Online at: https://researchtrendsjournal.com **ISSN No:** 2584-282X
Indexed Journal **Peer Reviewed Journal**

INTERNATIONAL JOURNAL OF TRENDS IN EMERGING RESEARCH AND DEVELOPMENT

Volume 3; Issue 1; 2025; Page No. 232-238

Received: 22-10-2024
Accepted: 30-11-2024

A Privacy-Preserving Federated Collaborative Filtering Framework for Secure Recommender Systems

[1]Vishal Trivedi and [2]Dr. Sunil Bhutoda

[1]Research Scholar, P.K. University, Shivpuri, Madhya Pradesh, India
[2]Professor, P.K. University, Shivpuri, Madhya Pradesh, India

DOI: https://doi.org/10.5281/zenodo.18414994

Corresponding Author: Vishal Trivedi

Abstract

This paper aims to enhance FL performance while protecting privacy by reducing noise during anonymization, using feature selection for dimensionality reduction, and generalizing data. Developing a predictive model for healthcare applications is the goal of this study, which also examines feature-based data separation rather than record-based data separation and assesses the suggested model's effectiveness using actual healthcare data. A recommender system is typically tailored to provide useful and effective suggestions for a specific type of item, such as CDs or news. Its design, graphical user interface, and core recommendation technique are all adjusted to cater to this specific type of item. Product recommendations, movie and TV program suggestions, article suggestions, and countless more examples are among the most common and significant use cases.

Keywords: Performance, recommender, integrates, learning and privacy-enhanced

1. Introduction

Integrates meta learning into a federated learning framework; meta learning is a paradigm in machine learning in which an AI at the highest level improves an AI at the lowest level. Clients train the model locally, while the server maintains a shared model by collecting locally updated models. Recommendations for users of edge devices are made using this approach. Their proposal, FedMeta, is based on. On an industrial recommendation assignment, for example, they beat both standalone models and those trained using a standard federated learning strategy [1].

Our study is motivated by the pioneering work in the field of federated collaborative filtering, which was published in. A collaborative filtering recommendation system that mimics the way a user engages with a collection of objects is trained using deep learning models. Local user updates occur on client devices, while the server changes the item factor vectors and distributes them to each client. For model updates, they employ gradient descent and alternating least squares. For their Federated Collaborative Filter, they use Adaptive Moment Estimation (Adam). We run the test using MovieLens' dataset of 25 million movies [8].

The federated collaborative filtering system proposed in takes inspiration from. Its approach is similar, but it differs from the Reptile meta-learning algorithm in that it does not necessitate sequential execution of the edge devices to update the central parameters or communication between them. Rather, training occurs simultaneously for all devices because they connect with the parameter server independently. For the purpose of considering the data's multi-view structure, a matrix factorization algorithm is presented [5].

For privacy-enhanced recommender systems, which do not allow service providers to learn any user's preferences, frameworks to integrate additively homomorphic encryption with recommender systems were originally introduced. The suggestion was made for a centralized system with the intention of creating suggestions using encrypted processing. Two years later, the same writers offered a more efficient method for achieving this [2]. They also published a paper that year that used homomorphic encryption to safely conduct content-based filtering. In a more recent study, homomorphic encryption was used to create a healthcare recommender system that prioritizes patient privacy via collaborative filtering. Additionally, a user-based recommender system that prioritizes patient privacy was proposed [6].

2. Literature Survey

Ankur Bansal et al. (2021) suggests a new privacy-preserving approach for neural network learning on partitioned datasets as a means to establish safe communication between participants. When training on a vertically partitioned dataset, the secure scalar product technique provides a safe method for neural networks to learn. In order to integrate several safe constructions for both horizontally and vertically partitioned data, two methods exist: the stochastic approach and the least squares technique.

Saeed Samat and Ali Miri (2022) [1] state that the unique algorithm is exceedingly safe and that no information about other parties is leaked. Updated privacy-protecting techniques are used by neural network learning systems like extreme learning machines and back-propagation. In circumstances involving several parties, these protocols are able to split the data both horizontally and vertically. So that no one can access the training data, the weight vector is shared between them. Predicting the output of the target data may involve many people working together.

