InFeMo: Flexible Big Data management through a federated Cloud system

CHRISTOS L. STERGIOU

Department of Applied Informatics University of Macedonia, Greece

KONSTANTINOS E. PSANNIS*

Department of Applied Informatics University of Macedonia, Greece

BRIJ B. GUPTA**

1 Department of Computer Engineering, National Institute of Technology, Kurukshetra, India

2 Department of Computer Science and Information Engineering, Asia University, Taiwan

This paper introduces and describes a novel architecture scenario based on Cloud Computing and count on the innovative model of Federated Learning. The proposed model named *Integrated Federated Model*, with acronym *InFeMo*. InFeMo incorporates all the existing Cloud models with a federated learning scenario, as well as other related technologies that may have integrated use with each other, offering a novel integrated scenario. In addition to this, proposed model is motivated to deliver a more energy efficient system architecture and environment for the users, which aims to the scope of data management. Also, by applying the InFeMo the user would have less waiting time in every procedure queue. Proposed system was built on the resources made available by Cloud Service Providers (CSPs), by using the PaaS (Platform as a Service) model, in order to be able to handle user requests better and faster. This research tries to fill a scientific gap in the field of federated Cloud systems. Thus, taking advantage of the existing scenarios of FedAvg and CO-OP, we keen to ended up to a new federated scenario that merges these two algorithms, and aiming to has a more efficient model, that it is able to select, depending on the occasion, if it "train" the model locally in client of globally in server.

* Corresponding author #1.

** Corresponding author #2.

http://dx.doi.org/10.1145/3426972

CCS CONCEPTS •

(1) Computing methodologies → *Modeling and simulation*; *Model development and analysis*;

(2) Information systems → *Data management systems*; *Database management system engines;*

(3) Security and privacy → *Database and storage security;*

Additional Keywords and Phrases: Cloud Computing 1, Federated Learning System 2, Management, Big Data 3, Secure 4

ACM Reference Format:

Christos L. Stergiou, Konstantinos E. Psannis, Brij B. Gupta. 2020. InFeMo: Flexible Big Data management through a federated Cloud system: ACM Transactions on Internet Technology, Special Issue "Deep Learning Algorithms and Systems for Enhancing Security in Cloud Services": This is the subtitle of the paper, this document both explains and embodies the submission format for authors using Word. NOTE: This block will be automatically generated when manuscripts are processed after acceptance.

1 INTRODUCTION

This work tries to study and integrate technologies in order to provide a more efficient environment for the academic users with the aim of managing the data. More specifically, through a system server assisted by Cloud providers, the academic user will have the opportunity to manage large-scale data, aka Big Data, from everywhere.

Big Data (BD) can be referred as a big thing in the field of modern technologies. As soon as we can decode the best use of Big Data we will have the opportunity to change the world completely by using all the extracted information of the data. To better understand the phenomenon of Big Data, we would have to find out the usage of their five major characteristics, which are widely known as five Vs of Big Data [1] [2]: 1) Volume, 2) Velocity, 3) Variety, 4) Veracity, 5) Value. Specifically, Volume of Big Data refers to the vast amounts of data which are generated every second, Velocity of Big Data refers to the speed at which the new data sets are generated and also the speed at which the data sets move around, Variety of Big Data refers to the various different types of data that can be used, Veracity of Big Data refers to the messiness or trustworthiness of the data, and Value of Big Data refers to the worth of the data which have being extracted.

Big Data management could be used with the aim to customize the consistency level. More generally, everyone can perform more relaxed consistency-based replication systems on top of particular database storage systems count on stricter transactional semantics. The customized replication and consistency enforcements could be considered a useful aspect of the applications, in which a number of updates might require higher integrity and some might require the higher scalability of relaxed consistency [3] [4].

Additionally, management will probably be the most difficult problem to address with big data. This is not a new problem in this field. It occurred years ago, where some scientists realized that data was distributed geographically and owned and managed by multiple entities [5] [6]. Analyzing the Data as Big Data and taking into account their features we can reach to some conclusions. Particularly, the richness of digital data representation forbids an unveiled methodology for data collection. Data specification often focuses more on the missing data than trying to attest every item. As regards the data volume, it is purposeless to attest every data item such as new approaches to data specification and ratification [5] [7].

Moreover, Cloud Computing (CC) additionally could be used as a base technology due to its type of services for other relative to the communication field technologies [4] [8]. Big Data is a relative technology of the field of communications that could rely on Cloud Computing. It is known from literature that BD refers to the description of the stunningly rise of data volume either of structured or unstructured form. In addition to this, the term BD describes a specific amount of data set [9] [10]. Therefore, the major problem arises not relies on the gain of large amounts of data, but whether these data have any value or not. Hopefully, by envisaging that the companies of IT field would be able to extract information from any source, also utilize the pertinent data and analyze the data aiming to get immediate answers, we will achieve reducing cost and time, producing new products and optimizing offerings, and more intelligent decisions making [7] [8].

In addition, Cloud Computing could be referred as an extremely successful example oriented IT services. Also, CC has brought a new revolution in the way in which the computing infrastructure used, and in addition it could be extended to Database as either Service or Storage. Moreover, CC could be an omnipresent example due to its characteristics by setting up innovative applications. These applications were not currently economically feasible on traditional businesses. Thus, through scalable DataBase Management Systems (DBMS), which is CC's infrastructure critical part, could be achieved an update on intensive application workloads, such as decision support systems [11].

Furthermore, due to its unique use of Cloud's environment, the providers and the customers of Cloud Computing are keen to share the responsibility for security and privacy in CC environments; with the limitation however of that the sharing levels will differ for different delivery models, which in turn affect the Cloud extensibility.

The following delivery data models are offered in the Cloud Computing environment [8] [12] [13]: 1) SaaS, 2) PaaS, 3) IaaS. These models provide relation to software, platform and the infrastructure as cloud services. Specifically, SaaS is the delivery model which could offer typically enabled services by providing a large number of integrated features, which could lead to less extensibility for the customers, PaaS is delivery model which aims to enable developers in order to build their own applications on top of the provided platforms, and IaaS is the delivery model which is the most extensible of the tree. In this delivery model, the Cloud providers must provide some basic, low-level data protection capabilities [12].

Moreover, a novel technique that offers new opportunities in order to manage and operate the data in cooperative environments makes its appearance. This novel technique is called federated learning. Thus, according to the literature of federated learning, the main objective of it is to train a model from data $\{X^t,...,X^K\}$ produced by K distributed clients. Every device represented as a client. Each client, $t \in [K]$, produces data in a Non-IID manner, which means the data distribution on client t, $X^t \sim P^t$, is not a uniform sample of the whole distribution [14]. Based on the literature, the federated learning technique bases on distributed machine learning to which a global model is learned by aggregating models that have been trained locally on data-generating clients [15]. Additionally, the algorithms of federated learning scenarios reckon with the fact that communication with edge devices occurs over unreliable networks with very limited upload speeds [15]. As a result, federated learning can significantly decrease the privacy and the security risks by limiting the attack surface only to the device, and not to both the device and the Cloud [16]. Particularly, Federated learning tries to give solutions to problems such as: 1) The distinct advantage on training on proxy data that is generally vacant in the data center could be provided by training on real world data delivered by mobile devices. 2) Trained data is better not to attain it to the data center wholly for the intention of model

training, due it is privacy sensitive or large in size. 3) User interaction could infer naturally labels on the data for supervised tasks [16].

The optimization problem of federated learning could be defined with an algorithm that optimizes the finite-sum objective:

$$\min_{w \in R^d} f(w) \text{ where } f(w) = \frac{1}{n} \sum_{i=1}^{n} f_i(w) \quad \textbf{(1)}$$

In equation (1) the w is a vector that contains d model parameters. In machine learning scenarios, we treat the function $f_i(w)$ as a loss function $f_i(w) = \ell(x_i, y_i; w)$, where an input-output pair of (x_i, y_i) is one of the n given labeled examples, in most times referred to as a *training examples*. One problem could be interpreted as finding the w which minimizes the average loss over all n training examples such as those in [15]. Moreover, another scenario is assuming that there are K clients over which the data is partitioned, with P_k the set of indexes of data points on client k, with $n_k = |P_k|$. This could lead us to produce a new equation from (1) [17]. Furthermore, regarding the Big Data context, which is the main technology apart of the Cloud Computing we focus in this work, we can state that the number of training examples is too large to be stored on one computer, and thus we need to distribute the computation to many computers. Concluding these we could have:

$$f(w) = \sum_{k=1}^{K} \frac{n_k}{n} F_k(w) \quad \text{where} \quad F_k(w) = \frac{1}{n_k} \sum_{i \in P_k} f_i(w) \quad \textbf{(2)}$$

From equation (1) and (2) we could have the federated learning parameters, such as: 1) the number of federated learning rounds, represented as T, 2) the total number of nodes used in the process, represented as K, 3) the fraction of nodes that used at each iteration for each node, represented as C, 4) the local batch size which used at each learning iteration, represented as B, 5) the number of iterations for local training before the pooling, represented as N, and 6) the local learning rate, represented as η. These parameters need to be optimized in accordance with the constraints of the machine learning applications. Also, in addition to the above parameters, we can describe the main strategies of federated learning with some notations, such as the K, which is the total number of clients, the k, which illustrates the index of clients, n_k, which refers to number of data samples usable during training for client k, $w_{k,t}$ which demonstrates the model's weight vector on client k, at the federated round t, $l(w,b)$, which illustrates the loss function for weights w and batch b, and E, which represents the number of local epochs.

Thereafter, there is also the Federated Stochastic Gradient Descent, or widely known as FedSGD of SGD, which is a deep learning training primarily count on variants of stochastic gradient descent, at which place gradients are computed on a random subset of the total dataset and afterwards used to make one step of the gradient descent. FedSGD which initially proposed by Shokri & Shmatikov [17] is the direct transposition of FedSGD algorithm to the federated setting. However, the gradients are averaged by the server proportionally to the number of training samples on each node, and used to make a gradient descent step.

