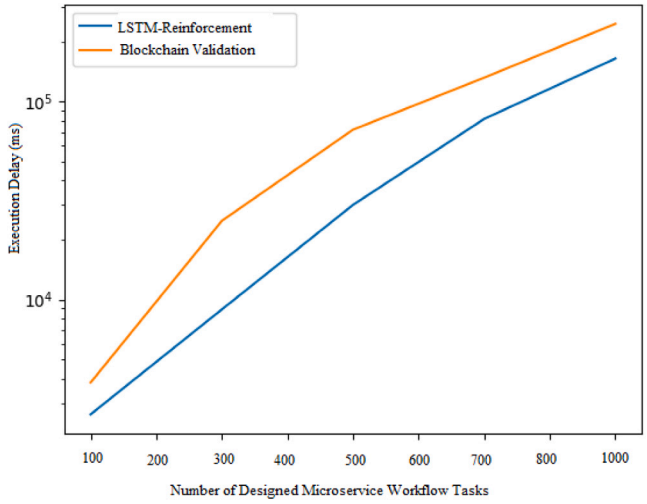


Fig. 6. Number of Epochs and Microservices Workflows Reward.

5.2. Algorithm comparison

In this session, the study obtained the numerical results of all methods with the different microservice workflows with the different numbers of malware and their total validation time. The study implemented the different nodes such as patient, fog node, and cloud computing and observed their performances during execution in the experimental as shown in Table 5. Fig. 6 shows the number of epochs microservices workflows reward performance in the proposed system and improved from time to time. Hence, it is a good strategy to choose adaptive scheduling that improves the performances in different states and actions with the different number of epochs in the system. Fig. 7 shows the performances of the Q-learning method in the different states during the execution of microservices workflows in the system. In the work, the states are adding microservices workflows that can be added or deleted according to tasks and services based on performances in the network. Therefore, it is a good idea to obtain optimal performances with Q-learning and validate the workflow data among distributed services.

The existing prior blockchain healthcare works (Kayan et al., 2022; Zhang et al., 2021; Agrawal and Kumar, 2022) closely compared the performance of reinforcement learning algorithms (e.g., State action reward state action (SARSA) proof of work (PoW) (Zheng et al., 2022), Q-Learning (Yang et al., 2022) and Proposed Q-learning PoPMV) on the

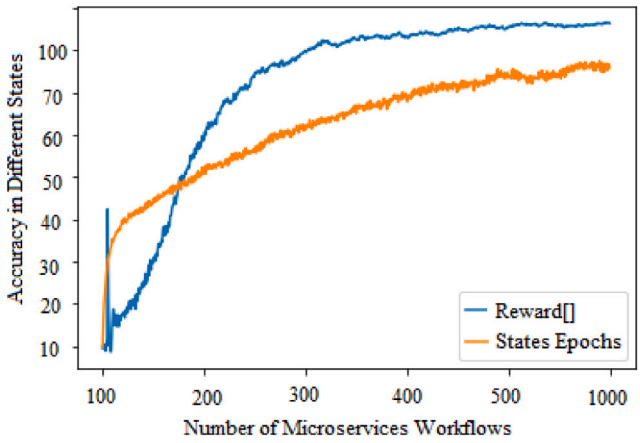


Fig. 7. Number of Epochs and Microservices Workflows Reward in Different States.

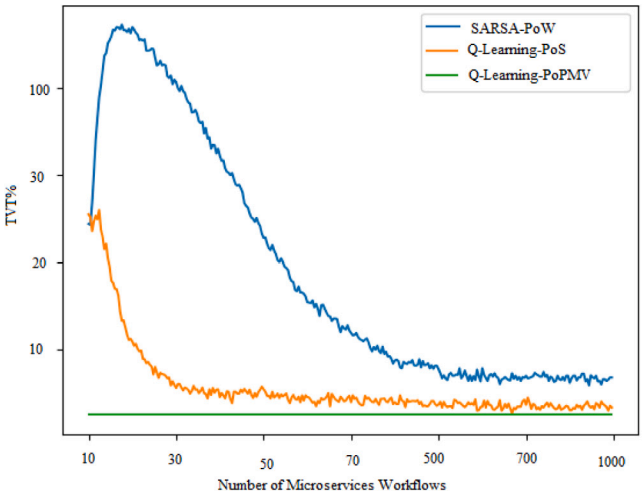


Fig. 8. Number of Epochs and TVT performances of Microservices Workflows in Different States.

