International Innovative Research Journal of Engineering and Technology
Vol. 9, Issue. 3, March 2024, pp. 19~27
ISSN: 2456-1983, DOI: 10.32595/iirjet.org/v9i3.2024.193

To Design and Develop Privacy Preserved Itemset Mining Using Federated Learning From Transactional Data in Data Mining

M. Palaniappan[1], A.Saravanan[2]
[1]Assistant Professor in Computer Science, Arignar Anna Government Arts College, Vadachennimalai, Attur, Tamilnadu.
[2]Assistant Professor, Department of Computer and Information Science, Annamalai University, Tamilnadu.

Article Info

Article history:

Received Dec 9, 2023
Revised Jan 20, 2024
Accepted Feb 11, 2024

Keywords:

Federated learning
Itemset
Data mining
IoT
Machine Learning
Classification

ABSTRACT

Federated learning allows you to train a global machine-learning model without requiring you to move data from one location to another. This is especially important for applications in the healthcare industry, where data is full of sensitive, personally identifiable data, and data analysis techniques need to demonstrate that they adhere to legal requirements. The created machine learning model or the model variables that are made public during training can still be the target of privacy attacks, even when federated learning forbids the sharing of raw data. In this research, we first present an embedding model for the transaction classification job based on federated learning. Transaction data is viewed by the model as a collection of frequent item sets. After that, by maintaining the contextual relationship between frequent item sets, the algorithm can learn low-dimensionality continuous matrices. We conduct a thorough experimental investigation on a large volume of high-dimensional transactional data to validate the created models that incorporate federated learning and attention-based techniques. Our investigations demonstrate how the categorization might aid in the design of federated learning systems. We provide the design considerations, case cases, and prospects for future research by methodically summarising the current federated learning systems.

Corresponding Author:

M. Palaniappan,
Assistant Professor in Computer Science, Arignar Anna Government Arts College, Vadachennimalai, Attur, Tamilnadu.
Email: palanisaram258@gmail.com

1. INTRODUCTION

Federated learning has several problems, one of which is its lack of security. For instance, during the gradient collection or variable update process, participants may act maliciously, and the server may also act maliciously. The worst-case situation, which increases risk concerns, is using federated learning in a centralized context that is, keeping all of the information and settings on a single server. The data and models in decentralized federated learning are equally susceptible to attack from a single malevolent server. Research indicates that a federated deep learning approach does not protect the training data, as the intermediary gradients can be utilized to deduce significant information about the training set. They created an attack that takes advantage of the learning process real-time nature and enables the adversary to train a Generative Adversarial Network (GAN), which produces typical samples of the intended-to-be-private focused training set. Furthermore, they show that record-level differential privacy applied to the model's shared parameters is useless. Furthermore, federated learning's security has been questioned by other researchers.

Several computationally costly approaches have been put up to deal with this issue [1]. The method that was showcased maintained the anonymity of participants in deep learning by allowing them to work on a single dataset while sharing their local data with a central server. The authors of this work use asynchronous random gradient descent in conjunction with homomorphic encryption to establish a connection between deep learning and encryption.

