iFaaSBus: A Security- and Privacy-Based Lightweight Framework for Serverless Computing Using IoT and Machine Learning

Muhammed Golec, Ridvan Ozturac, Zahra Pooranian, *Member, IEEE*, Sukhpal Singh Gill, and Rajkumar Buyya, *Fellow, IEEE*

***Abstract*—As data of COVID-19 patients is increasing, the new framework is required to secure the data collected from various Internet of Things (IoT) devices and predict the trend of disease to reduce its spreading. This article proposes security- and privacy-based lightweight framework called iFaaSBus, which uses the concept of IoT, machine learning (ML), and function as a service (FaaS) or serverless computing to diagnose the COVID-19 disease and manages resources automatically to enable dynamic scalability. iFaaSBus offers OAuth-2.0 Authorization protocol-based privacy and JSON Web Token & Transport Layer Socket protocol-based security to secure the patient's health data. iFaaSBus outperforms response time compared to nonserverless computing while responding to up to 1100 concurrent requests. Further, the performance of various ML models is evaluated based on accuracy, precision, recall, F-score, and area under the curve (AUC) values, and the K-nearest neighbor model gives the highest accuracy rate of 97.51%.**

***Index Terms*—Artificial intelligence (AI), function as a service (FaaS), Internet of Things (IoT), machine learning (ML), security and privacy, serverless computing.**

I. Introduction

COVID-19 is a disease that is thought to have started in Wuhan, China and that have been declared a pandemic by WHO [1]. COVID-19 caused 169 597 415 confirmed cases and 3 530 582 deaths worldwide till today (30th May 2021) [1]. Studies to find vaccines have been ongoing since December 2019, when the epidemic began, and in February 2021, a total of 66 vaccine studies are ongoing [2]. The trials of four vaccine companies have now ended, and they are effective up to 95%. As can be understood, it takes time to find a cure for the pandemic. If the rate of spread of a pandemic can be controlled for as long as the time required for the treatment to be found and applied, the increase in mortality can be prevented, and the global economic and sociological effects of the pandemic can be reduced. Here, Internet of Things (IoT) applications come into play with their wide range of use and ease of use. Data from sensors and medical devices can be transmitted to servers via an Internet connection [3] and can be made sense with machine learning (ML) models. In this way, people suspected of COVID-19 are quickly identified, the necessary authorities are informed, and the spread of the disease to more people can be prevented by isolation [4]. On the other hand, there are challenges such as user privacy, information security, scalability, and dynamic network topology issues that need to be overcome to diagnose and monitor pandemics with patient health data using IoT [5]. If the patient's personal information and health data fall into the wrong hands, the user privacy problem arises. Governments may also want to permanently use the proposed IoT-based studies to control people in the postpandemic period [6]. On the other hand, if secure communication channels are not used, individuals' health data can be changed or fall into the wrong hands. Finally, the system's scalability should be considered for future systems to meet the demands, considering the number of users and economic concerns [7].

1) Motivation and Our Contributions: This article presents a security- and privacy-based serverless lightweight framework called *iFaaSBus*. This study assumes that health data such as "Dry Cough," "Breathing Problem," and "Fever" are received from users and sent to the cloud via the IoT device using wearable devices and sensors. In addition, periodic data such as "Abroad Travel," "Sore Throat," "Attended Large Gathering," "Contact with COVID Patient," "Visited Public Exposed Places," "Headache," and "Gastrointestinal" are received via the mobile application and sent to the cloud. Using the transport layer security (TLS) and JSON Web Token (JWT) protocols in communication channels guarantees users' data security. In

Manuscript received March 15, 2021; revised June 2, 2021 and June 28, 2021; accepted July 2, 2021. Date of publication July 7, 2021; date of current version February 2, 2022. Paper no. TII-21-1218. *(Corresponding author: Muhammed Golec.)*

Muhammed Golec is with the Electrical and Electronics Engineering Department, Bursa Uludag University, Bursa 16370, Turkey, and also with Sisecam Company, 34947 İstanbul, Turkey (e-mail: ml181223@qmul.ac.uk).

Ridvan Ozturac is with Hepsiburada, 34387 Istanbul, Turkey (e-mail: ridvan.ozturac@hepsiburada.com).

Zahra Pooranian is with 5G & 6GIC, University of Surrey, GU2 7XH Guildford, U.K. (e-mail: zahra.pooranian@gmail.com).

Sukhpal Singh Gill is with the School of Electronic Engineering and Computer Science, Queen Mary University of London, E1 4NS London, U.K. (e-mail: s.s.gill@qmul.ac.uk).

Rajkumar Buyya is with the Cloud Computing and Distributed Systems (CLOUDS) Laboratory, School of Computing and Information Systems, The University of Melbourne, Melbourne, VIC 3010, Australia (e-mail: rbuyya@unimelb.edu.au).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TII.2021.3095466.