Methods that carry out two processes, such as selecting private neighbors and perturbing them, are known as private neighbor collaborative filtering algorithms (Tianqing Zhua et al., 2014) [2]. Differential privacy is the foundation upon which private neighbor selection is built. That is, the target data's neighbors are selected secretly according to their commonalities with others. This technique improves privacy protection while simultaneously reducing the quantity of noise. Xinyu Yang et al. (2015) [2] noted that lightweight apps based on temporal perturbations may also use privacy preservation to save sensor readings and modify them in a way that complicates the data's temporal information [5].

User k-anonymity was addressed by Guillermo Navarro Arribas et al. (2022) via the anonymization of query logs. Previous to the release of such sensitive information, it is essential to anonymize querylogs. It safeguards the data usefulness and user profile while ensuring that users' k-anonymity in the query log is preserved. The actual query logs are subjected to this suggested method. Methods for data mining based on clusters analyze the information.

A web-based system for discrimination prevention, categorization, and privacy protection was presented by Sant Gadge (2016) [3]. It does data discretization, classification, rule generalization, rule protection, and data preparation. Automatic decision-making is possible with the use of rule mining techniques for categorization and association. When deciding between candidates based on sensitive or non-sensitive traits, discrimination may be done either directly or indirectly. Machine learning's high dimensionality challenges are addressed by feature selection. When dealing with massive datasets, this is a necessary pre-processing step. From the data collection, it extracts just the characteristics that are informative. From the dataset, it extracts only the most important attributes.

3. Research Methodology

As previously mentioned, the model will be conducting a federated collaborative filtering through matrix decomposition with the added security of homomorphic encryption. The intuition behind this is based on decentralized matrix factorization where each user profile is updated locally and the recommendation profiles of items are aggregated and updated by the server. The item recommendation gradients are encrypted by homomorphic encryption and the parameter server then aggregates these gradients [8].

We first implement the case of a regular collaborative filtering system using single value decomposition, which is a matrix factorization method. Collaborative filtering relies on user similarity and the baseline assumption being made is that users that are similar to each other in terms of movies they have previously rated, will be similar in the movies they will be rating in the future. This way, we're making filtering decisions for individual users based on the preferences and judgements of other users. If there are n users and m items, and each user rated a subset of m, M = n x m would be the matrix denoting user-item rating pairs and jMj would be the total number of ratings. User-profile and item-profile matrices are denoted by U and V respectively where U is the user profile matrix which consists of the left singular vectors with n users and d concepts for movie, giving it a dimension of n x d. V is the movie profile matrix consisting of the right singular vectors with m movies and d concepts, making it a m x d dimension matrix. d can be any number between 1 and the smaller number among n and m [11].

The movies and users are mapped into a 3D space where they are represented by points on it. The afixes of this latent subspace are called factors. Through this mapping, we find low dimensional representations of users and movies such that the users that like these movies are closer to each other in this latent subspace.

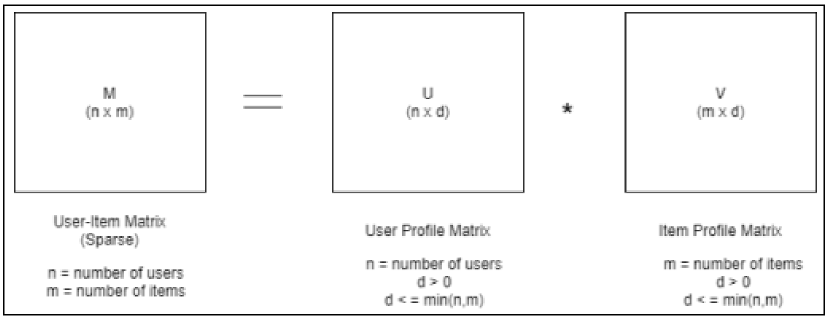

Fig 1: The matrix decomposition

Typically, a user only interacts with a small subset of the total number of movies, and the resulting matrix is therefore largely of a sparse nature. These matrices are used to predict user i's rating for item j, which is < u_i; v_j >. This can be posed as an optimization problem:

$$\min_{U,V} = \frac{1}{|M|}(r_{i,j} - < u_i, v_j >)^2 + \lambda||U||^2 + \mu||V||^2$$

where λ and μ are small positive values used for regularization purposes.

