IMPLEMENTATION OF COMPUTING INFRASTRUCTURES FOR BLOCKCHAIN BASED DISTRIBUTED LEARNING MODELS

Remzi GÜRFİDAN[1] and Mevlüt Ersoy[1]

[1]Suleyman Demirel Universitesi Muhendislik Fakultesi

November 29, 2022

Abstract

Reaching a sufficient number of data sets, learning past experiences from many systems and using this experience in instant or future predictions are among the capabilities of artificial intelligence. The horizontal and vertical growth of industrial systems and the transfer of experience from each location to all other locations increase the quality of the process. However, the rapid growth of IoT (Internet of Things) and OT (Operational Technology) assets in recent years raises questions about data integrity, confidentiality and accessibility. It deploys edge computing and blockchain-based solutions for data security and secure transmission in the IoT ecosystem. In this study, a four-layer IoT ecosystem network is proposed that combines the learning capabilities of artificial intelligence-based systems used in different locations and offers a blockchain-based storage system for data security. These layers consist of node layer, edge layer, decision layer, and training and blockchain layer, respectively. The lowest layer, the node layer, is responsible for collecting the temperature and humidity values in different locations with the developed node devices in order to evaluate them. The data generated in the node devices is transferred to the communicating edge device in the edge layer. The edge layer collects the data from the nodes in the edge system and transfers it to the server centrally. The training and blockchain layer provide the collection of data from edge devices, training the artificial intelligence model and transferring the weights to the decision layer. At the same time, the blockchain-based storage system works at this layer to securely store the processed data. As a result, with this study, it is aimed to develop a framework for transferring the local learning experiences of distributed IoT devices to all IoT devices and for the secure storage of data.

IMPLEMENTATION OF COMPUTING INFRASTRUCTURES FOR BLOCKCHAIN BASED DISTRIBUTED LEARNING MODELS

Remzi GÜRFİDAN[1], Mevlüt ERSOY[2]

[1]Süleyman Demirel University, Computer Engineering,*remzigurfidan@isparta.edu.tr*;

0000-0002-4899-2219, Isparta, Turkey

[2]Süleyman Demirel University, Computer Engineering,*mevlutersoy@sdu.edu.tr*;

0000-0003-2963-7729 Isparta, Turkey

Abstract: Reaching a sufficient number of data sets, learning past experiences from many systems and using this experience in instant or future predictions are among the capabilities of artificial intelligence. The horizontal and vertical growth of industrial systems and the transfer of experience from each location to all other locations increase the quality of the process. However, the rapid growth of IoT (Internet of Things) and OT (Operational Technology) assets in recent years raises questions about data integrity, confidentiality and accessibility. It deploys edge computing and blockchain-based solutions for data security and secure transmission in the IoT ecosystem. In this study, a four-layer IoT ecosystem network is proposed that

combines the learning capabilities of artificial intelligence-based systems used in different locations and offers a blockchain-based storage system for data security. These layers consist of node layer, edge layer, decision layer, and training and blockchain layer, respectively. The lowest layer, the node layer, is responsible for collecting the temperature and humidity values in different locations with the developed node devices in order to evaluate them. The data generated in the node devices is transferred to the communicating edge device in the edge layer. The edge layer collects the data from the nodes in the edge system and transfers it to the server centrally. The training and blockchain layer provide the collection of data from edge devices, training the artificial intelligence model and transferring the weights to the decision layer. At the same time, the blockchain-based storage system works at this layer to securely store the processed data. As a result, with this study, it is aimed to develop a framework for transferring the local learning experiences of distributed IoT devices to all IoT devices and for the secure storage of data.

Keywords: Blockchain, Edge Computing, Distributed Learning, IoT

INTRODUCTION

IoT technology, which is one of the assets of the Industry 4.0 ecosystem, is becoming more and more widespread in our daily lives and is consolidating its place in the field of technology. Capable IoT devices for different purposes have been developed to meet the various needs of technology and people [1]. In developed IoT technologies, hardware has limited capacities and limited processing power. For this reason, various methods are preferred for the security and performance of IoT devices. While the developed IoT devices were combined with edge computing technology, they preferred server-client architecture with security protocols such as SSL (Secure Socket Layer) and TLS (Transport Layer Security). However, this situation creates a bottleneck threat due to continuous growth in the process and may cause delays and malfunctions as a result of blockages in network traffic [6]. With these developments, many problem areas that need to be developed in terms of efficiency, security requirements, resource usage and user security have emerged [3, 5]. Many IoT devices used to gather information from the environment do not have enough resources to deal with malicious cyber-attacks. Manipulating the data collected by these devices or intentionally uploading unwanted data disrupts the integrity of the system set up in terms of security [2].

Considering all these problem areas and processes, valuable features such as getting rid of single point centralization, data immateriality and transaction transparency have made the combined use of edge computing and blockchain technologies popular [4]. Security problems that may be encountered during the collection and distribution of data obtained from IoT devices using edge computing technology are tried to be eliminated by using blockchain technology [1, 6, 9]. In particular, keeping and securing transaction records has attracted the attention of many researchers. While some of the researchers care about the security dimension in storing transaction records [6], some of them have deepened their studies on analysis and scalability [8]. Many of the studies have focused on keeping and protecting the transaction records produced by IoT devices.

More than fifty countries have officially published strategy papers/regulations summarizing their official positions on cyberspace, cybercrime and cybersecurity [10, 11]. Prevention and detection of computer crimes are among the main objectives of cyber security and information security. The authorities take constitutional measures on this issue. In Turkey, Law No. 5651 came into force for "regulating the broadcasts made on the Internet and combating the crimes committed through these broadcasts". With the General Data Protection Regulation (GDPR), a regulation on data protection and privacy has been prepared for individuals. GDPR primarily aims to give individuals control of their personal information and to bring companies in the EU into compliance with these regulations [12].

In order to ensure the confidentiality, integrity and security of the data in the proposed security models in the studies carried out, specific methods are presented for that study. In the healthcare field, patients' health information is critical to the privacy of personal data. In studies built on this basis, alternative methods have been developed by combining blockchain technology to ensure the confidentiality of data collected from health devices and encryption algorithms to ensure security [7, 8]. Therapy, diagnostic and analytical data of bedridden patients, disabled individuals or individuals with mobility restrictions due to old age are monitored remotely. This has made it important to protect the ownership, storage and sharing of therapeutic data during the home therapy service, which has become popular during the pandemic period. The most suitable solution to this problem can be provided with blockchain technology [13]. In addition to data privacy, when considering data security and performance criteria, blockchain technology can be preferred in order not to resort to complex cryptographic methods. In order to provide low-time latency and real-time service, effective solutions to the secure communication problem in smart grid systems are also realized with blockchain technology [14]. Consensus and agreement between nodes is defined as a serious problem in resource allocation processes in wireless networks. This problem can be solved by optimizing the spectrum allocation, block sizes and block numbers with the consortium feature of blockchain technology to improve the service quality of users [15]. In order to increase the efficiency of authentication systems and collaborative sharing designs, systems in which edge computing and blockchain technology are used together have been proposed [16]. In Table 1, the years of studies using blockchain and edge computing technology together, the preferred consensus algorithms, the purpose of the study and literature contributions are shown in a table.