To sum up, the presentation of the basic theoretical information of the research field of this work provides the main contributions of our work:

- ✓ Provide a more energy efficient system architecture and environment for the academic users with the aim to data management.
- ✓ Decrease number of rounds of communication needed to train a scenario model by using a federated Cloud system. Thus the users have to wait less.
- ✓ Try to fill a scientific gap in the field of federated cloud systems management.
- ✓ Proposes an innovative architecture model, InFeMo – Integrated Federated Model, that incorporates all Cloud models with a federated learning scenario, as well as other technologies that may have integrated use with each other, in a novel integrated scenario.

The rest of this work is organized as follows. Section 2 discusses the background research which has been made in the field of data management in Cloud environment and Federated Learning Systems. In addition to this, Section 3 gives details about the system formulation and its analysis. Specifically, Section 3 presents an analysis and an evaluation of our approach. Section 4 illustrates the scheme implementation of our work. Initially, it presents the Fundamental Procedure Scenarios, then the Model Explanation, and concluding with a comparative analysis with former works. On the other hand, Section 5 provides the algorithm approach of our work by presenting the algorithmic representation of our system. The experimental results count on a practical system is demonstrated in Section 6. Finally, Section 7 concludes this paper and the whole research work.

2 RELATED WORK REVIEW

2.1 Big Data Management in Cloud

During the last years, several works have been made in order to manage data, and more specifically Big Data, in Cloud environments. Thus, for the purpose of this research we have studied and analyzed previous literature researches that have been made in the field of data management in Cloud environment [18-23]. The following paragraphs present the previous to our study research papers.

To begin with, Thakur et al. [18] present a Robust reputation management mechanism, that tries to encourage the Cloud Providers (CPs) in a federated cloud to separate users between good and malicious, and grant resources in such a way that they do not share them.

Moreover, Cai et al. [19] present a novel in-memory data management system, called Memepic, that unifies both online data query and data analytics functionality, permitting low-latency storage service and efficient in-situ data analytics.

Another work in this field is presented by Pasquier et al. [20], which introduce Information Flow Control (IFC) model and describe and evaluate this IFC architecture and implementation (CamFlow) that compromises an OS level execution of IFC with support for application management, in cooperation with an IFC-enabled middleware.

To continue with, Zhu et al. [21] present a controllable blockchain data management (CBDM) model that can be deployed in a Cloud environment, which it can evaluate its security and performance, in order to demonstrate utility.

A heterogeneous data storage management scheme that flexibly provides simultaneously deduplication management and access control over innumerable CSPs is introduced by Yan et al. [22].

Finally, a trust based federated identity management as a Cloud based utility service is presented by Premarathne et al. [23]. Furthermore, this work proposes a Cloud-based utility service model for federated identity-based trust negotiations management and a novel trust based evaluation method to access the cooperativeness of the identity providers to improve the reliability.

2.2 Federated Learning Scenarios

On the other hand, there are a number of remarkable works associated with the novel scenarios of Federated Learning Systems [14-16] [24-28]. The following paragraphs present the relative research papers.

Initially, a general distributed multiquery processing problem motivated by the need to speedup data acquisition in federated databases using evolutionary algorithm presented by Mansha & Kamiran [24].

Furthermore, Yao et al. [14] through the experiments presented in their work show that baseline methods could outperformed by their proposed model, especially in Non-IID data distributions, and accomplishes a compression of more than 20% in required communication rounds.

In another related work, Wang et al. [25] propose an algorithm that adapts to real-time system dynamics, derived from theoretical analysis. Then, it defines that every specific iteration contains a local update step which is possibly followed by a global aggregation step.

Nilsson et al. [15] benchmarks three federated learning algorithms *(1) Federated Averaging (FedAvg)* [26], *(2) Federated Stochastic Variance Reduced Gradient (FSVRG)* [26], *(3) CO-OP* [27] and compare their performance opposed to a centralized technique in which data resides on the server.

Moreover, Young et al. [28] presents an approach to computing the covariance matrix with federated databases. Also, it computes the exact covariance matrix rather than an approximation.

Finally, McMahan et al. [16] advocate a different scenario that leaves the training data distributed on the mobile devices, and learns a shared model by aggregating locally-computed updates. Thus, this work introduces the Federated Averaging algorithm, which integrates local stochastic gradient descent (SGD) on each client with a server that performs model averaging.

3 SYSTEM FORMULATION & ANALYSIS

The first goal of this work is to introduce a flexible data management system that it is set up in an academic server and it operates with the useful help of the cooperative Cloud Providers. The system set up does not differ to any other structure of server set regarding the hardware. The main purpose is to be set a cooperative system, a federated scenario, in which the cooperative CSPs could “help” the load-balance of data management and transmission by having the load of user authentication.

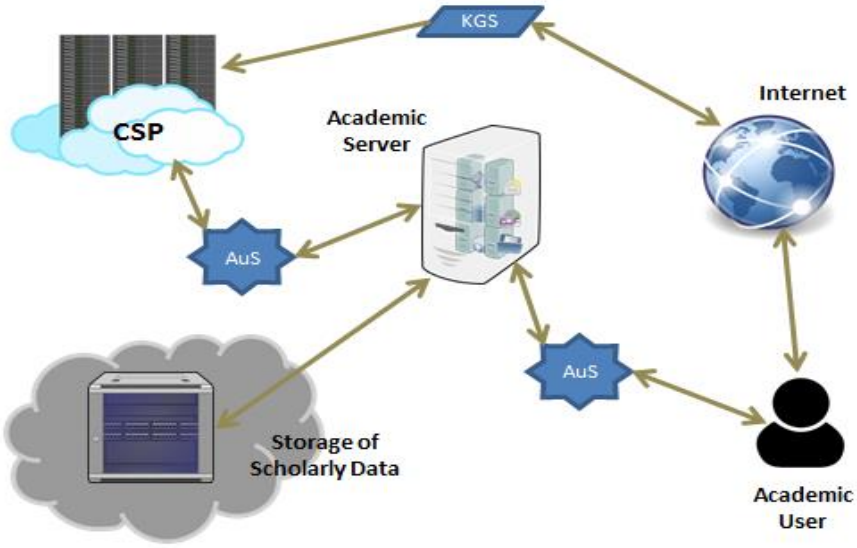


Figure 1: System architecture.

Figure 1 shows the operation of our proposed scenario implementation. More specifically, the academic user will be able to access academic data via a CSP (Cloud Service Provider) that is authorized to access the server of the academic institution. This will enable instantly and faster the data management through the benefits of the CSP Cloud platform.

The user authentication will be done in two parts. Initially, through the CSP with the KG (Key Generation System) that will give the unique key until the connection expires. Then, it will be authenticated through the academic server, where the user will be authenticated, as well as the level of permissions granted to the user.

There will be some states of rights: 1) Data owner, 2) Data researcher, 3) student. Depending on the property that results from the authentication process taking place on the CSP platform, the level of data management will be obtained.

3.1 Evaluation Approach Scenario

Depending on the system introduced above, we can clearly explain its function with figure 2. More specifically, we will try to explain the operation of the proposed system when multiple users try to have access to the Academic Server.

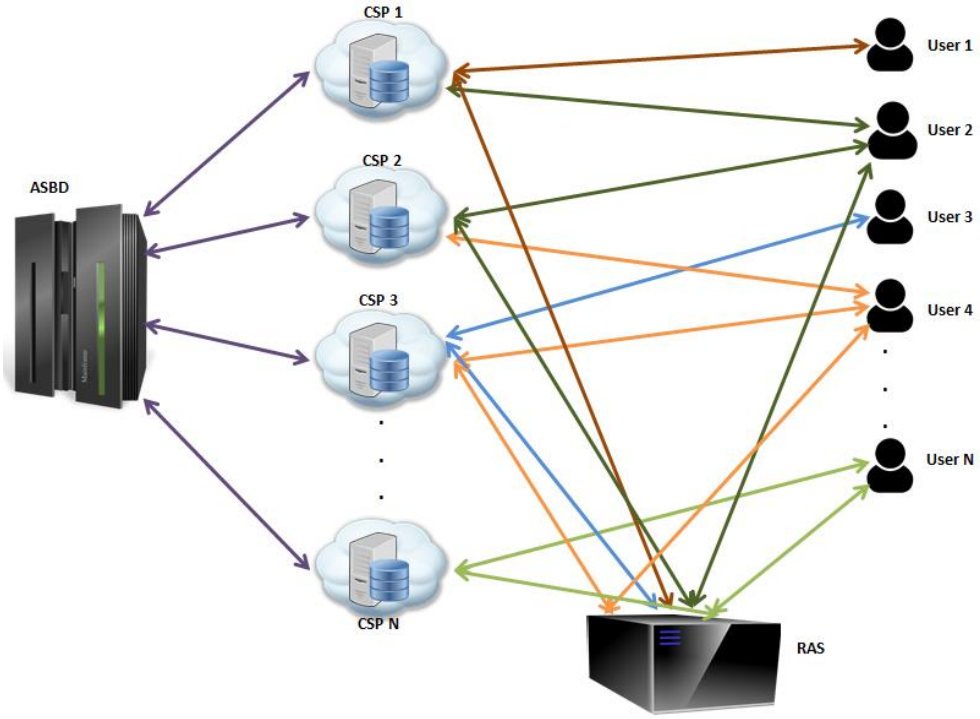


Figure 2: Academic Server evaluation and procedure.

Figure 2 show the operation and procedure of the Academic Server evaluation scenario. Accurately, the server follows the exact steps in its operation:

1) Connection request to CSP (1 or 2 or 3).
2) Control - User authentication in RAS (Reputation Authentication System)
3) Response of RAS through the open communication channel, directly to the user.
4) Depending on the user permissions level (1 or 2 or 3) the corresponding CSP data center will be used.
5) Communication of the CSP with the ASBD (Academic - Scholarly Big Data) repository to open a direct user communication channel.