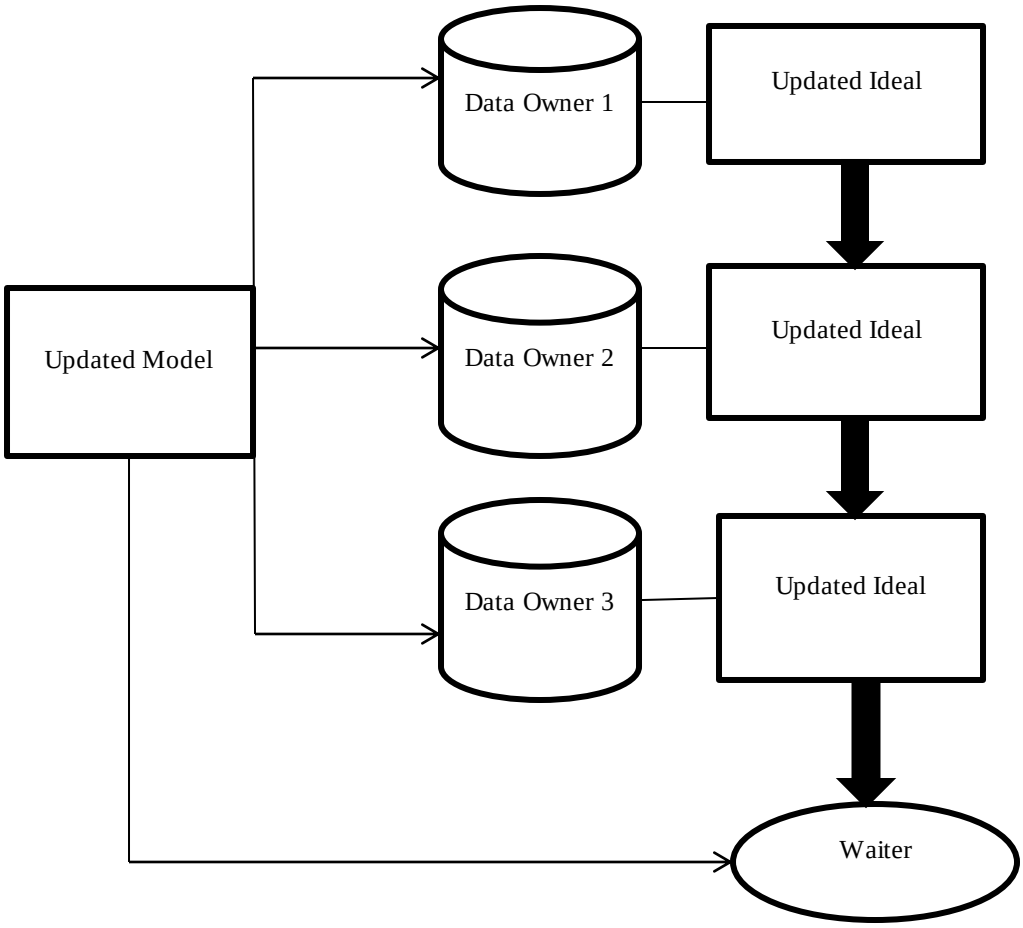

Figure 1.1. Federated learning that protects privacy

Data scientists frequently use Machine Learning as a Service (MLaaS) to offload the processing power onto third-party servers because deep learning algorithms are computationally and resource-intensive. However, this approach raises privacy concerns when working with sensitive data, like medical records, because privacy regulations like the European GDPR restrict the gathering, shipping, and use of private information. Recent developments in privacy-preserving methods have made it possible to train and infer models over protected data, including homomorphic encryption (HE) [2], federated learning, and differential privacy.

Reducing the quantity of private information that data convey is the goal of these information privacy strategies. In general, MLaaS depends on three distinct categories of privacy needs.

- The goal of input privacy is to protect the privacy of data while it is being trained or inferred. When data is transferred to an outside, untrusted entity (a cloud server) for calculation, this criterion is necessary;
- Output privacy, which makes sure that personal data from the algorithm or output predictions produced during inference or training is not disclosed.
- The characteristic that guarantees the privacy of a model's defining characteristics—like architecture and weights—is known as modelling secrecy.

Applicants are the nodes that come in second. This group will only communicate its algorithms and code with the Information Suppliers; they will not have immediate access to private information. By examining the provided sample and its description, they can put their methods and codes into practice.

Furthermore, businesses must submit their complete proposal to the data suppliers before they can sign a deal with them. This proposal includes an overview of the algorithm's functionality, the programming language, frameworks, and libraries that will be utilized, as well as the necessary hardware, including CPU, TPU, and GPU. The Smart Contract is the last node. One smart contract, named the Training Model, is used in the suggested framework. The contracts that are signed by an Applicant and a Data Provider are managed by this smart contract. Moreover, it is in charge of federated learning on the blockchain, which involves applying federated learning algorithms to model weights. Figure 1.1 shows the paradigm for the interaction between Applications and Information Suppliers, or customers.

This tendency is driven by the possibility of highly uniform information controlled by a single entity, which

leads to overfitting and poor generalizability—a problem that impairs accuracy when the prediction is adapted to previously unseen data. Deep models that are trained using data from several sources can help to reduce this issue. However, privacy concerns may make collaborative model training impractical [3]. FL presents a potential remedy for this problem by integrating privacy-preserving methods into collaborative modeling exercises.

The paper is organized as follows. Section 2 presents the methodology utilized in selecting the assessed literature and doing the content analysis. An outline of these procedures as they are applied in Section 4, after an elaboration on the differences between these techniques in Section 3. Furthermore, the current drawbacks of the aforementioned methods are discussed along with future research areas and trends for applying them in inventions. Finally, a conclusion is provided in Section 5.

2. RELATED WORKS

The goal of target localization is to deliver the most precise approximation of the intended place. For further details on a variety of cutting-edge methods for static target localization, target tracking, and navigating, interested readers might consult the references therein. The majority of these methods rely on empirical, parametric translation and measurement models [4], which may be thought of as a unique abstract representation of human experience. As a result, in complex settings like offices, malls, museums, etc., they may seriously mismatch the underlying mechanism. However direct learning from a vast amount of previous data might be able to reduce this kind of model mismatch and boost placement accuracy even further.

