EdgeLeakage: Membership Information Leakage in Distributed Edge Intelligence Systems

Kongyang Chen, Yi Lin, Hui Luo, Bing Mi, Yatie Xiao, Chao Ma, and Jorge Sá Silva

***Abstract*—In contemporary edge computing systems, decentralized edge nodes aggregate unprocessed data and facilitate data analytics to uphold low transmission latency and real-time data processing capabilities. Recently, these edge nodes have evolved to facilitate the implementation of distributed machine learning models, utilizing their computational resources to enable intelligent decision-making, thereby giving rise to an emerging domain referred to as edge intelligence. However, within the realm of edge intelligence, susceptibility to numerous security and privacy threats against machine learning models becomes evident. This paper addresses the issue of membership inference leakage in distributed edge intelligence systems. Specifically, our focus is on an autonomous scenario wherein edge nodes collaboratively generate a global model. The utilization of membership inference attacks serves to elucidate the potential data leakage in this particular context. Furthermore, we delve into the examination of several defense mechanisms aimed at mitigating the aforementioned data leakage problem. Experimental results affirm that our approach is effective in detecting data leakage within edge intelligence systems, and the implementation of our defense methods proves instrumental in alleviating this security threat. Consequently, our findings contribute to safeguarding data privacy in the context of edge intelligence systems.**

***Index Terms*—Distributed Edge Intelligence, Membership Information Leakage, Data Privacy**

I. Introduction

In modern edge computing systems, distributed edge nodes collect raw information and provide data analytics to support low transmission latency and real-time data processing. Recently, edge nodes can provide distributed machine learning models with their available computation resources to support an intelligent decision making, inspiring an emerging area called edge intelligence. Traditional machine learning depends on a large amount of data samples to support its training, and it usually needs a central server for data collection, model training or aggregation, etc. Considering the critical privacy concerns, many organizations are not allowed to share their individual data, which thus significantly decreases the overall model accuracy. Therefore, Federated Learning (FL) is proposed to server as a novel distributed learning paradigm, where each participant client user keeps its own individual data locally, and only shares its model parameters (or gradient updates) to a centralized server for model aggregation [1]. However, existing solutions show that Federated learning still suffers from serious information leakage when the centralized server is attacked. With a remote central server, it is thus hard to build a connection-frequency model training such as Federated learning. To improve the privacy and security, each client has a chance to be chosen as a temporary server to aggregate model updates from participant clients.

In the realm of model security, it has been established that machine learning models face vulnerability to various model attacks, such as membership inference attacks [2], [3], [4], model inversion attacks [5], and property inference attacks [6]. These attacks have the potential to result in the leakage of sensitive information from the training dataset. For instance, a membership inference attack determines whether a given data sample was utilized in the previous model training process. Such knowledge is advantageous for adversaries seeking to exploit model security, posing potential severe ramifications, especially in the deployment of Machine Learning as a Service (MLaaS) [2]. Concerning the membership inference attack (MIA), Shokri et al.[7] introduced this attack against machine learning models employing shadow models in black-box scenarios, effectively transforming MIA into a binary-classification problem. Additionally, efforts have been made to address the cost associated with such attacks. Specifically, research by Yeom et al.[8] and Song and Mittal [9] delved into metric-based membership inference attacks, focusing on factors such as prediction confidence and prediction entropy, respectively, aiming to mitigate the attack's computational expense.

In this study, we investigate security threats within distributed edge intelligence systems. We specifically focus on employing membership inference attacks to elucidate potential data leakage, encompassing NN-based attacks, Metric-based attacks, and Differential attacks. Furthermore, we assess the performance of these attacks on diverse participant client users, with experimental results substantiating the existence of potential data leakage. Finally, we introduce several defense mechanisms aimed at preempting the aforementioned attacks. The principal contributions of our research are delineated as follows:

- We analyze the security model within distributed edge intelligence systems and demonstrate the integration of various membership inference attack methods, including NN-based attacks, Metric-based attacks, and Differential attacks. Additionally, we propose several defense strategies to counteract these attacks.
- Experimental findings validate the efficacy of our approach in detecting data leakage issues within edge intelligence systems, while also highlighting the utility of

K. Chen, Y. Lin, H. Luo, and Y. Xiao are with Guangzhou University, Guangzhou, China. K. Chen is also with Pazhou Lab, Guangzhou, China.

B. Mi is with Guangdong University of Finance and Economics, and also with Pazhou Lab, Guangzhou, China.

C. Ma is with Wuhan University, Wuhan, China.

J. Sá Silva is with University of Coimbra, INESC Coimbra, Portugal.

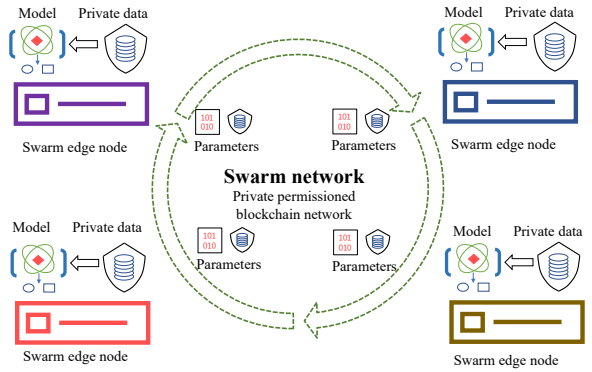


Fig. 1: Distributed Edge Intelligence Systems.

our defense mechanisms in mitigating this security threat.

The subsequent sections of this paper are structured as follows: Section II provides an overview of the framework underpinning our membership inference attacks. Section III presents the results obtained from our experiments. Section IV outlines several defense methodologies devised to combat these attacks. Section V delves into related research, while Section VI furnishes a comprehensive conclusion to our study.

II. Membership Inference Leakage against Distributed Edge Intelligence Systems

In this section, we present the system framework of our distributed edge intelligence system and propose our membership inference attack model against it.

