AIBLOCK: Blockchain based Lightweight Framework for Serverless Computing using AI

Muhammed Golec, Deepraj Chowdhury, Shivam Jaglan, Sukhpal Singh Gill and Steve Uhlig

2022 22nd International Symposium on Cluster, Cloud and Internet Computing (CCGrid) | 978-1-6654-9956-9/22/$31.00 ©2022 IEEE | DOI: 10.1109/CCGRID54584.2022.00106

***Abstract*—Artificial intelligence (AI)-based studies have been carried out recently for the early detection of COVID-19. The goal is to prevent the spread of the disease and the number of fatal cases. In AI-based COVID-19 diagnostic studies, the integrity of the data is critical to obtain reliable results. In this paper, we propose a Blockchain-based framework called AIBLOCK, to offer the data integrity required for applications such as Industry 4.0, healthcare, and online banking. In addition, the proposed framework is integrated with Google Cloud Platform (GCP)-Cloud Functions, a serverless computing platform that automatically manages resources by offering dynamic scalability. The performance of five different machine learning models is evaluated and compared in terms of Accuracy, Precision, Recall, F-Score and Area under the curve (AUC). The experimental results show that decision trees gives the best results in terms of accuracy (98.4%). Further, it has been identified that utilization of Blockchain technology can increase the load on memory.**

***Index Terms*—Blockchain, Artificial Intelligence, Serverless Computing, Data Integrity.**

I. Introduction

The coronavirus (COVID-19) epidemic caused by severe acute respiratory syndrome (SARS) is a pandemic that is in effect since December 2019 [1]. Worldwide, this pandemic has had devastating effects both socially and economically. The number of confirmed fatal cases reported by World Health Organization (WHO) is 5,705,754 as of February 2022 [2]. With isolation rules and the roll-out of vaccines, death rates and the spread of the disease have been reduced. On the other hand, studies have been carried out on promising developments in the field of computer science and on the early detection and prevention of the spread of this pandemic and possible future pandemics. One of these developments is the Internet of Things (IoT) applications, which is a communication network of physical objects that can be connected to each other over the Internet [3]. It can be used in a wide range of areas with e-health applications such as Industry 4.0, smart city, and home care [4]. With the accuracy rate in Machine Learning (ML) techniques reaching satisfactory levels, health studies using IoT and ML started to increase [5]. The data received from users through sensors are trained with ML models, and patients are directed to the nearest health institutions in case of a possible pandemic detection [6].

M. Golec, S. S. Gill and S. Uhlig are with School of Electronic Engineering and Computer Science, Queen Mary University of London, United Kingdom. Email: m.golec@qmul.ac.uk, s.s.gill@qmul.ac.uk, steve.uhlig@qmul.ac.uk

D. Chowdhury is with the department of Electronics and communication engineering, International Institute of Information Technology, Naya Raipur, India. Email: deepraj19101@iiitnr.edu.in

S. Jaglan is with Department of Electrical Engineering, Indian Institute of Technology (IIT), Delhi, India. Email: ee1190053@iitd.ac.in

Corresponding author: Muhammed Golec

In healthcare applications using IoT, it is difficult to ensure data integrity due to the nature of IoT [7]. The data of the user stored in the database should not be changed (immutability), so misdiagnosis and negative problems that may arise as a result should be prevented. Here, Blockchain is one of the most useful technology to offer information security [7]. Blockchain consists of blocks created by digitally signing each block and appending this block with its hash value to the previous block [8]. Blockchain has a wide range of uses; for example, it has been utilized in financial systems with digital currencies such as Bitcoin and in areas where smart contracts such as Hyperledger are used [9]. Each new block added to the Blockchain is processed together with the hash of the previous block and added to the end of the chain. Any attempt at modifying any block will affect the hash value of each block in a chain. That's why blockchain provides immutable data storage. Blockchain is promising in IoT applications where data integrity is important.