cloud computing machines for the microservices workflows healthcare for the execution. Fig. 8 shows TVT performances of microservices workflows for healthcare tasks with the different algorithms in the network. Fig. 8 shows that Q-learning PoPMV outperformed as compared to existing State action reward state action (SARSA) PoW and Q-learning algorithms regarding performances, accuracy, cost, and execution time for the microservices workflows healthcare tasks in the network.

5.3. Finding and limitation of PoPMV

In proposed blockchain validation schemes, we suggested the PoPMV to validate the transactions of data among different computing nodes. The proposed identified the both known and unknown attacks in distributed fog cloud microservices for healthcare workflow applications. However, PoPMV computation time becomes higher when the number of workflow tasks failed in different computing nodes. In future work, we will improve the proposed method with the federated Artificial Intelligence (AI) edge analysis to minimize the energy, time, and storage costs in the system.

6. Conclusion

The study presented Pattern Attacks Efficient Hashing Scheme of Blockchain for the Internet of Medical Things Healthcare Run time Microservices Workflows in this paper. The goal is to design a novel

pattern of malware validation scheme based on blockchain technology and design the microservice workflow application at runtime instead of design time. The study designed the malware detection system based on cloud computing and the system's distributed fog nodes and patient devices. The simulation results showed that the proposed outperforms in terms of malware detection and their patterns and execution time for the microservice workflows compared to existing studies. The study designed the SARSA-PoW, Q-learning, and Q-learning PoPMV algorithms to run the microservices workflows healthcare tasks and showed their performances with the blockchain and cloud computing technologies in work. The study showed that Q-learning-PoPMV outperformed all existing algorithms regarding microservices workflows for the healthcare tasks in the network.

However, our proposed model's computation time on the decision-maker nodes becomes higher. In future work, we will improve the proposed method with the federated Artificial Intelligence (AI) edge analysis to minimize the energy, time, and storage costs in the system.

Declaration of competing interest

There is not conflict of interest.

Data availability

The malware dataset is available on the following link: https://github.com/ABDULLAH-RAZA/Assignment-

Funding statements

This article was co-funded by the European Union under the REFRESH – Research Excellence For REgion Sustainability and High-tech Industries project number CZ.10.03.01/00/22_003/0000048 via the Operational Programme Just Transition. Also, this work was supported by the Ministry of Education, Youth, and Sports of the Chezk Republic conducted by VSB–the Technical University of Ostrava, Czechia, under Grants SP2023/039 and SP2023/042. All authors approved the final version of the manuscript.

Data statements

In this study, the proposed blockchain-enabled healthcare works are implemented based on a real-time research tool, and the source and dataset are publically available on the following github. The malware dataset is available on the following link: https://github.com/ABDULLAH-RAZA/Assignment-. The given URL is designed data which is shown in Table 3 with their properties and characteristics during the experimental part in the network.

References

Abdelmoneem, R.M., Benslimane, A., Shaaban, E., 2020. Mobility-aware task scheduling in cloud-fog IoT-based healthcare architectures. Comput. Netw. 107348.

Agrawal, N., Kumar, R., 2022. Security perspective analysis of industrial cyber physical systems (I-CPS): A decade-wide survey. ISA Trans.

Almaiah, M.A., Ali, A., Hajjej, F., Pasha, M.F., Alohali, M.A., 2022. A lightweight hybrid deep learning privacy preserving model for FC-based industrial internet of medical things. Sensors 22 (6), 2112.