A. Distributed Edge Intelligence Systems

As depicted in Figure 1, our distributed edge intelligence system distributes data samples across edge clients, with each client sharing and aggregating local model updates for joint training. Unlike centralized data aggregation methods such as federated learning, our approach does not necessitate a specific central server for parameter sharing and aggregation. Instead, each client user can function as a central client, selected before each aggregation through an internally determined server selection mechanism. This strategy mitigates various potential attacks and failures, including central client failures or privacy breaches. Additionally, our system leverages edge computing and blockchain technology to bolster transmission security. Given that edge nodes learn in a distributed manner without central server assistance, we refer to this approach as "swarm learning" for brevity.

B. MIA against Distributed Edge Intelligence Systems

In the membership inference attack (MIA) against our system, we consider the attacker to be one of the internal clients participating in our distributed edge system, with the objective of targeting other clients within the same system. It is assumed, without loss of generality, that the attacker has access to certain information about the distributed edge system, such as the model architecture. Consequently, both white-box and black-box attacks are viable within this context. We employ the membership inference attack technique described in [8], which utilizes a single shadow for conducting the attack.

For the attacking client, the shadow training set comprises its local training and test sets, while the shadow model corresponds to its local model. Thus, there is no necessity to train any additional model aside from the attack model. The architecture of our attack is illustrated in Figure 2. It is important to note that while the attacking client participates in the global model aggregation as a regular client, it also endeavors to gather local data information from other clients in a malicious manner.

We will introduce three membership inference attack against our distributed edge system In this following.

C. Attack 1: NN-based Attack

In this section, we train an attack model at the attacking client to target other clients. Illustrated in Figure 3, our refined shadow model attack architecture is depicted. To elucidate our attack methodology, we assume the presence of N clients in the distributed edge system, where only the last client (i.e., with client ID N) acts as the malicious attacking client. Thus, the objective is for the last client to target the first client, represented simply as N *attack 1*, or $N \rightarrow 1$. Furthermore, we assess the transmission of the attack across these N clients, with client i acting as the malicious attacker, where i ranges from 2 to $N-1$. These attack scenarios can be denoted as i *attack 1*, or $i \rightarrow 1$, where i ranges from 2 to $N-1$. Additionally, we utilize both balanced and unbalanced datasets to evaluate the attack performance of the attack model.

D. Attack 2: Metric-based Attack

In this experiment, we implement two metric-based attacks on our distributed edge system: attacks based on prediction confidence and attacks based on prediction entropy. Unlike the NN-based attacks discussed in Section II-C, these attacks rely on metrics that incur lower costs and overhead. As their names suggest, we train an attack model and observe the differences in predictions made by the model for various data samples. Specifically, we conduct membership inference attacks on our distributed edge system with 2, 3, and 4 clients.

We assess the effectiveness of metric-based attacks using the CIFAR-10 dataset, dividing it into $D_{n,SL}^{Train}$ and D_{SL}^{Test}. Here, $D^{Train}n, SL$ represents the training sets of client n, while all clients utilize the same test set D_{SL}^{Test}. Given that each client within the same edge framework employs identical model architectures, we utilize the model of the attacking client itself as the basis for measuring attack-based metrics. Consequently, we designate $D_{n,SL}^{Train}$ as positive samples (i.e., member labels) in the training sets for the attack model, and D_{SL}^{Test} as negative samples (i.e., non-member labels).

For this experiment, we employ ResNet50, a widely used model in image recognition, as the training model for our distributed edge clients. We conduct membership inference attacks based on prediction entropy and prediction confidence on our edge frameworks with 2, 3, and 4 clients. All experiments in this section are categorized into two classes for the attack model: distinguishing only between members and non-members.

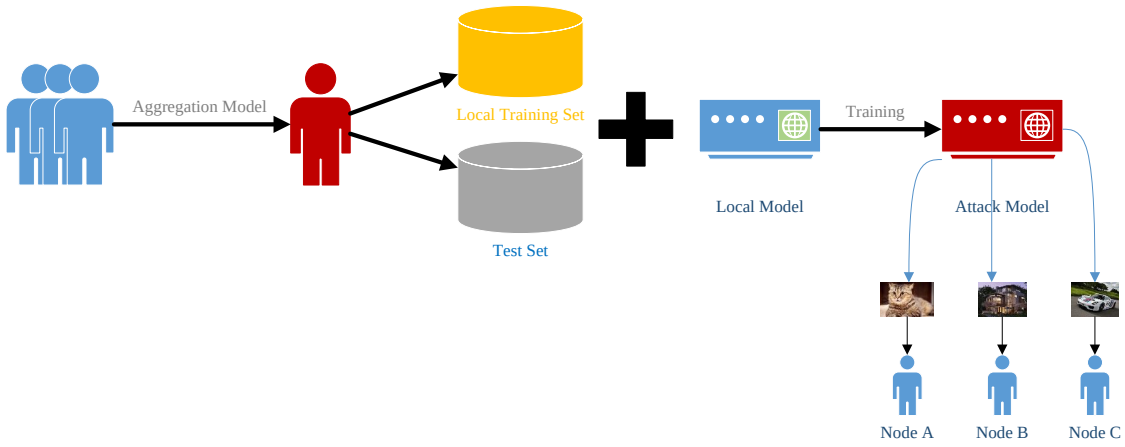


Fig. 2: Our Attack Architecture.

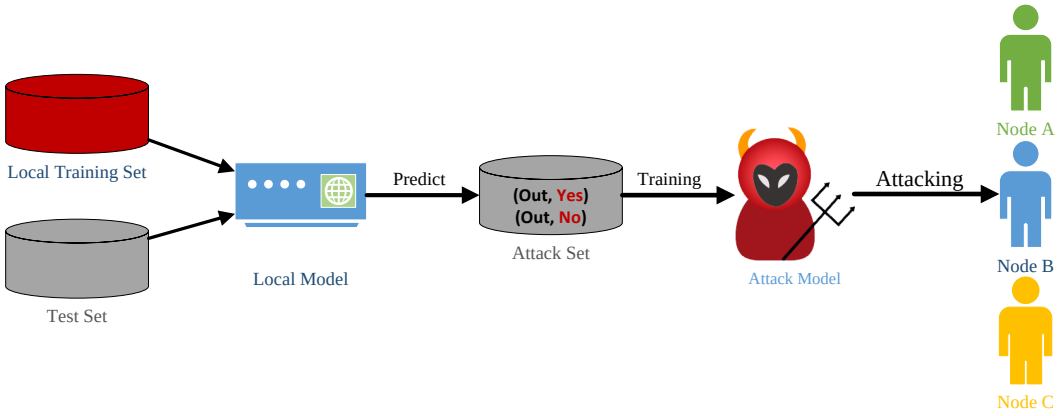


Fig. 3: Improved Shadow Model Attack Architecture.