We solve this optimization problem by minimizing the loss function using Stochastic Gradient Descent with the following update steps:

$$u_i^t = u_i^{t-1} - \eta * (-2 * v_j * (r_{i,j} - < u_i, v_j >) + 2 * \lambda u_i$$

$$v_j^t = v_j^{t-1} - \eta * (-2 * u_i * (r_{i,j} - < u_i, v_j >) + 2 * \lambda v_j$$

Then we build a federated scenario where rating information is stored on user's local devices and the model is trained on joint data and the iterative updating of gradients is split in two parts that are performed on the client side and server side. The stochastic gradient descent for user-pro le matrix U, (4.3) is performed on each user's device (the local update) while the stochastic gradient descent for item-pro le matrix V, (4.4) is performed on the parameter server (the global update). The separation of these processes and the resulting decomposition of the matrix prevents the server from directly having raw data available to it. The diagram below shows how the federated collaborative filtering done through this matrix decomposition has been set up [6].

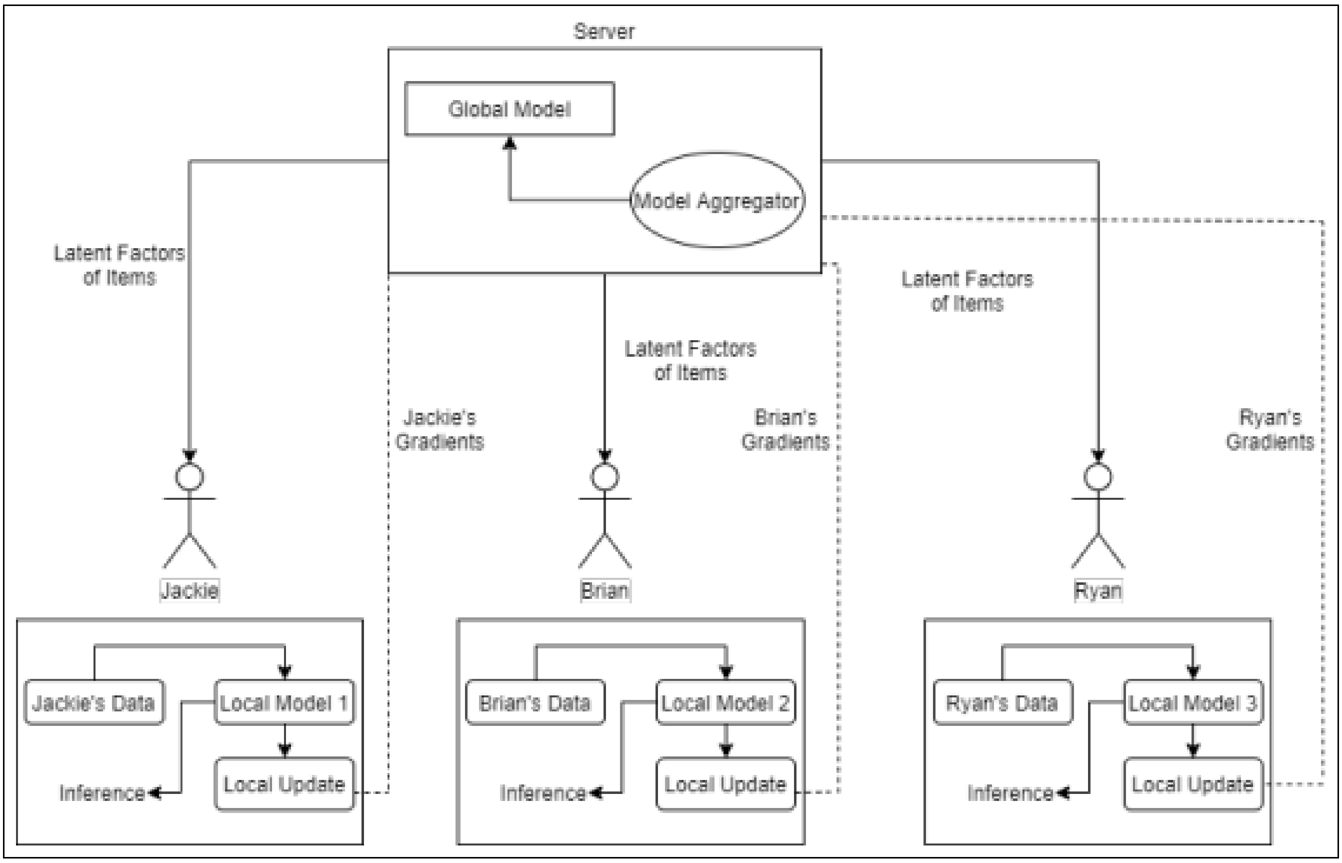

Fig 2: Federated collaborative filtering

The latent factors of items are sent from the server to the users to download, following which the users begin their local computation of gradients. These gradients are then sent back to the server for the next global updation of item factors [15].