The number of IoT devices, which is approximately 22 billion today, is expected to be around 50 billion by 2030 [10]. As the number of IoT devices increases day by day, the amount of data that needs to be processed produced by IoT devices is becoming very large. IoT devices are known to have processing capacity and energy limitations and need external processing power and resources to process the huge amounts of data they generate [11]. One of the methods used to meet this need is cloud computing. When cloud computing was first introduced, users were charged according to the resources they allocated, not the resources they consumed, leading to extra cost [12]. To reduce these costs, it is necessary to rely on dynamic scaling of the resources. To solve this scalability problem, the concept of serverless computing was introduced recently. Serverless computing is a model that scales automatically depending on the customer's resource needs and works with a pay-as-you-go policy as a pricing policy [13]. The customer does not need to allocate resources in advance and the system is scaled as needed. In this way, the processing power and resources required for processing the large amount of data produced in IoT applications can be provided, thanks to serverless computing.

A. *Motivation and Our Contributions*

High accuracy rates have been achieved in the detection of COVID-19 using ML techniques. In these studies, the data received through the IoT and sensors are sent to the cloud

978-1-6654-9956-9/22/$31.00 ©2022 IEEE
DOI 10.1109/CCGrid54584.2022.00106

via a secure communication channel, and recognition COVID-19 is made in the deployed ML model here. The security of the communication channel between the IoT and the Cloud is important for data security and privacy. On the other hand, the data stored in the Cloud database can be changed by malicious people, thus causing negative consequences such as misdiagnosis. To prevent this, there is a need for a technique that ensures the integrity of health data. Blockchain is seen as a promising approach due to its features such as ensuring data integrity and immutability with the functions and methods it uses to meet this need. In this work, we propose novel IoT-based COVID-19 detection framework called AIBLOCK, which uses Serverless Computing and Blockchain. Literature reported that, there is no existing solution which can handle dynamic scalability and data integrity simultaneously for IoT-based COVID-19 detection. The main contributions of this work are:

- We propose a lightweight framework, called AIBLOCK, which leverages serverless computing to offer scalability for IoT-based systems,
- AIBLOCK guarantees the immutability of the data collected from users and uses Blockchain to ensure data integrity,
- ML models are evaluated using accuracy, recall, precision and AUC parameters, and disease detection is performed with high performance by using the model with the highest accuracy in AIBLOCK.
- Google Cloud Platform (GCP) is used to evaluate the performance of AIBLOCK in Serverless computing environment and demonstrate that AIBLOCK gives better performance in terms of Quality of Service (QoS) parameters such as throughput and Average Response Rate (ARR).
- Experimental results show that AIBLOCK offers dynamic scalability by responding to increasing number of user requests.

The rest of the article is organized as follows. Section II shows the studies found in the literature. In Section III, the working mechanism of the proposed system and its background is explained. Performance measures of the proposed study are presented in Section IV. In Section V, the article is completed by giving the conclusion and future studies.

II. Related Works

Since the COVID-19 pandemic first appeared in December 2019, the health community has worked to reduce the spread of the disease and the number of fatal cases. Besides, capitalising on the developments in the field of AI, many studies have addressed the detection and tracking of COVID-19. Nasser et al. [14] proposed a study that detects COVID-19 from Deep Learning (DL) and computer tomography (CT) images by integrating IoT and cloud technologies. The study also makes it possible to monitor the condition of patients in real-time. The authors of [15] presented a new method to measure fever, which is a symptom of COVID-19. In addition, they aimed to solve the communication latency using Cloud in their studies. Mukherjee et al. [16] tried to detect COVID-19 by using an advanced k-Nearest Neighborhood (KNN) algorithm in their study. This algorithm has better performance than the normal c algorithm. In addition, the authors tried to improve the accuracy in detecting COVID-19 using Ant Colony Optimization (ACO). Otoom et al. [17] proposed an early detection framework for COVID-19 using eight different ML models. The study also monitors the treatment processes of patients who survived the disease, to understand the nature of COVID-19. In another study, the authors proposed a blockchain-based system to track COVID-19 infected patients globally [18]. Accordingly, while protecting the data security and privacy of the patients, it aimed to keep the spread of the disease under control by following the patients.Golec et al. [6] suggested a Covid-19 detection study using IoT and ML in the framework called iFaasBus. In the study, they used a serverless platform for the processing power and storage required for the IoT. The study, it is aimed to protect the security and privacy of users by using OAuth 2.0 and Transport Layer Security (TLS) protocols. To the best of our knowledge, there is no solution existing in the literature which deals simultaneously with scalability and data integrity for IoT-based COVID-19 detection. Table I shows the comparison of AIBLOCK with existing works based on important parameters, which clearly shows the novelty and superiority of this work