Table 1. Review of studies using blockchain and edge computing together

References	**Year**	**Consensus**	**Purpose of the Study**	**Contribution**
[17]	2020	Ethereum	Resource Allocation	A secure and distributed platform is proposed that will pro
[18]	2018	PoW	Resource Allocation	A reliable model has been proposed that can perform smar
[19]	2019	PoW	Resource Allocation	A safe energy trading framework in SDN-enabled V2G is d
[20]	2020	DPoS	Resource Allocation	A scheme is proposed for solving the tamper-proof, static r
[21]	2017	PoW	Access Management	A security update flow rule table scheme based on blockcha
[22]	2019	PoW	Access Management	A lightweight, enhanced voting mechanism is introduced to
[23]	2020	PoR	Attack Detection	A distributed, self-organizing voting mechanism based on v
[24]	2020	-	Attack Protection	An edge-cloud and SDN blockchain distributed security fra
[25]	2020	PBFT	Attack Protection	A Geographical-based, scalable, improved consensus mecha

In this study, an IoT framework with a four-layer and blockchain storage system has been developed for the secure and high-accuracy operation of IoT assets with distributed-based learning systems. In the second part of the study, firstly, an IoT device was developed for processing nodes that collect data at endpoints. This IoT end device is responsible for transmitting temperature and humidity data to the upper layer. In the second step, edge devices were developed to transfer the collected node data to the server. These edge devices are responsible for both transferring the data to the server and transmitting the learning results to the nodes. In the last step, the data collected in the center is learned with the LSTM learning algorithm and six different order classes are decided. In addition, it is explained how the processing data is stored in the blockchain. As a result, it is predicted that the proposed secure and blockchain-based IoT framework will contribute to the literature with its contribution as follows.

* It has been shown that distributed learning-based systems will provide higher learning performance by combining datasets on the central server and updating their weights.

* It has been shown that preferring edge devices in distributed IoT-based systems will use network traffic more efficiently.

* It has been observed that the transmission of data transfer with smart contract and storage in blockchain-based systems ensure data integrity and security.

* In IoT systems with high data security, solution-specific development of edge and node devices has enabled the use of faster and more secure cryptographic methods.

In the second part of the study, the technical details of the artificial intelligence-based and secure IoT infrastructure are explained. In this section, structures such as the IoT device developed, the general working architecture of the study, the prepared blockchain and smart contract structure, and the developed machine learning model are explained. In the third part, there are the findings obtained from the study. In this section, the network traffic measurements of the developed infrastructure, the speed and resource usage measurements of the infrastructure are clearly shown. In the fourth chapter, the results obtained from the study are explained.

ARTIFICIAL INTELLIGENCE BASED AND SECURED IOT FRAMEWORK

In this study, it has been tried to develop a blockchain-based distributed edge computing backbone. The study will be examined under three main headings in itself. The first part of the developed backbone consists of the technical features and visuals of the IoT card, which provides data from the environment. In the second part, the details of the designed edge computing structure and artificial intelligence processes will be discussed. In the last part, how the blockchain technology is integrated into the first and second parts will be explained.

Designed IoT Device: IoT_TH

We chose to develop the IoT card ourselves, which we will use within the layered architecture of the infrastructure we will develop. We placed sensors on the IoT card we developed to collect the temperature and humidity data of the environment. We named the card we developed using the initials of this environment data and the abbreviation of the concept of internet of things, IoT_TH. The purpose of designing this card is to measure the temperature and humidity values of the environment where the card is located and to send the values obtained to the remote server via the Wi-Fi module. ESP8266 is used as hardware, MCU and Wi-Fi module. DHT11 temperature and humidity sensor is used as the sensor. There is a Relay and a Buzzer on the board to activate when necessary. The reason for placing a relay on the board is to give this board the ability to control motors and similar equipment. The developed board is fed with 5V power. The relay and buzzer on the board work with 5V. The DHT11 sensor works with 3.3V, which is the same value as the ESP8266. There is 1 test pin on the PCB board to measure the total consumed current and voltage. With this pin, it is possible to measure how much power the card consumes in which situations. With the SS34 protection diode, when reverse polarity is applied to the board, damage to the circuit elements is prevented.

Under normal circumstances, when the location of the card changes or the SSID and password information of the Wi-Fi device to which it will be connected changes, the card must be reprogrammed and the SSID and password information must be reconfigured as hardware code. However, we can easily do this by dynamically adjusting the Wi-Fi configurations of the card through the switch we placed on the card. Thus, when the location of the card is changed, the information of the new Wi-Fi device to be connected is entered with the Wi-Fi configuration button and it can be easily adapted to the new location. Thanks to these features, the card has a more effective quality. The developed card is an ergonomic card with dimensions of 70mm - 30mm - 18mm. The internal structure, connection diagram and design of the developed card are shown in Figure 1.

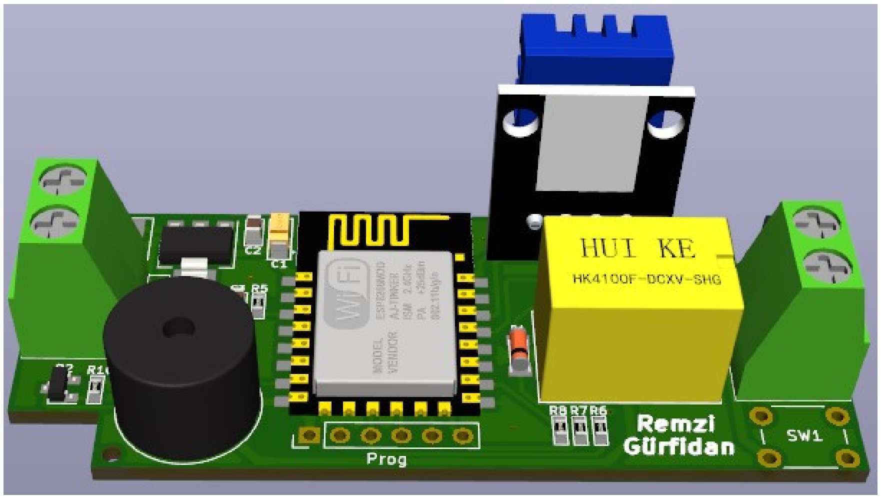


Figure 1. IoT board design developed to measure environmental values

The developed card has the ability to both collect and send media data to the upper layer and fulfill the order from the upper layer. For this reason, it can act as both a client and a server. The energy consumption and data transmission rates of the developed card are given in detail in the findings part of the study, both numerically and graphically.

Designed Edge Computing Architecture

It would be appropriate to examine the general architecture of the study in four main sections. The entire architecture of the study and the designed layers are shown in Figure 2. If we order these four parts hierarchically starting from the lowest layer, we can say that the node layer, the edge layer, the decision layer and the training layer. There is a semantic structure in this hierarchical order. It has been designed in line with the needs of the greenhouse areas selected for the test environment of the developed infrastructure. The configuration of the node layer, which is the most basic layer and created in the real greenhouse environment, will be explained. There are IoT_TH cards (N_1, N_2, N_n) in the node layer. As ?(N_x) - ?!(TH_x), each development board has a temperature and humidity sensor (TH_1, TH_2, TH_n). IoT_TH cards developed to receive environmental data are located at different locations of the greenhouses. In addition, there are heating and cooling systems connected to the IoT_TH cards that will be activated in line with the information coming from the upper layers.