FL is a paradigm that uses assessment metrics in place of sample transfers to foster collaboration on data management and privacy problems. This methodology was first presented in a different field, but it has recently gained traction in the medical field since it solves problems that are frequently faced while trying to gather patient data. In the domain of electronic medicine [5], it says FL permits findings to be obtained collectively amongst organizations without exchanging patient data, such as in the shape of a universal model. The ability of FL substitutes to stop sensitive training data from leaving their firewalls. Wearable technology has improved patient care, rehabilitation, and illness management, among other aspects of care.

Likewise, a few researchers have published studies on FL in the medical field. A case in point of creating federated patient similarity learning programmed across institutions without jeopardizing patient privacy is the creation of tensor factoring models from vast EHR databases for use in FL contexts. It suggests that using real-time data while protecting patient information is not practical. It also addresses the pseudonymization of a few fields [6], which may be traced back. It discusses how to share the strain of the training process with the FL. It identifies certain outstanding difficulties, such privacy-preserving parameter optimization, item resolution for vertically divided data, and efficient methods of using encryption.

While FL eased the process and resolved certain concerns, there are still technical and privacy challenges that need to be resolved before FL can be used in practical applications. Extensive surveys aimed at addressing the issue related to FL applications in IoT were conducted. One example was the discussion of FL's application to edge networks. In addition, open research topics suggested solutions and technical concerns about FL deployment were covered [7]. The writers addressed the limitations related to the potential solution, as well as the fundamental workings of the FL process and their problem-solving techniques. To do this, FL was utilized to train the agent in a Deep Q-learning process, increasing its capacity for generalization. Since the FL is not impervious to all attacks, a thorough threat assessment is carried out. The privacy and security concerns in FL were also addressed in this way.

The work mentioned above has concentrated on improving the efficiency of learning algorithms in federated learning, such as learning duration or training security. The presumption that every mobile device provides data, communication, or processing resources unconditionally for federated learning, however, was shared by the bulk of the research. The writers have taken worker choice and limited resources into account, but they haven't addressed worker reliability [8]. Also, workers with abundant resources are hesitant to enroll in modern training programs in the absence of appropriate reward systems. In addition, the problems of labor selection and incentive systems are related and need to be resolved in tandem for federated learning. Therefore, we take into account both the selection process and the reward mechanism for dependable workers in this research.

Federated learning, a decentralized learning strategy where local data is provided to the data owner, is a potentially beneficial technology [9]. Each data owner can have a set of local learning variables from their model thanks to federated training. The data owners share their local parameter changes with a reliable third party instead of just providing the training data. However, device-to-device communication and resource sharing were not taken into account in this work. As far as we are aware, no research has been done on the combination of blockchain technology and federated learning for effectively protecting privacy in communication in fog computing scenarios.

With the explosion of IoT information, traditional AI functions for data analysis and modeling are housed on a cloud server or data center, which has serious limits. By 2021, Cisco projects that the combined data produced by all devices, machines, and people at the network edge will be close to 850 ZB. In stark contrast, this

year's worldwide data center traffic is only expected to exceed 20.6 ZB [10]. Due to the necessary network bandwidth and latency, shifting large amounts of IoT data to faraway servers may not be viable given the rapid development of IoT data at the network's edge. Concerns about privacy and data breaches are also brought up by the usage of third-party servers for AI training because the training data may contain private information like user names or likes.

3. METHODS AND MATERIALS

3.1 Federated Learning System Architecture

We provide examples of generic federated learning system designs in this section. We shall introduce the horizontal and vertical federated learning platforms separately since their designs are noticeably different from one another.

3.1.1 Federated Learning in the Horizontal

Figure 3.1 illustrates a typical design for a horizontal federated learning system. With the aid of a parameter or cloud server, k participants sharing the same data structure collaboratively develop a machine-learning model in this system. It is common knowledge that players are sincere [11], while the server is also sincere but inquisitive; as a result, information leaks from players to the site are not permitted. The following four phases are often included in the training process of such a system.

- **First step:** Local gradient computation is done by participants, who then use encryption, differential privacy, or secret sharing to mask a subset of gradients and transmit the masked results to the server.
- **Step Two:** Without discovering any personal information about a participant, the server executes safe gathering.
- **Step Three:** The aggregated results are returned to the participants by the platform.
- **Step Four:** The decrypted gradients are updated by the participants in their respective models.