We then enrich this algorithm with encryption whereby gradients are communicated safely in the manner de need by Fig 3 and operated on using partially homomorphic encryption. This addresses the problem of gradient leakage. To implement homomorphic encryption, we used Python's Paillier encryption protocol (https://github.com/data61/python-paillier) with their recommended library gmpy2 being used to accelerate the encryption process. The diagram for this combined process is as follows:

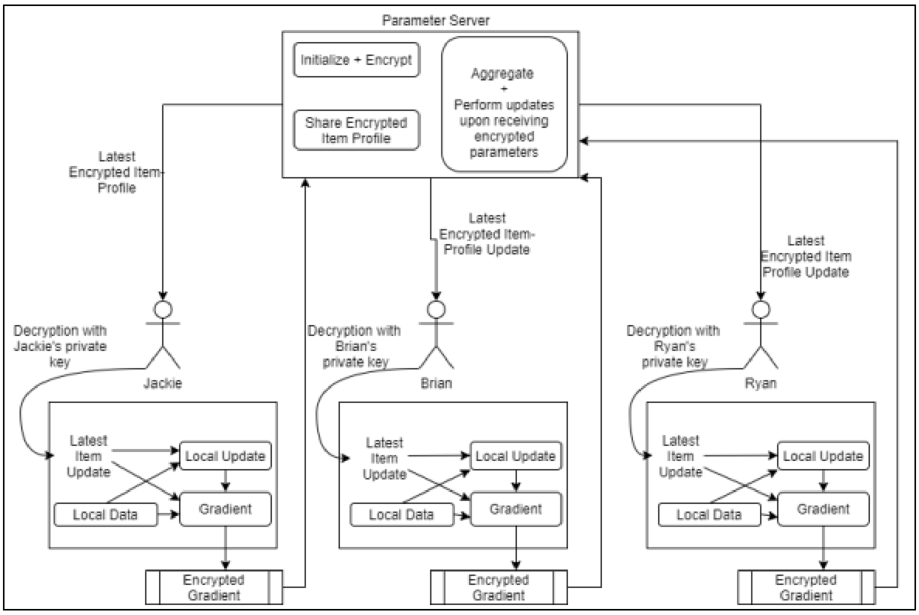
Parameter Server
Initialize + Encrypt
Share Encrypted Item Profile
Aggregate + Perform updates upon receiving encrypted parameters
Latest Encrypted Item Profile
Latest Encrypted Item-Profile Update
Latest Encrypted Item Profile Update
Decryption with Jackie's private key
Jackie
Decryption with Brian's private key
Brian
Decryption with Ryan's private key
Ryan
Latest Item Update
Local Update
Local Data
Gradient
Latest Item Update
Local Update
Local Data
Gradient
Latest Item Update
Local Update
Local Data
Gradient
Encrypted Gradient
Encrypted Gradient
Encrypted Gradient

Fig 3: Federated collaborative filtering with encryption

4. Results

Table 1: The Latency and MSE Loss Outcomes about Various Secure Nodes and Model Owners presents the condensed findings of the test, which assesses the correctness and efficacy of the approach across various combinations of secure nodes and model owners [7].

Table 1: The Latency and MSE Loss Outcomes about Various Secure Nodes and Model Owners