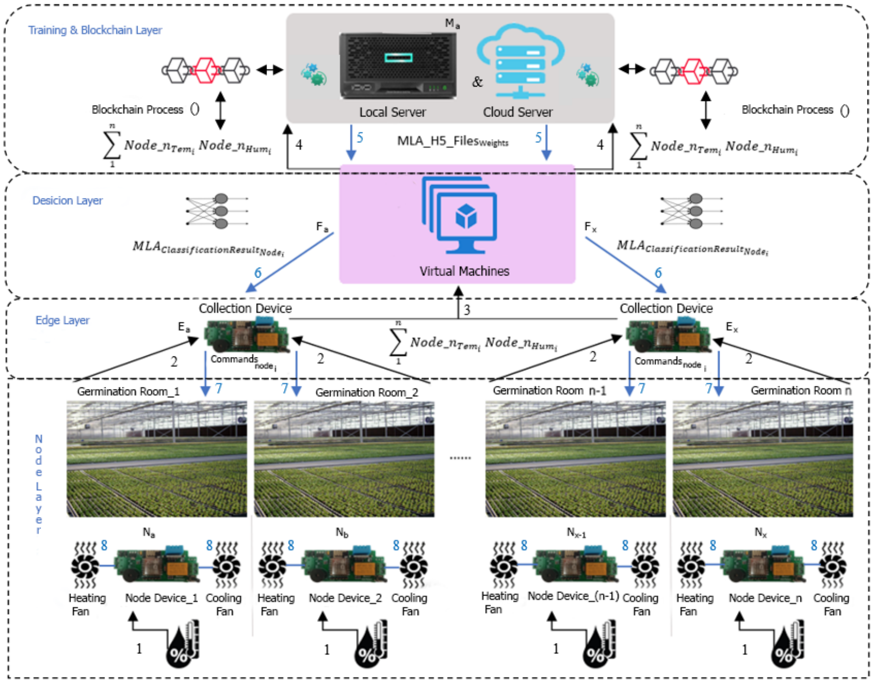


Figure 2. The general architecture of the study

Our next top layer is the edge layer, hierarchically. The purpose of this layer is to reduce the danger of network traffic that may occur on the server if the data from a lower layer is transmitted to the central server at once. Another purpose is to prevent data sending devices from being suspended while waiting for a response in return for the data they send. A simply modified version of the developed IoT_TH board is used to build this layer. This change can be described as deactivating the original sensors. In this layer, there are potentially different number of node devices under the responsibility of IoT_TH devices. ?($\mathrm{TH_x}$) transmits the values in its environment to the $\mathrm{N_x}$ device it is connected to. What is expected from this layer in the whole organization is that the collecting devices in the layer correctly label and accumulate the media data coming from the node devices they are responsible for and periodically direct them to the upper layer, the decision layer. The obtained values are sent to the collection devices in the upper layer, the Edge Layer, in a format ($[?]_1$,$[?]_2$,. . . ,$[?]_\mathrm{t}$) prepared in accordance with the smart contract format prepared for the blockchain. Each collection device $[?]_\mathrm{t}$ then stores the data sent to it by the $\mathrm{N_n}$ devices as shown in Equation 1. The data that is encrypted and stored in each $[?]_\mathrm{t}$device is sent to the Decision Layer, which is the upper layer, in fixed periods.

$$t = \sum_{n=1}^{N_n} \left(N_{\mathrm{Temp}_n}\right)_n \bigcup \ \left(N_{\mathrm{Hum}_n}\right)_n (1)$$

The detailing of the layers in the hierarchical structure will be continued with the last layer, not the next layer. We can explain the reason for this as the training part and the classification part in machine learning algorithms are inverse sequential in the architecture realized in this study.

In the last layer, Education and blockchain layer, there is one local server and one cloud server. Two main operations are performed in this layer. The first of these processes is to carry out the training process of the artificial intelligence model with data from different locations. Another process is to convert the incoming data into a blockchain structure and store it. In the first operation, a model is trained using the LSTM algorithm. The weight values of the input variables formed after the training are carefully extracted from the model. these extracted weights are meaningful for the developed model and meaningless for the human mind. The weights obtained are transferred to the decision layer, which is a lower layer, in order to perform the classification process. In the second main transaction, the incoming data is safely stored in its raw form, under the guarantee of blockchain, in a re-readable form when necessary. It is configured for both local and cloud server storage in the blockchain layer, which is the last layer where data reaches. The necessity of a dual storage system can be decided depending on the importance of the data to be obtained and transferred as a requirement of the infrastructure developed in this study. In this layer, a single piece of operation that can be performed through a machine learning algorithm is fragmented. this distribution process has brought much more effective and successful results on heterogeneous data. We proved this with numerical data in section 3.3.

Virtual machines (J_1, J_2,. . . , J_k) are located in the Decision Layer. J_k collects the encrypted data sent to it from $[?]_\mathrm{t}$ and sends it to the final blockchain storage layer as shown in Equation 2.

$J_k = \sum_{t=1}^{t} \left({}_{N_x}\right)_t$ (2)

The decision layer contains virtual machines (J_1, J_2,. . . , J_k). Jk accumulates the encrypted data sent to it from $[?]_\mathrm{t}$ and sends it to the last layer as shown in Equation 2. The decision layer completes the work of the fragmented machine learning algorithm thanks to the trained model weights it receives from the training layer. Classification processes are carried out without the need for new training only thanks to the weights, and the results are sent to the lower layer, the edge layer, for implementation.

2.3. Blockchain Architecture

There are many platforms such as Ethereum, IBM Blockchain, Hyperledger Fabric, R3 Corda that can be used to create a blockchain structure. In the blockchain structure realized in this study, Hyperledger Fabric was preferred as the infrastructure. The smart contract structure, use of docker container technology, support for a common programming language such as Java in smart contract preparation [26], and the use of (Practical Byzantine Fault Tolerance) PBFT consensus algorithm [27] were effective in choosing this infrastructure.

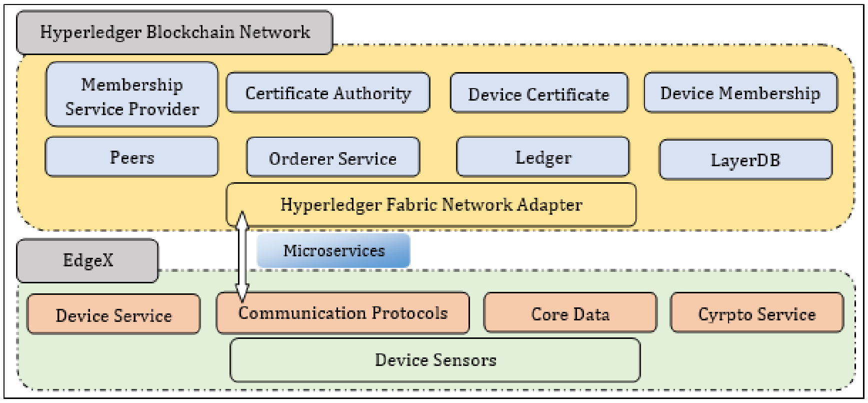


Figure 3. Hyperledger Blockchain Network Architecture [28]

Figure 3 shows the internal structure of the Hyperledger Blockchain network. In the Hyperledger Blockchain network, there are service providers, certificate authority, device certification center, device membership module, peers, administrator services, registry and database to network members. Hyperledger Fabric Network Adapter is available for communication with end devices. Thanks to microservices written with edge devices, communication is provided by following communication protocols. Edge devices have device service, device sensors, kernel information and crypto services. In the study referenced in Figure 3, while blockchain technology was combined with edge computing technology, microservices and messaging systems were prepared as separate layers and used as middleware. Message systems such as Kafka and Zookeeper transmit the data it listens from the microservice layer to the blockchain structure. In the infrastructure we designed, the edge structure and blockchain structure communicate with the prepared microservice.

2.3.1. Smart Contract Structure

A smart contract is executable code running on a designed blockchain, prepared to facilitate the terms of an agreement between parties experiencing a mutual trust problem, to execute and execute transactions. It can be thought of as a system that allows digital asset transactions to all or some of the relevant parties after the predefined rules in smart contracts are fulfilled by the parties [29]. Compared to traditional contracts, smart contracts do not allow a trusted third party to operate, resulting in low transaction costs. Prepared smart contracts will be assigned to a unique address of 20 bytes. Once the smart contract is paired with a blockchain, the code assigned to the contract cannot be changed. In order for a contract to be run, it is sufficient for users to send the transaction they want to perform to the address of the contract. The requested transaction is then evaluated by each consensus node in the network to arrive at a consensus. The status of the contract will then be updated based on the result achieved [30].