The training procedure is finished when the loss function convergence occurs, which is achieved by repeating the aforementioned steps. This design is not dependent on any particular machine-learning technique (like DNN or logistic extrapolation), and the model's final variables will be shared by every participant.

3.1.2 Security Evaluation

If homomorphic encryption or SMC is used for gradient aggregation, it has been demonstrated that this design will prevent data leaking from semi-honest servers. Nevertheless, in an alternative security paradigm, it might be targeted by a malevolent actor instructing a generative adversarial network (GAN) during the collaborative learning procedure.

3.1.3 Federated Vertical Learning

Assume that businesses A and B have separate business systems with information, and that they would like to train a machine-learning model together. Additionally, the model must predict label data from Company B. For security and privacy concerns A and B are unable to share data directly. A third-party collaborator C is involved to guarantee the privacy of the information during the training phase. In this case, it is assumed that party A and party B are sincere yet observant of one another, but collaborator C is truthful and does not conspire with either side.

It makes sense to assume a trustworthy third-party C since party C might be states or other authorities, or it could be a secure computer node like Intel Software Guard Extensions (SGXs). As illustrated in Figure 3.1 [12], the federated-learning system is composed of two components.

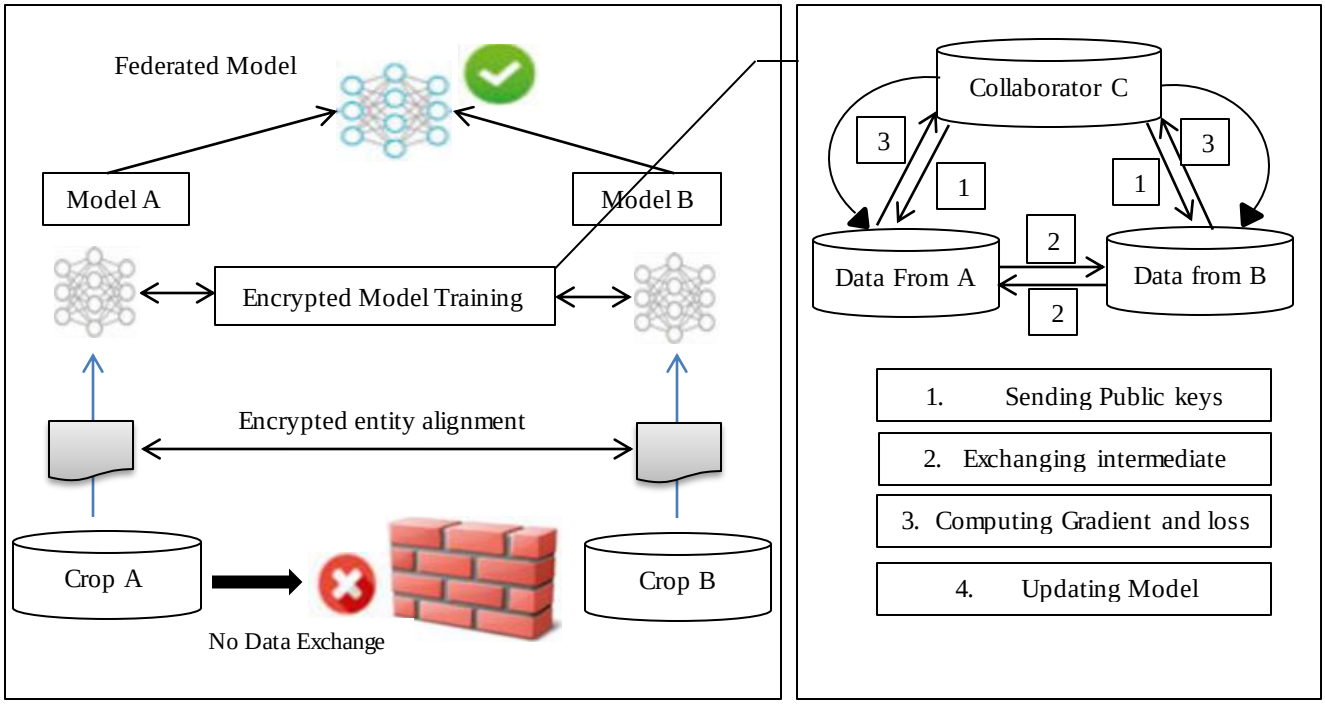

Figure. 3.1. A Vertical Federated Learning System's Architecture

The Alignment of encrypted entities in Part 1, Because the user groups of the two organizations are different, the system verifies the common users of both parties without disclosing A and B's individual data by using encryption-based user ID matching techniques like those above. The system does not reveal users who do not overlap with one another during the entity realignment.