HW	No. of SN	No. of MO	Performance (sec.)	Accuracy/Loss
2vCPU, 8 GB RAM	2	5	0.206405878	20.31747246
		10	0.166790009	20.18802261
		15	0.18239975	20.30132675
		20	0.180090189	20.24121284
	3	5	0.623690128	20.1605072
		10	0.643671751	20.20030594
		15	0.633473635	20.30055428
		20	0.613318205	20.23125458
	4	5	1.142184496	20.22513199
		10	1.15519762	20.28224564
		15	1.156814575	20.25988197
		20	1.150216818	20.21783066
	5	5	1.788914204	20.20358849
		10	1.991049051	20.2536335
		15	1.904865742	20.30569077
		20		
	6	5	2.572024107	20.25515747
		10	2.58530283	20.25584221
		15	2.640356779	20.24275398
		20	2.559947968	20.24914742
	7	5	3.520589828	20.23002434
		10	3.668842077	20.29376221
		15	3.531092882	20.21783066
		20		
	8	5	3.435461998	20.20701218
		10	3.57885313	20.19074821
		15	3.466508627	20.26420021
		20	3.486571312	20.26704407
	9	5	4.246810913	20.19329834
		10	4.306902647	20.25208664
		15	4.299203157	20.23276711
		20	4.311759949	20.31554985
	10	5	5.377040625	20.26992607
		10	5.529221296	20.27954102
		15	5.333623171	20.22979164
		20	5.429213047	20.23600006

4.1 Insights

4.1.1 Accuracy vs. Privacy

The experimental results indicate that after putting the model parameters and inference values through an SMPC cluster across 2 to 10 secure nodes, the overall output produced the mean squared error (MSE) loss within the range of 20-21, benchmarked against the non-private MSE also in the same range. Hence it can be safely ascertained that the method preserves privacy while not affecting accuracy [14].

4.1.2 Model vs. Accuracy

The experimental findings indicate that an increase in the number of models led to a minimal Mean Squared Error (MSE) loss, with a variance of no more than one point, consistently falling within the range of 20 to 21 which again is benchmarked against the non-private model accuracy in the same range. The mean squared error (MSE) loss shows a consistent value (as shown in Figure 4: Empirical comparison of varying secure nodes), indicating the overall correctness of the framework stays unaffected by the growth in the number of models. The nodes involved in secure multi-party computing can do computations on many models simultaneously while maintaining a high level of accuracy [9].