The consensus algorithms used largely determine the performance of the distributed system for blockchains, with variables such as throughput, latency, node scalability [31]. In the prepared smart contract, the initLedger method is run to perform the initial settings that need to be made at the start of the ledger. Before new data is written to the Ledger, it is checked whether there is data with the same id. When a positive answer is received, a new object is created and the ledger registration process is started and the new record is returned at the end of the process. GetAllAsset method can be executed to read the records. After the necessary permissions are checked, the data registered in the ledger can be read and listed with the help of an iterator. The pseudo code of the smart contract created in Algorithm 1 is shown.

Algorithm 1 Smart Contract Pseudo Code

```
function initLedger () config LedgerStandarts () function CreateAsset (ctx, p
```

2.4. Developed Long-Short Term Memory (LSTM) Model

In the developed study, the data collectors shown in Figure 1, located in 3 different locations, transferred the humidity and temperature data of the environment they were in for six months to the server. Since the locations are used for different purposes, it is seen that the temperature and humidity values remain within certain ranges. It is seen that the first location has 0-15° degrees, the second location has 15-25° degrees, and the third location has 23-30° degrees. There are six different classifications for heating and cooling settings within the model. It has datasets containing only two classifications in three locations. Modeling of systems that take such narrow data ranges and classification with artificial intelligence systems will be limited. The model learned with data from a single location will not produce correct results when it encounters data from other classifications. For this reason, in our study, it is aimed to train the data of different locations with narrow data ranges on central servers and to transfer their weights to all locations. Thus, classification will be performed against a temperature and humidity value encountered for the first time in the number one location, and the system will continue to operate without any problems. The data obtained after the data collection process was stored on the central server. As given in Table 2, six different order classifications have been determined for the operation of the heating and cooling system needed according to the ranges of temperature and humidity values.

Table 2. Temperature ranges and classification groups to be used for model training

Humidity Range (g/m3 - %)	Temperature Range ([?]C)	Class Type
68-71	0-9	1
70-73	9,1- 15	2
71-74	15,1- 21	3

Humidity Range (g/m3 - %)	Temperature Range ([?]C)	Class Type
73-75	22,1- 23	4
74-77	23,1 – 27	5
75-80	27,1- more	6

The data set was created by combining the collected data in a single center. Long short-term memory (LSTM) algorithm is preferred in the model developed to predict the classification type according to temperature and humidity values. LSTM is a recurrent neural network (RNN) architecture that remembers values at random intervals [32]. An LSTM is well suited for classifying, processing and predicting time series given the time delays of unknown size and duration between significant events [33].

Basically, the internal structure of the LSTM architecture; It consists of input (Equation 4), forget (Equation 5) and output gates (Equation 7) and input layer (Equation 6). In the LSTM architecture, first of all, ?? and ?-1, which are used as inputs, are decided which information will be deleted. This is done by a sigmoid layer (Equation 3) called the forget the door layer [34].

$\sigma\left(x\right)=\left(1+e^{-x}\right)^{-1}$ (3)

$i_t=\sigma\left(W_i x_t+R_i h_{t-1}+b_i\right)$ (4)

$f_t=\sigma\left(W_f x_t+R_f h_{t-1}+b_f\right)$ (5)

$g_t=tanh\left(W_g x_t+R_g h_{t-1}+b_g\right)$ (6)

$o_t=\sigma\left(W_o x_t+R_o h_{t-1}+b_o\right)$ (7)

In the LSTM model, the first step after dataset partitioning is to decide which information to discard in the cell. This decision is made by the sigmoid layer called the forget gate layer in the LSTM model. It is then decided which new information will be stored in the cell state. For this, the input gate layer is used. A vector of the new values to be generated is also produced with the tanh layer. The output layer is created by combining the generated information and vectors. This output is filtered by cell state. Here, first, a SoftMax layer is run, which decides which parts of the cell state it will output. The SoftMax layer extracts the ordered order classifications.

In the experiments conducted to develop the proposed LSTM model and evaluate its performance; Intel (R) I9 3.2 Ghz processor hardware with 64 GB RAM and Python programming language is used in the Spyder interface. The dataset was randomly partitioned as 80% for training and 20% for testing. The hyperparameters used in the LSTM model are given in Table 3.

Table 3. Hyperparameters and values of the model

Hyperparameters	**Value**
Training set size	675
Test set size	220
Initial learning rate	0.005
Dropout rate	0.5
Batch size	4
LearnRateDropPeriod	125
LearnRateDropFactor	0,2
Max Epoch Iterations	200
fullyConnectedLayer	20
dropoutLayer	0,5

FINDINGS

Obtained performance measurements were carried out on a computer with an Intel Xeon 3.40 Ghz processor and 32 GB RAM.

3.1. Network Traffic Measurements with Edge Computing

Two different systems have been established to measure the performance of the edge computing structure. The purpose of the two systems is to reveal the advantages and disadvantages of the proposed edge computing model over standard delivery methods. For this purpose, four node points were created and ten thousand fixed data were sent directly to the server from each of them. During this process, the total number of packets sent, total packet sizes and delivery times were measured. In the second system, four nodes were created and the nodes were grouped in pairs and associated with an end device. The end devices, on the other hand, collect the data coming from the node devices connected to them and pack every ten data sets and send them to the server. During this process, the total number of packets sent, total packet sizes and delivery times were measured. Obtained measurement results are shown in Figure 4.

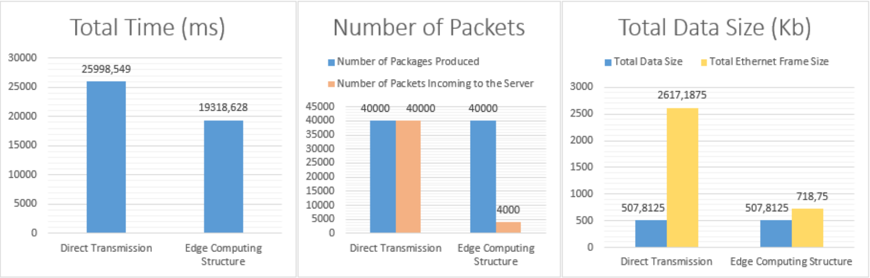


Figure 4. Network Performance Measurements of Edge Computing System

When the total number of packets sent to the server is examined, 40,000 packets were produced in the standard direct data transmission system and 40,000 packets were sent to the server. In the edge computing system, since the data to be sent is accumulated on the end devices, 40,000 packets are produced, but 4,000 packets are sent to the server. In addition, although the raw data size to be sent to the server is 507.8125 Kilobytes (KB) in both systems, the total frame size reaching the server is 2617.1875 KB in the standard direct transmission system and 718.75 KB in the edge computing system. The main reason for this difference is the source address, destination address, etc. added to the package during package creation apart from the raw data are standard sections. As a natural consequence of the smaller number of packets sent in the Edge computing system, the total data size sent has made the Edge Computing system advantageous. As a result of these sub-reasons, the Edge computing system with a value of 19318.628 ms in the temporal measurement of data transmission is 7.5% superior to the standard direct data transmission system with a value of 259998.549 ms in terms of temporal cost.

3.2. Blockchain Performance Metrics

The performance metrics of the blockchain structure used in the developed system were measured. The purpose of this measurement is to provide data security, data integrity and data tampering, while revealing the time cost of the transactions performed. Figure 5 shows the amount of work that can be done in response to the data sent per unit time.