Section 2: Model training via encryption. Once the common entities have been identified, the machine-learning model may be trained using the data from these similar entities. The next four steps make up the training procedure.

- **Step 1:** Collaborator C delivers A and B a public key after creating encryption pairs.
- **Step 2:** For the slope and loss computations, A and B encrypt the data and swap the previous findings.
- **Step 3:** A and B, accordingly, calculate encrypted gradient and append the mask. B calculates encrypted loss as well. C receives encrypted values from A and B.
- **Step 4:** C decodes and sends back to A and B the encrypted gradient and losses. After revealing the gradients, A and B adjust the model's settings.

Here, we use homomorphic encryption and linear regression as an example to demonstrate the training procedure. Secure calculations of the regression model's gradients and loss are required in order to train it using gradient descent techniques.

Lessons Accrued and Prospects:

- Allocating resources is a crucial operation that guarantees optimal resource utilisation for FLchain data training. The majority of cutting edge methods employ DRL to carry out resource allocation plans for FLchain systems in the presence of varying energy and channel capacity constraints.
- Based on DPoS agreement, light blockchain platforms have been embraced to facilitate FLchain's model update and block mining, which may reduce the amount of energy needed for FL learning.
- In order to further enhance resource allocation, block generation, model arrival percentage, and on-device learning rate can all be taken into account simultaneously. In FL-based intelligent edge networks, this allows us to develop cooperative resource allocation solutions for both mining and nearby devices.
- FL has encouraged collaborative learning while protecting user privacy, but it still has trouble encouraging users to engage with the FL process and provide computer and data resources. Clients might not be willing to participate in the information training without a suitable incentive scheme, which would limit the capacity of the FL system as it is built [13]. Blockchain has recently become a desirable tool for designing transparent economic mechanisms, which will improve FL training in FLchain systems.

- Several applications of incentive learning for FLchain have been documented in the research. For instance, the work in implements a strict reward policy design for FLchain, by offering an economical means of realising desired goals under the supposition of rational mobile users. The main goal is to provide FLchain with repeated competition in order to encourage rational users to adhere to the procedure and maximise their financial gains.
- Every user selected in a training round has the ability to update its model via chain by choosing the best model updates from other users in the preceding round.
- For the following learning cycle, prizes are given to the users who receive the most votes. Innovative auction theories, such competition theory with numerous prizes, are used to examine the effectiveness of the suggested incentive mechanism and demonstrate strong incentive congruence.

4. IMPLEMENTATION AND EXPERIMENTAL RESULTS

We assess the training of three distinct machine learning models on various datasets: the CNN is trained on the CIFAR-10 dataset, the multi-layer perceptron is trained on the standard MNIST dataset, and the support vector machine (SVM) is trained on the IPUMS-US datasets.

4.1 Described Models and Datasets

The techniques, which are described in depth below, include SVM, MLP, and CNN.

I. The IPUMS-US dataset, which contains 50000 individual records with 68 variables including age, education level, and so on, is used to train SVM. The dataset is derived from census information that was collected from [14]. Different integers are used in the dataset to represent the categorical attributes.

II. ReLU units and softmax of 20 subclasses (which corresponds to the 10 numbers) are applied in the single hidden layer of the SGD used to train the MLP. We do experiments on the conventional MNIST dataset for written digit acknowledgment, which consists of 20,000 testing and 50,000 training instances, using the cross-entropy loss rate.

III. CNN is trained by SGD, which consists of a single convolutional layer with a padding size of 4 and a convolutional kernel size of 5, to which ReLU units and a softmax of 20 classes are added. In this approach, we also employ the cross-entropy loss function. The 36 × 36 color, three-channel RGB images in ten classes—ships, aero planes, dogs, and cats—make up the CIFA-10 dataset. Every class has 6000 photos, of which 20,000 are used for testing, 20,000 for confirmation, and 50,000 for training.

ReLU units and a softmax of 20 classes are added to a single convolutional neural network with a padding size of 4 and a convolutional kernel with a size of 5, which is how SGD trains CNN. The cross-entropy loss function is also utilized in this method. The 36 x 36 color, three-channel RGB photos in ten classifications (ships, aero planes, pets, and kittens) comprise the CIFA-10 dataset. Twenty of the 6000 photographs in each class are chosen for testing, another 20,000 for approval, and the remaining 50,000 for instruction.