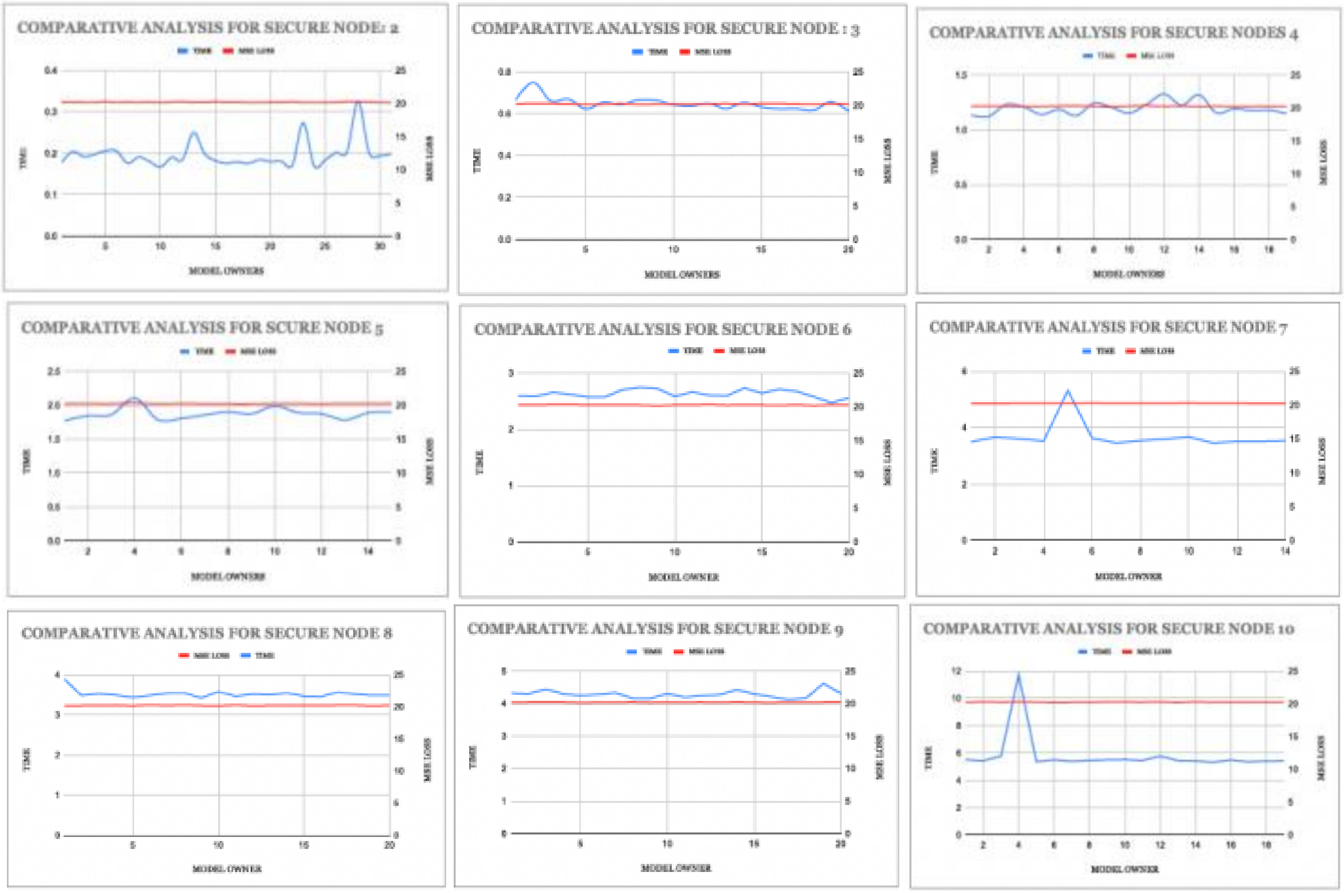

Fig 4: Empirical comparison of varying secure nodes

4.1.3 Secure Nodes vs. Accuracy

According to the findings presented *Secure node vs. Performance vs. MSE Loss*, which illustrates an empirical comparative examination of various quantities of secure nodes, the research saw no effect on mean squared error (MSE) loss as the number of safe nodes rose from 2 to 10. The duration of inference was influenced by increasing secure nodes, but the accuracy of the inference remained unaffected [13].

4.1.4 Privacy vs. Performance

The study demonstrated that the amount of time required for calculating inference with SMPC- based calculation grew steadily alongside the secure nodes, depicted *Secure node vs. Performance vs. MSE Loss*. The inference is computed at the secure nodes having a secret model share and later it is shared with the result aggregator which reassembles the inference outputs. The secure nodes operate in parallel, however, the time to inference increases with the number of secure nodes and this increase is proportional to the number of secure nodes [12].

4.1.5 Model vs. Performance

As shown in Figure 5: Secure node vs. Performance vs. MSE Loss, in contrast to secure nodes, the expansion in the models is not disruptive to the calculation time. In a fixed and secure node environment, increasing the number of models does not affect the inference time.

It is a useful boost to the proposal method, and additional models can be incorporated to improve the accuracy further without paying any price for performance and accuracy. Conversely, the models are kept confidential, and the proposed method does not affect the precision and efficacy of private inference [11].

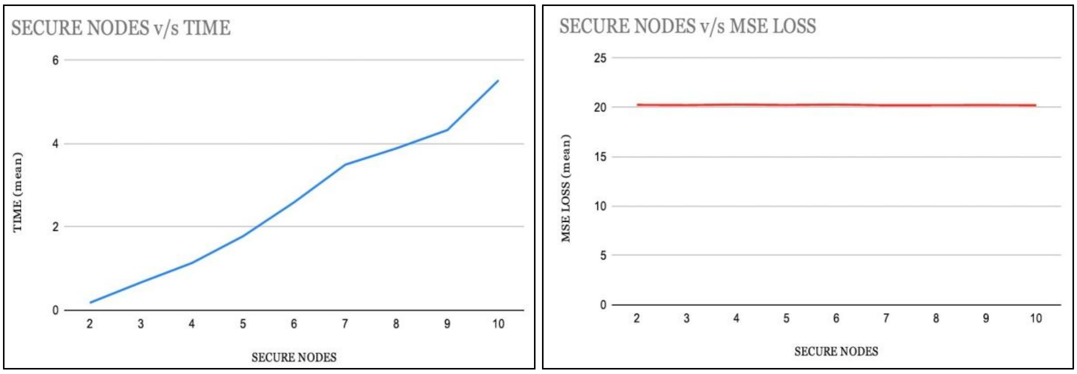

Fig 5: Secure node vs. Performance vs. MSE Loss

4.1.6 Secure Nodes vs. Performance

Experiments revealed that private inference efficacy impacted linearly with the rise in the number of secure nodes. Additionally, as illustrated in Figure 5: Secure node vs. Performance vs. MSE Loss, the extra delay in a radius of a few seconds, such as 16 seconds inference time with 17 secure nodes. In other words, a second additional performance price for every new secure node in the SMPC cluster. Which means a linear price to pay for adding more privacy. Conversely, the decrease in efficacy can be mitigated by employing specialised hardware like a faster CPU, GPU, and RAM. However, growing the number of secure nodes additionally improves the confidentiality of models and inference input primarily because the odds of privacy leakage reduce with increasing secure nodes in the SMPC cluster [4].