Digital Object Identifier 10.1109/TII.2021.3095466

addition, the OAuth 2.0 protocol was used to ensure user privacy. Therefore, it was ensured that only authorized health personnel could access the information of users who were diagnosed with the disease. In our study, K-nearest neighbor (KNN), support vector machine (SVM), artificial neural network (ANN) ML algorithms that have high-performance rates in the previous study [4] were used for the detection of COVID-19. In addition, in a study aimed at predicting the demand for health services by associating the development of cardiovascular diseases with environmental factors, linear regression (LR) and light gradient boosting machine (LightGBM) models were added to our study because the accuracy rates were satisfactory [8]. The models created with these five ML algorithms were compared by calculating accuracy, precision, recall, F-score, and AUC metrics. According to the results, KNN is the ML model with the highest accuracy with 97.51%. In addition, the operating speed of these five different ML models was compared, as the proposed study can be used for scenarios such as heart attack detection where the response time is very critical. The fastest running model was determined to be LR with a response time of about 0.2 ms against 1000 concurrent requests. The proposed system can be used not only for COVID-19 detection but also where user privacy and information security are critical, such as heart disease detection [9], chronic kidney disease detection [10], and dental disease detection [11]. With the increase in the number of users using the system, problems such as latency and response time may occur depending on the number of requests and processing density on the server. Serverless computing or function as a service (FaaS) has been used to solve these problems by scaling the system automatically with the increase in the number of users [7]. Thus, as the number of users using the system increases, the increasing number of requests can be processed by the server. In order to achieve this, our artificial intelligence (AI) server was deployed on a platform using serverless computing, and experiments were conducted to test it. No studies have been performed in the literature to identify COVID-19 using ML, which considers user privacy, personal health data security, and system scalability. The main contributions of this article are as follows:

- ensuring the privacy and data security of users;
- providing scalability using a serverless architecture;
- using ML models for disease diagnosis;
- keeping the disease's spread and the number of fatal cases under control by identifying suspicious cases in possible pandemics; and
- providing security and reliability for IoT communication networks using JWT, TLS, and OAuth-2.0 Authorization protocols.

2) Lightweight Testbed: iFaaSBus was created to build a small-scale testbed for conducting research in security and privacy for serverless computing. Future researchers can simulate their policies using iFaaSBus to test/validate their approach before implementing it on a real serverless cloud.

The rest of this article is organized as follows. Section II discusses related work, Section III presents the methodology, Section IV presents the performance evaluation. Finally, Section V concludes this article.

II. Related Work

When we look at the literature, various studies diagnose COVID-19 with some symptoms using IoT technology. Karmore *et al.* [12] proposed a medical diagnosis humanoid (MDH), which takes parameters such as X-ray images and blood values through the sensors and can diagnose the disease using an ML algorithm in their study. In the proposed system, the humanoid robot can go to the specified targets and sterilize itself. In Nenad and Dorde's study, a system is recommended as a precaution against COVID-19 for working environments such as offices [13]. According to this system, the body temperature is taken from the people, it is checked whether they wear a mask or not, and it is determined whether they comply with social distance. Users who are determined not to meet one of these three situations cannot enter the workplace. Otoom *et al.* [4] proposed a method for obtaining real-time patient health data through wearable sensors and performing COVID-19 detection with eight different ML models. Ahmed *et al.* [14] proposed a system that COVID-19 is detected by combining two different deep learning (DL) algorithms from chest X-ray images taken via sensors. The results are still being sent to a doctor for specialist approval. In the study suggested by Vedaei *et al.* [6], IoT, mobile application, and fog layer are used. IoT layer collects health data and a mobile app controls social distance. Moreover, the fog layer collects environmental information to minimize exposure to COVID-19. In the proposed framework, users are diagnosed with Covid with health data, and environmental risk factors are displayed on the application screen with the mobile application. In [15], a system for the business environment is proposed. The person's temperature is measured with the infrared sensor connected to the IoT device, and if it is above a certain threshold, the competent authority will be notified. Also, it is determined whether the employees are wearing masks with the camera in the system. An application called Arogya Setu, which is used in India to access people's health history in the system, is accessible via a QR code. In this way, it is checked whether the patient is in contact or not. To the best of our knowledge, no study has considered data privacy and security along with scalability in a single framework to detect COVID-19 disease in the literature. Table I shows the comparison of our study with existing works, which did not consider dynamic scalability, information security, and user privacy.

A. COVID-19 Contact Tracking Applications

Today, many governments and healthcare organizations use contact tracking applications to control the spread of COVID-19 disease. These practices have brought some concerns, such as the privacy and security of the person, which has been discussed in the literature [16]. Most COVID-19 contact tracking applications are monitoring the spread of the disease with GPS, the literature reported [17], [18]. Therefore, COVID-19 applications are still an area that needs to be investigated regarding user privacy and security. Some applications are thought to provide privacy but have security and privacy vulnerabilities, including using unsafe cryptographic algorithms (e.g., Message-Digest

TABLE I
COMPARISONS OF IFAASBUS WITH EXISTING STUDIES

Study	COVID-19 Symptoms	COVID-19 Detection	Dynamic Scalability	Information Security	User Privacy
Otoom et al. [4]	Fever, Cough, Fatigue, Sore Throat, Shortness of Breath	ML	○	○	○
Nenad and Dorde [13]	Fever	○	○	○	○
Karmore et al. [12]	Chest X-rays, Fever	ML + DL	○	○	○
Ahmet et al. [14]	Chest X-rays	DL	○	○	○
Vedaei et al. [6]	Age, Gender, Cough Rate, Shortness of Breath,Temperature, SpO2	ML	○	○	○
Baskaran et al. [15]	Temperature	ML	○	○	○
iFaaSBus (this paper)	Breathing Problems, Dry Cough, Fever, Gastrointestinal, Abroad Travel, Sore Throat, Attended the Large Gathering, Contact with Covid Patient, Headache, Visited Public Exposed Places	ML	●	●	●

○, Method does not support the property; ●, method supports the property.