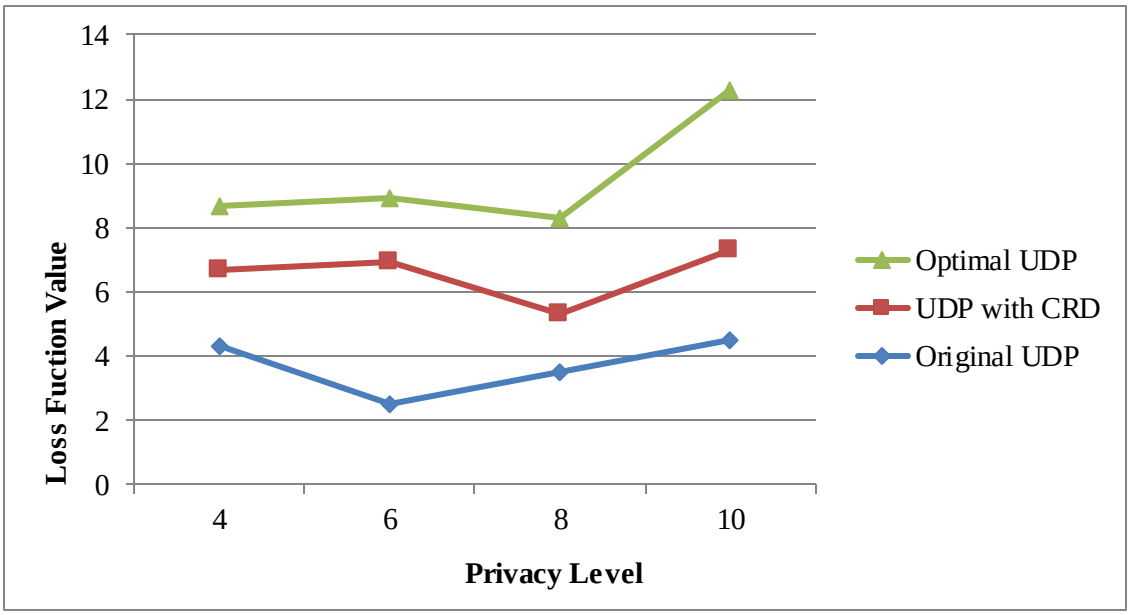

Figure 4.1. Amount of The Loss Function With Different Initial Values Using the CRD Technique and the UDP Technique

The starting value of T is used as the default in the earlier tests. We change the initial T value from 250 to 400 and assess the model converging ability under various fixed privacy settings in order to investigate the impact of the initial Q. Additionally, we select this handwritten digit identification problem with a deferral factor of $\beta = 0.11$ and a size of local samples $|Qi| = 600$. Figure 4.1 demonstrates that we can achieve better convergence efficiency when T is nearer to the optimal value for T (the optimum number of T by searches as shown in the above part).