E. Attack 3: Differential Attack

In the section, we introduce Maximum Mean Discrepancy (MMD), which is crucial for differential attacks. Furthermore, we present the dataset utilized in our attack.

The core concept behind differential attacks lies in Maximum Mean Discrepancy (MMD). MMD, introduced by Gretton et al. [10], is a smooth function that tests whether distributions p and q differ. A large value of this function suggests potential differences between the distributions. Mathematically, it is defined as:

$$F(D_{SL}^{Mem}, D_{SL}^{Nonmem}) = ||\frac{1}{n}\sum_{i=1}^{n_t}\phi(y_i) - \frac{1}{n_s}\sum_{j=1}^{n_s}\phi(y_j^{'})||_v$$

where $y_i \in D_{SL}^{Mem}$, $y_j^{'} \in D_{SL}^{Nonmem}$, n_t and n_s represent the sizes of D_{SL}^{Mem} and D_{SL}^{Nonmem} respectively, v denotes the dimension of the kernel space, and ϕ is a feature space map defined as $k \to v$. In our experiment, we employ the Gaussian kernel function $k(y, y') = \langle\phi(y), \phi(y')\rangle = exp(-\frac{||y-y'||}{2\sigma^2})$.

With the proliferation of defense methods such as Adversarial Regularization [11], MemGuard [12], and differential privacy [13], effectively acting on models, it becomes challenging to directly differentiate between members and non-members through the model prediction probability space. Hence, we map it to the Reproducing Kernel Hilbert Space (RKHS) [14] and then calculate the distance between two centroids in the kernel space.

Differential comparison, an idea applied to machine learning in recent years [15], has been introduced to membership inference attacks. In this experiment, we propose an enhanced differential attack against our distributed edge system. For the training datasets of distributed edge clients, we utilize both independently and non-independently identically distributed cases respectively to evaluate the impact of distribution variations on distributed edge privacy leaks.

The differential attack employs Maximum Mean Discrepancy to measure the characteristic distance between two groups of samples: one group closely resembling the training sample, and the other closely resembling the non-training sample. In the first variant of the differential attack, denoted as differential attack one, we iteratively add the target sample to D_{SL}^{Mem} and determine the target sample's membership by comparing the distance between D_{SL}^{Mem} and D_{SL}^{Nonmem}. Details of the differential attack are shown in Algorithm 1. Lines 1-5 of Algorithm 1 initialize some variables, while lines 6-14 represent the iterative calculation process for determining

Algorithm 1 Differential attack against our system.

Require: $D_{n,SL}^{Train}$, D_{SL}^{Test}, $D_{Diff,SL}^{Pre}$
Ensure: T_{Pre}^{Mem}, T_{Pre}^{Nonmem}
1: $D_{Diff,SL}^{Pre}$←empty
2: flag←n
3: **while** flag **do**
4: state←0
5: statev←0
6: **for** $D_{SL}^{Mem} \in D_{flag,SL}^{Train}$, $D_{SL}^{Nonmem} \in D_{SL}^{Test}$ **do**
7: $d' \leftarrow F(D_{SL}^{Mem}, D_{SL}^{Nonmem})$
8: **if** d' ¿ statev **then**
9: state←i
10: statev←d'
11: **end if**
12: **end for**
13: flag←flag-1
14: **end while**
15: **if** state¿0 **then**
16: $T_{Pre}^{Mem} \leftarrow (D_{Diff,SL}^{Pre}$, state)
17: **else**
18: $T_{Pre}^{Nonmem} \leftarrow D_{Diff,SL}^{Pre}$
19: **end if**

Algorithm 2 Differential attack against our system.

Require: $D_{n,SL}^{Train}$, D_{SL}^{Test}, $D_{Diff,SL}^{Pre}$
Ensure: $T_{Pre,i}^{Mem}$, T_{Pre}^{Nonmem}
1: $D_{Diff,SL}^{Pre}$←empty
2: flag←n
3: **while** flag **do**
4: state←0
5: statev←0
6: **for** $D_{SL1}^{Pre} \in D_{state,SL}^{Train}$, $D_{SL2}^{Pre} \in D_{flag,SL}^{Train}$ **do**
7: $d' \leftarrow F(D_{SL}^{Mem}, D_{SL}^{Nonmem})$
8: **if** d'¿statev **then**
9: state←i
10: statev←d'
11: **end if**
12: **end for**
13: flag←flag-1
14: **end while**
15: **if** statev¿$F(D_{state,SL}^{Pre}, D_{SL}^{Test})$ **then**
16: $T_{Pre,state}^{Mem} \leftarrow D_{Diff,SL}^{Pre}$
17: **else**
18: $T_{Pre}^{Nonmem} \leftarrow D_{Diff,SL}^{Pre}$
19: **end if**

the target sample's membership. Subsequent lines detail the discrimination methods employed in the attack.

In the second differential attack, we employ a different strategy. We iteratively calculate the membership of the target sample to differential groups (each group is closer to $D_{i,SL}^{Train}$). Algorithm 2 details the procedure. Lines 1-5 prepare initial variables, lines 6-14 outline the iterative calculation process for determining the target sample's membership, and subsequent lines specify the discrimination methods employed in the attack. The key distinction between this algorithm and the previous one (Algorithm 2) lies in the direct calculation of the membership of $D_{i,SL}^{Train}$ to $D_{k,SL}^{Train}$ (where i is not equal to k).