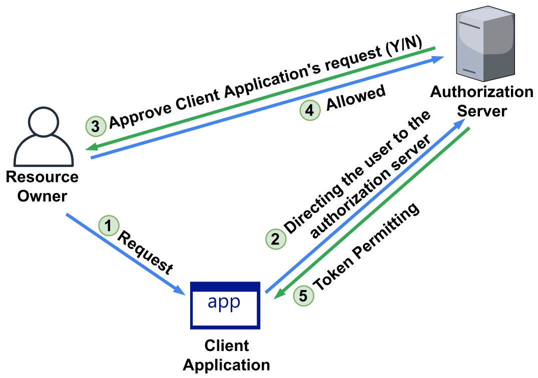

3 Approve Client Application's request (Y/N)
4 Allowed
Authorization Server
Resource Owner
2 Directing the user to the authorization server
1 Request
5 Token Permitting
app
Client Application

Fig. 1. OAuth 2.0 diagram.

Algorithm 5 and Secure Hash Algorithm) and storing user information in plain text [19].

III. METHODOLOGY

This section discusses security and privacy protocols, the working mechanism of the iFaaSBus framework, and performance evaluation metrics.

A. Security and Privacy Protocols

This section discusses OAuth 2.0 and TLS protocols, which are used for user privacy and information security.

1) OAuth-2.0 Authorization Protocol: The OAuth2 protocol is an authorization protocol designed to allow users to access their information on a service provider by third-party websites [20]. Fig. 1 shows the diagram of OAuth2. The working logic of OAuth2 is as follows: i) The resource owner (user) sends a request to the client application that he/she wants to use. ii) The client program asks the user to obtain a JWT from the authorization server. It automatically redirects the user to the authorization server. iii) The authorization server asks the user if he/she wants to approve the client application request. When the user approves this request, the authorization server returns a token to the client application and allows the license server to use its authorized resources.

2) JSON Web Token Structure: JWT is a request for comments (RFC) standard for user authentication and information security. It consists of three main parts: header, payload, and signature. The header section contains information about the token type and the algorithm used. There are claims such as sub (subject) and iat (Issued at) in the payload section. These claims contain information such as token validity period and token owner. Since user privacy is taken into account in our study, this section contains only an id representing the user. The signature part is created by encrypting the header and payload part with a private key with the header's algorithm. More details about JWT are given in [21].

3) Transport Layer Socket (TLS) Protocol: It is a protocol discovered by Netscape where X.509 certificates are used, allowing data to be transmitted securely between server–client using cryptographic methods [22]. The literature [23] reports that the use of the TLS protocol with JWT does not affect system performance. Accordingly, the working steps of the TLS handshake are as follows: i) The client section of the Secure Sockets Layer (SSL) version sends its session data and password settings to communicate with the server. ii) The server part sends its SSL version, session data, password setting, and certificate to the client. iii) The client checks the certificate's validity from the server over the certificate authority. If the certificate is invalid, the handshake operation will fail. If the certificate is valid, the process continues with the next step. iv) The client part sends its secret key to the server by encrypting it with the public key obtained from the server's certificate in the second step. v) On the server-side, this encrypted data received from the client is deciphered with the server's private key, and the key obtained is now used as a session key to be used in secure communication between client-server. vi) The client part tells the server that the handshake process is over with a message encrypted with the session key. Likewise, the server part transmits that the handshake part belongs to it finished with the encrypted message. Communication between client and server is now done securely using this private key (the session key).

B. Recommended System

In this section, the security- and privacy-based serverless (iFaaSBus) framework is explained, and its general working scheme is shown in Fig. 2. Accordingly, we assume that information on "dry cough," "breathing problem," and "fever" was obtained from wearable devices and sensors connected to IoT devices. Furthermore, "abroad travel," "sore throat," "attended large gathering," "contact with COVID patient," "visited public exposed places," "headache," and "gastrointestinal" periodic

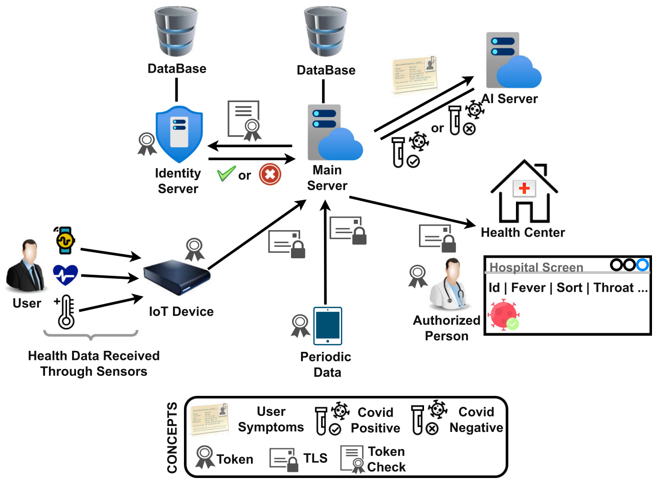
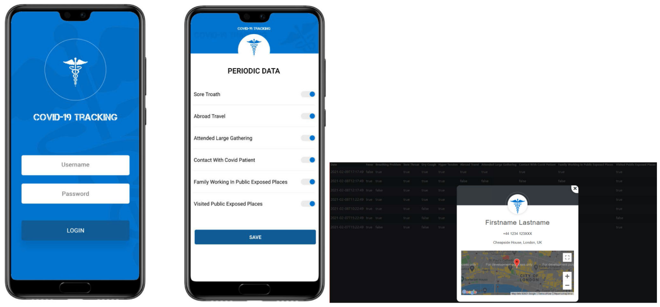


Fig. 2. iFaaSBus framework.



