Journal of Discrete Mathematical Sciences & Cryptography
ISSN 0972-0529 (Print), ISSN 2169-0065 (Online)
Vol. 27 (2024), No. 2-A, pp. 249–259
DOI : 10.47974/JDMSC-1879

Secure multi-party computation in deep learning : Enhancing privacy in distributed neural networks

P. Vidya Sagar *
Department of Computer Science & Engineering
Koneru Lakshmaiah Education Foundation
Vaddeswaram
Andhra Pradesh
India

Hayder M. A. Ghanimi †
Department of Information Technology
College of Science
University of Warith Al-Anbiyaa
Karbala
Iraq

and

Department of Computer Science
College of Computer Science and Information Technology
University of Kerbala
Karbala
Iraq

L. Arokia Jesu Prabhu §
Department of Computer Science and Engineering
CMR Institute of Technology
Hyderabad
Telangana
India

* *E-mail:* **pvsagar20@gmail.com (Corresponding Author)**
† *E-mail:* **hayder.alghanami@uowa.edu.iq**
§ *E-mail:* **arokiajeruprabhu@gmail.com**

L. Raja [‡]
Department of Electronics and Communication Engineering
Sri Eshwar College of Engineering
Coimbatore
Tamil Nadu
India

Pankaj Dadheech [@]
Department of Computer Science & Engineering
Swami Keshvanand Institute of Technology
Management & Gramothan (SKIT)
Jaipur 302017
Rajasthan
India

Sudhakar Sengan [#]
Department of Computer Science and Engineering
PSN College of Engineering and Technology
Tirunelveli 627152
Tamil Nadu
India

Abstract

Ensuring data privacy while applying Deep Learning (DL) on distributed datasets represents an essential task in the current period of critical data security. Data privacy and accuracy of models are typically impacted by traditional methods. Data privacy is of the most tremendous significance in distributed data settings, and the current research presents a novel model for DL termed Secure Multi-Party Computation (SMPC). The accuracy of the mathematical models and confidentiality of the data are frequently compelled to coexist in conventional methods. In order to enable collaborative DL without compromising private information, the recommended system uses the Paillier Homomorphic Encryption Scheme (PHES). By using innovative cryptographic methods, this decentralized method secures the confidentiality of data without utilizing a Trusted Authority (TA). By performing a thorough assessment of the CIFAR-10 and IMDB datasets, the present study demonstrates that the system the author uses is simple and scalable and that it offers accuracy on par with conventional methods. By presenting an approach for achieving a balance between the two competing demands of data security and computational performance, this method signifies a vast advance forward with a confidentiality DL.

Subject Classification: *94B35.*

Keywords: *Deep learning, Encryption, Data security, Multi-Party computation, Scalability, Privacy.*

[‡] *E-mail:* **raja.lece@sece.ac.in**
[@] *E-mail:* **pankajdadheech777@gmail.com**
[#] *E-mail:* **sudhasengan@gmail.com**

1. Introduction

The invention of Deep Learning (DL) has triggered an explosion in several sectors, driving Artificial Intelligence (AI) research to new forms of intelligence. Access to vast volumes of data is vital to these successful models. Still, privacy and security are significant alarms, mainly if the data initiates from multiple resources that are not all in the exact location. Information security breaches and misuse of confidential data are two significant risks related to the traditional hierarchical method of training models. Finding methods that are proficient in the achievement of both data providers' severe privacy and the demanding data requirements of DL algorithms is, thus, a significant challenge. Currently, available methods ensure a certain level of data protection—For instance, using anonymization of data and asymmetrical privacy—but this security is frequently compromised for the purpose of the precision of models.

Additionally, scenarios involving multiple users working together while securing their private data from exposure to one another or a central server might not be appropriate to the methods mentioned previously. Secure Multi-Party Computation (SMPC) presents a method for this test by describing an architecture that makes it possible for users to work together to execute computations on their assumed data secretly. The implementation of SMPC in DL does, however, present some challenges. Standard SMPC solutions are impractical for large-scale transmission of information tasks due to the cost of computation triggered by sophisticated cryptographic procedures and the cost-effectiveness of distributed learning. The present article introduces a novel SMPC system framework that uses the Paillier Homomorphic Encryption Scheme (PHES) to secure personally identifiable information while ensuring multi-party data input, thus solving the mathematical and security challenges in networked DL. The model supports a decentralized framework that avoids a Trusted Authority (TA) requirement. Data security is provided by advanced SMPC protocols and cryptography methods.

