FL-TIA: Novel Time Inference Attacks on Federated Learning

Chamara Sandeepa*, Bartlomiej Siniarski†, Shen Wang‡, Madhusanka Liyanage§¶
*†‡§School of Computer Science, University College Dublin, Ireland
Email: *abeysinghe.sandeepa@ucdconnect.ie, †bartlomiej.siniarski@ucd.ie, ‡shen.wang@ucd.ie, §madhusanka@ucd.ie

***Abstract*—Federated Learning (FL) is an emerging privacy-preserved distributed Machine Learning (ML) technique where multiple clients can contribute to training an ML model without sharing private data. Even though FL offers a certain level of privacy by design, recent works show that FL is vulnerable to numerous privacy attacks. One of the key features of FL is the continuous training of FL models over many cycles through time. Observing changes in FL models over time can lead to inferring information on changes to private and sensitive data used in the FL process. However, this potential leakage of private information is not yet investigated significantly. Therefore, this paper introduces a new form of inference-based privacy attacks called FL Time Inference Attacks (FL-TIA). These attacks can reveal private time-related properties such as the presence or absence of a sensitive feature over time and if it is periodical. We consider two forms of such FL-TIA: i.e. identifying changes in membership of target data records over training rounds and detecting significant events in clients over time by observing differences in FL models. We use the network Intrusion Detection System (IDS) as a use case to demonstrate the impact of our attack. We propose a continuous updating attack model method for membership variation detection by sustaining the accuracy of the attack. Furthermore, we provide an efficient detection method that can identify model changes using cosine similarity metric and one-shot mapping on shadow model training.**

***Index Terms*—Federated Learning, Inference Attacks, Privacy, Intrusion Detection, Privacy Attack, Communication Networks**

I. Introduction

The ever-increasing communication capabilities and new modes of technologies for collecting data have already led to the rapid growth of big data. This can be expected to increase rapidly in the near future. In Beyond 5G (B5G)/6G, there will be an expansion of automated services with zero-touch networking and management. These future networks and associated services will require to continuously improve their Artificial Intelligence (AI) systems to provide adaptable services over time for new changes in data. However, due to rising data privacy concerns, the best option is to record and process data locally without transmitting it to a third party [1]. Yet these locally trained ML models may suffer from issues such as overfitting to local data and, thus, may lead to less accurate results. Therefore, it may still require more accurate ML predictions with support from third-party.

FL comes as a solution to improve the quality of AI systems in a collaborative manner by allowing them to share ML model updates instead with third parties, meanwhile keeping the original data private. However, with collaborative model updates shared from these end clients in the FL model training, there is a possibility of leaking sensitive information on data.

A. Related Works

Many related works describe FL-based privacy attacks, including membership inference, determining class representatives, property inference on data, and model inversion attacks. The survey in [2] classifies different privacy leakage scenarios based on insiders, like malicious clients and servers, and outsiders, like consumers and eavesdroppers. Recent works identify different types of threats and attacks in FL. Such attacks include reconstruction attacks, data and model poisoning, backdoor attacks, and inference attacks [3].

Inference attacks are adopted in FL from general ML attacks. They aim to reveal information on training data used to train the models through exploiting the model parameters [3]. Several works focus on identifying specific properties of clients. One such example is the work in [4], which proposes source inference, a technique to identify the source that added specific data for training the FL model. The authors in [5] discuss another type of inference attack called category inference, where an attacker attempts to determine the category information of the data used by the clients. They use a multi-train classifier inference model with an approximate model update technique. The works [6], [7] use property inference for identifying the first appearance of specific features in training data; however, they may be applicable only for certain distinct properties on the dataset and may not be applicable for two data with the same property.