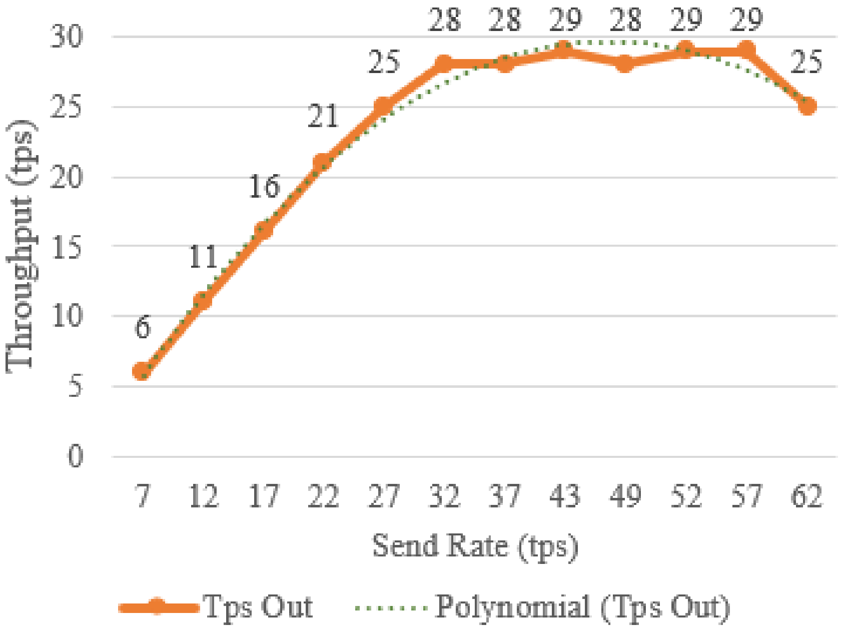


Figure 5. Number of transactions per unit time

Figure 6 shows the measurement values of the delay times of the blockchain transaction performed. Minimum delay, maximum delay and average delay are given in the same graph. As shown in the graph, the increase in the delivery rate per unit time causes the delay time to increase. Considering the amount of work sent per unit time in the performance test environment to determine the limits of the system, and considering the security contribution obtained, the delay time is considered to be a quite acceptable value.

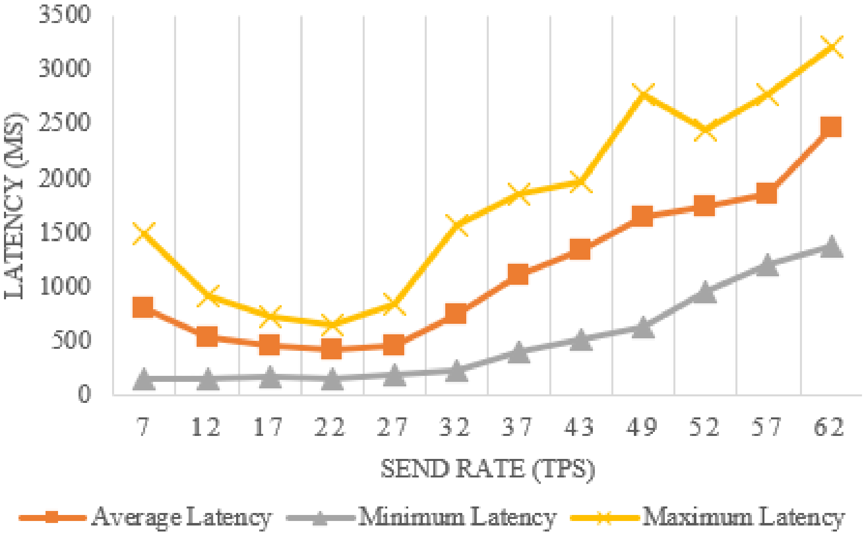


Figure 6. Delay time per unit time

In Figure 7, the usage rates of the resources of the computer where the test is performed at the time the blockchain transactions are carried out are shown. In our computer, which has a strong processing power and RAM capacity, the upper limits of the system have been tried to be measured by increasing the amount of work demand per unit time. In the results obtained, it was determined that the RAM capacity remained constant, and the processor power increased linearly up to 200 tps after the polynomial 200 tps value.

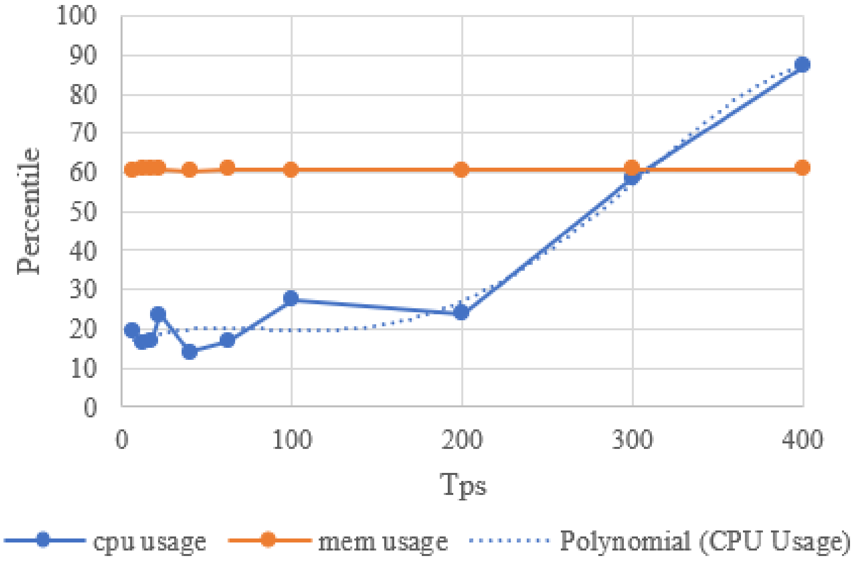


Figure 7. Blockchain Process Resource Usage

3.3. Evaluation of Model Results

The classification model is primarily carried out by separately training 3 different datasets from the locations. In the second step, a single dataset was obtained as a result of combining all datasets. In the developed LSTM model, this dataset was also trained and the results were obtained.

First of all, the training of three different locations was carried out separately and the results are shown in Figure 8. The model was completed with 200 training rounds, but the accuracy values were provided in a maximum of 100 training rounds, and then the model accuracy was fixed. As a result of training the data from the locations, respectively, 1st location 85%, 2nd location 84% and 3rd location 84% accuracy value. As a result, the fact that the temperature and humidity values are in the data set that he learned kept the order classification accuracies high. Afterwards, all datasets were combined and the training of the model was performed again and 94% accuracy was obtained. The most important reason for the increase in the accuracy of the model is due to the significant increase in the amount of data in the data set.

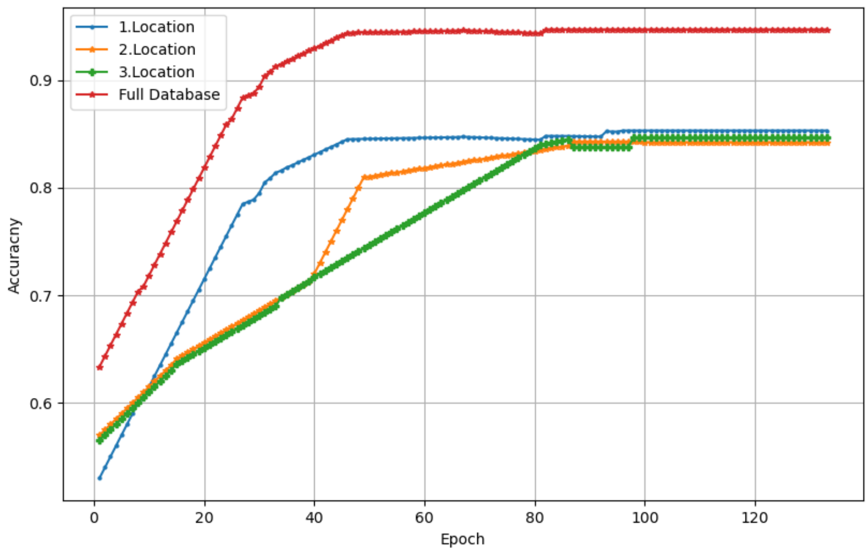


Figure 8. Training accuracy evaluation of the model separately for three locations and common dataset

An average of 94% accuracy was achieved in the classification of data from three different locations according to six different order classes. As shown in Figure 9, in the classification of the data from the locations, it was observed that the temperature and humidity values of the location shifted to the next or previous order class due to the proximity.