This comprehensive methodology ensures that the mathematical model meets without exposing raw data by including data setup, encryption, secure server aggregation, and decryption. In order to solve the issues of SMPC and the use of encryption, the approach's functional time and effectiveness are optimized, enabling a solution that is both conceptually reliable and robust in practice. Researchers concluded that our SMPC approach maintains its precision on par with comparable approaches and controls confidentiality effectively across different input

scales in thorough assessments using Neural Network (NN) models on the CIFAR-10 and IMDB datasets. The proposed method has contributed a vital improvement to privacy-preserving DL with its empirical results, which verify its basic principles and prove its scalability and practical use. The paper is organized as follows: the literature reviewed is presented in Section 2, the problem is defined in Section 3, Section 4 presents the system model, Section 5 presents the experiment analysis, and Section 6 concludes the work.

2. Literature Review

Research in the past few decades focused on SMPC's potential collaboration with DL, with multiple research projects addressing numerous issues and suggesting new approaches [1]. [2] exploited SMPC to develop a privacy-preserving technique for cloud-based Machine Learning (ML) applications. Applying the PyTorch and PySyft library resources, they conducted studies [3] that improved accuracy and data security on the dataset provided by MNIST, and they utilized secret sharing to address problems with confidentiality in the data they used for training [4]. The lack of scalable software platforms is a contributing factor to why [5] discovered a void in the overview of secure MPC in ML. Authors suggested CrypTen, an approach for ML investigators, to help close this area of research by permitting private, efficient evaluation of state-of-the-art models [6] employing a semi-honest attack model. The researchers utilized text classification and speech recognition as instances of tests that showed [7] how CrypTen can effectively use GPU functionality to allow confidential analysis. [8] projected a framework for privacy-preserving multi-party DL in cloud computing, focusing on collaborative learning while protecting data privacy against the cloud server. Their system [9], applicable to privacy-sensitive areas, verified manageable computational efficiency and met verifiability and privacy security requirements. [10] addressed the limitations of Federated Learning (FL) regarding data privacy through a protocol combining differential privacy and SMPC. They applied this to logistic regression on a real-world credit card fraud dataset, demonstrating the potential for more accurate models without exposing sensitive client data. Lastly, [11] tackled the network-based performance limitations of SMPC by proposing a pipelining-like approach for each round's computation and communication in DL applications [12-13]. The study reveals that matrix multiplication in distributed cloud deployments significantly reduces execution time, highlighting the

evolving landscape of SMPC in distributed computing and methods to improve privacy, efficiency, and practicality.

3. Problem Definition

The problem statement emphasizes developing a privacy-preserving computational model in a scenario where 'n' distinct parties, each possessing a dataset 'D_i' comprising pairs (x_i, y_i), engage in a collective training process. Here, 'x_i' represents the input features, and 'yi' represents the corresponding labels. The primary mathematical challenge is to compute a global model function '$F(D)$'- akin to training a neural network so that no information about any individual dataset 'D_i', other than what can be deduced from the output of 'F' is disclosed. Additionally, the aim is to optimize the parameters θ of the NN to minimize the loss function $L(\theta;D)$, ensuring that the model's accuracy is on par with that of a model trained on a centrally aggregated dataset. This optimization must occur under the stringent condition of maintaining data privacy [11-12]. Another critical aspect is computational efficiency. The processes, including the SMPC protocols for encryption and decryption, must be efficient. We define the total computation time as $T = f(C_{smpc}, n, |D|)$, where C_{smpc} denotes the computational complexity of the SMPC operations, n represents the number of parties, and $|D|$ is the cumulative size of the datasets involved

4. System Model

In the system model that applies SMPC to DL for securing data privacy in distributed networks, 'n' parties exist. Each holds a confidential dataset 'D_i' for $i=1,2,..,n$. Within every dataset 'D_i', there are data points (x_i, y_i) with 'x_i' representing the input features and 'y_i' as the corresponding labels; the collective goal is to train a neural network model collaboratively while maintaining the confidentiality of each dataset. The parties include clients and a server, with the clients tasked to encrypt their datasets using a secure encryption function. For enhanced specificity, the clients employ Paillier encryption, a form of homomorphic encryption that allows computations to be carried out on ciphertexts and obtain an encrypted result that, when decrypted, matches the result of operations performed on the plaintext. This results in $E(D_i)$ for each dataset, which is securely transmitted to the server. The server coordinates the SMPC process, managing the encrypted data without needing a Trusted Authority (TA)