4.1.7 Accuracy vs. Performance

Figure 5: Secure node vs. Performance vs. MSE Loss, clearly shows the outcomes of the experiment when accuracy stays unchanged whereas inference performance showed little impact of hardware configuration. The accuracy did not change with a higher set of hardware configurations and it remained in the range of 20-21 MSE Loss, although the inference time decreased with higher CPU and RAM configuration. Therefore, it can be determined that accuracy and performance have no relationship within CoInMPro. Conversely, accuracy is more dependent on the existence of several models. Performance relies upon the presence of nodes in the SMPC cluster.

5. Conclusion

This paper has been an interdisciplinary effort which explores Federated Learning, recommender Systems and cryptography. We started by giving an overview of Federated Learning and exploring the most pertinent works in the field as of early 2020, which is important to note because of the pace at which this very new domain is expanding. It then touches upon recommender systems and different cases of privacy that were the foundations of the novel work that was done here. An example of another such application would be a restaurant recommender based on Yelp dataset using Natural Language Processing and techniques. It would take user inputs based on the kind of food they are looking for and run it through the algorithm that parsed through the reviews, using it as a corpus and finally outpufitting suggestions. It is easy to imagine this process being made much more accurate with the local updates and improvements being made on the user's device, much like what Google did with its implementation of mobile keyboard prediction which is currently being used in GBoard.

6. References

1. Samet S, Miri A. Privacy-preserving back-propagation and extreme learning machine algorithms. IEEE Transactions on Knowledge and Data Engineering. 2022;79–80:40–61.
2. Yang X, Ren X, Yang S, McCann J. A novel temporal perturbation-based privacy-preserving scheme for real-time monitoring systems. Computer Networks. 2015;88:72–88.
3. Gadge S. Discrimination prevention with classification and privacy preservation in data mining. Procedia Computer Science. 2016;79:244–253.
4. Wang F, Liang J. An efficient feature selection algorithm for hybrid data. Neurocomputing. 2016;193:33–41.
5. Tian H, Zhang W, Xu S, Sharkey P. A knowledge model sharing–based approach to privacy-preserving data mining. ACM Transactions on Data Privacy. 2022;5(2):433–467.
6. Huang Z, Du W, Chen B. Deriving private information from randomized data. In: Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD); 2015.
7. Yu H, Kim J, Kim Y, Hwang S, Lee YH. An efficient method for learning nonlinear ranking SVM functions. Information Sciences. 2022;209:37–48.
8. Yakut I, Polat H. Arbitrarily distributed data-based recommendations with privacy. Data & Knowledge Engineering. 2022;72:239–256.
9. Khan I, Kulkarni A. Knowledge extraction from survey data using neural networks. Procedia Computer Science. 2023;20:433–438.
10. Vaidya J, Kantarcioglu M, Clifton C. Privacy-preserving Naïve Bayes classification. The VLDB Journal. 2018;17(4):879–898.
11. Li J, Lu K, Bates PW. Geometric singular perturbation theory with real noise. Journal of Differential Equations. 2015;259(10):5137–5167.
12. Miao J, Niu L. A survey on feature selection. Procedia Computer Science. 2016;91:919–926.
13. Xiao J, Cao H, Jiang X, Gu X, Xie L. GMDH-based semi-supervised feature selection for customer

classification. Knowledge-Based Systems. 2017;132:236–248.

14. Bhaduri K, Stefanski MD, Srivastava AN. Privacy-preserving outlier detection through random nonlinear data distortion. IEEE Transactions on Systems, Man, and Cybernetics. 2021;41(1):260–272.
15. Kantarcioglu M, Jiang W. Incentive-compatible privacy-preserving data analysis. IEEE Transactions on Knowledge and Data Engineering. 2023;25(6):1323–1335.