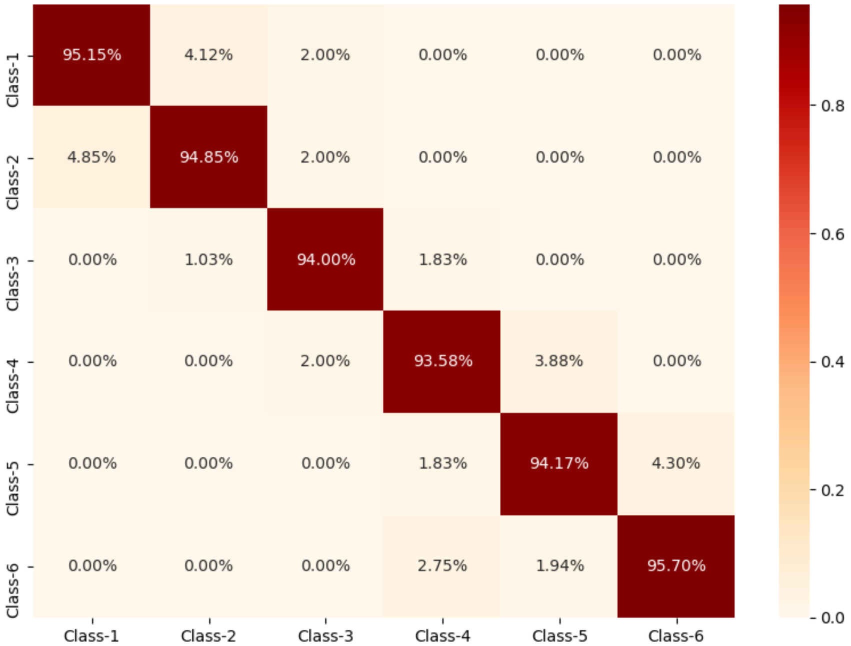


Figure 9. Confusion matrix of the main model's classification success

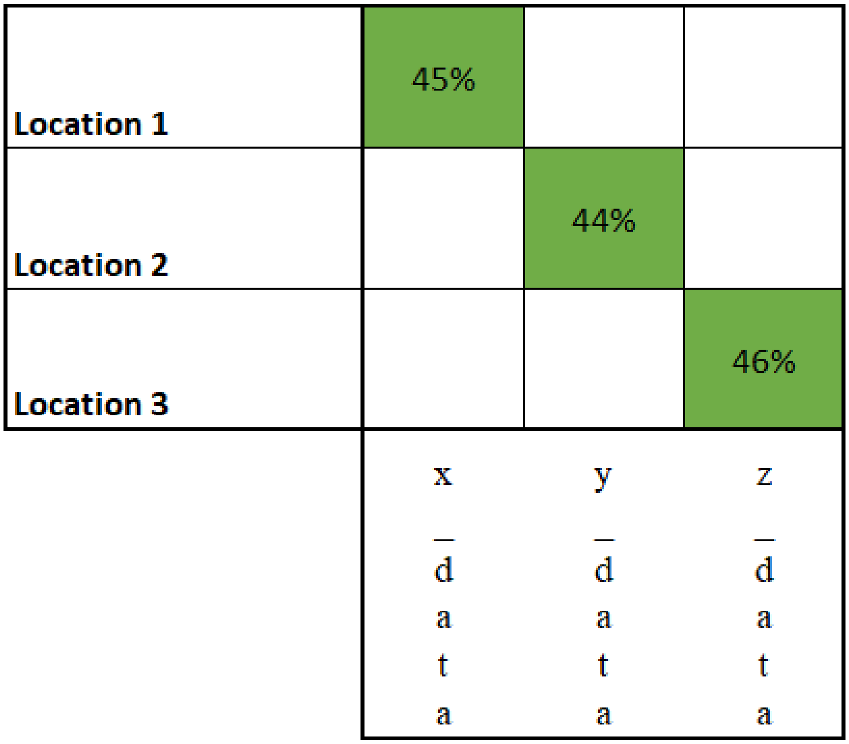
In order to determine the advantages of the federated learning model, after creating models with the training data of the datasets for each location, test data and data from other locations were used as shown in Table 4.

The aim here is to show that the success of classification made with data ranges that are not in the training dataset will be low.

Table 4. Dataset mapping of locations

Model	Test Dataset	Dataset Name
Location 1 Weight	Location 2 and Location 3	x_data
Location 2 Weight	Location 1 and Location 3	y_data
Location 3 Weight	Location 1 and Location 2	z_data

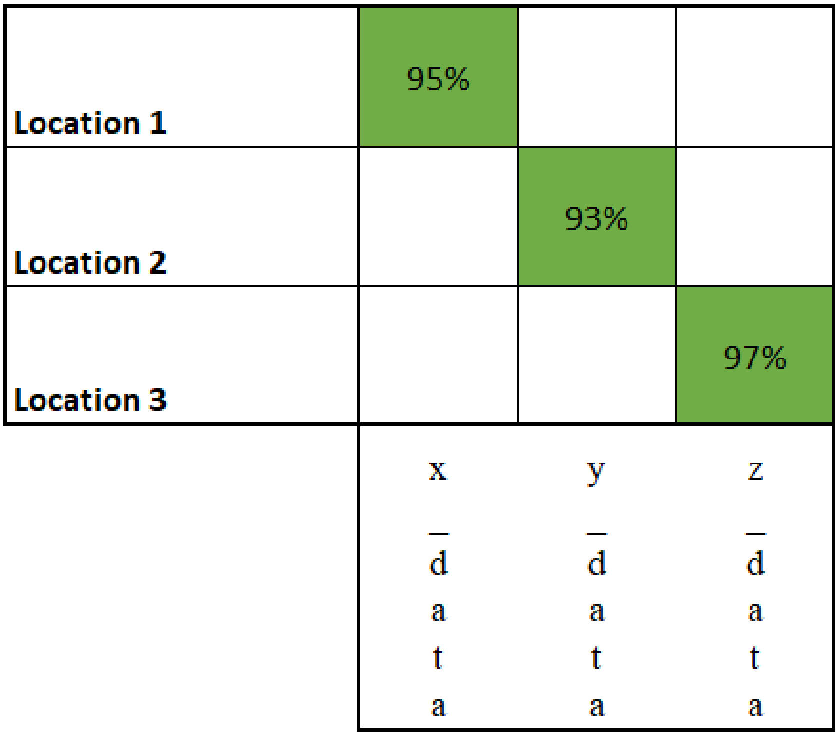
As seen in Figure 10, three different locations were trained with their own dataset and the test was performed with data from different locations. Model classification success decreased from 85% to 44% as an independent location. If the regional data used in the training of the models do not provide the classification diversity of the proposed system, it can cause serious problems in classification predictions.



Location 1	45%		
Location 2		44%	
Location 3			46%
	x_data	y_data	z_data

Figure 10. Classification success of models in different datasets

In the final stage of the performance evaluation of the proposed model, the data set from three locations and collected in the central server and the weights of the trained model were transferred to the edge systems and evaluated. Here, it has been observed that the models in the locations have never been encountered before, and that the data ranges can be classified by combining the weights obtained from the training of other locations. First of all, the weights of the main model trained with the central dataset were distributed to the edge locations and the weight was updated. Afterwards, each location was asked to classify the data ranges they encountered for the first time, as shown in Table 5. As shown in Figure 11, data ranges encountered for the first time were recognized by the weights learned in the main model, and order classification was performed with an average of 95% accuracy.