TABLE I
Comparison of AIBLOCK with existing works. $\times$:= method does not support the property

Work	COVID Detection	Data Collection	Scalability	Data Integrity
Nasser et al. [14]	DL	IoT	$\times$	$\times$
Hasan et al.[15]	$\times$	IoT	$\times$	$\times$
Mukherjee et al.[16]	ML	IoT	$\times$	$\times$
Otoom et al.[17]	ML	IoT	$\times$	$\times$
Rimsan et al.[18]	$\times$	$\times$	$\times$	Blockchain
Golec et al.[6]	ML	IoT	✓	$\times$
AIBLOCK	ML	IoT	✓	Blockchain

III. AIBLOCK: Proposed Technique

In this section, we provide the background information about the hardware and software components used to design AIBLOCK. Further, we provide the details of the dataset and discuss the working mechanism of AIBLOCK. Finally, we explain the performance metrics to evaluate ML models.

A. Hardware Instruments & Software Components

Hardware Components used in the AIBLOCK Framework: IoT layer, Gateway Node and Serverless Data Center Layer. Software Components: Gateway Node and Serverless Node.

- **IoT Layer:** The IoT layer forms the first layer of the framework. IoT devices can be used to receive and transmit data from sensors via communication protocols such as Zigbee, Bluetooth, WiFi, and Bluetooth Low Energy (BLE). In our study, we assume that the user data is taken from the sensors using the Raspberry Pi-4, an IoT device [19]. This data is sent to the server over the secure communication channel using the TLS protocol.

- **Hardware Gateway Node:** It is the mid-point between the IoT layer and the serverless platform. The data received by the sensor of the IoT layer are sent to the gateway node, then the gateway node sends the data to the serverless Datacenter (GCP - Cloud Functions). AIBLOCK also uses the gateway node as a user interface so that the user can communicate with the application to obtain the output, i.e., the predicted result by AI and the current block's hash.
- **Serverless Datacenter Layer:** In AIBLOCK, the serverless datacenter can be regarded as the most important part of the framework, as in this part every computational work is carried out. Data from the IoT is processed in this layer by a previously deployed ML model to find the COVID-19 detection result. In addition, blockchain transactions for data integration are also performed here. Blockchain is responsible for data integrity, it is present as a serverless platform, which creates a block with input data and the predicted AI output. The final hash formed with data and previous block has is sent to gateway for validation. In our study, GCP - Cloud Functions with serverless computing architecture was used [20].
- **Software Gateway Node:** It is responsible for transmitting the raw data to the serverless platform and getting the results after the calculation from the serverless platform. A mobile application or a local computer can be considered a gateway node. In AIBLOCK, the node consists of an API that receives data from the IoT and returns the result. In the study, an API was written using the Python - Flask framework [21].
- **Serverless Node:** When the serverless receives a request from the gateway node, it collects the data from the gateway node and uses the ML model to generate the COVID-19 results. Using blockchain in Serverless Node, the results from the ML model and the data used are recorded in blocks. These blocks are stored at both the server database and the user gateway to ensure data integrity.

B. Dataset

The COVID-19 dataset published by WHO was used in the study [22]. There are 20 health-related data (i.e., symptoms) and one variable showing the COVID-19 infection status in the dataset. After computing the correlation matrix, data analysis, and Active Shape Models (ASM), feature selection methods are used before the ML model is established. The 10 variables with the highest correlation with the COVID-19 infection status were used. These variables are: "Dry Cough", "Breathing Problem", "Fever", "Abroad Travel", "Sore Throat", "Attended Large Gathering", "Contact with COVID Patient", "Visited Public Exposed Places", "Headache", and "Gastrointestinal".

10 variables in the dataset used in Figure 1 and their statistical information are shown. Since the dataset is a dataset consisting of categorical variables consisting of only 1s and 0s, the mean values of the data are in the same decimal base. Therefore, we do not need to do normalization and standardization. In addition, the standard deviation values of the variables are close to each other and there is no big difference in terms of mean values. The max and min values are equal for all variables.