Date	Fever	Breathing Problem	Sore Throat	Dry Cough	Hyper Tension	Abroad Travel	Attended Large Gathering	Contact With Covid Patient	Family Working In Public Exposed Places	Visited Public Exposed Places
2021-02-09T17:17:49	false	true	true	true	true	true	true	true	true	true
2021-02-08T12:17:49	true	true	true	true	true	false	false	false	false	true
2021-02-08T12:17:49	true	false	true	true	true					true
2021-02-08T11:22:49	true	true	true	true	true					true
2021-02-08T10:22:49	true	true	true	false	true					true
2021-02-07T15:22:49	true	true	true	false	true					false
2021-02-07T15:22:49	true	false	true	false	true					true

(a) (b)

Fig. 4. GUI of iFaaSBus. (a) Mobile application screen. (b) Hospital panel.

data are also entered periodically once a day through a mobile application. There is an identity server in our software architecture, the main server where health data is collected and recorded, an AI server where ML is diagnosed, and a mobile application that users periodically enter their data. Finally, a hospital screen for expert control and emergency intervention is available. In the following, we examine these sections.

1) Collection and Uploading of Data: Two types of data are used in our system. The first is health data, including "dry cough," "breathing problem," and "fever" data. Other data types are periodic ones such as "abroad travel," "sore throat," "attended large gathering," "contact with COVID patient," "visited public exposed places," "headache," and "gastrointestinal". It is assumed that health data is received from users with wearable devices and sensors connected to IoT devices and sent to the main server. Each IoT device is given a token containing the user's Id by the identity server. Periodic data is sent to the main server via a mobile application as shown in the Fig. 4. Users first log in to the identity server with the mobile application. The identity server gives each user a token containing id information. This token is also the same as the token assigned to users' IoT devices. Users send their periodic data to the main server and the tokens defined for them via this mobile application. Users should enter their periodic data every day and once a day. Fig. 4(a) shows the mobile application interface. In our study, a dataset created from data containing COVID-19 symptoms published by the WHO is used [24].

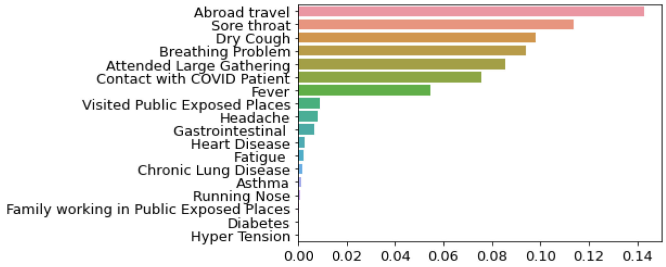


Fig. 3. Order of importance according to the correlation values of the variables.

2) Identity Server: The identity server issues a token containing the user Id to users who register with the mobile app and the IoT device. During the study, each user's identifying information is used instead of personal information, such as the name and address of all users. Only authorized persons (e.g. doctor, medical officers) can access information such as the name and home address of users suspected of having the disease with a special password given to the server authorities to ensure user privacy.

3) Main Server: The users' health data and periodic data are sent to the main server along with the tokens received from the identity server over the communication channel secured by TLS. Before the main server stores this data in the database, it checks whether the user is registered on the identity server. If the identity server approves the user, periodic data and health data are recorded in database (DB) and sent to AI server. In case of detection of COVID-19 with an ML model on the AI server, feedback is sent to main server, and user information is sent to medical server from the main server. In our study, identity server, main server, and medical server are created using NetCore software language; and the AI server is created using Python software language and a Microservice structure. MicroServices are services that work together that communicate with lightweight mechanisms that have advantages such as technology variety, flexibility, and scaling [25]. We used event-driven architecture to communicate between these services in Microservices. Event-based programming generally means developing software to react to external influences (events) [26]. Event-driven architecture has advantages such as improving the performance by increasing the system's scalability as there will be asynchronous communication between the services. In Algorithm 1, the working principle of the main server is explained with a pseudo code. Event-driven architecture is used in the main server. In the first step, health data from IoT devices are sent to the main server over the secure TLS channel, and the token received from the identity server. The main server first prevents unauthorized persons from entering the system by verifying the incoming token "CheckToken(token)" on the identity server. If it is detected that the token is registered on the identity server (authentication) and the relevant user

Algorithm 1: Main Server Event-Driven Architecture.

```
Input: CreateHealthDataCommand and token
Output: Success/Unsuccess
Begin
  CreateHealth-
  Data(token,CreateHealthDataCommand):
    identity_result = CheckToken(token)
    if identity_result == true:
      Push CreateHealthDataCommand
    else:
      return UnAuthorize
  Handler CreateHealthDataCommandHandler:
    Insert the HealthData data to DB
    Push HealthDataCreatedEvent
  Handler HealthDataCreatedEventHandler:
    combined_data = Push GetCombinedDataQuery
    ai_result =
    CheckDataByAIServer(combined_data)
    if ai_result == true:
      Push CovidCaseDetectedEvent
  Handler CovidCaseDetectedEventHandler:
    combined_data = Push GetCombinedDataQuery
    SendMedicalServer (combined_data)
  Handler GetCombinedDataQueryHandler:
    return combined_data from DB
End
```

has the authorization for the required operation (authorization), the "CreateGealthDataCommand" command is pushed. In the handler of this command, incoming health data is first recorded in the DB. The processes so far are repeated in periodic data received from the mobile application in the same way. Then, the "HealthDataCreatedEvent" event is pushed. With this event, the data containing current periodic and health data are assigned to the variable "combined_data" by pushing the "GetCombinedDataQuery" event. This received data is sent to the AI Server "CheckDataByAIServer(combined_data)" and if COVID-19 is detected on the AI Server, "CovidCaseDetectedEvent" event is pushed. The patient's information is sent to the medical server through this event. No operation in Algorithm 1 depends on the user request and resources in the system, making it a constant time method. Hence, the overall complexity of the algorithm is O (1).