Arsyad, A.A., Widayat, I.W., Köppen, M., 2022. Supporting farming smart documentation system by modular blockchain solutions. Decis. Mak. Appl. Manag. Eng. 5 (1), 1–26.

Aujla, G.S., Jindal, A., 2020. A decoupled blockchain approach for edge-envisioned IoT-based healthcare monitoring. IEEE J. Sel. Areas Commun.

Baysal, M.V., Özcan-Top, Ö., Betin-Can, A., 2023. Blockchain technology applications in the health domain: a multivocal literature review. J. Supercomput. 79 (3), 3112–3156.

Chandra, Y.P., Matuska, T., 2022. Intelligent data systems for building energy workflow: Data pipelines, LSTM efficiency prediction and more. Energy Build. 267, 112135.

Elhoseny, M., Alshehri, M.D., Abdulkareem, K.H., 2022. Blockchain multi-objective optimization approach-enabled secure and cost-efficient scheduling for the Internet of Medical Things (IoMT) in fog-cloud system. Soft Comput. 1–14.

Erol, I., Oztel, A., Searcy, C., Medeni, İ.T., 2023. Selecting the most suitable blockchain platform: A case study on the healthcare industry using a novel rough MCDM framework. Technol. Forecast. Soc. Change 186, 122132.

Jayanetti, A., Halgamuge, S., Buyya, R., 2022. Deep reinforcement learning for energy and time optimized scheduling of precedence-constrained tasks in edge–cloud computing environments. Future Gener. Comput. Syst. 137, 14–30.

Kaur, A., Singh, P., Singh Batth, R., Peng Lim, C., 2022. Deep-Q learning-based heterogeneous earliest finish time scheduling algorithm for scientific workflows in cloud. Softw. - Pract. Exp. 52 (3), 689–709.

Kayan, H., Nunes, M., Rana, O., Burnap, P., Perera, C., 2022. Cybersecurity of industrial cyber-physical systems: a review. ACM Comput. Surv. 54 (11s), 1–35.

Kumar, P., Kumar, R., Gupta, G.P., Tripathi, R., Jolfaei, A., Islam, A.N., 2023. A blockchain-orchestrated deep learning approach for secure data transmission in IoT-enabled healthcare system. J. Parallel Distrib. Comput. 172, 69–83.

Lakhan, A., Lateef, A.A.A., Abd Ghani, M.K., Abdulkareem, K.H., Mohammed, M.A., Nedoma, J., Martinek, R., Garcia-Zapirain, B., 2023a. Secure-fault-tolerant efficient industrial internet of healthcare things framework based on digital twin federated fog-cloud networks. J. King Saud Univ.-Comput. Inf. Sci. 35 (9), 101747.

Lakhan, A., Li, X., 2019. Mobility and fault aware adaptive task offloading in heterogeneous mobile cloud environments. EAI Endorsed Trans. Mob. Commun. Appl. 5 (16).

Lakhan, A., Mohammed, M.A., Abdulkareem, K.H., khanapi Abd Ghani, M., Marhoon, H.A., Nedoma, J., Martinek, R., Garcia-Zapirain, B., 2023b. Secure blockchain assisted internet of medical things architecture for data fusion enabled cancer workflow. Internet Things 24, 100928.

Lakhan, A., Mohammed, M.A., Nedoma, J., Martinek, R., Tiwari, P., Kumar, N., 2022. Blockchain-enabled cybersecurity efficient iioht cyber-physical system for medical applications. IEEE Trans. Netw. Sci. Eng.

Li, H., Huang, J., Wang, B., Fan, Y., 2022. Weighted double deep Q-network based reinforcement learning for bi-objective multi-workflow scheduling in the cloud. Cluster Comput. 25 (2), 751–768.

Manthiramoorthy, C., Khan, K.M.S., et al., 2024. Comparing several encrypted cloud storage platforms. Int. J. Math. Stat. Comput. Sci. 2, 44–62.