	ABROAD TRAVEL	SORE THROAT	DRY COUGH	BREATHING PROBLEM	ATTENDED LARGE GATHERING	CONTACT WITH COVID PATIENT	FEVER	VISITED PUBLIC EXPOSED PLACES	HEADACHE	GASTROINTESTINAL
count	5434.000000	5434.000000	5434.00000	5434.000000	5434.000000	5434.000000	5434.0000	5434.000000	5434.000000	5434.000000
mean	0.451049	0.727457	0.792602	0.666176	0.461907	0.501656	0.786345	0.518955	0.503497	0.469452
std	0.497644	0.445309	0.405480	0.471621	0.498593	0.500043	0.409924	0.499687	0.500034	0.499112
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	0.000000	0.000000	1.000000	0.000000	0.000000	0.000000	1.000000	0.000000	0.000000	0.000000
50%	0.000000	1.000000	1.000000	1.000000	0.000000	1.000000	1.000000	1.000000	1.000000	0.000000
75%	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
max	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000

Fig. 1. The Statistical Description of the Dataset

C. AIBLOCK Mechanism

In AIBLOCK, the data related to the 10 previously mentioned variables is first obtained from the user via sensors. The corresponding data is then sent to the IoT. This data is passed to the Gateway Node and then the data is forwarded to the serverless node. In the serverless node, two different jobs are performed. One is to determine whether users have COVID-19 through the ML model that has been deployed before. The other is to process the result indicating the COVID-19 status, together with the ten health variables from the user, into blockchain. The purpose of doing this is to prevent an unwanted external intervention against the COVID-19 detection status and health data by providing data integrity with blockchain. The COVID-19 detection status and the current hash of the generated blockchain are sent to the gateway node, which also acts as a user interface and displays the output to the user. This hash is saved in the server database, and allow data verification by comparing it to the hash data of the user gateway. If any attempt at changing the data takes place, the hash values in the blockchain will also change and the data held in the two databases will not match. In this way, data integrity will be guaranteed. Figure 2 shows the main diagram of the proposed operation AIBLOCK.

In Algorithm 1, the PseudoCode showing the operation mechanism of the AIBLOCK operation is given. Here, the health data (Hi) is received by the sensors as input. And as Output, 'Match' and 'Non-Match' results, which are the result of data integrity control, are returned. Initially, health data is sent to the IoT and collected there ($\sum$). Health data collected in IoT is sent to the Cloud via a reliable communication channel. The ML model previously deployed in the Cloud uses this data to return the COVID-19 result (Δ). A hash value is obtained by processing the COVID-19 result with health data in the blockchain module. The obtained hash value and the COVID-19 result are sent to both Cloud DB and IoT. Hash values kept in these two different environments are

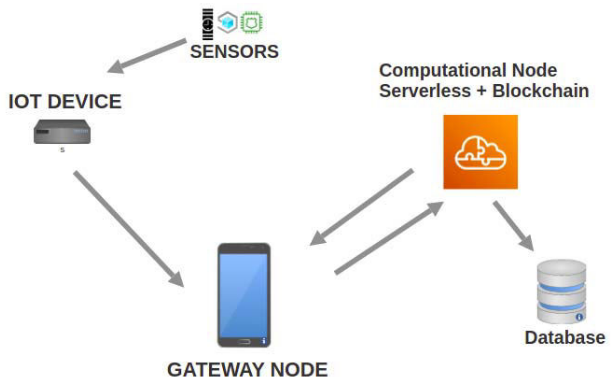

Fig. 2. Architecture of AIBLOCK

compared and in case of any inconsistency, the non-match result is returned. In this way, both data integrity is ensured and the user is informed by showing the COVID-19 result over the IoT. As can be seen from the pseudocode, time complexity is O(1) since there is no loop in the AIBLOCK mechanism and there is only an if-else condition.