[13]. The system's decentralized nature is reinforced by the absence of robust SMPC protocols and encryption mechanisms like Paillier, which ensure data security. The NN's training process is optimized for encrypted data, using SMPC techniques instead of direct loss function minimization for encrypted calculations. Aggregating encrypted updates allows for the optimization of the model parameters θ indirectly. Regarding computational efficiency, the model is tuned to optimize operation time, acknowledging the inherent complexity of 'C_{smp}', which was added by SMPC and Paillier encryption. The computation time 'T' is thus formulated to reflect the advanced cryptographic operations, ensuring the process remains tractable for the number of participating parties n and the aggregate size of the datasets $|D|$.

The system model workflow is presented in the following steps:

Step 1. **Initialization:** Each of the n clients initializes their local model using their private dataset D_i, which contains data points x_i, y_i

Step 2. **Local Encryption:** Clients encrypt their datasets using the Paillier homomorphic encryption function, resulting in encrypted datasets $E(D_i)$.

Step 3. **Secure Upload:** The encrypted datasets are securely transmitted to the server, ensuring that the raw data remains confidential.

Step 4. **Encrypted Aggregation:** The server aggregates the encrypted data from all clients using homomorphic properties that allow for computations on ciphertexts.

Step 5. **Computation on Encrypted Data:** The server performs model training computations on aggregated encrypted data, using Paillier encryption's homomorphic addition for additive operations for training.

Step 6. **Decryption of Results:** Once the server has completed the computations, the results (still in encrypted form) are returned to the clients.

Step 7. **Local Decryption and Model Update:** Clients decrypt the results using their private keys and update their local models accordingly.

Step 8. **Iteration:** Steps 2-7 are repeated for several iterations until the model converges or a predefined stopping criterion is met.

Step 9. **Model Convergence:** Once the model has converged, clients can use their updated, locally held model to make predictions or refine it as needed.

5. Experimental Analysis

The empirical investigation of the SMPC framework in DL is performed by utilizing two completely separate neural network frameworks, each optimized for an entirely distinct type of data. Considering the CIFAR-10 dataset—which includes 60,000 32x32 color images over ten classes—the initial model is the LeNet-5 Convolutional Neural Network (CNN), which is appropriate for image classification tasks. The result is an entire review of CNN's image processing techniques. Additionally, there is an RNN model termed a Long Short-Term Memory (LSTM) network, which is suitable for analyzing information consecutively. An enormous amount of reviews of films for sentiment analysis constitutes a component of the IMDB movie reviews dataset, namely the LSTM dataset. GPUs with the NVIDIA GeForce RTX 2080 Ti platform are used in the testing setup. Intel core i7 processor and 32GB of RAM to efficiently manage computational demands, and TensorFlow 2x is chosen as the DL method. The evaluation was designed to measure three critical aspects, scalability, privacy, and learning efficiency, against three models from the literature Model 1 [Ref. [2]], Model 2 [Ref. [3]] and Model 3 [Ref. [4]]. Further, the time complexity of the encryption-decryption on the client side and the server-side aggregation for the two datasets were also evaluated.

The results for the CIFAR-10 dataset in Figure 1 show that the SMPC-LeNET model consistently increases accuracy with more participants, peaking at 93.03% with N=45, suggesting the SMPC's scalability. Other