The combination of 1 to 5 depends on the demands of each user and in the available Cloud provider at the moment of the user's request. Then, depending on the internet connection and the availability of the CP, the RAS authenticates the user and the level of user's permissions to the Academic Server. With this proposed system the user authentication takes place in the federated CSP environment and as a result the Academic Server is not burdened with this load. Thus, we can achieve an energy and computational efficient scenario for an academic server that will serve thousands of users, with different demands each one.

The whole proposed system set up is based on a federated learning system between the multiple CSPs, due to the general principle of federated learning systems. The general principle composed in training local models on local data samples and exchanging parameters between various local models at some frequency in order to generate a wide-spread model.

4 SCHEME IMPLEMENTATION

In this section we try to introduce with more details the operation of the proposed system and the implementation design of it. All the information pertaining to the system's functionality are revealed in the following subsections.

4.1 Fundamental Procedure Scenarios

In this subsection, we introduce a number of fundamental algorithms of the proposed system. The operation of the Fundamental Procedure Scenario took part on an academic server for academic users. The whole system illustrates how some major processes could be completed based on our proposed scenario.

4.1.1 Data management through access by different CSPs.

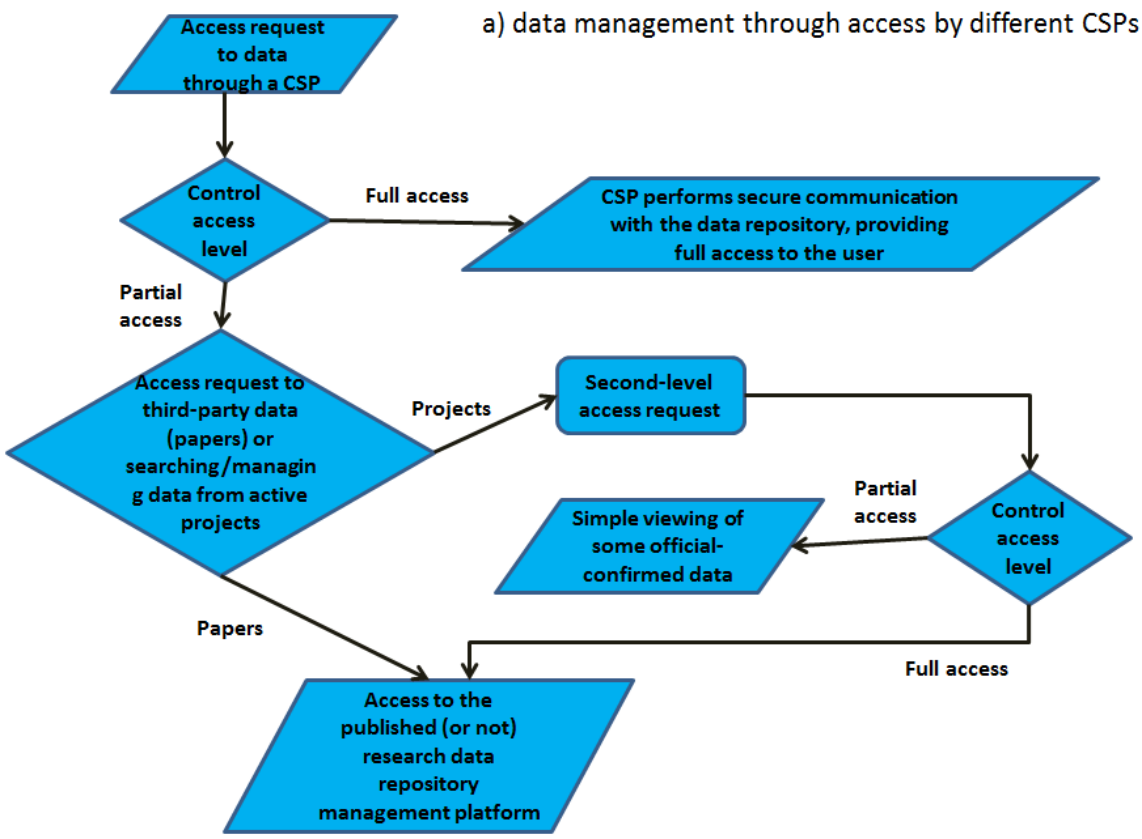


Figure 3: Data management through access by different CSPs.

Figure 3 shows the procedure of the management of data through the access by using multiple CSPs. Each step demonstrated in figure 3 is analyzed more clearly in the following steps, which depict how an academic user could request access to data and how this user could manage these data depending on the rights granted to the respective user. Particularly, the procedure follows the following steps:

Step 1 - The user makes a request for data access through a CSP.

Step 2 - A check is made on the CSP for the level of user access. There are two cases here, one is the user to have full access to the data, and the other is the user to have partial access to the data.

Step 3 - Full access - The CSP performs a secure communication channel with the data repository (academic server), providing full access to the user.

Step 3 - Partial access - A new request is sent by the user through the CSP for access to third-party data (scientific papers and works), or data from search / management related to participation in active projects. There are two cases here, one is related to the access to scientific papers and works, and the other is related to the access to data from finished and open academic projects.

Step 4 - Scientific papers & works - Access to the repository research platform of published research data located on the academic server.

Step 4 - Finished & open academic projects - A new second-level access request is made to authenticate the user and their rights.

Step 5 - A new second-level access level control for user rights is performed. There are two cases here, the one concern the full access of the user to the data from finished and open academic projects in the academic server, and the other concerns the partial access of the user to the data from finished and open academic projects in the academic server.

Step 6 - Full access - Provided by the academic server, through the CSP, full user access, and user access to the management platform.

Step 6 - Partial access - It is provided by the academic server, through the CSP, a simple view of certain, official-confirmed data to the user, from the management platform.

4.1.2 Deletion of data through access by different CSPs.

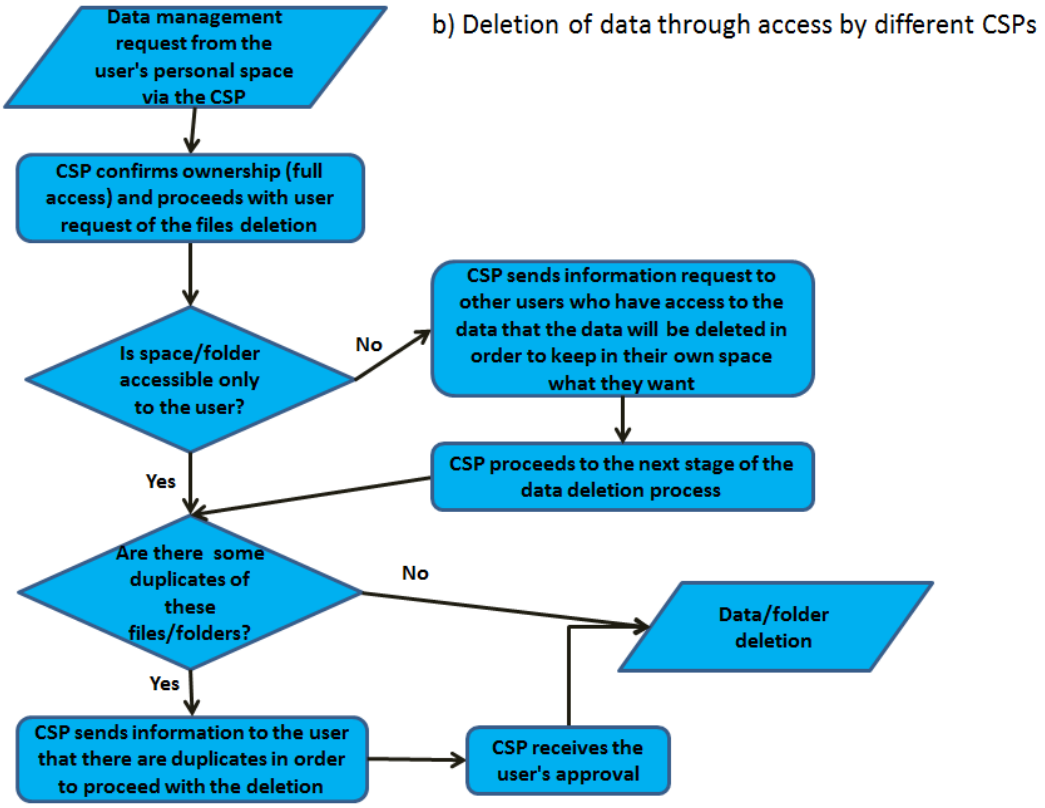


Figure 4: Deletion of data through access by different CSPs.

Figure 4 shows the procedure of deleting data on the academic server, through access from multiple CSPs. Each step demonstrated in figure 4 is analyzed more clearly in the following steps, which depict how an academic user could access data through the academic server and how this user could manage in order to delete these data depending on the rights granted to the respective user. Particularly, the procedure follows the following steps:

Step 1 - Request to delete data from the user in his/her personal space via the CSP.

Step 2 - The CSP confirms to the user that he has full access to and ownership of the content, and then deletes the data requested by the user.

Step 3 - User data rights control is performed. These are data/files only accessed by the user who requested the deletion or accessed by other users. There are two cases here, yes and no.

Step 4 - No - The CSP sends an update request to other users who have access to the data/files that the data/files will be deleted so that if they wish to keep copies in their own space on the system.

Step 5 - No - The CSP proceeds to the deletion of data/files. (*Then, the procedure follows the next step in a row: Step 4 - Yes*)

Step 4 - Yes - A data management system checks if there are any duplicates of the data/files that were requested to be deleted. Two cases arise in this step, yes and no.