Mastoi, Q.-U.-A., Elhoseny, M., Memon, M.S., Mohammed, M.A., 2021. Deep neural network-based application partitioning and scheduling for hospitals and medical enterprises using IoT assisted mobile fog cloud. Enterp. Inf. Syst. 1–23.

McGibney, A., Bharti, S., 2022. DISTiL: DIStributed industrial computing environment for trustworthy digital workflows: a design perspective. In: International Symposium on Leveraging Applications of Formal Methods. Springer, pp. 219–226.

Mohammed, M.A., Lakhan, A., Abdulkareem, K.H., Zebari, D.A., Nedoma, J., Martinek, R., Kadry, S., Garcia-Zapirain, B., 2023. Energy-efficient distributed federated learning offloading and scheduling healthcare system in blockchain based networks. Internet Things 22, 100815.

Qu, Y., Gao, L., Xiang, Y., Shen, S., Yu, S., 2022. FedTwin: Blockchain-enabled adaptive asynchronous federated learning for digital twin networks. IEEE Netw.

Rana, S.K., Rana, S.K., Nisar, K., Ag Ibrahim, A.A., Rana, A.K., Goyal, N., Chawla, P., 2022. Blockchain technology and artificial intelligence based decentralized access control model to enable secure interoperability for healthcare. Sustainability 14 (15), 9471.

Rasool, R.U., Ahmad, H.F., Rafique, W., Qayyum, A., Qadir, J., 2022. Security and privacy of internet of medical things: A contemporary review in the age of surveillance, botnets, and adversarial ML. J. Netw. Comput. Appl. 103332.

Ren, L., Ning, X., Wang, Z., 2022. A competitive Markov decision process model and a recursive reinforcement-learning algorithm for fairness scheduling of agile satellites. Comput. Ind. Eng. 108242.

Rosado, D.G., Santos-Olmo, A., Sánchez, L.E., Serrano, M.A., Blanco, C., Mouratidis, H., Fernández-Medina, E., 2022. Managing cybersecurity risks of cyber-physical systems: The MARISMA-CPS pattern. Comput. Ind. 142, 103715.

Tanha, F.E., Hasani, A., Hakak, S., Gadekallu, T.R., 2022. Blockchain-based cyber physical systems: Comprehensive model for challenge assessment. Comput. Electr. Eng. 103, 108347.

Utku, A., 2023. Deep learning based an efficient hybrid prediction model for Covid-19 cross-country spread among E7 and G7 countries. Decis. Mak. Appl. Manag. Eng. 6 (1), 502–534.

Wang, H., 2020. IoT based clinical sensor data management and transfer using blockchain technology. J. ISMAC 2 (03), 154–159.

Yang, Z., Yang, R., Yu, F.R., Li, M., Zhang, Y., Teng, Y., 2022. Sharded blockchain for collaborative computing in the Internet of Things: Combined of dynamic clustering and deep reinforcement learning approach. IEEE Internet Things J.

Yu, J., Gao, M., Li, Y., Zhang, Z., Ip, W.H., Yung, K.L., 2022. Workflow performance prediction based on graph structure aware deep attention neural network. J. Ind. Inf. Integr. 27, 100337.

Zhang, D., Wang, Q.-G., Feng, G., Shi, Y., Vasilakos, A.V., 2021. A survey on attack detection, estimation and control of industrial cyber–physical systems. ISA Trans. 116, 1–16.

Zheng, R., Wang, Q., Lin, Z., Jiang, Z., Fu, J., Peng, G., 2022. Cryptocurrency malware detection in real-world environment: Based on multi-results stacking learning. Appl. Soft Comput. 109044.

Zhu, K., Zhang, Z., Sun, F., Shen, B., 2022. Workflow makespan minimization for partially connected edge network: A deep reinforcement learning-based approach. IEEE Open J. Commun. Soc. 3, 518–529.