Authors in [8] attempt to identify information leaks through membership inference in sequential FL, which is a form of FL that only accounts for one model at a time. Authors in [9] use *shadow models* for improving the accuracy of membership inference attacks in general ML scenarios. Work in [6] uses shadow models for property inference attacks in FL. However, these works do not identify that attack models will loose their accuracy over continuous FL iterations since they are not able to infer on new data. These shadow models are copies of the original ML algorithm. They are trained using artificially generated dataset or noisy real data. A GAN-based membership inference attack without shadow models for FL is proposed in [10]. Using a GAN, an attacker can generate any amount of artificial data samples. GAN consists of two components, a discriminator and a generator. Representative data is fed to the discriminator. The generator uses random noise to generate new samples similar to the input dataset. In

our work, we present a combined approach of using GAN and shadow models. The Fig. 1 provides an overview of the GAN and shadow model training process and outlines their potential benefits and weaknesses when applying them separately from the attacker side.

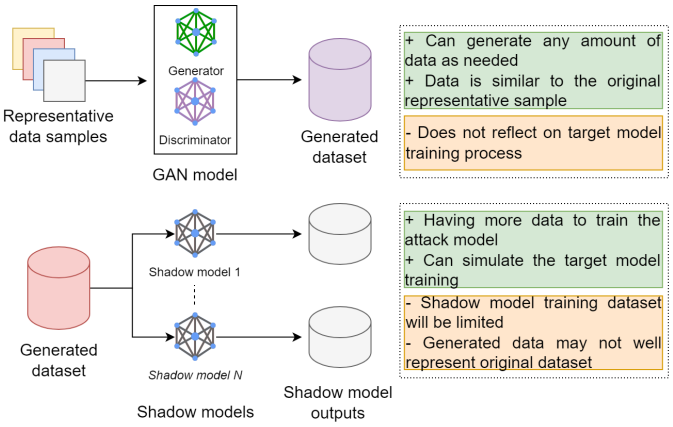


Fig. 1: GAN data generation and shadow model training process. Each technique has disadvantages for the attacker when using the two techniques isolated in attack model training. Our method combines both approaches for better attack accuracy.

The inference attacks are designed to exploit ML model parameters. In FL, since FL is sharing model parameters with third parties by design, there is a high possibility of being subjected to an inference attack for clients participating in FL training. However, existing types of inference attacks were designed to reveal different properties such as membership, source or any other property of the data only at a particular snapshot in the FL lifecycle [4], [5], [10]. None of these works consider or assess the severity of attacks with the variation over the property *time*.

Therefore in this paper, we present a new type of inference attack called FL-TIA/TIA to infer the changes that occur over time in a scenario of IDS. We show the significance of privacy leakage in FL systems with *time inference*. We provide a real-world use case scenario of IDS to showcase the practicality and severity of the attack. We can identify time periods of data generation as a critical property that can impact the privacy of individuals and organisations in many domains. This may include examples such as network operations, IoT, transportation, business, and healthcare industries. The time periods of performing actions and occurrence of events can be highly confidential information in many applications. For example, considering emergency situations [11] in healthcare, the time interval that an emergency has occurred can be used as potential data to track an anonymized individual when correlated with public or other de-anonymized data.

Furthermore, revealing time through IoT devices will cause information leakages, such as tracking user habits and identifying device usage and idle times. It may also leak information about available device resources and processing limitations. Then, adversaries can use this information to their advantage. Examples include launching a physical assault on data owners or selling private information for purposes like personalized advertising. However, we identified that no considerable discussion on time inference attacks is currently performed for FL, which can periodically send information to untrusted third parties over public channels. Therefore, it makes FL more vulnerable to the mentioned and other potential privacy attacks with leakage of time information unless this threat is carefully considered beforehand.