Step 5 - No - Files are deleted from the file system.

Step 5 - Yes - The CSP sends the user information that there are duplicates elsewhere in the file system, and that it will delete them too.

Step 6 - After receiving approval from the user that it has been updated and accepts the duplicate deletion, it proceeds to delete the duplicate files from the system.

4.1.3 Adding data through access by different CSPs.

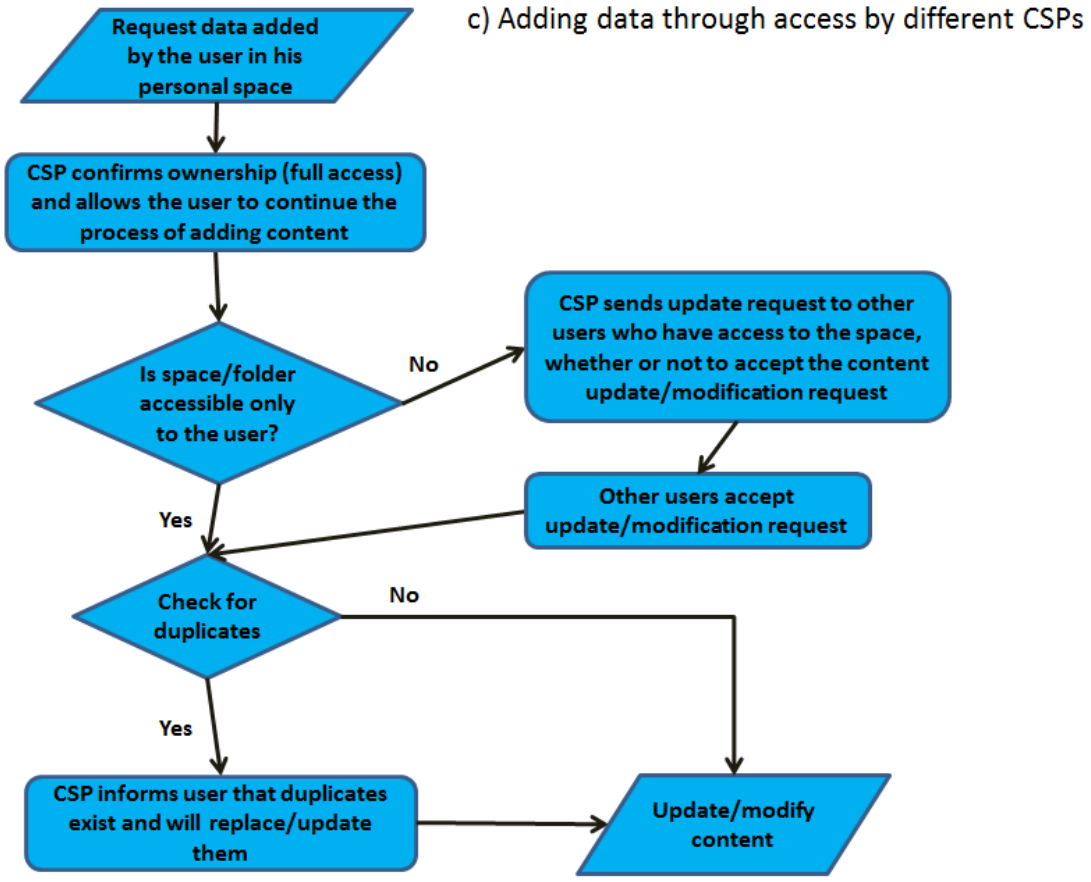


Figure 5: Adding data through access by different CSPs.

Figure 5 shows the procedure of adding data to the file system through access from multiple CSPs. Each step demonstrated in figure 5 is analyzed more clearly in the following steps, which depict how an academic user could access the owned disk space on the academic server and how this user could add an manage these data depending on the rights granted to the respective user. More particular, the procedure follows the following steps:

Step 1 - User request to add data to their personal space in the CSP system.

Step 2 - The CSP confirms authorized content ownership (full user access), and allows the content to be added by the authorized user.

Step 3 - The system checks whether the space the user is trying to modify/add content is a space/folder that is only accessible by the user or additional from someone else. Two cases arise in this step, yes and no.

Step 4 - No - The CSP sends an update (update request) to other users who have access to that space/folder for acceptance / approval / update access / content modification. (*Then, the procedure follows the next step in a row: Step 4 - Yes*)

Step 4 - Yes - Checking the file system, if any files already exist in the file system that the user wants to add (duplicate check). Two cases arise in this step, yes and no.

Step 5 - Yes - The CSP informs the user that duplicates exist and will replace/update existing ones with the new ones.

Step 6/Step 5 - No - Updating space/folder content.

4.2 Model Explanation

The operation of our proposed system is presented in this subsection. Generally, through the traffic that the central server receives from the requests of the various CSPs coming from the users, a system can be developed to support the communication of the CSPs with the academic server in order to follow the safest and fastest authentication methods through federated methods. The result of this process will be to allow the user to select the most appropriate CSP for the task the user wants to perform, and to develop a machine learning system through the communication of the CSPs with the academic server. This system will build on the resources made available by CSPs (using the PaaS - Platform as a Service) format to be able to handle user requests better and faster. This will develop a more efficient management system in a federated Cloud environment.

Thus, the user could have a more immediate communication with the academic server through the safe environment provided by the cooperative CSPs. Assuming the user is a client (*k*) that have contacted the server several times for a specific folder containing various type of data (n_k), so the cooperative CSPs could learn a scenario about this user in order to make the authentication procedure instantly and to navigate the user exactly to the most used files. This learning method could be established in the edge of communication of the each client and the collaborated CSPs. As a result, the academic server could reduce the computational ability for user services and focus on the most important, research process. Additionally, based on the 3 procedures presented in subsection 4.1, the system could also be applied to the training process taking place in the CSPs on federated learning concept. Thus, the users will be able to bypass some of the "easy" steps after a continuous and advanced use.

Moreover, the proposed Cloud model collaborates very well with the academic server and the various clients. The SaaS model assists in the storage of the amounts of data in the academic server through the assistance of the multiple collaborative CSPs. Also, the IaaS model is used due to the interface scenario that consist of the communication of the users with the academic server through the assistance of the multiple collaborative CSPs.

5 ALGORITHM APPROACH

The evaluation of our work can be additionally demonstrated through an algorithm analysis is inferred from the synchronous and asynchronous algorithms of federated learning scenarios. Studying the literature in the field of federated learning we ended up with two predominant algorithmic models, one of each category. The one is *Federated Averaging (FedAvg)* algorithm which is a synchronous algorithm and the other is *CO-OP* algorithm which is an asynchronous algorithm.

5.1 Federated Averaging (FedAvg) Algorithm

The Federated Averaging algorithm, or as briefly mentioned FedAvg, was initially introduced by A. Nilsson et al. [16]. This algorithm orchestrates training through a central server which hosts the shared global model w_t, where t is the communication round. Nevertheless, the actual optimization is done locally on clients using, for example, the Stochastic Gradient Decent (SGD). Moreover, FedAvg algorithm has five "*hyper-parameters*", directly related to the general federated learning parameters, previously mention in the Introduction Section.

The parameters B, E, η, and λ are commonly used when training with SGD [17]. However, in FedAvg algorithm the variable E stands for the total number of iterations through the local data *before* the global model is updated [15].

In its operation, Federated Averaging algorithm begins with randomly initializing the global model of w_0. Specifically, one communication round of FedAvg algorithm drives to the consisting of the following aspect: (*algorithm operation procedure*) The server selects a subset of clients S_t, $|S_t| = C \cdot K \geq 1$, and distributes the current global model w_t to all clients in S_t. After updating their local models w_t^k to the shared model, $w_t^k \leftarrow w_t$, each client partitions its local data into batches of size *B* and performs *E* epochs of SGD. At the end, the clients upload their trained local models w_{t+1}^k to the central server, which subsequently generates the new global model, w_{t+1} by computing a weighted sum of all received local models. The weighting scheme depends on the number of local training examples, as described through a pseudocode in Algorithm 1, and particularly in equation (3) below [15] [16].

$$w_{t+1} = \sum_{k \in S_t} \frac{n_k}{n_\sigma} w_{t+1}^k \quad (3) \quad \text{where} \quad n_\sigma = \sum_{k \in S_t} n_k$$

ALGORITHM 1: FedAvg

Operation on the server side:

initialize w_0

for each round t = 0, 1, ... do

$m \leftarrow \max([C \cdot K], 1)$

S_t = random set of m clients

for each client $k \in S_t$ **in parallel do**

$w_{i+1}^k \leftarrow$ ClientUpdate(k, w_t)

run *equation (3)*

Operation on the client side [*ClientUpdate(k, w_t)*]:

$B \leftarrow$ (split P_k into batches of size B)
for each local epoch i from 1 to E **do**
  for batch $b \in B$ **do**
    $w \leftarrow w - \eta \nabla \ell(w; b)$
return w to server

In Algorithm 1 (FedAvg algorithm) the K clients are indexed by k. B is the local mini-batch size, E is the number of local epochs, and finally η is the learning rate.

Particularly, FedAvg count on the operation of FedSGD. As a typical representation of the FederatedSGD (FedSGD) we could set $C = 1$ and then implemented a fixed learning rate of η which depicts to each client k the computation of $g_k = \nabla F_k(w_t)$, representing the average gradient on its local data at the state model w_t, and also the central server aggregates the given gradients and then applies the new $w_{t+1} \leftarrow w_t - \eta \sum_{k=1}^{K} \frac{n_k}{n} g_k$, based on $\sum_{k=1}^{K} \frac{n_k}{n} g_k = \nabla f(w_t)$. Another similar update of the model produced by the following equation, $\forall k,\ w_{i+1}^k \leftarrow w_t - \eta g_k$, which then becomes $w_{t+1} \leftarrow \sum_{k=1}^{K} \frac{n_k}{n} w_{i+1}^k$. The last aforementioned equation reflects each client that locally takes one step of gradient descent on the current model with the use of its local data, and continuously the server takes a weighted average of the resulting models. As a result, if the algorithm represented that way, the user be able to add more computation weight to each client by rehearsing the new local data, converted by $w^k \leftarrow w^k - \eta \nabla F_k(w^k)$ a several times before the step of averaging. This approach termed by H. B. McMahan et al. [16] as *Federated Averaging* approach, of better known as *FedAvg*.