Location 1	95%		
Location 2		93%	
Location 3			97%
	x_data	y_data	z_data

Figure 11. Performance evaluation of the test sets shown in Table 5 after the combined dataset and the trained model weights are distributed to the edge locations.

By applying the federated learning architecture of the classification estimation model developed with the LSTM architecture, the weights of the model are distributed over different locations. As a result of this, as seen in Table 5, the improvement was followed and the following results were obtained in this framework.

* The low number of datasets and low data diversity of local models developed in distributed locations negatively affect the accuracy.

* An average increase in accuracy performance from 85% to 95% accuracy was observed at each location.

* Classification accuracy drops to 45% at a temperature and humidity outside of its own temperature and humidity range in each location's own dataset.

* Updating the weights of the locations centrally has enabled both the widening of the temperature and humidity ranges perceived by the locations, and higher accuracy with the increase in the number of datasets.

Table 5. Evaluation of federated learning architecture of different locations

	Test Your Own Datasets	Testing with Different Datasets	Cross-test after combining the weights of the thr
Location 1	85%	45%	95%
Location 2	84%	44%	93%
Location 3	84%	46%	97%

CONCLUSION

The most important factors affecting performance in learning-based systems are the number of data and data diversity. In running learning-based systems at distributed points in a system, accuracy performance may not reach the desired level due to data set diversity and insufficient amount of data. Collecting, storing and training data in a central location brings certain risks and precautions.

When the features of the infrastructure proposed in this study are compared with the studies carried out in the literature, Table 6 shows the advantages of our infrastructure. GDPR policies, log immutability, storage strategy, edge computing system prepared for network performance stand out as important positive differences. In addition, basic features such as data security, non-tamperability, distributed structure, and identification, which are also present in other studies, are also available in our infrastructure.

Table 6. Comparison of the use of Blockchain to secure data in the cloud ecosystem.

References / Features	**[35]**	**[36]**	**[37]**	**[38]**	**[39]**	**[40]**	**[41]**	**[42]**	**[43]**	**[44]**
Artefact Identification		-			-	-	-	-	-	
Permissioned Blockchain	-								-	
Securing Data Integrity									-	
Blockchain Trustworthiness									-	
Tamperproof Digital Data										
Distributed										
GDPR Challenges	-	-	-	-	-	-	-	-	-	
Log Immutability	-	-	-	-	-		-	-	-	
Edge Devices Collaboration	-	-	-	-	-	-	-	-	-	-
Storage	C	C	C	C	C	C	C	C	C	C
Framework	-	E	HF	-	-	HF	HF	HF	-	HF
Year	2017	2018	2019	2019	2019	2019	2019	2019	2020	2021

References / Features	**[35]**	**[36]**	**[37]**	**[38]**	**[39]**	**[40]**	**[41]**	**[42]**	**[43]**	**[44]**
C: Cloud, L: Local, E: Exonum, HF: Hyperledger Fabric	C: Cloud, L: Local, E: Exonum, HF: Hyperledger Fabric	C: Cloud, L: Local, E: Exonum, HF: Hyperledger Fabric	C: Cloud, L: Local, E: Exonum, HF: Hyperledger Fabric	C: Cloud, L: Local, E: Exonum, HF: Hyperledger Fabric	C: Cloud, L: Local, E: Exonum, HF: Hyperledger Fabric	C: Cloud, L: Local, E: Exonum, HF: Hyperledger Fabric	C: Cloud, L: Local, E: Exonum, HF: Hyperledger Fabric	C: Cloud, L: Local, E: Exonum, HF: Hyperledger Fabric	C: Cloud, L: Local, E: Exonum, HF: Hyperledger Fabric	C: Cloud, L: Local, E: Exonum, HF: Hyperledger Fabric

Although different infrastructures are used to create the blockchain structure in the studies, Hyperledger Fabric infrastructure is generally preferred. In addition, data integrity, data security, tamper resistance and distributed structure have been successfully carried out in many studies.

In this study, a four-layer architecture is presented to increase the learning success of data processing systems in distributed locations, to achieve higher accuracy and to create a secure storage. The data from the nodes are incrementally collected to the central server and trained with the LSTM deep learning algorithm. The weights containing the learning results are transferred to the decision layer and all node devices are used. In addition, post-training data is safely stored in a blockchain-based storage system. Thus, learning capabilities have been transferred to all devices. In the tests and evaluations, each node showed high learning performance in new data, apart from its own data diversity. As a result, it is aimed to solve the problems experienced in cases where the data obtained from a location, the number of data and the diversity are insufficient, with the central architecture. In particular, it is foreseen that distributed systems will be processed securely and with higher accuracy, and that IoT systems will be used widely and effectively.

REFERENCES

[1] Zhang, L., Zou, Y., Wang, W., Jin, Z., Su, Y., & Chen, H. (2021). Resource allocation and trust computing for blockchain-enabled edge computing system. *Computers & Security* , *105* , 102249.

[2] Liu, J., Gong, B., & Wang, Q. (2022). A trusted proof mechanism of data source for smart city. *Future Generation Computer Systems* , *128* , 349-364.

[3] Rocha, A. S., Pinheiro, B. A., & Borges, V. C. (2021). Secure D2D caching framework inspired on trust management and blockchain for Mobile Edge Caching. *Pervasive and Mobile Computing* , *77* , 101481.

[4] Gadekallu, T. R., Pham, Q. V., Nguyen, D. C., Maddikunta, P. K. R., Deepa, N., Prabadevi, B., . . . & Hwang, W. J. (2021). Blockchain for edge of things: applications, opportunities, and challenges. *IEEE Internet of Things Journal* , *9* (2), 964-988.

[5] Song, J., Gu, T., & Mohapatra, P. (2021). How BlockChain Can Help Enhance The Security And Privacy in Edge Computing? *arXiv preprint arXiv:2111.00416* .

[6] Pahl, C., El Ioini, N., & Helmer, S. (2018, March). A Decision Framework for Blockchain Platforms for IoT and Edge Computing. In *IoTBDS* (pp. 105-113).

[7] Christo, M. S., Jesi, V. E., Priyadarsini, U., Anbarasu, V., Venugopal, H., & Karuppiah, M. (2021). Ensuring Improved Security in Medical Data Using ECC and Blockchain Technology with Edge Devices. *Security and Communication Networks* , *2021* .

[8] Kumar, G., Saha, R., Lal, C., & Conti, M. (2021). Internet-of-Forensic (IoF): A blockchain based digital forensics framework for IoT applications. *Future Generation Computer Systems* , *120* , 13-25.

[9] Faiyaz, F. R., Lisa, A. S., Rahat, L., Tabassum, N., & Istiaq, W. B. (2021). *Blockchain-based edge computing for medical data storage & processing using federated learning* (Doctoral dissertation, Brac University).

[10]Hathaway, M., & Klimburg, A. (2012). Preliminary considerations: on national cyber security. National Cyber Security Framework Manual. NATO Cooperative Cyber Defence Centre of Excellence, Tallinn.

[11]Von Solms, R., & Van Niekerk, J. (2013). From information security to cyber security. computers & security, 38, 97-102.

[12] ”Proposal for a Regulation of the European Parliament and of the Council on the protection of individuals with regard to the processing of personal data and on the free movement of such data (General Data Protection Regulation)”. Council of the European Union. 2019.

[13] Rahman, M. A., Hossain, M. S., Loukas, G., Hassanain, E., Rahman, S. S., Alhamid, M. F., & Guizani, M. (2018). Blockchain-based mobile edge computing framework for secure therapy applications. *IEEE Access* , *6* , 72469-72478.

[14] Wang, J., Wu, L., Choo, K. K. R., & He, D. (2019). Blockchain-based anonymous authentication with key management for smart grid edge computing infrastructure. *IEEE Transactions on Industrial Informatics* , *16* (3), 1984-1992.