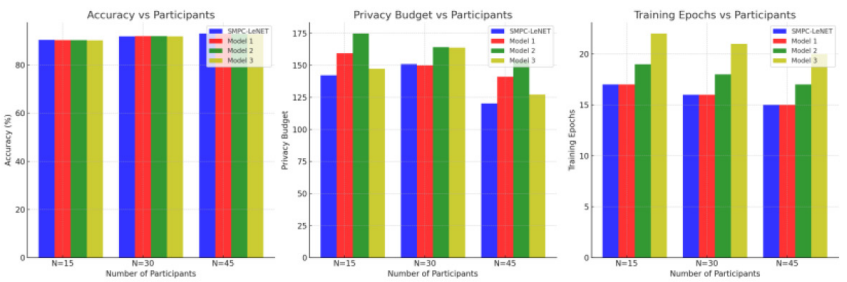

Figure 1

Analysis of the CIFAR-10 dataset

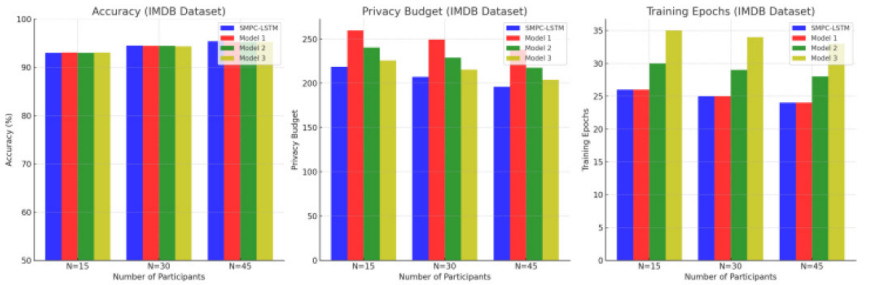

Figure 2

Analysis of the IMDB dataset

models also show a positive trend in accuracy as the number of users grows, with Model 2 incurring the highest privacy cost across all sizes. Intriguingly, SMPC-LeNET's privacy budget increases at N=30 before dropping at N=45, suggesting a non-linear relationship with the number of users. Training epochs for SMPC-LeNET decrease with more users, indicating enhanced learning efficiency, while Models 2 and 3 also require fewer epochs as contribution increases, except for Model 1, which remains constant. Model 3 consistently needs more epochs, likely reflecting a more complex model. These patterns underscore the benefits of increased participation in the SMPC framework and point to an intricate interplay between privacy cost and participant numbers.

The IMDB dataset results (Figure 2) for the SMPC system utilizing LSTM models indicate a clear trend of increasing accuracy and decreasing privacy costs with growing users (N=15 to N=45). The SMPC-LSTM model shows a steady rise in accuracy from 93.02% to 95.39%, closely followed by the other three models, suggesting that all models benefit similarly from larger datasets. Notably, Model 1 starts with slightly higher accuracy at N=15 but is marginally outperformed by SMPC-LSTM at N=45. The privacy cost, reflecting the amount of noise added for privacy, decreases across all models as user numbers increase. SMPC-LSTM starts with the lowest cost and maintains this trend, indicating a more efficient privacy-to-accuracy trade-off. Training epochs decrease uniformly for SMPC-LSTM and Model 1, suggesting an improvement in learning efficiency as the number of users increases, while Models 2 and 3, requiring more epochs overall, exhibit a gradual decrease, pointing towards a higher computational complexity or potentially more robust learning with added data. This analysis underscores the effectiveness of SMPC-LSTM in handling privacy and learning efficiency at scale within the SMPC framework.

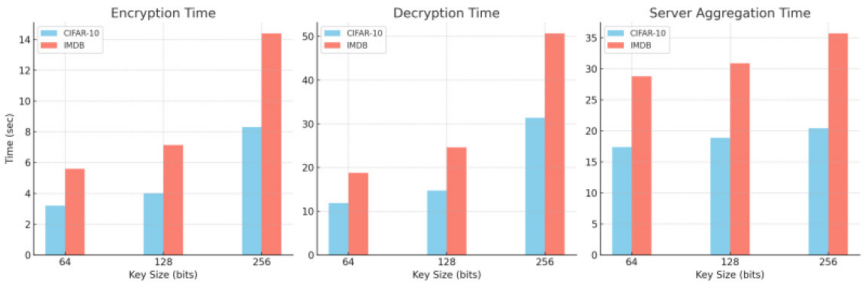

Figure 3

Time complexity analysis

The PHES performance analysis for CIFAR-10 and IMDB datasets reveals a direct correlation between key sizes and the time required for encryption and decryption processes (Figure 3). As key sizes escalate from 64 to 256 bits, encryption times more than double, and decryption times nearly triple, highlighting a significant computational cost. Although linearly dependent on the number of users, server aggregation times exhibit a more minor yet consistent increase. For instance, CIFAR-10 aggregation time increases from 0.58xN to 0.68xN. At the same time, the IMDB dataset sees a more marked increase from 0.96xN to 1.19xN, recommending that the IMDB dataset's complexity or volume may affect the computational overhead. Higher keys, which improve security but enhance the time required for processing, have impacted SMPC operations.