4) AI Server: The user's Id number and health data are sent through a secure communication channel from the main server to the AI server, and the COVID-19 detection is done by giving it to the ML algorithm. If COVID-19 is detected in the user, the result is reported to the main server, and the necessary actions are taken. Serverless architecture is used in our AI, identity, and core servers. Serverless cloud computing architecture is a type of cloud computing that is scaled by the provider company and is based on pay-as-you-go logic [7]. The application used by the provider company is provided, and at the same time, software developers do not need to deal with servers, hardware, and software management. In this way, by ensuring the scalability of the system we recommend, it can provide services to many users simultaneously, and it offers a much more economical service with the logic of pay-as-you-go.

5) Hospital Screen: The hospital panel contains a page that immediately displays the ID number and symptoms of a suspected COVID-19 patient. Only authorized people can access the hospital panel, and they will be redirected to an identity server that uses the OAuth 2.0 protocol. Authorities who want to enter the hospital panel first log into the identity server with their user name and password. The identity server returns a token that contains the information that they are allowed to authorized persons. In this way, the patient's personal information, whose Id appears on the hospital panel, can only be seen by the authorities (e.g. doctor, medical officers). Fig. 4(b) shows the hospital panel.

C. Machine Learning Models

In this article, ten data received from the main server were trained with five different ML algorithms, and COVID-19 patients were diagnosed. The following ML models have been used for prediction.

1) Logistic Regression (LR): LR is a statistical model for building the relationship between the dependent and independent variables to find a binary variable type (Yes–No, 1–0). It is used in classification problems. It calculates the parameters so that the sample values' observation probability is high instead of minimizing the sum of square root errors in normal regression problems.

2) K-Nearest Neighbors (KNN): KNN is one of the most used and simple types of classification algorithms. It includes the "lazy learning" algorithm that memorizes training data instead of learning. A value of K is determined for our algorithm. The K value represents the nearest K neighbor's result to look for when finding a result for a value. Euclid, Manhattan, Hamming, and Minkowski functions can be used to find the closest K value. After calculating the KNN, the desired value is assigned to the most appropriate class.

3) Support Vector Machine (SVM): SVM is a supervised learning method that draws a line between points to separate the plane points. Adding ± 1 to the drawn line creates a field called the margin, and the success of the model is directly proportional to the width of the margin. The wider the margin, the clearer the distinction between the two classes.

4) Artificial Neural Network (ANN): ANN is an ML method created by imitating the human brain's learning structure. Human neuron cells connect to form networks. Thanks to these networks, people acquire abilities such as observation, thinking, and learning. The bond between neurons is called synaptic, and these bonds are adjusted as they learn, and new bonds are formed if necessary. In this way, learning is carried out in the human brain. In the ANN algorithm, the coefficients (weights) between neurons are adjusted according to the mathematical model, and the learning process is performed.

5) Light Gradient Boosting Machine (LightGBM): It is an alternative to the gradient boosting algorithm, and it is aimed to obtain faster training and higher accuracy. LightGBM, a

TABLE II
FEATURE SELECTION METHODS' PERFORMANCE

Feature Selection Method	ML Model	Accuracy (%)
Pearson Correlation Coefficient	LightGBM	97.24
Chi-Squared Test	KNN	97.51
Mutual Information	LightGBM	97.14

histogram-based algorithm, provides ease of operation by converting continuous variables into discrete values. Two factors that determine the education time in decision trees are the number of calculations and the number of divisions. The leaf-oriented strategy is used for learning in LightGBM. In this way, it learns faster and with less error rate.

D. Evaluation Metrics of Machine Learning Algorithms

We used five performance parameters such as accuracy, precision, recall, F-score, and ROC curve to evaluate the performance of ML models, and the details of evaluation metrics are given in [27].

IV. RESULTS AND DISCUSSION

In this section, we analyze the performance of various ML models for the diagnosis of COVID-19 diseases. Further, we compare the performance of the proposed framework for both serverless computing and nonserverless computing.

A. Performance of Machine Learning Algorithms

The dataset consists of 22 dependent variables and the "Covid-19" independent variable representing the COVID-19 status. According to the preliminary examination, all variables in the dataset consist of "YES" and "NO" categorical values. "YES" represents the condition that the mentioned variable is present in the patient, and "NO" represents the condition that the mentioned variable is not present in the patient. There is no missing or meaningless data in the dataset. Since all states are represented in two ways as "YES" and "NO" only, they are converted to numeric values 1 for "YES" and 0 for "NO" using "LabelEncoder." Using the most unrelated variables in the dataset in an ML model causes the ML model to be unable to define the relationships between variables and causes an overfitting problem in the model [28]. Using three different feature selection methods, Pearson correlation coefficient, Chi-squared test, and mutual information, ten variables with the highest correlation value were determined. Then, the accuracy obtained with five different ML models, namely, LR, KNN, SVM, ANN, and LightGBM, was determined. Table II shows the highest accuracy ratio for these three different feature selection methods and the ML models used to obtain this ratio. According to this result, the model using the "Chi-squared test" and KNN has the highest accuracy rate. We used the "Chi-squared test" as feature selection in our dataset. Fig. 3 shows the variables' correlation values found using the mutual information technique in the dataset. In Table III, the first ten, first eight, first six, and first four variables with the highest correlation value obtained with the "Chi-squared test" were found, and accuracy values were found for each of them using the KNN ML model. Accordingly, the highest accuracy value was obtained using the first 10 variables with the highest correlation value. Next, the correlation matrix was created to find the correlation relations of the selected 10 variables, and it was determined that the correlations of the variables were independent of each other.