5.2 CO-OP Algorithm

On the other hand, regarding the asynchronous approach, there is the CO-OP algorithm [20] which we have distinguished. With this approach it is possible to immediately combine any received client model with the global model. Particularly, each client k has an age z_k related with its model and the global model has age z. The model age difference, $z - z_k$, is used to calculate a weight when combing models. This scenario approach is motivated by the fact that in an asynchronous framework, some clients will "*train on outdated models while others will train on more up-to-date*" models [15].

In CO-OP algorithm, a local model will only be combined if $b_l \le z - z_k \le b_u$, for some choice of integers $b_l < b_u$. The intuition behind this rule of common acceptance is that we neither want to merge outdated models $(z - z_k < b_u)$ nor models from overactive clients $(z - z_k < b_l)$. The lower and upper bounds, b_l and b_u can therefore be thought of as an *age filter*. In addition to this, CO-OP inherits all "*hyper-parameters*" from its underlying optimization algorithm, which may be the SGD algorithm, such as FedAvg [15] [27].

The operation of training in CO-OP can be declared as follows: Each client has its own training data, and performs E rounds of an optimization algorithm before requesting the current global model age z from the

server. In this aspect, the client has to decide whether or not its age variation meets the restrictions. If the local model is outdated, the client reconciles with the global model and starts over. Otherwise, if the client is active, he simply continues his training. Differently, the local model is uploaded to the server for merging. The pseudocode of CO-OP algorithm is presented in Algorithm 2 [15].

ALGORITHM 2: CO-OP

$w = w_1 = ... = w_K \leftarrow w_0$

$z \leftarrow b_l$

$z_1 = ... = z_K \leftarrow 0$

Each client performs k independently runs:

while *true* **do**

Conglomerate a new batch of B samples D_k

$w_k \leftarrow ClientUpdate(w_k)$

Connection between client – server is ready

Request and receive the model age z from the server

if $z - z_k < b_u$ **then**

// Client is outdated

Fetch w, z from the server

$w_k \leftarrow w \ , \ z_k \leftarrow z$

else if $z - z_k < b_l$ **then**

// Client is overactive

continue

else

// Normal update

$w_k, z_k \leftarrow UpdateServer(w_{k,} z_k) = \{$

$w \leftarrow (1-z) \cdot w + z \cdot w_k \ , \ z \leftarrow (z - z_k + 1)^{-\frac{1}{2}}$

$z \leftarrow z + 1$

return & download w, z

}

There are some *age filter restrictions* on CO-OP. Moreover, CO-OP algorithm introduces two additional parameters in its procedures, namely b_l and b_u, but little guidance is provided in order to explain how one should choose these values. Only the intuitive constraint $b_l < b_u$ is given in the original paper that proposes CO-OP [27]. However, arbitrarily choosing these parameters by setting only this limitation in the mind can cause deadlock [15].

If all the clients are considered overactive, thus the algorithm deadlocks. For this reason two additional constraints that should be fulfilled to avoid this deadlock are identified: $b_l < K$ and $b_u < 2b_l$. If the first constraint is unfulfilled, CO-OP is guaranteed to deadlock after K updates. This follows from the intuition of

b_l ; at least b_l normal updates must be performed by distinct clients before a client is allowed another normal update. The second constraint says that a deadlock might occur if the difference between b_l and b_u is too small [15].

5.3 Proposed Method

As we can infer, the major advantage of FedAvg algorithm is that orchestrates training through a central server which hosts the shared the global. Additionally, the major advantage of CO-OP algorithm is that makes possible to immediately merge any received client model with the global model. Taking advantage of those two different scenarios we ended up to a scenario that merge these two algorithms in order to have a better efficient model, that selects depending on the occasion if it train the model locally in client or as global in server. Our proposed model named *InFeMo - Integrated Federation Model*.

The "*hyper-parameters*" of our proposal are the same used as the previous models: 1) the fraction of clients C to choose for training, 2) the local mini-batch size *B*, 3) the number of local epochs E, 4) the learning rate η, and 5) the learning rate decay λ. The parameters B, E, η, and λ are typically used when training with SGD, identically with FedAvg and CO-OP models.

In its operation, the IFM algorithm begins with randomly initializing the global model of w_0. Particularly, the operation procedure of the first round of our proposed model demonstrates as: The central-academic server chooses a subset of clients S_i, where will be over 1. Then, the global model which is selected at this time is distributed to all the connected clients S_i. Thereafter, a local model will only be merged if $x_i \le y - y_k \le x_u$, for some choice of integers $x_i < x_u$. The common accepted rule of our scenario is that we want to merge outdated models $(z - z_k < b_u)$. Subsequently, the client updates their local models in order to be shared model, $w_k \leftarrow w$, $y_k \leftarrow y$, each client partitions its local data into batches of size *B* and performs *local updates*. At the end, the clients upload their trained local models w_k and y_k to the central academic server, which subsequently generates the new global model, w_{t+1} by computing a weighted sum of all received local models. The overall weighting scheme is dependent on the number of local training updates, as described through a pseudocode in Algorithm 3, and particularly in equation (3) below.

$$w_{i+1} = \sum_{k \in S_i} \frac{n_k}{n_\sigma} w_{i+1}^k \qquad (4) \qquad \text{where} \quad n_\sigma = \sum_{k \in S_i} n_k$$

Equation (4) exceeds the already defined equation (3).

ALGORITHM 3: Proposed model - InFeMo

Client-side operation

$B \leftarrow$ (split P_k into batches of size B)
for each local update produced i from 1 to E **do**
  if ($y - y_k < x_u$) **then**
    Outdate client Client
    $w_k \leftarrow w$, $y_k \leftarrow y$

for batch $b \in B$ **do**

$w \leftarrow w - \eta \nabla \ell(w; b)$

else

update normally

$w_k, y_k \leftarrow UpdateServer(k, w_i)$

return to server w_k, y_k

Server-side operation

initialize w_0

for each round $i++, \ldots$ **do**

$m \leftarrow \max(S_t, 1)$

S_i = random set of nc clients

for each client $k \in S_i$ **in parallel do**

$w_{i+1}^k \leftarrow UpdateClient(k, w_i)$

run *equation (4)*

Particularly, proposed IFM algorithm keens to provide to the user less waiting time in the queue of the network for each procedure. Due to the decision system of the model the relative data could be decided to be trained locally or globally depending the priority of each occasion. This weighting scheme of the proposed model mainly depends on the number of local updates that could be done in each process.

6 EXPERIMENTAL RESULTS

We have made multiple experimental scenarios in order to compare and justify the operation of InFeMo model. Thus, through the experimental scenarios which we have made we have strengthened our suggestion that our proposed architecture is more efficient than the former works. We perform a number of simulations and measurements through which we can realize that we have done a good effort.

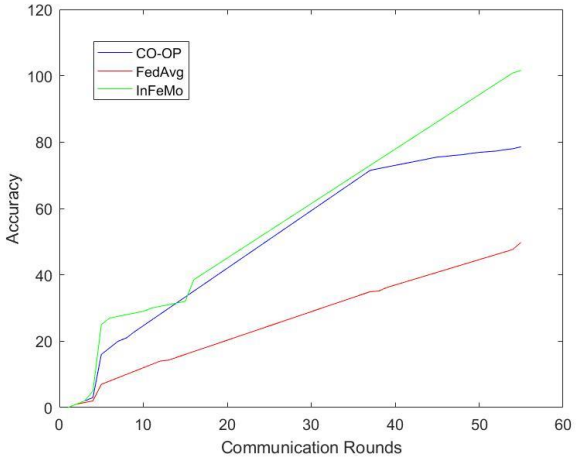


Figure 6: Performance comparison of the federated models InFeMo, FedAvg, and CO-OP.

Figure 6 describes the better efficient operation provided to the user by applying the InFeMo algorithm in the federated system architecture. As we can observe our proposed model offers more accuracy as long as the communication rounds rises instead of the other two models, the FedAvg and the CO-OP. This means that it could offer a better option of time needed for the user to contact and operate with the academic server. In Figure 6, the vertical axis shows the system's accuracy and the horizontal axis shows the communication rounds that have been examined.

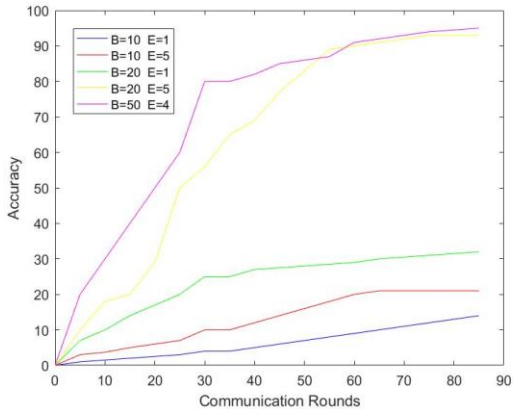


Figure 7: Test system's accuracy vs. communication rounds (scenario A).