B. Our Contributions

We consider two scenarios of FL-TIA in our IDS use case to infer the training round/iteration and to recover IDS attack events. by observing major changes in models and inferring its direction over training rounds. The timestamp of these training rounds can be recorded meanwhile capturing the local models. Thus, with the information about the actual rounds and timestamps, the attacker can reveal the time intervals. We perform the attacks for the use case of FL-based IDS, where an organisation can collaboratively train their local IDS ML models with third parties or an FL-based IDS implemented at different nodes or locations of the same organisation. TIA can result in multiple adverse effects for the victim IDS, including 1) exposure of the identity of the organisation and the period when a specific attack event has occurred from membership variation in anonymised and timestamp removed data records, 2) revealing the severity of the attack events even without any data records, 3) identification of vulnerabilities and peak vulnerability periods in the security system of the organisation or a specific node with IDS. In the paper, we further summarise ***our contributions*** as follows:

- Present a new variation of inference attacks on FL, called time inference, for a use case of IDS.
- Propose a mechanism that combines GAN and shadow models to determine and maintain the accuracy of membership variation over time.
- Design a novel approach for detecting the occurrence and change of network attack events over time by continuous observation of target model similarity metric.
- Discuss potential defence techniques for TIAs and limitations of these defences.

C. Outline

The rest of this paper is arranged as follows. Section II presents an overview of the system model of our attack scenario. Section III describes details of the proposing time inference attacks. The experiments conducted are discussed in Section IV. In Section V, we present a discussion on potential defence techniques. Finally, we conclude the paper with a summary and future directions in Section VI.

II. System Model

We propose two types of TIA to identify 1) membership variation and 2) major events over time. They are launched

against a network-related use case scenario of IDS, where multiple organisations jointly compute FL models for intrusion detection. Fig. 2 provides an overview of the use case.

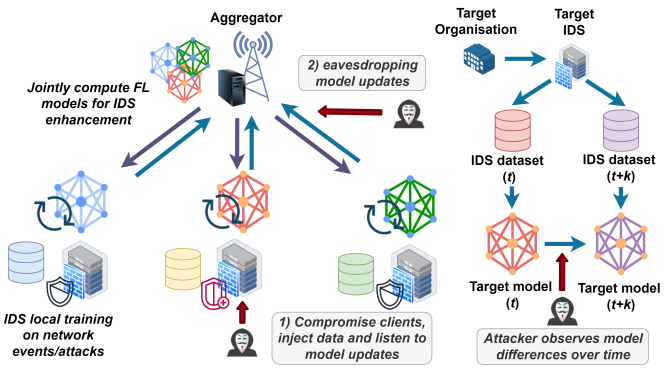


Fig. 2: IDS collaborative FL among multiple organisations

Here, TIA can be done by any party who can observe the variation of FL models over time: 1) in the membership variation detection attack, a malicious client, and 2) for the significant events detection, an eavesdropper who compromises the communication channel. Similar to [9], [10], our work considers that the attacker can obtain model parameters. Then, they can observe how the model parameters change over time to infer the properties of the original dataset. For the attacks, the attacker may also possess a small sample of the original training data. The attacker can obtain such data from compromised clients. In a real-world scenario, it may be possible to get data by attacking a few low-end IoT devices [12] or an IDS with a weaker security mechanism.

III. Proposed Time Inference Attacks

This section introduces our approaches for launching the FL-TIA in multiple steps, including algorithms of the attack.

A. Exploiting intrusion detection

The attacks such as DDoS can last between a few seconds to hours or even days [13]. Long-lasting attacks can result in significant amounts of data that can be collected by IDS model training. Also, shorter bursts of attacks may lead to periodic increases in attack data used to train IDS. In our use case, we consider IDS trained through FL by several organisations, where the TIA adversary needs to identify when these attacks occur. Revealing such information can lead to identifying potential vulnerability periods, which the attacker can use against the organisation to launch a security or privacy attack since the protection system is busy and vulnerable. Also, attack duration can leak organisation's internal information, like how quickly the organisation responds to the attacks.