Maximum Mean Discrepancy (MMD) serves as one of the bases for differential attacks, increasing with distributional differences. Intuitively, the distribution of local datasets among distributed edge clients within the same architecture is likely to vary. For instance, in medical applications of distributed edge systems, clients may span the globe, leading to disparate data distributions between locations like Guangzhou and New York, potentially driven by ethnic disparities or differences in lifestyle. Consequently, assessing privacy risks of our distributed edge system under non-IID conditions may better reflect reality. Lastly, we experiment with altering the aggregation weight of each client to evaluate its impact on the attack results.

III. Experiment Evaluation

In this section, we implement the aforementioned membership inference attack against our distributed edge intelligence system and evaluate its performance.

A. Experiment Datasets

In our experiments, we utilize several common datasets to assess the privacy risks associated with MIA, including CIFAR-10, CIFAR-100, and News. We adhere to the preprocessing methods outlined in [8] and [16].

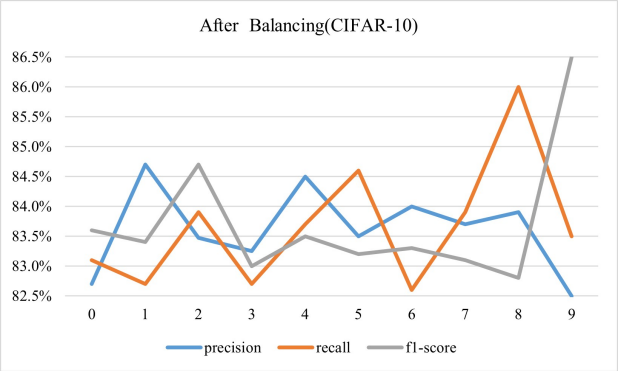
CIFAR-10: CIFAR-10 is a widely used dataset in the field of image recognition, comprising images with a resolution of 32×32 pixels. It consists of 50,000 training images and 10,000 test images categorized into 10 classes. Notably, each class contains 5,000 images, ensuring even distribution across the dataset.

CIFAR-100: Similar to CIFAR-10, CIFAR-100 serves as a benchmark dataset for image recognition tasks. It comprises 50,000 images classified into 100 classes. Each class contains 500 training images and 100 test images, maintaining an even distribution across the dataset.

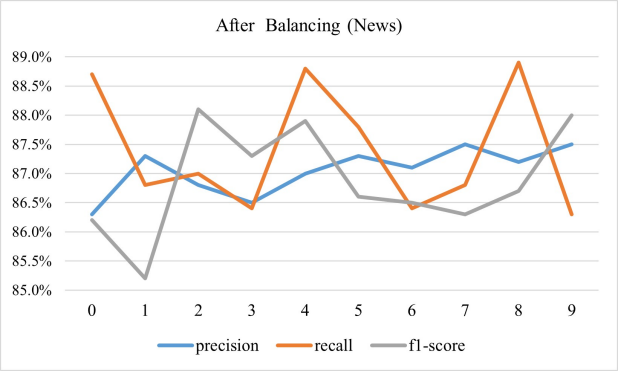
News: The News dataset is an internationally recognized standard dataset commonly used in classification, data mining, and information retrieval research. It consists of approximately 20,000 newsgroup documents categorized into 20 distinct newsgroups covering various topics.

B. Experiment Settings

We assess the privacy risks of the distributed edge system on CIFAR-10, CIFAR-100, and News datasets. For each dataset, we divide it into two parts: D_{SL}^{Train} (comprising $D_{1,SL}^{Train}$, ..., $D_{n,SL}^{Train}$) and D_{MIA}^{Train}. According to the attack strategy, D_{MIA}^{Train} is further subdivided into D_{Shadow}^{Train} and D_{Shadow}^{Test}. D_{Shadow}^{Train}, categorized into members and non-members based on whether they participate in the training of the attacked client, is utilized to train the attack model. For image datasets, we employ convolutional neural networks (CNNs) to construct the client's model [8]. For text datasets, linear neural networks (NNs) are used [8].



(a) CIFAR-10 dataset.



(b) News dataset.

Fig. 4: The attack results of *One-to-One attack*.

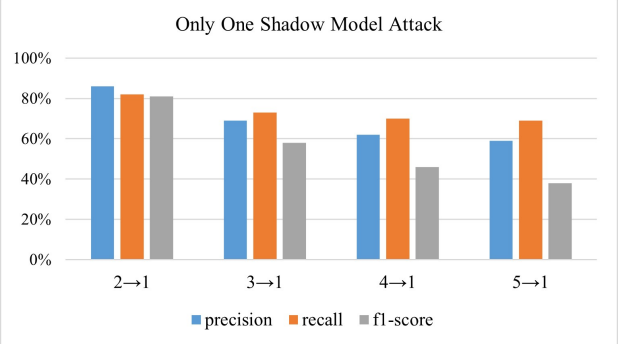


Fig. 5: The attack results of *One-to-One attack* on the CIFAR-10 dataset, where $N \rightarrow 1$ means that client N attacks client 1.

We consider three typical membership inference attacks in our system. The first is the *One-to-One attack*. In a distributed edge system with N clients, this implies that the client with client ID N is the malicious attacking client. For instance, $N \rightarrow 1$ signifies that client N will attack client 1. The second attack is the *Multi-to-One attack*. In a distributed edge system with N clients, this denotes that the client with client ID i is the malicious attacking client, where $i = 2, 3, \cdots, N-1$. For example, $i \rightarrow 1$ indicates that client i will attack client 1. The third attack is the *One-to-Multi attack*. In a distributed edge scenario with N clients, this indicates that the client with client ID i, where $i = 2, 3, \cdots, N-1$, is the malicious attacking client and will target other clients.