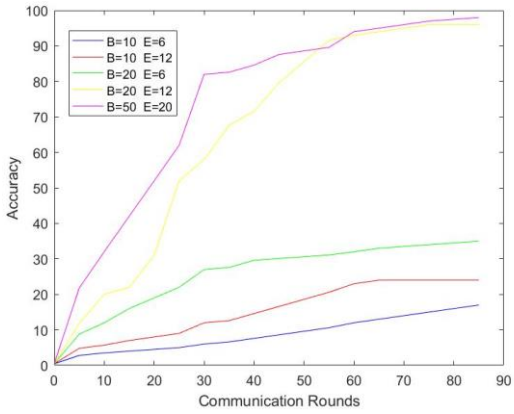
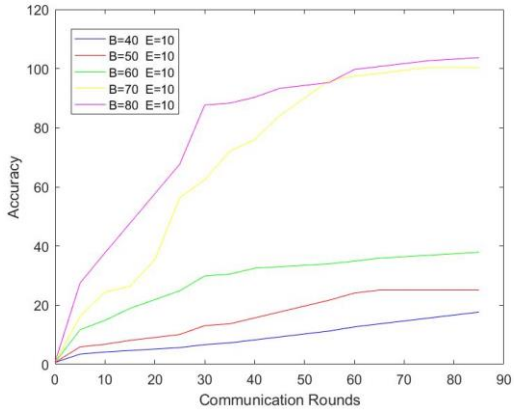


Figure 8: Test system's accuracy vs. communication rounds (scenario B).



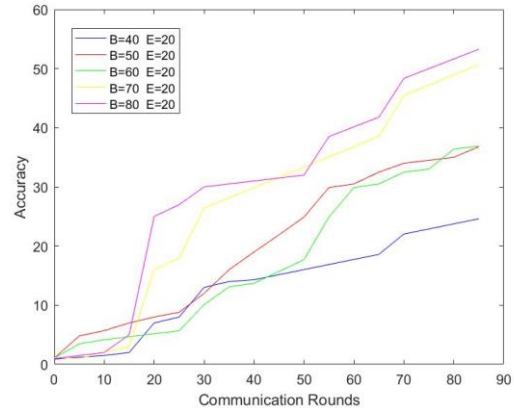


Figure 9: Test system's accuracy vs. communication rounds (scenario C).

Figure 10: Test system's accuracy vs. communication rounds (scenario D).

Figures 7, 8, 9, and 10 demonstrate four experimental scenarios that considering the efficiency of different measurements in time. Through these scenarios we can observe that adding more local SGD updates per round could assemble a dramatic reduction in communication costs. The vertical axis shows the system's accuracy and the horizontal axis shows the communication rounds that have been examined for these scenarios. More specific, the expected number of updates per client and per round here is $u = (E\left[n_k\right]/B)*E = (n*E)/(K*B)$, where the expectation is over the draw of a random client k. Thus, we can observe that increasing u by varying both E and B is more effective.

7 COMPARATIVE ANALYSIS

In order to analyze the functionality of our proposed model we have made comparison analysis with some relative previous projects.

The comparative analysis that takes into account here based on two aspects, the architectural model and the CSP-academic server-user communication-authentication model. As regards the architectural model we try to clarify the features of Topology, Encryption Method, Affiliated Technologies, and Cloud Model the works use and include in their function. On the other hand, regarding the CSP-academic server-user communication-authentication model we try to clarify the features of Computation, Authentication, Vulnerability, Trust, and Accessibility of each work compared here.

Table 1: Comparison of architectural model with other former ones

Work	Topology - Architecture	Encryption method/model	Affiliated Technologies	Cloud Model
Thakur et al. [17]	-	-	-	IaaS
Cai et al. [19]	MemepiC - Traditional Analytics Architecture	-	-	-
Pasquier et al. [20]	Cambridge Flow Control Architecture	IFC	IoT	-
Zhu et al. [21]	Blockchain architecture	Bilinear Pairing Generator	-	SaaS
Yan et al. [22]	-	Attribute-Based Encryption	-	-
Premarathne et al. [23]	-	Security Threat Vulnerability	-	-
Mansha & Kamiran [24]	Mixed topology	-	-	-
Yao et al. [14]	AlexNet Architecture	-	-	-
Wang et al. [25]	Edge Computing Architecture	-	Mobile Edge Computing, IoT	-

Nilsson et al. [15]	Star topology - Artificial neural network architecture	-	-	-
Young et al. [28]	-	-	-	-
McMahan et al. [16]	TensorFlow Architecture	-	-	-
Proposed Model	Mixed Cloud Architecture	AES	Big Data, IoT	SaaS, PaaS & IaaS

Table 1 presents the architecture model characteristics of former related works, compared with our proposed model. The main aspects that studied in order to produce our conclusions are *Topology/Architecture*, *Encryption method or model*, *Affiliated Technologies* integrated in each scenario, and which *Cloud Model* used in each scenario. More specifically, we can observe that most of the works related to Federated Learning Systems (5 of the 6) propose system architecture. Also, regarding the works related to Federated Learning Systems, only one work [25] contributed with another affiliated technology. On the other hand, only the works related to data management in Cloud environment contributed to an Encryption method or model (4 of the 6). This could be resulted because the major goal of this works related to the data, and its usage. Subsequently, through the illustrated findings of Table 1 we can observe that there are not many works in this field that contribute Federated Learning Systems with another affiliated technologies, and at the same time, proposing a new data management architecture/model.

Table 2: Comparison of architectural model with other former ones

Work	**Computation**	**Authentication**	**Vulnerability**	**Trust**	**Accessibility**
Thakur et al. [17]			**X**	**X**	
Cai et al. [19]	**X**				**X**
Pasquier et al. [20]		**X**		**X**	**X**
Zhu et al. [21]				**X**	**X**
Yan et al. [22]	**X**			**X**	**X**
Premarathne et al. [23]	**X**	**X**	**X**		**X**
Mansha & Kamiran [24]					**X**
Yao et al. [14]	**X**				
Wang et al. [25]	**X**				
Nilsson et al. [15]	**X**				**X**
Young et al. [28]	**X**				**X**
McMahan et al. [16]	**X**			**X**	**X**
Proposed Model	**X**	**X**	**X**	**X**	**X**

Table 2 lists the basic characteristics studied in this work compared with related previous works analyzed in Section 2, which are Computation, Authentication, Vulnerability, Trust, and Accessibility. As we can observe from Table 2 the most contributed characteristic is the “Accessibility”, contributed by 9 of 12 works,

with most of them contributing the topic "*Big Data management in Cloud*" (5 of 6) works. Additionally, the "Computation" characteristic also contributed most, by 8 of 12 works, with the most works contributed on the topic of "*Federated Learning Scenarios*" (5 of 6 works). Moreover, regarding the characteristics, the "Authentication" is the less contributed characteristic in the related previous works, contributed by 2 of 12, which contributed only from works of the topic "*Big Data management in Cloud*". Furthermore, the previous related work that contributes the most of the characteristics is U. S. Premarathne et al. work [14], from the topic of "*Big Data management in Cloud*", which contributes 5 of the 6 characteristics, "Computation", "Authentication", "Vulnerability", and "Accessibility". Summarizing, the aspects of Table 2 we can observe that arise a gap that our study tries to "fill up" by proposing and presenting a novel system that contributes five major characteristics ("Computation", "Authentication", "Vulnerability", "Trust", and "Accessibility") in this field.

Resulting in our findings, as shown by both tables, there is no prior work dealing with integrating specific Cloud models through the federated learning model. Also, none of the earlier work clarifies the encryption model it uses to authenticate users and communicate with the central server. In addition, the proposed model makes grouped and unified use of technologies, as much of the data that is transferred and managed is derived from Internet of Things technology and because of their unique nature, much of the data is characterized as Big Data. In general, there is no mention of consolidated use of technologies in previous work in this field. Further, from the study of the data obtained from the Table 2, it seems that very few of the previous papers studied here deal with Authentication and Vulnerability, as well as very few of the previous papers involve all the features listed in Table 2 in their study.

On the basis of these data, it seems that the present work is going to fill a scientific gap existing in this field of research. On the one hand, no other architecture model has been studied and proposed so far, which incorporates all Cloud models with a federated scenario, as well as other technologies that may have integrated use with each other. Therefore, we believe that this proposal introduces an innovative idea in the field of federated cloud systems, both in the field of administration, as well as end-user communication with the server, which is also related to security and immediacy.

7.1 Distributed Cloud vs. Federated Cloud

In this subsection will present the main differences between Distributed and Federated technique. These key differences also extend to their use in Cloud environments.

The fundamental dissimilarity among federated learning and distributed learning counts on the pretensions made on the features of the local datasets [29] [30]. On the one hand the distributed learning inventively targets at parallelizing computing power while on the other federated learning inventively targets at training on heterogeneous datasets. Additionally, distributed learning targets at training a single model on innumerable servers, a common underlying hypothesis is that the local datasets are i.i.d. and roughly have the same size. Federated learning does not count on speculations such these, rather the datasets are typically heterogeneous and their sizes might span various orders of magnitude [31].

Moreover, through the federated technique a problem that arises in the distributed technique could be solved. The problem is: "*Given a set of overlapping distributed queries bound to perform multiple aggregation operations on a given set of data sources, place the aggregation operators within the communication network to minimize the cost of data movement across the communication edges of network*" as previously set by S.

Mansha & F. Kamiran [24]. This issue was solved by S. Mansha & F. Kamiran [24] in their work use an evolutionary algorithm that expands a federated learnig technique.

Also, regarding to former works in the field, such as those of A. Nilsson et al. [15], H. B. McMahan et al. [26], and J. Konecny et al. [32], typically considered that the distributed optimization algorithms achieve that:

- ✓ Data is regularly distributed over clients
- ✓ Client-side data are independent and identically distributed (widely known as i.i.d.) illustrations from the overall distribution
- ✓ The number of clients is much smaller than the average number of locally available training examples per client

As a result, the distributed data center optimization typically obligates control over the data distribution since these approaches count on balanced and i.i.d. data speculations [15]. In the other hand, the federated learning proposes to have a number of edge devices perform its procedure tasks locally and as a result only communicate an updated model to a collaborating server with them [15]. Also, count on the novel law commitments, federated learning utilizes the General Data Protection Regulation's (GDPR) data minimization principle [33] since only the learned model, and no raw data, is produced centrally [15].