Algorithm 1 The AIBLOCK Operating Mechanism

Input: $\mathbf{H_i}$
Output: **Match** or **Non-Match**

Begin
  Send $\mathbf{H_i} \longrightarrow$ IoT $\sum_0^i \mathrm{H_i} \longrightarrow$ Cloud **ML**($\sum_0^i \mathbf{H_i}$)
    return Δ
  Send Δ + Hash(Δ , $\mathbf{H_i}$) $\longrightarrow$ IoT and Cloud DB
  if Hash(Δ , $\mathbf{H_i}$) in IoT == Hash(Δ , $\mathbf{H_i}$) in DB**:**
      return Match
    else:
      return Non-Match
End

D. Performance Metrics

We evaluate the performance of our system from three different perspectives: Performance of ML Models, Server Performance, and blockchain Load.

Performance of ML Models: We use Accuracy, Precision, Recall, F-Score, and Area Under Curve (AUC) to evaluate the performance of the five different ML models used in COVID-19 detection.

Server Performance: We calculate the throughput and the average response rate (ARR) against the number of concurrent request (NCR) to measure the performance of the Serverless and Non-Serverless platforms. We used Google Cloud Functions[20] as the Serverless platform and Heroku[23] as the Non-Serverless Platform. We deployed AIBLOCK separately on these two platforms to measure performance and sent concurrent request with The Apache-Jmeter.

Throughput (Per / Second): It is the average of the successful delivery of messages passing over the communication channel. Usually measured in data packets per second (P/s) or bits per second (bit/s) [24].

The Average Response Rate (ARR) (milisecond): It is the average of the response time given by the server to the requests sent by the client, [25]. ARR is desired to be small in time sensitive IoT applications.

The Cold Start Latency: Serverless computing scales idle resources to zero, thus working on a pay-as-you-go basis. There is a certain latency in reallocating these resources, which are scaling to zero for functions. This latency is called cold start latency and can be problematic for time-sensitive IoT applications [26].

Blockchain Load: We calculate Average Memory (Byte) and Peak Memory (Byte) values to measure the additional load that the blockchain used in the study brings to the system on the server where it is installed.

IV. Performance Evaluation

In this section, we evaluate the performance of the proposed system, AIBLOCK, from three different perspectives. First, we will find the model with the highest accuracy rate among the five different ML models. We only need to find the ML model with highest accuracy one since only one ML model is used in the system. Later, we will deploy AIBLOCK in two different environments to show the superiority of the Serverless platform proposed in the system over the Non-Serverless platform. Finally, we will run the system without blockchain to measure the additional load that blockchain brings to the system.

Performance of ML Models: In our study, five different ML models, Logistic Regression(LR), Support Vector Machine (SVM), K Nearest Neighbour(KNN), Decision Tree (DT), Random Forest(RF) [27], are tested for the detection of COVID-19. These ML models and evaluation metrics are described in the previous section. The dataset used in the study is categorical but not numerical. Therefore, the values in the dataset are converted to 1-0 values using the "LabelEncoder" technique. Correlation values between independent variables and dependent variables were evaluated. In addition, the correlation values of the independent variables with each other are checked and one of the independent variables with a correlation value greater than 0.66 are removed from the dataset ("Sanitization from Market" and "Wearing Masks" were discarded). The independent variables are subjected to the Active Shape Models (ASM) feature selection method and their order of importance is determined. And the first 10 variables in order of importance are determined. To set up the ML model, we divided the dataset into train and test by 0.20. The value of "random_state" was taken as 0. In Table III, the Accuracy, Precision, Recall, F-Score, and AUC values of these five ML models are given as percentages, respectively. Accordingly, the model with the highest Accuracy is the RF model with 98.40%, and the model with the lowest Accuracy is the LR with 96%. Therefore, it is sufficient to deploy the RF model to the server for COVID-19 detection. In this way, COVID-19 patients are detected using RF, the nearest health institution is notified, and the spread of the disease is tried

TABLE II
ML HYPERPARAMETER

ML Models	HyperParameters
LR	Solver = saga, penalty = none, C = 1.0
KNN	Neighbour = 5, metric = euclidean, weights = distance
SVM	C =1.0,kernel = polynomial
DT	max_feature = sqrt, max_depth = none, splitter = best, criterion = gini
RF	criterion = gini, n_estimators: 15, max_features = auto

TABLE III
COMPARISON OF ML PERFORMANCES (%).