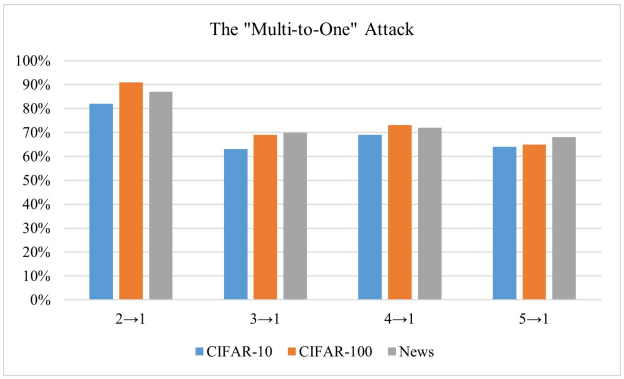


Fig. 6: The attack results of *Multi-to-One*, where the $i \rightarrow 1$ means that client i attacks client 1.

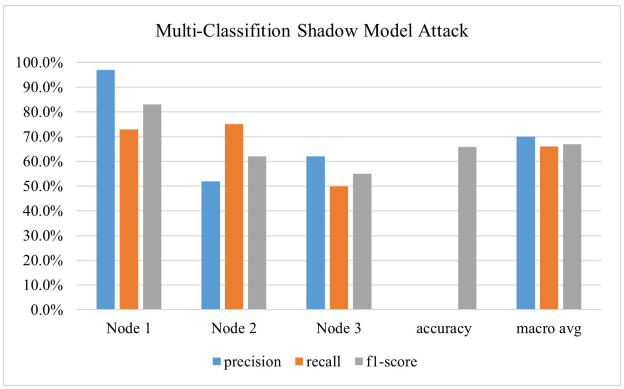


Fig. 7: The attack results of *One-to-Multi*, where *accuracy* and *macro-avg* are the general attack accuracy and macro-average result of the whole attack.

C. Experiment Results for Attack 1

The attack performance of the *One-to-One attack* is illustrated in Figure 5 and Figure 4. Figure 4 indicates that the attack results of the *One-to-One attack* on the CIFAR-10 dataset and News dataset are consistently stable, achieving high attack performance (¿82%) across varying numbers of clients in our distributed edge system. However, Figure 5 suggests that the effectiveness of the attack diminishes as the size of the distributed edge system increases. For instance, on the CIFAR-10 dataset, the attack results decrease from 83% to 40% (even lower than the blind guess baseline) as the client size increases from 2 to 5.

The attack performance of the *Multi-to-One attack* is depicted in Figure 6. It can be observed that each client in the distributed edge system can successfully execute the membership inference attack. However, there is a slight decrease in attack accuracy as the client size increases from 3 to 5.

The attack performance of the *One-to-Multi attack* is illustrated in Figure 7. In our distributed edge system, we achieve an accuracy of 66%, significantly surpassing the baseline of blind guessing (i.e., 33% for a client size of $N = 3$).

We analyze the variations in *One-to-One attack* results under balanced and unbalanced datasets and provide relevant explanations. On unbalanced datasets, the effectiveness of the attack decreases with an increase in the number of clients. This phenomenon seems reasonable because ordinary internal clients cannot observe the model aggregation process,

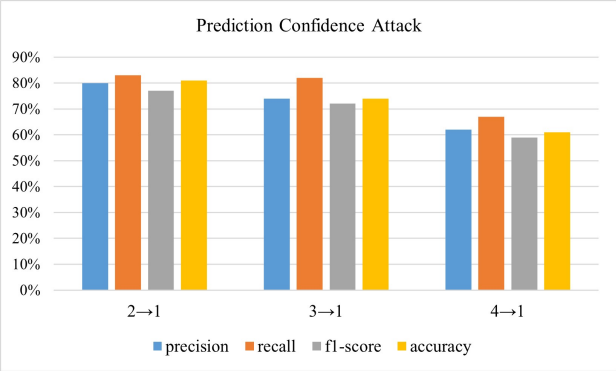


Fig. 8: Prediction confidence attack.

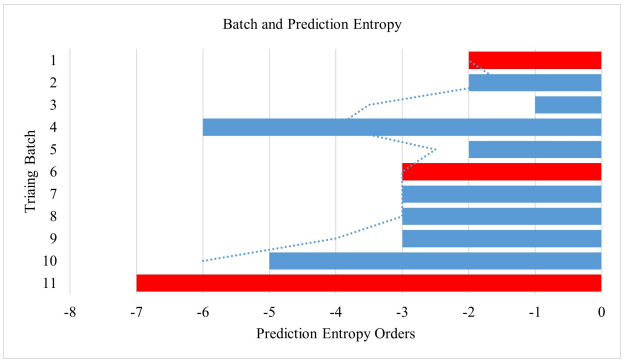


Fig. 9: Relationship between training epochs and prediction entropy. The abscissa represents the magnitude of prediction entropy, while the ordinate denotes the training batch. The red batch indicates where the target sample is added.

preventing the attacking client from obtaining specific information about each client's parameter contributions. However, membership inference attacks often exploit the model's lack of generalization ability to infer membership based on performance disparities between training and non-training data. Without knowledge of the specific generalization differences of the attacked client, our attack effectiveness weakens with an increasing number of client points.

But why does the attack effectiveness improve after balancing the datasets? According to [17], a comprehensive study of deep learning models reveals that a "perfect" model can be achieved if the model parameters exceed the number of training and test sets. Additionally, they argue that overfitting and generalization abilities are not as intuitive as commonly perceived. Moreover, techniques like dropout and regularization are not always necessary and may not be as effective as using a simpler model directly. As per [17], our attack model contains far more parameters than the number of training and test sets, potentially leading to the learning of faults and redundant information. For instance, if there are more negative samples than positive samples in the training sets, the model may lean towards classifying low-confidence samples as negative. Furthermore, due to dataset errors and learning method issues, no "perfect" model exists in machine learning. This means that any model will misclassify data, and models with good classification performance may struggle with datasets containing misjudged data. Consequently, training and evaluating attacks on unbalanced datasets are scientifically unsound, potentially yielding results lower than the blind guess baseline (50%). Balancing the positive and negative samples of the attack model leads to improvements in attack performance.

D. Experiment Results for Attack 2

For the attack based on prediction entropy, we observe poor results on our distributed edge system. Even on the distributed edge system with only 3 clients, we achieve only 57% accuracy, slightly better than the 50% blind guess baseline, indicating weak performance. Figure 8 illustrates the results of the attack based on prediction confidence on our distributed edge system. Remarkably, even with 4 clients in the distributed edge system, we achieve a 66% attack accuracy, significantly surpassing the 50% blind guess baseline.