[15] Guo, F., Yu, F. R., Zhang, H., Ji, H., Liu, M., & Leung, V. C. (2019). Adaptive resource allocation in future wireless networks with blockchain and mobile edge computing. *IEEE Transactions on Wireless Communications* , *19* (3), 1689-1703.

[16] Guo, S., Hu, X., Guo, S., Qiu, X., & Qi, F. (2019). Blockchain meets edge computing: A distributed and trusted authentication system. *IEEE Transactions on Industrial Informatics* , *16* (3), 1972-1983.

[17] L. Cui, S. Yang, Z. Chen, Y. Pan, Z. Ming, and M. Xu, “A Decentralized and Trusted Edge Computing Platform for Internet of Things,” *IEEE Internet of Things Journal* , vol. 7, no. 5, pp. 3910–3922, 2020

[18] N. Z. Aitzhan and D. Svetinovic, “Security and Privacy in Decentralized Energy Trading Through Multi-Signatures, Blockchain and Anonymous Messaging Streams,” *IEEE Transactions on Dependable and Secure Computing* , vol. 15, no. 5, pp. 840–852, 2018.

[19] A. Jindal, G. S. Aujla, and N. Kumar, “SURVIVOR: A blockchain based edge-as-a-service framework for secure energy trading in SDN-enabled vehicle-to-grid environment,” *Computer Networks* , vol. 153, no. 2019, pp. 36–48, 2019.

[20] W. Sun, J. Liu, Y. Yue, and P. Wang, “Joint Resource Allocation and Incentive Design for Blockchain-Based Mobile Edge Computing,” *IEEE Transactions on Wireless Communications* , pp. 1–1, jun 2020.

[21] P. K. Sharma, S. Singh, Y. S. Jeong, and J. H. Park, “DistBlockNet: A Distributed Blockchains-Based Secure SDN Architecture for IoT Networks,” *IEEE Communications Magazine* , vol. 55, no. 9, pp. 78–85, 2017

[22] S. Biswas, K. Sharif, F. Li, S. Maharjan, S. P. Mohanty, and Y. Wang, “PoBT: A Lightweight Consensus Algorithm for Scalable IoT Business Blockchain,” *IEEE Internet of Things Journal* , vol. 7, no. 3, pp. 2343–2355, 2020

[23] A. Asheralieva and D. Niyato, “Reputation-Based Coalition Formation for Secure Self-Organized and Scalable Sharding in IoT Blockchains with Mobile Edge Computing,” *IEEE Internet of Things Journal* , vol. 4662, no. c, pp. 1–1, 2020.

[24] D. V. Medhane, A. K. Sangaiah, M. S. Hossain, G. Muhammad, and J. Wang, “Blockchain-Enabled

Distributed Security Framework for Next-Generation IoT: An Edge Cloud and Software-Defined Network-Integrated Approach," *IEEE Internet of Things Journal* , vol. 7, no. 7,pp. 6143–6149, jul 2020.

[25] L. Lao, X. Dai, B. Xiao, and S. Guo, "G-PBFT: A Location-based and Scalable Consensus Protocol for IoT-Blockchain Applications,"*Proceedings - 2020 IEEE 34th International Parallel and Distributed Processing Symposium, IPDPS 2020* , pp. 664–673, 2020

[26] Sousa, J., Bessani, A., & Vukolic, M. (2018, June). A byzantine fault-tolerant ordering service for the hyperledger fabric blockchain platform. In *2018 48th annual IEEE/IFIP international conference on dependable systems and networks (DSN)* (pp. 51-58). IEEE.

[27] Xu, X., Zhu, D., Yang, X., Wang, S., Qi, L., & Dou, W. (2021). Concurrent practical byzantine fault tolerance for integration of blockchain and supply chain. *ACM Transactions on Internet Technology (TOIT)* , *21* (1), 1-17.

[28] Ajayi, O. J., Rafferty, J., Santos, J., Garcia-Constantino, M., & Cui, Z. (2021). BECA: A Blockchain-Based Edge Computing Architecture for Internet of Things Systems. *IoT* , *2* (4), 610-632.

[29] V. Buterin, "A next-generation smart contract and decentralized application platform.," Available online at: https://github.com/ethereum/wiki/wiki/White-Paper/ [Accessed 19/02/2017].

[30] Alharby, M., & Van Moorsel, A. (2017). Blockchain-based smart contracts: A systematic mapping study. *arXiv preprint arXiv:1710.06372* .

[31] Li, W., Feng, C., Zhang, L., Xu, H., Cao, B., & Imran, M. A. (2020). A scalable multi-layer pbft consensus for blockchain. *IEEE Transactions on Parallel and Distributed Systems* , *32* (5), 1146-1160.

[32] Gers, F. A., Schmidhuber, J., & Cummins, F. (2000). Learning to forget: Continual prediction with LSTM. *Neural computation* , *12* (10), 2451-2471.

[33] Jalayer, M., Orsenigo, C., & Vercellis, C. (2021). Fault detection and diagnosis for rotating machinery: A model based on convolutional LSTM, Fast Fourier and continuous wavelet transforms. *Computers in Industry* , *125* , 103378.

[34] Süzen, A. A., Yildiz, Z., & Yilmaz, T. (2019). Lstm Tabanlı Derin Sinir Ağı Ile Ayak Taban Basınç Verilerinden Vki Durumlarının Sınıflandırılması. *Bitlis Eren Üniversitesi Fen Bilimleri Dergisi* , *8* (4), 1392-1398.

[35] Liang, X., Shetty, S., Tosh, D., Kamhoua, C., Kwiat, K., & Njilla, L. (2017, May). Provchain: A blockchain-based data provenance architecture in cloud environment with enhanced privacy and availability. In *2017 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID)* (pp. 468-477). IEEE.

[36] Putz, B., Menges, F., & Pernul, G. (2019). A secure and auditable logging infrastructure based on a permissioned blockchain. *Computers & Security* , *87* , 101602.

[37] Lone, A. H., & Mir, R. N. (2019). Forensic-chain: Blockchain based digital forensics chain of custody with PoC in Hyperledger Composer. *Digital investigation* , *28* , 44-55.

[38] Cebe, M., Erdin, E., Akkaya, K., Aksu, H., & Uluagac, S. (2018). Block4forensic: An integrated lightweight blockchain framework for forensics applications of connected vehicles. *IEEE Communications Magazine* , *56* (10), 50-57.

[39] Tian, Z., Li, M., Qiu, M., Sun, Y., & Su, S. (2019). Block-DEF: A secure digital evidence framework using blockchain. *Information Sciences* , *491* , 151-165.

[40] Zheng, W., Zheng, Z., Chen, X., Dai, K., Li, P., & Chen, R. (2019). Nutbaas: A blockchain-as-a-service platform. *Ieee Access* , *7* , 134422-134433.

[41] Nyaletey, E., Parizi, R. M., Zhang, Q., & Choo, K. K. R. (2019, July). BlockIPFS-blockchain-enabled interplanetary file system for forensic and trusted data traceability. In *2019 IEEE International Conference on Blockchain (Blockchain)* (pp. 18-25). IEEE.

[42] Rane, S., & Dixit, A. (2019, January). BlockSLaaS: Blockchain assisted secure logging-as-a-service for cloud forensics. In *International Conference on Security & Privacy* (pp. 77-88). Springer, Singapore.

[43] Noura, H. N., Salman, O., Chehab, A., & Couturier, R. (2020). DistLog: A distributed logging scheme for IoT forensics. *Ad Hoc Networks* , *98* , 102061.

[44] Awuson-David, K., Al-Hadhrami, T., Alazab, M., Shah, N., & Shalaginov, A. (2021). BCFL logging: An approach to acquire and preserve admissible digital forensics evidence in cloud ecosystem. *Future Generation Computer Systems* , *122* , 1-13.