Models	Accuracy	Precision	Recall	F-Score	AUC
LR	96	91	86	89	91
KNN	97.70	95	92	93	95
SVM	96.44	99	79	88	94
DT	98	98	98	98	96.2
RF	98.40	89	89	91	96.40

to be reduced. Python software language was used for ML models [28]. Hyperparameter optimization was applied to all models to find the ML models with the highest performance. The optimum hyperparameter values found according to this process are given in Table II.

Serverless Performance: Within the framework of the AIBLOCK, we use a cloud infrastructure that automatically manages resources and includes serverless computing with scalability. In this way, we examine the parameters that take into account customer satisfaction, such as QoS, in response to the increasing number of users. Therefore, by testing our proposed work in a Serverless and a Non-Serverless platform, we compare the performance of the two platforms and show the superiority of Serverless. In another comparison, we examine ML performances on Serverless Cloud. ML speeds are important for places where reaction time is critical, such as mining operations and autonomous vehicle braking systems. In Figure 3, we deployed our ML model to GCP-Functions as a Serverless platform and deployed it to Heroku as a Non-Serverless platform. The configuration settings for the GCP-Functions used are Europe-west2 region, Python 3.7 runtime environment, and 512 MB allocated memory, respectively. We used Heroku to deploy AIBLOCK on a Non-Serverless platform. For this, we first uploaded ML method to a repository on Github. Next, we linked from the Heroku server to Github. Configuration settings for the Heroku server are as follows: Region "Europe", Framework "Python", Slug size "500 MiB". To measure the performance of the two platforms, we measured the number of concurrent requests (NCR) versus the throughput.

We created post requests using J-Meter in order to send the number of concurrent requests (NCR) to the platforms. And we gradually increased NCR. NCR is for modelling the number of concurrent users. Therefore, figure 3 shows how much throughput the two platforms are able to respond to these users as the number of users using the system increases. As can be seen from the figure, the Serverless Platform has much higher throughput.

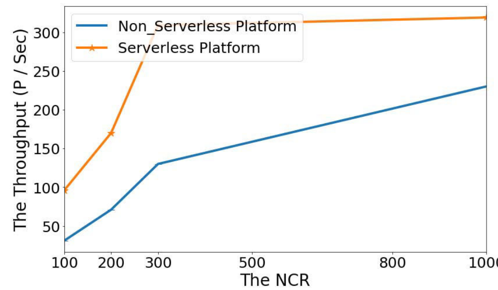

Fig. 3. Comparison of Throughput for Platforms

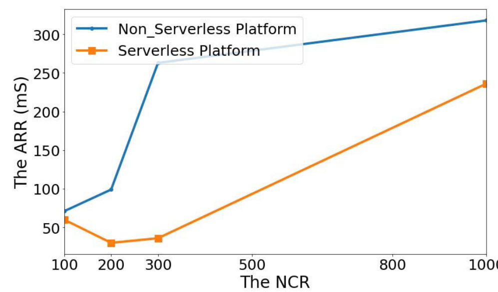

Fig. 4. Comparison of ARR for Platforms

To make another performance comparison, we measured the server's average response rate (ARR) versus NCR. Our aim in doing this is to measure the performance of two different platforms for latency-sensitive IoT applications such as autonomous vehicles, simultaneous patient monitoring. Figure 4 shows the ARR of the platforms versus gradually increasing NCR. As can be seen from the figure, the Serverless Platform can respond to requests sent by the client much faster thanks to its dynamic scalability feature. Another thing to note is that 100 NCRs has a higher response rate than 200 NCRs on the Serverless platform. The reason for this is the cold start problem seen in serverless platforms due to its zero scaling feature [26]. Cold start latency has been measured to be approximately 5 ms.

Blockchain Payload: Within the framework of AIBLOCK, we use blockchain in the cloud to ensure data integrity and an ML model in the cloud for COVID-19 detection. We measure the additional load that these two methods bring to the system. To measure this, ML and blockchain modules were deployed locally on a personal computer. The memory consumed by the system in RAM was measured when only the blockchain module was running, only the ML module was running, and both modules were running. Table IV shows the experimental results of memory usage. As it can be understood from the results, when designing systems using blockchain, the amount of memory that will be needed beforehand should be taken into account.

TABLE IV
MEMORY USAGE.