We attempt to elucidate and analyze the experimental results. Figure 9 illustrates the relationship between training epochs and prediction entropy. The red epochs indicate where target samples are added, while the blue epochs signify their removal. After each epoch of training, the post-training model is employed to predict these samples and obtain their prediction entropy. The experimental findings reveal that the relationship between sample prediction entropy and training epochs is not consistently clear. While prediction entropy generally decreases with increasing training epochs, there are instances where the addition of target samples actually increases the prediction entropy of the model. Traditional membership inference attacks typically target conventional machine learning models that enhance prediction accuracy and generalization ability through numerous epochs. This aligns with the basis of attacks relying on prediction entropy.

However, our distributed edge system adopts a strategy of fewer epochs and more aggregation to enhance model prediction accuracy and generalization ability. Consequently, it exhibits a robust defense against membership inference attacks based on prediction entropy. In contrast, attacks based on prediction confidence yield relatively better results. This may be attributed to the more complex relationship between model prediction performance and sample training status, compared to a simple entropy relationship. Although this relationship cannot be explicitly expressed currently, the attack model is able to capture it. Thus, attacks based on prediction confidence prove more effective against our distributed edge system.

E. Experiment Results for Attack 3

We utilize the CIFAR-100 dataset to evaluate and observe the performance of the differential attack against our distributed edge system. In the independent identically distributed (IID) experiment, we partition the dataset into two parts: $D_{n,SL}^{Train}$ and D_{SL}^{Test}, consistent with the settings in the previous experiment. For the non-IID experiment, we employ a method proposed by [2] to partition the dataset, generating non-IID datasets using Dirichlet distribution. Privacy evaluation is conducted on the distributed edge framework with 4 clients, where the datasets are divided into $2n$ parts: $D_{n,SL}^{Train}$ and D_{SL}^{Test}. Each client's test set differs, and the size of the training

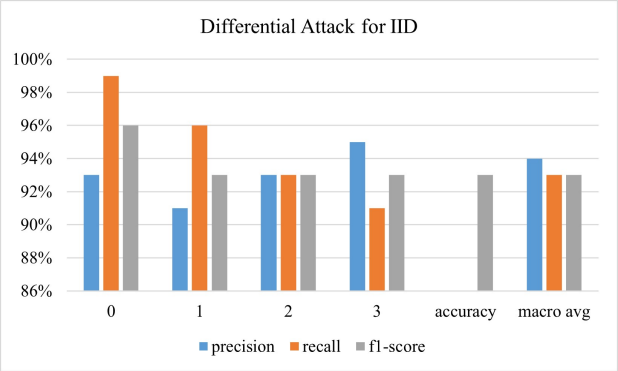


Fig. 10: Differential Attack for IID.

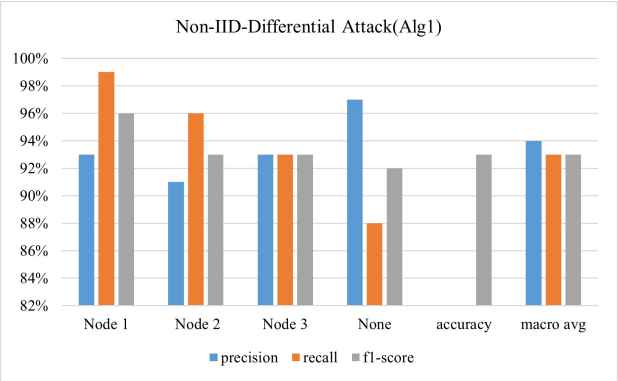


Fig. 11: Differential Attack Algorithm 1 for Non-IID.

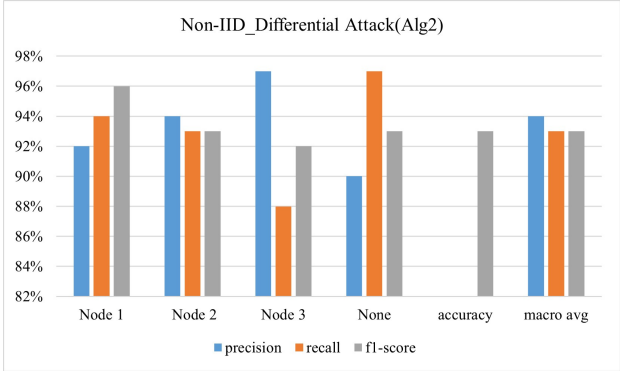


Fig. 12: Differential Attack Algorithm 2 for Non-IID.

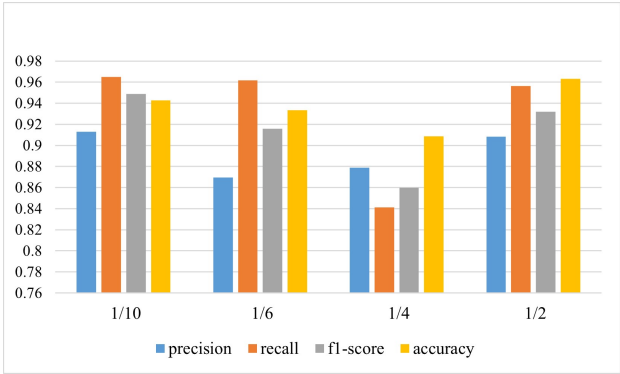


Fig. 13: Differential Attack Algorithm 2 for Different Aggregation.

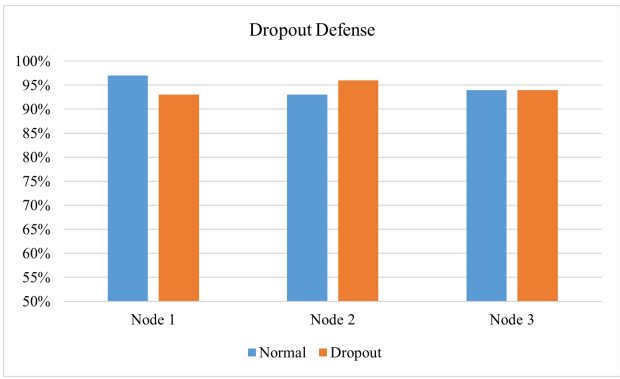


Fig. 14: The results of the Dropout defense method.

set for each client is similar. Table I illustrates the distribution of the training set and test set for each client.