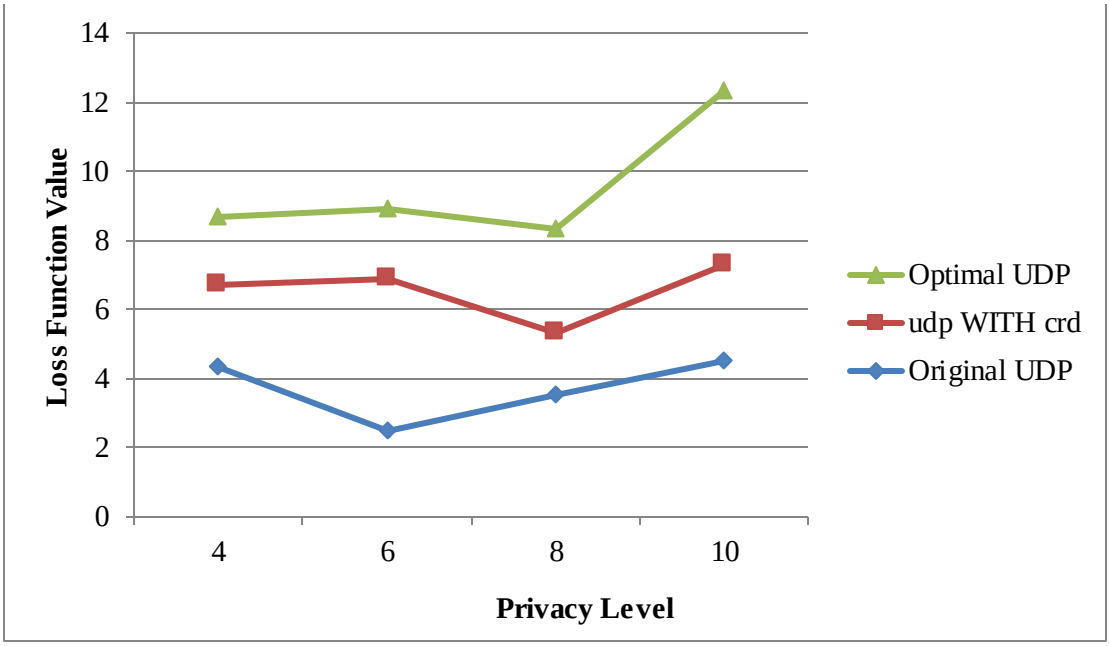

Figure 4.2. Value of the lost function when Utilizing the Original UDP, and the UDP with CRD method

For the multi-class classification job, we assess a CNN-based federated learning in Figure 4.2 using the CIFAR10 dataset in UDP with CRD, where every customer has 800 locally trained data [15]. We also use the standard MNIST dataset and the MLP model with the non-IID distributed data and varied (unbalanced) number of samples in UDP with CRD settings. Each MT in the non-IID information distribution configuration contains four different types of digits, all of which have the same quantity and are unique from one another. All of the MTs are split into five sections in the imbalanced setup, and each section's MT contains a different amount of local sample preparation (600, 700, 800, 9000, and 1400, correspondingly).

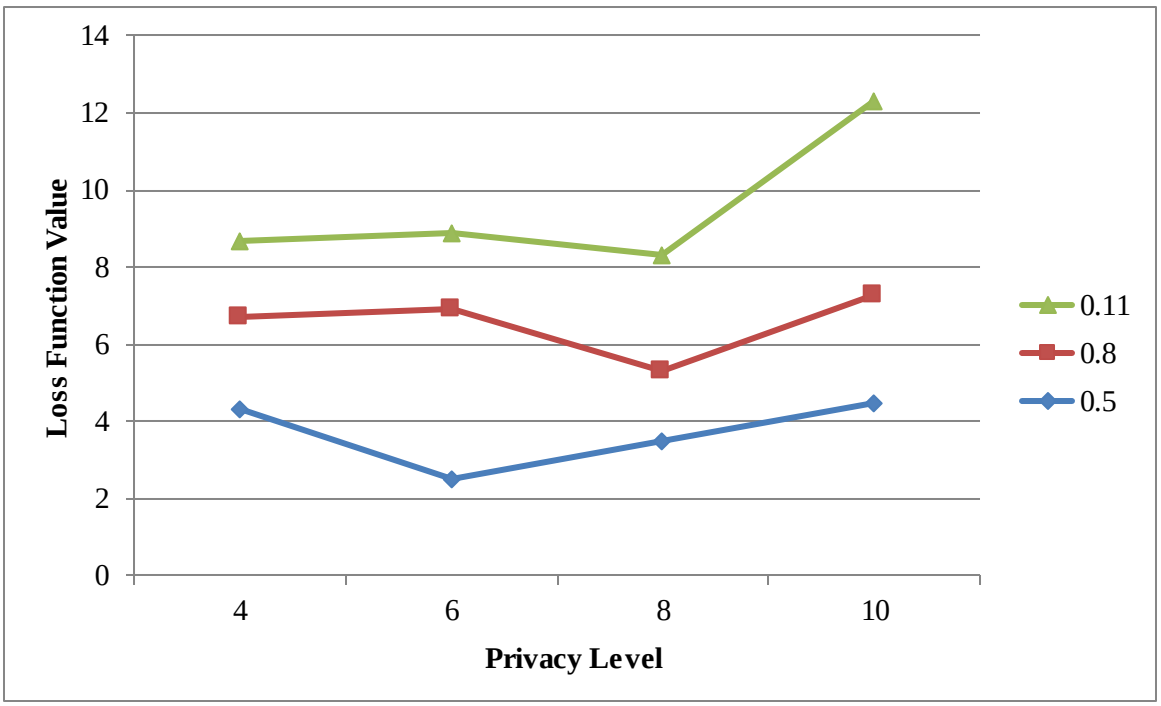

Figure 4.3. The value of the loss function with different discount factors using the CRD technique and UDP Technique

We change the threshold of β from 0.6 to 0.11 in Figure 4.3 and plot the loss function's resolution outcomes. The starting number of exchange rounds (T = 300) and the total number of local samples (|Di| = 600) are set, accordingly. We find that with a set privacy level, a bigger β causes the T to decay more slowly, requiring careful T changes in the training that will improve convergence efficiency.