Statue	Average Memory (Byte)	Peak Memory (Byte)
Blockchain	9551	67276
ML	3162	13450
ML + Blockchain	10236	68945

V. CONCLUSIONS AND FUTURE WORK

This article proposes a new framework for IoT called AIBLOCK, which is resource-limited and inherently difficult to provide data integrity, that includes serverless computing and blockchain paradigm. We test AIBLOCK by applying a study that detects COVID-19 using ML. Accordingly, the data received from the patients through the sensors are collected in the IoT and sent to the cloud via a secure communication channel. The ML model, previously deployed in the cloud, processes health data and returns a result indicating the COVID-19 status. This result and the health data from the IoT are processed together in the blockchain module and a copy of the resulting hash is sent to the IoT and cloud database. In case of an intervention on health data or COVID-19 results, hashes will be changed and external interventions on health data will be deterred by ensuring data integrity. Since the proposed work is deployed to a server with serverless computing, resources are automatically scaled if needed. In the study, five different ML models were tested and the model with the highest accuracy was deployed to the cloud. The study was deployed on serverless and non-serverless platforms, performance tests were made separately and the superiority of serverless computing was demonstrated. Finally, we evaluated the serverless performance and the additional load brought by blockchain. The performance results showed that the AIBLOCK works effectively in applications where IoT is used.

A. Future Directions

In this work, only conventional ML models have been considered for performance optimization. In future, more compute intensive ML models such as Long short-term memory (LSTM) and transfer learning model can be used to achieve hyper-parameter optimization. In the future, a mobile application can be designed for AIBLOCK to collect information from the patient automatically from the wearable sensors. AIBLOCK can be used for other disease diagnoses such as diabetes, heart disease or cancer. Further, AIBLOCK can be used for other IoT applications such as agriculture, traffic management, weather forecasting or smart city. Also, by applying cold start latency reducing techniques inherent in the serverless computing paradigm, this framework will give much better performance for time-sensitive IoT applications.

ACKNOWLEDGEMENTS

Muhammed Golec would express his thanks to the Ministry of Education of the Turkish Republic, for their support and funding. We gratefully acknowledge Google for their support of this research through their GCP research credit program.

REFERENCES

[1] L. Yang, S. Liu, J. Liu, Z. Zhang, X. Wan, B. Huang, Y. Chen, and Y. Zhang, “Covid-19: immunopathogenesis and immunotherapeutics,” *Signal transduction and targeted therapy*, vol. 5, no. 1, pp. 1–8, 2020.

[2] “Coronavirus disease (covid-19).” [Online]. Available: https://www.who.int/emergencies/diseases/novel-coronavirus-2019

[3] S. Chen, H. Xu, D. Liu, B. Hu, and H. Wang, “A vision of iot: Applications, challenges, and opportunities with china perspective,” *IEEE Internet of Things journal*, vol. 1, no. 4, pp. 349–359, 2014.

[4] Y. K. Teoh, S. S. Gill, and A. K. Parlikad, “Iot and fog computing based predictive maintenance model for effective asset management in industry 4.0 using machine learning,” *IEEE Internet of Things Journal*, 2021.

[5] S. S. Gill *et al.*, “Ai for next generation computing: Emerging trends and future directions,” *Internet of Things*, vol. 19, p. 100514, 2022.

[6] M. Golec, R. Ozturac, Z. Pooranian, S. S. Gill, and R. Buyya, “ifaasbus: A security and privacy based lightweight framework for serverless computing using iot and machine learning,” *IEEE Transactions on Industrial Informatics*, 2021.

[7] B. Liu, X. L. Yu, S. Chen, X. Xu, and L. Zhu, “Blockchain based data integrity service framework for iot data,” in *2017 IEEE International Conference on Web Services (ICWS)*. IEEE, 2017, pp. 468–475.

[8] P. Wei, D. Wang, Y. Zhao, S. K. S. Tyagi, and N. Kumar, “Blockchain data-based cloud data integrity protection mechanism,” *Future Generation Computer Systems*, vol. 102, pp. 902–911, 2020.

[9] P. Tasatanattakool and C. Techapanupreeda, “Blockchain: Challenges and applications,” in *2018 International Conference on Information Networking (ICOIN)*, 2018, pp. 473–475.