6. Conclusion and Future Work

The study ends with the outline of a new Secure Multi-Party Computation (SMPC) methodology that is appropriate for Deep Learning (DL) applications that involve highly confidential contexts. The structure provides for the collaborative training of neural network models while securing private information employing the Paillier Homomorphic Encryption Scheme (PHES). The suggested architecture ensures robust data privacy since it does not use a Trusted Authority (TA) and employs reliable cryptographic protocols. Multiple tests on CIFAR-10 and IMDB datasets demonstrated the security of the entire system model, involving encrypted data aggregation and decryption. These results indicate that the approach is both efficient and scalable and that it produces levels of precision that are on the same level as standard DL models. The SMPC approach achieves a great compromise between the security of data and

quantitative performance, which is how these results highlight its efficacy as a revolutionary method for secure ML.

In the future, investigators are going to attempt to make cryptography and decryption simpler, make methods that function with numerous datasets and DL systems, and make these mathematical models more accessible to use in real-world applications.

References

[1] Sayyad, S., Privacy-preserving deep learning using secure multi-party computation. 2nd IEEE International Conference on Inventive Research in Computing Applications, p. 139-142 (2020).

[2] B. Knott et al., Crypten: Secure multi-party computation meets machine learning—advances in Neural Information Processing Systems, 34, 4961-4973 (2021).

[3] Sharma, Sandeep Kumar, Kumar, Anil, Ashtagi, Rashmi & Jain, Rekha. OCA: An intelligent model for improving security breach of biometric based authentication systems, *Journal of Discrete Mathematical Sciences and Cryptography*, 26:5, 1415–1425 (2023), DOI: 10.47974/JDMSC-1765.

[4] Byrd, D., & Polychroniadou, A. Differentially private, secure multi-party computation for federated learning in financial applications. In Proceedings of the 1st ACM International Conference on AI in Finance, pp. 1-9 (2020).

[5] Bautista, O. G., & Akkaya, K. Network-efficient pipelining-based secure multiparty computation for machine learning applications. IEEE 47th Conference on Local Computer Networks (LCN), pp. 205-213 (2022).

[6] E. Marquet, J. Moeyersons, E. Pohle, M. V. Kenhove, A. Abidin and B. Volckaert, "Secure Key Management for Multi-Party Computation in MOZAIK," IEEE European Symposium on Security and Privacy Workshops, Delft, Netherlands, pp. 133-140 (2023).

[7] S. P. Sanon, R. Reddy, C. Lipps and H. D. Schotten, "Secure Federated Learning: An Evaluation of Homomorphic Encrypted Network Traffic Prediction," IEEE 20th Consumer Communications & Networking Conference, Las Vegas, NV, USA, pp. 1-6 (2023).

[8] C. Dong et al., "Maliciously Secure and Efficient Large-Scale Genome-Wide Association Study With Multi-Party Computation,"

IEEE Transactions on Dependable and Secure Computing, vol. 20, no. 2, pp. 1243-1257 (2023).

[9] B. Rajalakshmi, "Exploring Cryptographic Paradigms for Secure Cloud Computing," 2nd International Conference on Augmented Intelligence and Sustainable Systems, Trichy, India, pp. 1290-1294 (2023).

[10] K. Iwamura and A. A. A. M. Kamal, "Communication-Efficient Secure Computation of Encrypted Inputs Using (k, n) Threshold Secret Sharing," IEEE Access, vol. 11, pp. 51166-51184 (2023).

[11] A. V. Kumar, K. Monica, and K. Mandadi, "Data Privacy Over Cloud Computing using Multi-Party Computation," International Conference on Intelligent Data Communication Technologies and Internet of Things (IDCIoT), Bengaluru, India, pp. 262-267 (2023).

[12] W. Ruan, M. Xu, W. Fang, L. Wang, L. Wang, and W. Han, "Private, Efficient, and Accurate: Protecting Models Trained by Multi-party Learning with Differential Privacy," 2023 IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA, pp. 1926-1943 (2023).

[13] W. J. Liu and Z. X. Li, "Secure and Efficient Two-Party Quantum Scalar Product Protocol With Application to Privacy-Preserving Matrix Multiplication," IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 70, no. 11, pp. 4456-4469 (2023).