8 CONCLUSION & FUTURE WORK

Cloud Computing could be used to be a base technology for many technologies due to its type of services. Cloud Computing provides new generation of services which aims to offer accessibility to information, applications and data from any place at any time. Moreover, this work presented and described a new system architecture based on Cloud Computing, and count on the novel scenario of Federated Learning, which called Integrated Federated Model - InFeMo. Our model incorporates all Cloud models with a federated learning scenario, as well as other technologies that may have integrated use with each other. The major motivation of InFeMo is to offer provide a more efficient system architecture and environment for the academic users with the aim to data management. This efficiency of our proposal counts on its operation, because it decreases the number of rounds of communication that needed to train a scenario model by using a federated Cloud system, and as a result it makes the user that uses this system to wait less. System's federated algorithm relies on the advantages of the former models of FedAvg and CO-OP algorithms. Consequently, we ended up to our new scenario that merges these two algorithms aiming to have a more efficient model, which selects the training model depending on each occasion.

As a result, due to our work tries to fill a scientific gap in the field of federated cloud systems, we can make more researches and experiments in order to achieve and explore the new opportunities arising in this field of study. Based on our research and the comparative analysis, no other architecture model has been studied and proposed so far, which incorporates all Cloud models with a federated scenario, as well as other technologies that may have integrated use with each other. So, we keen on to work on this field trying to find out new aspects that could lead us to efficient and more secure communication between the user and the central server. Also, we could try to involve an IoT scenario of sensors and a Smart Building scenario in order to find out new methods and aspects that arise here.

There are also other areas where this proposed model could be applied, beyond the academic community, offering multiple benefits. As already mentioned above, the academic community can offer users a more efficient environment, offering less waiting time and ease in the mass management of their data. Regarding

the health sector, where the proposed model could also be applied, as it would mainly facilitate the medical staff in the easier access to the sensitive medical data and their analysis, which can be performed in the specific Cloud environment of this system, more immediate and efficient. Thus, for example, the doctor will be able to receive the data needed through the smart phone directly, and also due to the use of the federated scenario to enable data analysis applications in order to provide data immediately and quickly. Another area of application of the proposed model could be industry. There it would facilitate the most efficient supervision of the production process of a production chain. Based on the federated scenario of the proposed model, the managers of each related part of the production will be able to receive data analysis components, but also to have faster and more direct access to them. In addition, even in the administrative part of an industry it could help in better and faster access to data, by giving the users the ease of using a Cloud infrastructure without the necessary choice of a specific provider. These could be the next areas of the future continuation of our current research.

ACKNOWLEDGMENTS

The authors would like to thank the anonymous reviewers for their valuable comments and feedback which was extremely helpful in improving the quality of the paper. Also, the whole work is a part of the PhD project contacted from C. L. Stergiou.

REFERENCES

[1] C. Stergiou, K. E. Psannis, “Algorithms for Big Data in Advanced Communication Systems and Cloud Computing”, in Proceedings of 19th IEEE Conference on Business Informatics 2017 (CBI2017), Doctoral Consortium, 24-26 July 2017, Thessaloniki, Greece. [DOI: 10.1109/CBI.2017.28]

[2] B. Marr, “Big Data: The 5 Vs Everyone Must Know”, LinkedIn article, 6 March 2014. Retrieved: 17/12/2018. Link: https://www.linkedin.com/pulse/20140306073407-64875646-big-data-the-5-vs-everyone-must-know

[3] Z. Lv, A. K. Singh, “Big Data Analysis of Internet of Things System”, ACM Transactions on Internet Technology, vol. 0, issue: ja, Accepted on March 2020. [DOI: 10.1145/3389250]

[4] C. Stergiou, K. E. Psannis, “Recent advances delivered by Mobile Cloud Computing and Internet of Things for Big Data applications: a survey”, Wiley Online Library, International Journal of Network Management, vol. 27, issue: 3, pp. 1-12, May 2016.

[5] M. M. Rathore, A. Paul, A. Ahmad, M. Anisetti, G. Jeon, “Hadoop-Based Intelligent Care System (HICS): Analytical Approach for Big Data in IoT”, ACM Transactions on Internet Technology, vol. 18, issue: 1, No. 8, 24 pages, November 2017. [DOI: 10.1145/3108936]

[6] H. Yu, J. Yang, C. Fung, “Fine-grained Cloud Resource Provisioning for Virtual Network Function”, IEEE Transactions on Network and Service Management, in Press, 2020.

[7] C. Stergiou, K. E. Psannis, “Efficient and secure BIG data delivery in Cloud Computing”, Springer, Multimedia Tools and Applications, vol. 76, issue: 21, pp. 22803–22822, November 2017.

[8] C. Stergiou, K. E. Psannis, B.-G. Kim, B. Gupta, “Secure integration of IoT and Cloud Computing”, Elsevier, Future Generation Computer Systems, vol. 78, part 3, pp. 964-975, January 2018. [DOI:10.1016/j.future.2016.11.031]

[9] M. Hilbert, P. López, “The World’s Technological Capacity to Store, Communicate, and Compute Information” Science, vol. 332, issue: 6025, pp. 60–65, April 2011. [doi:10.1126/science.1200970].

[10] Z. Fu, K. Ren, J. Shu, X. Sun, F. Huang, “Enabling Personalized Search over Encrypted Outsourced Data with Efficiency Improvement”, IEEE Transactions on Parallel and Distributed Systems, vol. 27, issue: 9, September 2016. [DOI: 10.1109/TPDS.2015.2506573]

[11] D. Agrawal, S. Das, A. El Abbadi, “Big Data and Cloud Computing: Current State and Future Opportunities”, pp. 530-533, in Proceedings of 14th International Conference on Extending Database Technology, EDBT 2011, 21-24 March 2011, Uppsala, Sweden.

[12] C. Pahl, P. Jamshidi, O. Zimmermann, “Architectural Principles for Cloud Software”, ACM Transactions on Internet Technology, vol. 18, issue: 2, No. 17, 23 pages, February 2018. [DOI: 10.1145/3104028]

[13] N. Ferry, F. Chauvel, H. Song, A. Rossini, M. Lushpenko, A. Solberg, “CloudMF: Model-Driven Management of Multi-Cloud Applications”, ACM Transactions on Internet Technology, vol. 18, issue: 2, No. 16, 23 pages, January 2018. [DOI: 10.1145/3125621]

[14] X. Yao, C. Huang, L. Sun, “Two-Stream Federated Learning: Reduce the Communication Costs”, in Proceedings of 2018 IEEE Visual Communications and Image Processing (VCIP), 9-12 December 2018, Taichung, Taiwan, Taiwan. [DOI: 10.1109/VCIP.2018.8698609]

[15] A. Nilsson, S. Smith, G. Ulm, E. Gustavsson, M. Jirstrand, “A Performance Evaluation of Federated Learning Algorithms”, in Proceedings of DIDL '18: Proceedings of the Second Workshop on Distributed Infrastructures for Deep Learning, December 2018, pp.

1-8, Middleware '18: 19th International Middleware Conference Rennes France. [DOI: 10.1145/3286490.3286559]

[16] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, B. A. y Arcas, “Communication-Efficient Learning of Deep Networks from Decentralized Data”, in Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS) 2017, JMLR: W&CP, volume 54, 20-22 April 2017, Fort Lauderdale, Florida, USA. [arXiv:1602.05629]

[17] R. Shokri, V. Shmatikov, “Privacy-preserving deep learning”, in Proceedings of 53rd Annual Allerton Conference on Communication, Control, and Computing (Allerton), 30 September – 2 October 2015, Allerton Park and Conference Center, USA.

[18] S. Thakur, J. G. Breslin, “A Robust Reputation Management Mechanism in the Federated Cloud”, IEEE Transactions on Cloud Computing, vol. 7, issue: 3, pp. 625-637, July-September 2019. [DOI: 10.1109/TCC.2017.2689020]

[19] Q. Cai, H. Zhang, W. Guo, G. Chen, B. Chin Ooi, K.-L. Tan, W.-F. Wong, “MemepiC: Towards a Unified In-Memory Big Data Management System”, IEEE Transactions on Big Data, vol. 5, issue: 1, pp. 4-17, March 2019. [DOI: 10.1109/TBDATA.2017.2789286]

[20] T. F. J.-M. Pasquier, J. Singh, D. Eyers, J. Bacon, “CamFlow: Managed Data-sharing for Cloud Services”, IEEE Transactions on Cloud Computing, vol. 5, issue: 3, pp. 472 - 484, July-September 2017. [DOI: 10.1109/TCC.2015.2489211]

[21] L. Zhu, Y. Wu, K. Gai, K.-K. R. Choo, “Controllable and trustworthy blockchain-based Cloud data management”, Elsevier, Future Generation Computer Systems, vol. 91, pp. 527-535, February 2019. [DOI: 10.1016/j.future.2018.09.019]

[22] Z. Yan, L. Zhang, W. DING, Q. Zheng, “Heterogeneous Data Storage Management with Deduplication in Cloud Computing”, IEEE Transactions on Big Data, vol. 5, Issue: 3, pp. 393-407, September 2019. [DOI: 10.1109/TBDATA.2017.2701352]

[23] U. S. Premarathne, I. Khalil, Z. Tari, A. Zomaya, “Cloud-Based Utility Service Framework for Trust Negotiations Using Federated Identity Management”, IEEE Transactions on Cloud Computing, vol. 5, Issue: 2, pp. 290-302, April-June 2017. [DOI: 10.1109/TCC.2015.2404816]

[24] S. Mansha, F. Kamiran, “Multi-Query Optimization in Federated Databases using Evolutionary Algorithm”, in Proceedings of the 2015 IEEE 14th International Conference on Machine Learning and Applications, 9-11 December 2015, Miami, FL, USA. [DOI: 10.1109/ICMLA.2015.125]

[25] S. Wang, T. Tuor, T. Salonidis, K. K. Leung, C. Makaya, T. He, K. Chan, “Adaptive Federated Learning in Resource Constrained Edge Computing Systems”, IEEE Journal on Selected Areas in Communications, ver. 99, pp. 1-1, March 2019. [DOI: 10.1109/JSAC.2019.2904348]

[26] H. Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, et al. 2016. Communication-efficient learing of deep networks from decentralized data. arXiv: 1602.05629

[27] Yushi Wang. 2017. CO-OP: Cooperative Machine Learning from Mobile Devices. Master’s thesis. Dept. Elect. And Comput. Eng., Univ. Alberta, Edmonton, Canada.