TABLE III
CLASSIFICATIONS ON A DIFFERENT NUMBER OF FEATURES

Feature Number	A (%)	P (%)	R (%)	F (%)
10	97.51	99.65	97.31	98.47
8	96.50	98.30	97.43	97.86
6	95.58	98.17	96.42	97.29
4	92.27	97.42	93.07	95.20

A: accuracy; P: precision; R: recall; F: F-score.

TABLE IV
COMPARISON IN PERCENT (%) OF ML PERFORMANCES

Models	Accuracy	Precision	Recall	F-Score	AUC
LR	96.04	98.08	97.09	97.58	94.12
KNN	97.51	99.65	97.31	98.47	97.88
SVM	96.22	95.71	99.88	97.75	89.53
ANN	96.68	96.33	99.77	98.02	94.64
LightGBM	96.87	97.46	98.77	98.11	93.40

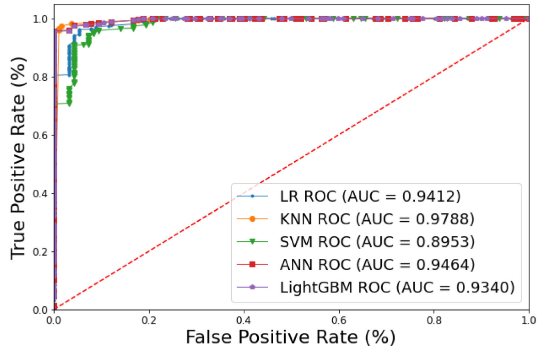


Fig. 5. Roc curves of ML algorithms.

The "GridSearchCV" function was used to find the most optimal hyperparameters for ML models. The hyperparameter values of the ML models used are as follows: **LR** ("C": 1.0, "solver": "liblinear"), **KNN** ("n_neighbors": 6), **SVM** ("C": 1, "kernel": "poly"), **ANN** ("activation": "relu," "alpha": 0.005, "hidden_layer_sizes": (100, 100), "solver": "adam"), **LightGBM** ("learning_rate": 0.1, "max_depth": 2, "n_estimators": 1000). The accuracy, precision, recall, F-score, and AUC values of the five different ML algorithms we have used are given in Table IV. Fig. 5 shows the ROC curves of the five different ML algorithms used in our study. The area under the ROC curve called the AUC information is also shown in Fig. 5. As can be seen, all five different algorithms have very high accuracy values. However, since only one algorithm in our model is sufficient, the KNN algorithm with the highest accuracy was preferred.

B. Performance of Clouds

The performance of the proposed work is tested on both serverless and nonserverless computing environments, which clearly shows the superiority of serverless computing. Then, five different ML models are deployed on the server using serverless

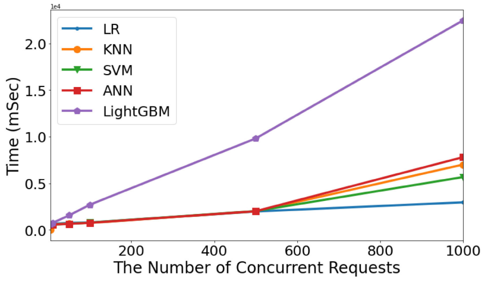


Fig. 6. Performance comparison of ML models on the serverless cloud.

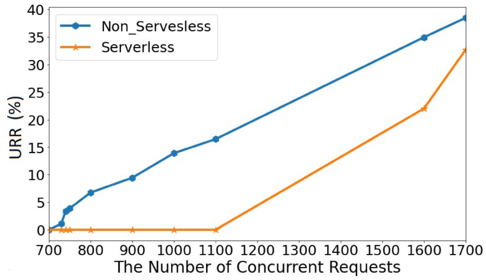


Fig. 7. Comparison of serverless and nonserverless clouds. URR, unsuccessful response rate.

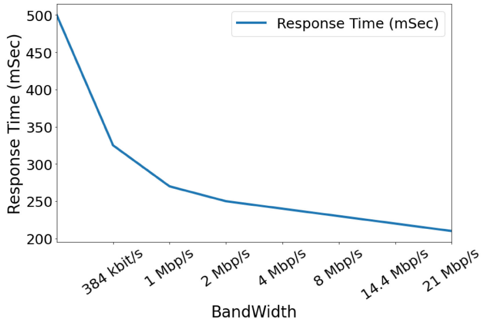


Fig. 8. Response time of the server according to the changing bandwidth.

computing to find out the fastest ML model for scenarios where response time is critical.

1) Serverless vs. Nonserverless Computing: In traditional cloud computing, it is challenging to enable dynamic scalability in over- and underutilization of resources due to the changing demand of users at runtime. Further, it leads to the wastage of money and user dissatisfaction. Because of all these needs, the concept of serverless computing was born. Thanks to its scalability feature, serverless computing automatically scales up the required processing power and memory in parallel with the increasing number of users. Heroku is used for the implementation of serverless computing [29]. For a nonserverless environment, an environment is created with a personal computer using i5-9300H CPU and 8 GB DDR4 RAM. The performance of both server and serverless clouds is evaluated by simultaneously submitting requests to calculate the unsuccessful response rate to these requests. Apache JMeter program was used in the measurements, and the results are shown in Fig. 7. As seen in the figure, after 700 simultaneous requests, i.e., 700 users, a nonserverless server cannot respond to the requests. However, in a cloud using serverless computing, this number goes up to 1100. Because a serverless cloud is scaled depending on the number of requests, it can successfully return to large numbers of simultaneous requests. This enables a large number of users to use the system simultaneously.