Regarding the differential attack under independent identical distribution (IID), we achieve remarkably accurate results, as depicted in Figure 10. This experiment employs a distributed edge configuration with four clients. Figure 11 and Figure 13 present the results of differential attack algorithms 1 and 2, respectively, under the non-IID condition. These results are equally impressive. Even in our distributed edge system with four clients, we achieve 80% accuracy, significantly surpassing the blind guess baseline (25%). Notably, our distributed edge system exhibits lower resistance to differential attacks, particularly under non-IID conditions. Subsequently, the attack results under different aggregation weights are illustrated in Figure 13.

IV. Defense Strategies

In this section, we employ several defense methods against membership inference attacks, including Regularization [11], [18], and Dropout [19], to assess their effectiveness in mitigating privacy risks. We utilize a distributed edge configuration with four clients and evaluate the defenses against the best-performing differential attack identified in previous experiments.

Specifically, we use the CIFAR-100 dataset with four clients in this section. For Dropout, we adopt the strategy outlined in [16], incorporating dropout mechanisms after each max-pooling layer of the model. Each dropout mechanism is assigned a differential weight, typically 0.25 or 0.5. Regarding Regularization [11], we employ L2-Regularization with a weight of 0.001. Figure 14 presents the attack results before and after the inclusion of dropout mechanisms, while Figure 15 illustrates the defense effect of regularization. It's notable that regularization does not yield significant defense results. However, it appears that the differential attack itself may have a better defense against regularization, as evidenced by the improved defense results obtained when employing conventional attacks, as shown in Figure 15.

V. Related Works

Distributed Edge Intelligence. Traditional machine learning methods rely heavily on massive data for training, posing challenges in data collection and privacy preservation during

Client ID	Labels	Training Size	Test Size
1	4. 5. 6. 7. 9. 10. 14. 15. 16. 17. 19. 22. 31. 32. 34. 36. 37. 39. 42. 43. 46. 48. 50. 51. 54. 57. 58. 59. 60. 62. 63. 65. 66. 67. 68. 73. 74. 75. 76. 78. 79. 82. 83. 88. 90. 93. 94. 95. 97. 98.	11360	3787
2	0. 3. 9. 11. 12. 13. 15. 17. 18. 19. 20. 22. 23. 25. 26. 27. 30. 32. 33. 35. 37. 41. 42. 44. 46. 47. 50. 51. 52. 55. 56. 58. 59. 60. 64. 69. 70. 72. 74. 75. 77. 80. 81. 85. 86. 88. 89. 90. 95. 96. 99.	10365	3455
3	0. 1. 2. 4. 6. 7. 11. 13. 14. 16. 19. 20. 21. 22. 24. 26. 28. 30. 31. 33. 34. 35. 37. 38. 39. 40. 41. 44. 45. 47. 48. 49. 50. 51. 53. 56. 61. 64. 68. 69. 71. 73. 76. 78. 81. 82. 86. 87. 89. 93. 94. 96. 97. 98. 99.	13269	4424
4	1. 2. 3. 4. 6. 8. 14. 17. 18. 21. 25. 27. 29. 31. 32. 33. 35. 40. 42. 46. 52. 54. 55. 56. 59. 60. 67. 68. 69. 70. 72. 74. 75. 76. 78. 82. 83. 84. 85. 86. 88. 89. 91. 92. 93. 94. 95. 97. 98. 99.	10005	3335

TABLE I: Non-IID distribution on each client.

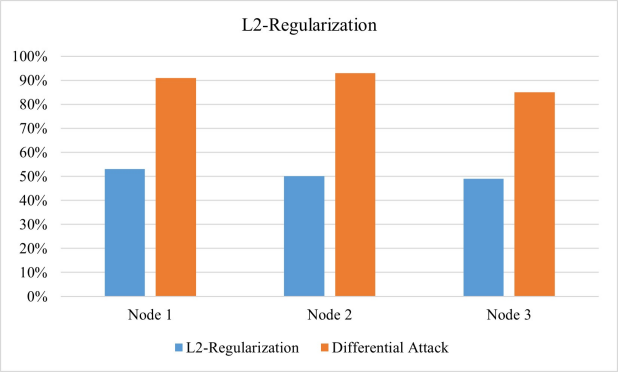


Fig. 15: The results of the L2-Regularization defense method.

data sharing and transmission. Federated Learning (FL)[4] offers a distributed learning approach to address this issue. However, FL still relies on a centralized server for model aggregation, leaving it vulnerable to information leakage during server-targeted attacks. In contrast, a novel distributed learning paradigm called swarm learning[16], [20] empowers participating client users to drive model training. In swarm learning, there is no centralized server; instead, each user forms a decentralized network to transmit local model updates sequentially. Prior to each aggregation cycle, a client is randomly selected to act as a temporary server, enhancing system robustness by aggregating updates from all client users.

Membership Inference Attack. Following model training, machine learning models retain information about the training data, making them susceptible to a technique known as membership inference attack. Initially proposed for genomic data [21], membership inference attacks have since demonstrated efficacy in contexts ranging from human mobility aggregation [3] to machine learning [7]. Numerous recent studies have addressed this vulnerability [8], [11], [4], [2], [15], [4], [9], [22]. For instance,[8] proposed an enhanced NN-based membership inference attack and introduced metric attacks. Additionally,[4] identified vulnerabilities in gradient descent and devised a white-box attack based on this insight. Moreover, [15] innovatively integrated the concept of differential comparison with membership inference attacks.

VI. Conclusion

In this study, we employ the membership inference attack to elucidate the latent data leakage inherent when edge nodes endeavor to collectively formulate a global model. Additionally, we address various defense mechanisms aimed at ameliorating this security vulnerability. It is our aspiration that this research will stimulate further investigations into data privacy within edge intelligence systems.