[28] B. Young, R. Bhatnagar, G. Tatavarty, H. Bian, “Covariance Matrix Computations with Federated Databases”, in Proceedings of ICMLA '07: Proceedings of the Sixth International Conference on Machine Learning and Applications, 13-15 December 2007, pp. 172-177, Cincinnati, OH, USA. [DOI: 10.1109/ICMLA.2007.36]

[29] J. Konecny, H. B. McMahan, D. Ramage, “Federated Optimization:Distributed Optimization Beyond the Datacenter”, ArXiv, pp. 1-38, November, 2015.[arXiv:1511.03575] [Retrieved March 2020] [link: https://arxiv.org/abs/1511.03575]

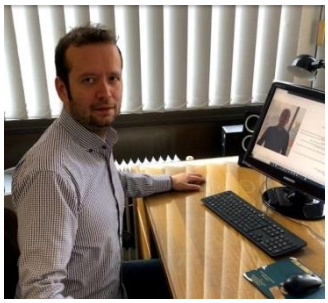
[30] J. Pei, P. Hong, K. Xue, D. Li, “Efficiently Embedding Service Function Chains with Dynamic Virtual Network Function Placement in Geo-Distributed Cloud System”, IEEE Transactions on Parallel and Distributed Systems, vol. 30, issue: 10, pp. 2179 – 2192, October 2019. [DOI: 10.1109/TPDS.2018.2880992]

[31] K.-Y. Chen, Y. Xu, K. Xi, H. J. Chao, “Intelligent virtual machine placement for cost efficiency in geo-distributed cloud systems”, in Proceedings of 2013 IEEE International Conference on Communications (ICC), 9-13 June 2013, Budapest, Hungary. [DOI: 10.1109/ICC.2013.6655092]

[32] Jakub Konecny, H. Brendan McMahan, Daniel Ramage, and Peter Richtarik. 2016. Federated Optimizetion: Distributed Machine Learning for On-Device Intelligence. arXiv: 1610.02527

[33] European Commission. 2018. What data can we process and under which conditions? Retrieved 14 March 2020, from https://ec.europa.eu/info/law/law-topic/data-protection/reform/rules-business-and-organisations/principles-gdpr/what-data-can-we-process-and-under-which-conditions_en

Christos L. Stergiou is currently a PhD candidate in the Department of Applied Informatics, School of Information Sciences, University of Macedonia, Greece. His main research interests include “Algorithms for Cloud Computing”, “Big Data Analytics”, “Artificial Intelligence”, “Machine Learning Techniques” and “Wireless Communication”. In 2017, he was awarded by the conference committee of 19th IEEE Conference on Business Informatics (CBI) for Doctoral Student Work titled “Algorithms for Big Data in Advanced Communication Systems and Cloud Computing”. He is a member of IEEE Industrial Electronics Society since 2017, and also he is a reviewer for several International Journals since 2016. Moreover, he has knowledge on Web Design and Development, Game Design and Editing, Graphics and 3D animation Design, Web and Mobile Application Development, Networking and Network Design, and Computer and Server Hardware.



Kostas E. Psannis was born in Thessaloniki, Greece. He is currently an Associate Professor in Communications Systems and Networking at the Department of Applied Informatics, School of Information Sciences, University of Macedonia, Greece, Director of Mobility2net Research & Development & Consulting JP-EU Lab and member of the EU-JAPAN Centre for Industrial Cooperation. Konstantinos received a degree in Physics (Department of Physics, founded in 1928), Faculty of Sciences, from Aristotle University of Thessaloniki (AUTH, founded in 1925), Greece, and the Ph.D. degree from the School of Engineering and Design, Department of Electronic and Computer Engineering of Brunel University, London, UK. From 2001 to 2002 he was awarded the British Chevening scholarship. The Chevening Scholarships are the UK government’s global scholarship programme, funded by the Foreign and Commonwealth Office (FCO) and partner organisations. The programme makes awards to outstanding scholars with leadership potential from around the world to study at universities in the UK. Dr. Psannis' research spans a wide range of Digital Media Communications, media coding/synchronization and transport over a variety of networks, both from the theoretical as well as the practical points of view. His recent work has been directed toward the demanding digital signals and systems problems arising from the various areas of ubiquitous big data/media and communications. This work is supported by research grants and contracts from various government organisations. Dr. Psannis has participated in joint research works funded by Grant-in-Aid for Scientific Research, Japan Society for the Promotion of Science (JSPS), KAKENHI Grant, The Telecommunications Advancement Foundation, International Information Science Foundation, as a Principal Investigator and Visiting Consultant Professor in Nagoya Institute of Technology, Japan. Konstantinos E. Psannis was invited to speak on the EU-Japan Co-ordinated Call Preparatory meeting, Green & Content Centric Networking (CCN), organized by European Commission (EC) and National Institute of Information and Communications Technology (NICT)/ Ministry of Internal Affairs and Communications (MIC), Japan (in the context of the upcoming ICT Work Programme 2013) and International Telecommunication Union. (ITU-founded in 1865), SG13 meeting on DAN/CCN, Berlin, July 2012, amongst other invited speakers. Konstantinos received a joint-research Award from the Institute of Electronics,

Information and Communication Engineers, Japan, Technical Committee on Communication Quality, July 2009 and joint-research Encouraging Prize from the IEICE Technical Committee on Communication Systems (CS), July 2011. Dr. Psannis has more than 60 publications in international scientific journals and more than 70 publications in international conferences. His published works has more than 2100 citations (h-index 24, i10-index 41). Dr. Psannis supervises a post-doc student and seven PhD students. Prof. Konstantinos E. Psannis serving as an Associate Editor for IEEE Access and IEEE Communications Letters. He is Lead Associate Editor for the Special Issue on Roadmap to 5G: rising to the challenge, IEEE Access, 2019. He is a Guest Editor for the Special Issue on Compressive Sensing-Based IoT Applications, Sensors, 2020. He is a Guest Editor for the Special Issue on Advances in Baseband Signal Processing, Circuit Designs, and Communications, Information, 2020. He is a Lead Guest Editor for the Special Issue on Artificial Intelligence for Cloud Based Big Data Analytics, Big Data Research, 2020. He is TPC Co-Chair at the International Conference on Computer Communications and the Internet (ICCCI 2020), Nagoya Institute of Technology Japan, ICCCI to be held in 2020 June 26-29 at Nagoya, Japan, and Conference Chair at the World Symposium on Communications Engineering (WSCE 2020- http://wsce.org/) to be held at University of Macedonia, Thessaloniki, Greece, October 9-11, 2020.

Brij B. Gupta received Ph.D. degree from Indian Institute of Technology Roorkee, India in the area of Information and Cyber Security. He has published more than 90 research papers (including 03 book and 14 chapters) in International Journals and Conferences of high repute including IEEE, Elsevier, ACM, Springer, Wiley Inderscience, etc. He has visited several countries, i.e. Canada, Japan, China, Malaysia, Hong-Kong, etc. to present his research work. His biography was selected and publishes in the 30th Edition of Marquis Who's Who in the World, 2012. He is also working principal investigator of various R&D projects. He is serving as associate editor of IEEE Access, Associate editor of IJICS, Inderscience and Executive editor of IJITCA, Inderscience, respectively. He is also serving as reviewer for Journals of IEEE, Springer, Wiley, Taylor & Francis, etc. Currently he is guiding 10 students for their Master's and Doctoral research work in the area of Information and Cyber Security. He is also serving as guest editor of various reputed Journals. Dr. Gupta is also holding position of editor of various International Journals and magazines. He has also served as Technical program committee (TPC) member of more than 20 International conferences worldwide. Dr. Gupta is member of IEEE, ACM, SIGCOMM, The Society of Digital Information and Wireless Communications (SDIWC), Internet Society, Institute of Nanotechnology, Life Member, International Association of Engineers (IAENG), Life Member, International Association of Computer Science and Information Technology (IACSIT). He was also visiting researcher with Yamaguchi University, Japan in January, 2015. His research interest includes Information security, Cyber Security, Mobile/Smartphone, Cloud Computing, Web security, Intrusion detection, Computer networks and Phishing.

```
\begin{CCSXML}
<ccs2012>
  <concept>
      <concept_id>10010147.10010341.10010342</concept_id>
      <concept_desc>Computing methodologies~Model development and
analysis</concept_desc>
      <concept_significance>500</concept_significance>
      </concept>
  <concept>
      <concept_id>10002951.10002952.10003190</concept_id>
      <concept_desc>Information systems~Database management system
engines</concept_desc>
      <concept_significance>500</concept_significance>
      </concept>
  <concept>
      <concept_id>10002978.10003018</concept_id>
      <concept_desc>Security and privacy~Database and storage security</concept_desc>
      <concept_significance>300</concept_significance>
      </concept>
 </ccs2012>
\end{CCSXML}

\ccsdesc[500]{Computing methodologies~Model development and analysis}
\ccsdesc[500]{Information systems~Database management system engines}
\ccsdesc[300]{Security and privacy~Database and storage security}
```