2) Execution Speed Comparison of ML Models on Cloud: Response time is critical in environments such as heart attack detection, traffic with autonomous vehicles, or closed-circuit television, where criminals are detected. Five different ML models are deployed to the cloud with serverless computing, and their response time is calculated depending on the number of concurrent requests, and experimental results are shown in Fig. 6. As the number of concurrent requests sent to the system, i.e. the number of users, increases, the response time also increases. Accordingly, the algorithm with the fastest response time on the cloud is LR, and the algorithm with the lowest response time is LightGBM. The performance has been evaluated on serverless computing with an Internet connection speed of 40 Mbit/s. As it is known, Internet speed can vary significantly in different continents, even in different regions of the same country, which can also impact the cloud's response time to a great extent. In Fig. 8, the response time performance of the cloud with serverless computing is calculated for the environment with seven different Internet bandwidth (BW) values. Experimental results show that the response time gets longer as the Internet BW is getting smaller.

C. Evaluation of Security and Privacy

In our study, TLS protocol and OAuth 2.0 protocol are used to secure health-related data. Health data and periodic data are sent via a communication channel using TLS to provide information security. In addition, JWTs are issued to registered devices (IoT + mobile devices) by identity server. In this way, the unauthorized device without JWT cannot send data to the system. The privacy of the users is provided with the OAuth 2.0 protocol. The identity server gives JWT to authorize the healthcare personnel to use the system. Healthcare personnel who want to access the disease-detected user information send a request to the identity server via these JWTs given to them. Identity server checks the JWTs and checks whether the health personnel who sent the request are registered in the system (authentication). Then, it checks whether it is authorized to receive user information (authorization). In this way, the user's information, such as name and home address, cannot be seen by anyone except authorized health personnel, and user privacy is ensured.

Limitations: Our study assumes that the users' health data are obtained from wearable devices without preprocessing. However, in real-life scenarios, this data may need to go through some preprocessing processes. In addition, all performance measurements were made in environments where the Internet connection is reliable. In environments where there is no Internet connection or a very weak Internet connection, the system's operation may be disrupted. Another limitation is that the Heroku service, which provides free service up to a certain number of users and processing power, is used in our system. Systems with many participants will require the use of much more professional companies such as Amazon Web Service and Google Cloud and Microsoft Azure. This can increase the cost of using the system. We used JWT for authentication, which is very secure [21], but there may be some risks, such as an unexpired JWT, that can be addressed using a signature [30].

V. Conclusion

In this article, a lightweight framework called iFaaSBus was proposed to detect COVID-19 disease using the concept of serverless computing. iFaaSBus processes the data coming from IoT devices and uses various ML models to predict the trend of COVID-19 patients. In the iFaaSBus framework, we used OAuth-2.0 protocol for users' privacy and JWT and TLS protocol to secure the user data. The concept of serverless computing is used to enable dynamic scalability for handling the changing number of requests with time. We used various ML models such as LR, KNN, SVM, ANN, and LightGBM and evaluated their performance in terms of accuracy, precision, recall, F-score, and AUC values. It has been identified that the LR model is the fastest, and the KNN model is the more accurate model with an accuracy rate of 97.51%. In future, iFaaSBus can be extended in the following ways. A mobile application can be developed for iFaaSBus to collect data from patients automatically using wearable sensors. Further, iFaaSBus can trace the location of the patients for future pandemics, which can help people maintain the required social distance to reduce the spread of disease. Furthermore, trust mechanisms such as Blockchain and biometric authentication can be incorporated within iFaaSBus to prevent future attacks such as denial of service. Moreover, iFaaSBus can be tested using other quality of services parameters such as energy, reliability, availability, and cost, and other issues such as platform migration and debugging. In this article, iFaaSBus has used tested for healthcare application with the COVID-19 domain, and it can be used for other healthcare domains such as heart disease, diabetes, cancer, and hepatitis. Finally, iFaaSBus can be generalized for other IoT applications such as agriculture, weather forecasting, traffic management, and smart city in the future.

Software Availability: iFaaSBus has been released as open-source software. The implementation code with experiment scripts and results can be found at the GitHub repository.[1]

References

[1] "WHO Coronavirus Disease (covid-19) Dashboard," 2021. [Online]. Available: https://covid19.who.int/

[2] "Covid-19 Vaccine Tracker," 2021. [Online]. Available: https://vac-lshtm.shinyapps.io/ncov_vaccine_landscape/

[3] M. Golec, S. S. Gill, R. Bahsoon, and O. Rana, "Biosec: A biometric authentication framework for secure and private communication among edge devices in IoT and industry 4.0," *IEEE Consum. Electron. Mag.*, to be published, doi: 10.1109/MCE.2020.3038040.

[4] 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," *Biomed. Signal Process. Control*, vol. 62, 2020, Art. no. 102149.

[5] M. Ndiaye, S. S. Oyewobi, A. M. Abu-Mahfouz, G. P. Hancke, A. M. Kurien, and K. Djouani, "IoT in the wake of Covid-19: A survey on contributions, challenges and evolution," *IEEE Access*, vol. 8, pp. 186821–186839, 2020.