[10] L. S. Vailshery, “Number of connected devices worldwide 2030,” Jan 2021. [Online]. Available: https://www.statista.com/statistics/802690/worldwide-connected-devices-by-access-technology/

[11] N. E. Nwogbaga, R. Latip, L. S. Affendey, and A. R. A. Rahiman, “Investigation into the effect of data reduction in offloadable task for distributed iot-fog-cloud computing,” *Journal of Cloud Computing*, vol. 10, no. 1, pp. 1–12, 2021.

[12] T. Lynn, P. Rosati, A. Lejeune, and V. Emeakaroha, “A preliminary review of enterprise serverless cloud computing (function-as-a-service) platforms,” in *2017 IEEE International Conference on Cloud Computing Technology and Science (CloudCom)*. IEEE, 2017, pp. 162–169.

[13] M. S. Aslanpour, A. N. Toosi, C. Cicconetti, B. Javadi, P. Sbarski, D. Taibi, M. Assuncao, S. S. Gill, R. Gaire, and S. Dustdar, “Serverless edge computing: vision and challenges,” in *2021 Australasian Computer Science Week Multiconference*, 2021, pp. 1–10.

[14] N. Nasser, Q. Emad-ul Haq, M. Imran, A. Ali, I. Razzak,

and A. Al-Helali, “A smart healthcare framework for detection and monitoring of covid-19 using iot and cloud computing,” *Neural Computing and Applications*, pp. 1–15, 2021.

[15] M. W. Hasan, “Covid-19 fever symptom detection based on iot cloud,” *International Journal of Electrical and Computer Engineering*, vol. 11, no. 2, p. 1823, 2021.

[16] R. Mukherjee, A. Kundu, I. Mukherjee, D. Gupta, P. Tiwari, A. Khanna, and M. Shorfuzzaman, “Iot-cloud based healthcare model for covid-19 detection: an enhanced k-nearest neighbour classifier based approach,” *Computing*, pp. 1–21, 2021.

[17] M. Otoom, N. Otoum, M. A. Alzubaidi, Y. Etoom, and R. Banihani, “An iot-based framework for early identification and monitoring of covid-19 cases,” *Biomedical signal processing and control*, vol. 62, p. 102149, 2020.

[18] M. Rimsan, A. K. Mahmood, M. Umair, and F. Hassan, “Covid-19: A novel framework to globally track coronavirus infected patients using blockchain,” in *2020 International Conference on Computational Intelligence (ICCI)*. IEEE, 2020, pp. 70–74.

[19] M. Siekkinen, M. Hiienkari, J. K. Nurminen, and J. Nieminen, “How low energy is bluetooth low energy? comparative measurements with zigbee/802.15. 4,” in *IEEE wireless communications and networking conference workshops*. IEEE, 2012, pp. 232–237.

[20] “Cloud functions nbsp;—nbsp; google cloud.” [Online]. Available: https://cloud.google.com/functions

[21] “Flask.” [Online]. Available: https://palletsprojects.com/p/flask/

[22] H. Hari, “Symptoms and covid presence,” Aug 2020. [Online]. Available: https://www.kaggle.com/hemanthhari/symptoms-and-covid-presence

[23] “Heroku,” 2021, [Online; accessed 23-October-2021]. [Online]. Available: https://www.heroku.com/

[24] Y. Xiao and J. Rosdahl, “Throughput and delay limits of ieee 802.11,” *IEEE Communications Letters*, vol. 6, no. 8, pp. 355–357, 2002.

[25] K. Xiong and H. Perros, “Service performance and analysis in cloud computing,” in *2009 Congress on Services-I*. IEEE, 2009, pp. 693–700.

[26] P. Vahidinia, B. Farahani, and F. S. Aliee, “Cold start in serverless computing: Current trends and mitigation strategies,” in *2020 International Conference on Omni-layer Intelligent Systems (COINS)*. IEEE, 2020, pp. 1–7.

[27] A. Singh, N. Thakur, and A. Sharma, “A review of supervised machine learning algorithms,” in *2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom)*. Ieee, 2016, pp. 1310–1315.

[28] “Welcome to python.org.” [Online]. Available: https://www.python.org/