References

[1] B. McMahan, E. Moore, D. Ramage, S. Hampson, B. A. y Arcas, Communication-efficient learning of deep networks from decentralized data, in: A. Singh, X. J. Zhu (Eds.), Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, AISTATS 2017, 20-22 April 2017, Fort Lauderdale, FL, USA, Vol. 54 of Proceedings of Machine Learning Research, PMLR, 2017, pp. 1273–1282.

[2] M. Yurochkin, M. Agarwal, S. Ghosh, K. H. Greenewald, T. N. Hoang, Y. Khazaeni, Bayesian nonparametric federated learning of neural networks, in: K. Chaudhuri, R. Salakhutdinov (Eds.), Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA, Vol. 97 of Proceedings of Machine Learning Research, PMLR, 2019, pp. 7252–7261.

[3] A. Pyrgelis, C. Troncoso, E. D. Cristofaro, Knock knock, who's there? membership inference on aggregate location data, CoRR abs/1708.06145.

[4] M. Nasr, R. Shokri, A. Houmansadr, Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning, in: 2019 IEEE Symposium on Security and Privacy, SP 2019, San Francisco, CA, USA, May 19-23, 2019, IEEE, 2019, pp. 739–753.

[5] Z. Yang, J. Zhang, E. Chang, Z. Liang, Neural network inversion in adversarial setting via background knowledge alignment, in: L. Cavallaro, J. Kinder, X. Wang, J. Katz (Eds.), Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, CCS 2019, London, UK, November 11-15, 2019, ACM, 2019, pp. 225–240.

[6] K. Ganju, Q. Wang, W. Yang, C. A. Gunter, N. Borisov, Property inference attacks on fully connected neural networks using permutation invariant representations, in: D. Lie, M. Mannan, M. Backes, X. Wang (Eds.), Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, CCS 2018, Toronto, ON, Canada, October 15-19, 2018, ACM, 2018, pp. 619–633.

[7] R. Shokri, M. Stronati, V. Shmatikov, Membership inference attacks against machine learning models, CoRR abs/1610.05820.

[8] A. Salem, Y. Zhang, M. Humbert, M. Fritz, M. Backes, Ml-leaks: Model and data independent membership inference attacks and defenses on machine learning models, CoRR abs/1806.01246.

[9] L. Song, P. Mittal, Systematic evaluation of privacy risks of machine learning models, in: M. Bailey, R. Greenstadt (Eds.), 30th USENIX Security Symposium, USENIX Security 2021, August 11-13, 2021, USENIX Association, 2021, pp. 2615–2632.

[10] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, A. J. Smola, A kernel two-sample test, J. Mach. Learn. Res. 13 (2012) 723–773.

[11] M. Nasr, R. Shokri, A. Houmansadr, Machine learning with membership privacy using adversarial regularization, in: D. Lie, M. Mannan, M. Backes, X. Wang (Eds.), Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, CCS 2018, Toronto, ON, Canada, October 15-19, 2018, ACM, 2018, pp. 634–646.

[12] J. Jia, A. Salem, M. Backes, Y. Zhang, N. Z. Gong, Memguard: Defending against black-box membership inference attacks via adversarial examples, in: L. Cavallaro, J. Kinder, X. Wang, J. Katz (Eds.), Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, CCS 2019, London, UK, November 11-15, 2019, ACM, 2019, pp. 259–274.

[13] M. Abadi, A. Chu, I. J. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, L. Zhang, Deep learning with differential privacy, CoRR abs/1607.00133.

[14] K. M. Borgwardt, A. Gretton, M. J. Rasch, H. Kriegel, B. Schölkopf, A. J. Smola, Integrating structured biological data by kernel maximum mean discrepancy, in: Proceedings 14th International Conference on Intelligent Systems for Molecular Biology 2006, Fortaleza, Brazil, August 6-10, 2006, 2006, pp. 49–57.

[15] B. Hui, Y. Yang, H. Yuan, P. Burlina, N. Z. Gong, Y. Cao, Practical blind membership inference attack via differential comparisons, CoRR abs/2101.01341.

[16] S. Warnat-Herresthal, H. Schultze, K. Shastry, S. Manamohan, S. Mukherjee, V. Garg, R. Sarveswara, K. Händler, P. Pickkers, N. Aziz, etc., Swarm learning for decentralized and confidential clinical machine learning, Nature 594 (7862) (2021) 265–270.

[17] C. Zhang, S. Bengio, M. Hardt, B. Recht, O. Vinyals, Understanding deep learning (still) requires rethinking generalization, Vol. 64, 2021, pp. 107–115.

[18] J. Li, N. Li, B. Ribeiro, Membership inference attacks and defenses in supervised learning via generalization gap, CoRR abs/2002.12062.

[19] S. Yeom, I. Giacomelli, M. Fredrikson, S. Jha, Privacy risk in machine learning: Analyzing the connection to overfitting, in: 31st IEEE Computer Security Foundations Symposium, CSF 2018, Oxford, United Kingdom, July 9-12, 2018, IEEE Computer Society, 2018, pp. 268–282.

[20] K. Chen, H. Zhang, X. Feng, X. Zhang, B. Mi, Z. Jin, Backdoor attacks against distributed swarm learning, ISA Transactions 141 (2023) 59–72.

[21] N. Homer, S. Szelinger, M. Redman, D. Duggan, W. Tembe, J. Muehling, J. V. Pearson, D. A. Stephan, S. F. Nelson, D. W. Craig, Resolving individuals contributing trace amounts of dna to highly complex mixtures using high-density snp genotyping microarrays, PLOS Genetics 4 (8) (2008) e1000167.

[22] D. Chen, N. Yu, Y. Zhang, M. Fritz, Gan-leaks: A taxonomy of membership inference attacks against generative models, in: J. Ligatti, X. Ou, J. Katz, G. Vigna (Eds.), CCS '20: 2020 ACM SIGSAC Conference on Computer and Communications Security, Virtual Event, USA, November 9-13, 2020, ACM, 2020, pp. 343–362.