[6] S. S. Vedaei *et al.*, "Covid-safe: An IoT-based system for automated health monitoring and surveillance in post-pandemic life," *IEEE Access*, vol. 8, pp. 188538–188551, 2020.

[7] M. S. Aslanpour *et al.*, "Serverless edge computing: Vision and challenges," in *Proc. Australasian Comput. Sci. Week Multiconf.,* 2021, pp. 1–10.

[8] H. Qiu, L. Luo, Z. Su, L. Zhou, L. Wang, and Y. Chen, "Machine learning approaches to predict peak demand days of cardiovascular admissions considering environmental exposure," *BMC Medical Info. Decision Making*, vol. 20, no. 1, pp. 1–11, 2020.

[9] K. Polat and S. Güneş, "A new feature selection method on classification of medical datasets: Kernel F-score feature selection," *Expert Syst. Appl.*, vol. 36, no. 7, pp. 10367–10373, 2009.

[10] A. Salekin and J. Stankovic, "Detection of chronic kidney disease and selecting important predictive attributes," in *Proc. IEEE Int. Conf. Healthcare Inf.*, 2016, pp. 262–270.

[11] G. Chitnis, V. Bhanushali, A. Ranade, T. Khadase, V. Pelagade, and J. Chavan, "A review of machine learning methodologies for dental disease detection," in *Proc. IEEE India Council Int. Subsections Conf.*, 2020, pp. 63–65.

[12] S. Karmore, R. Bodhe, F. Al-Turjman, R. L. Kumar, and S. Pillai, "IoT based humanoid software for identification and diagnosis of Covid-19 suspects," *IEEE Sensors J.*, to be published, doi: 10.1109/JSEN.2020.3030905.

[13] N. Petrovic and D. Kocić, "IoT-based system for COVID-19 indoor safety monitoring," *IcETRAN*, Belgrade, Serbia, 2020.

[14] I. Ahmed, A. Ahmad, and G. Jeon, "An IoT based deep learning framework for early assessment of Covid-19," *IEEE Int. Things J.*, to be published, doi: 10.1109/JIOT.2020.3034074.

[15] K. Baskaran, P. Baskaran, V. Rajaram, and N. Kumaratharan, "IoT based Covid preventive system for work environment," in *Proc. 4th Int. Conf. IoT Social, Mobile, Analytics Cloud*, 2020, pp. 65–71.

[16] H. R. Schmidtke, "Location-aware systems or location-based services: A survey with applications to Covid-19 contact tracking," *J. Reliable Intell. Environ.*, vol. 6, no. 4, pp. 191–214, 2020.

[17] B. S. L. Reichert and S. Brack, "Privacy-preserving contact tracing of Covid-19 patients," in *Proc. 41st IEEE Symp. Secur. Privacy.*, 2021, p. 375.

[18] C. Zhang, C. Xu, K. Sharif, and L. Zhu, "Privacy-preserving contact tracing in 5G-integrated and blockchain-based medical applications," *Comput. Standards Interfaces*, vol. 77, 2021, Art. no. 103520.

[19] R. Sun, W. Wang, M. Xue, G. Tyson, S. Camtepe, and D. Ranasinghe, "Vetting security and privacy of global COVID-19 contact tracing applications," 2020, *arXiv:2006.10933*.

[20] V. Sucasas, G. Mantas, A. Radwan, and J. Rodriguez, "An OAuth2-based protocol with strong user privacy preservation for smart city mobile e-health apps," in *Proc. IEEE Int. Conf. Commun.*, 2016, pp. 1–6.

[21] M. Jones, B. Campbell, and C. Mortimore, "JSON Web Token (JWT) profile for OAuth 2.0 client authentication and authorization grants," May 2015. [Online]. Available: https://tools. ietf. org/html/rfc7523

[22] W. Easttom, *Modern Cryptography: Applied Mathematics for Encryption and Information Security*. Salt Lake City, UT, USA: Springer Nature, 2021.

[23] L. Yan, X. Chen, H. Deng, and X. Ye, "A delegation token-based method to authenticate the third party in TLS," *Int. J. High Perform. Comput. Netw.*, vol. 13, no. 2, pp. 164–174, 2019.

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

[25] A. Krylovskiy, M. Jahn, and E. Patti, "Designing a smart city internet of things platform with microservice architecture," in *Proc. 3rd Int. Conf. Future Internet Things Cloud*, 2015, pp. 25–30.

[26] O. Etzion, "Towards an event-driven architecture: An infrastructure for event processing position paper," in *Rules and Rule Markup Languages for the Semantic Web*, A. Adi, S. Stoutenburg, and S. Tabet, Eds. Berlin, Germany: Springer, 2005, pp. 1–7.

[27] G. Abirami and R. Venkataraman, "Performance analysis of the dynamic trust model algorithm using the fuzzy inference system for access control," *Comput. Elect. Eng.*, vol. 92, 2021, Art. no. 107132.

[28] I. Jebli, F.-Z. Belouadha, M. I. Kabbaj, and A. Tilioua, "Prediction of solar energy guided by Pearson correlation using machine learning,*" Energy*, vol. 224, 2021, Art. no. 120109.

[29] "Heroku," 2021. Accessed Mar. 7, 2021. [Online]. Available: https://www.heroku.com/

[30] P. Solapurkar, "Building secure healthcare services using OAuth 2.0 and JSON web token in IoT cloud scenario," in *Proc. 2nd Int. Conf. Contemporary Comput. Inf.*, 2016, pp. 99–104.

[1] [Online]. Available: https://github.com/Muhammed1616/iFaaSBus