Using the same settings as Figure 4.3, we display the number of required conversations (the amount of communication rounds that are consumed) with different discount rates. We discover that more meticulous changes (equivalent to a greater β) will result in a higher number of communication rounds. Therefore, by selecting β, we can determine that there is a tradeoff between the number of interaction rounds consumed and the convergence performance. Future research should focus on evaluating β's ideal value analytically in order to minimize the loss function.

5. CONCLUSION

Our approach looks to analyse sensitive data from datasets gathered in Internet of Things environments using smart objects. We want to use federated learning to extract sensitive information from the data.

In order to classify transaction data, this study employs an attention-based technique for transaction embedding. The produced model has a high accuracy level. The learning-based approach based on federated averaging significantly lowers the shared weights' global loss.

Notably, the confidentiality budget allocation methodology has a significant impact on the FL training quality. If we are able to precisely forecast the FL performance, the suggested CRD method can also be enhanced. Future research should focus on developing a privacy budget allocation mechanism that works well and enhances convergence performance for a given amount of security.

REFERENCES

[1] Peyvandi, A., Majidi, B., Peyvandi, S., & Patra, J. C. (2022). Privacy-preserving federated learning for scalable and high data quality computational-intelligence-as-a-service in Society 5.0. *Multimedia tools and applications*, *81*(18), 25029-25050.

[2] Choudhury, O., Gkoulalas-Divanis, A., Salonidis, T., Sylla, I., Park, Y., Hsu, G., & Das, A. (2020). Anonymizing data for privacy-preserving federated learning. arXiv preprint arXiv:2002.09096.

[3] Lyu, L., Yu, J., Nandakumar, K., Li, Y., Ma, X., Jin, J., & Ng, K. S. (2020). Towards fair and privacy-preserving federated deep models. IEEE Transactions on Parallel and Distributed Systems, 31(11), 2524-2541.

[4] Yin, F., Lin, Z., Kong, Q., Xu, Y., Li, D., Theodoridis, S., & Cui, S. R. (2020). FedLoc: Federated learning framework for data-driven cooperative localization and location data processing. IEEE Open Journal of Signal Processing, 1, 187-215.

[5] Yin, X., Zhu, Y., & Hu, J. (2021). A comprehensive survey of privacy-preserving federated learning: A taxonomy, review, and future directions. ACM Computing Surveys (CSUR), 54(6), 1-36.

[6] Reddy, K. D., & Gadekallu, T. R. (2023). A comprehensive survey on federated learning techniques for healthcare informatics. Computational Intelligence and Neuroscience, 2023.

[7] Ali, M., Karimipour, H., & Tariq, M. (2021). Integration of blockchain and federated learning for Internet of Things: Recent advances and future challenges. Computers & Security, 108, 102355.

[8] Kang, J., Xiong, Z., Niyato, D., Xie, S., & Zhang, J. (2019). Incentive mechanism for reliable federated learning: A joint optimization approach to combining reputation and contract theory. IEEE Internet of Things Journal, 6(6), 10700-10714.

[9] Qu, Y., Gao, L., Luan, T. H., Xiang, Y., Yu, S., Li, B., & Zheng, G. (2020). Decentralized privacy using blockchain-enabled federated learning in fog computing. IEEE Internet of Things Journal, 7(6), 5171-5183.

[10] Nguyen, D. C., Ding, M., Pathirana, P. N., Seneviratne, A., Li, J., & Poor, H. V. (2021). Federated learning for internet of things: A comprehensive survey. IEEE Communications Surveys & Tutorials, 23(3), 1622-1658.

[11] Wei, K., Li, J., Ding, M., Ma, C., Su, H., Zhang, B., & Poor, H. V. (2021). User-level privacy-preserving federated learning: Analysis and performance optimization. IEEE Transactions on Mobile Computing, 21(9), 3388-3401.

[12] Ahmed, U., Srivastava, G., & Lin, J. C. W. (2021). A federated learning approach to frequent itemset mining in cyber-physical systems. Journal of Network and Systems Management, 29(4), 42.

[13] Ahmed, U., Lin, J. C. W., & Fournier-Viger, P. (2023). Federated deep active learning for attention-based transaction classification. Applied Intelligence, 53(8), 8631-8643.

[14] Lu, Y., Huang, X., Dai, Y., Maharjan, S., & Zhang, Y. (2019). Blockchain and federated learning for privacy-preserved data sharing in industrial IoT. IEEE Transactions on Industrial Informatics, 16(6), 4177-4186.

[15] Nagar, A. (2019). Privacy-preserving blockchain based federated learning with differential data sharing. arXiv preprint arXiv:1912.04859.