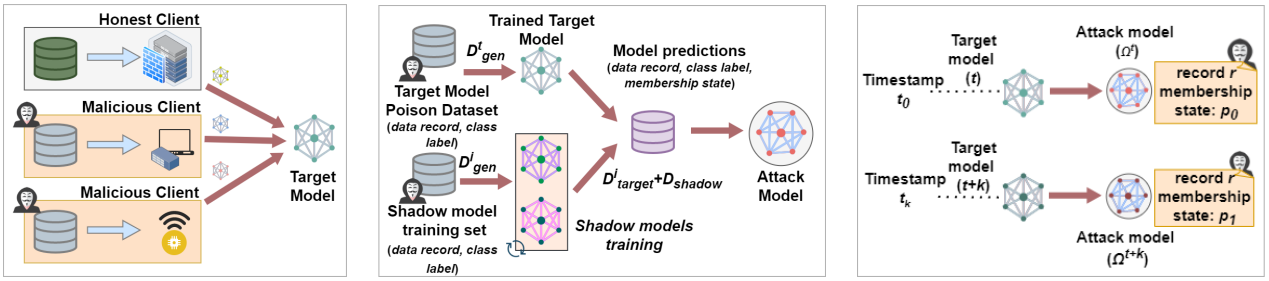
B. Attack Type 1: Membership Variation over Time

The interval when a given data record is utilised in FL model training is an unintended property that can be leaked in FL training iterations. Suppose an attacker obtains anonymised data records on the IDS, which is used as the training data of a certain organisation. The attacker wants to determine the owner of this data. This can be done using a membership inference [10] attack. However, even after detecting the owner's identity, knowing the exact time periods when the attack occurred is more valuable to the attacker. For example, with membership variation detection, the attacker can specifically identify when the members have been attacked more frequently. If attacks occur in a periodic pattern, the TIA attacker can predict future network attack possibilities of the target. Furthermore, they can identify the IDS FL model's vulnerabilities for any specific attack type. The benefit of this method is the attacker can pinpoint the exact occurrence time of a critical data record of interest in the victim organisation. Fig. 3 gives an overview of the steps of our proposed attack.

1) Attack model with GAN generated data and shadow model: The first step of the attacker is to enter to target model training process as a malicious client. In IDS systems, they may perform this by hijacking some nodes or injecting malicious updates over a compromised channel. Here, the attackers will need some data to train the malicious local models. For this, they can generate a representative dataset D_{gen} using GAN [10]. For a target model raining round t, the attacker injects $D^t_{gen} \subset D_{gen}$ to the target model and obtains D_{target} with class labels and membership state s. The attacker also trains a set of shadow models [9], with a similar architecture to the target models, using dataset $D^I_{gen} \subset D_{gen}$ and obtain D_{shadow} with class labels and membership state as well. Then, the predictions for test portions of D_{target} and D_{shadow} from both the target model and shadow models can be obtained. These predictions, along with the membership state s of both datasets, are fed into the training of another model named the attack model.

2) Continuously progressing attack model for training rounds: When using the same attack model trained for a previous round for predicting membership states, we observe that the accuracy of the attack model is gradually reduced. None of the previous works of the current state-of-the-art, such as [9], [10], considered the factor of accuracy loss over iterations. The attack model trained in FL training round t may not be as accurate in training round $t+1$ since new data may get added each round. To overcome this issue, we introduce the idea of the continuous progression of the attack model.

After getting the initial version of the attack model, the attacker uses Algorithm 1 to get the attack model Ω^{t+1} at next round $t+1$. Here, they get a representative sample $D^{t+1}_{gen} \subset D_{gen}$ for this new round $t+1$. Then, the attacker combines D^{t+1}_{gen} to the target model training dataset D^t_{gen} in the previous round. This combined dataset is used in the function POISON(), where the dataset is added to the FL training via the malicious nodes. It will return D^{t+1}_{target}. This output is combined with the initially trained shadow model dataset D_{shadow}. Since the attacker needs to perform shadow model training only once, the process is lightweight. It is then used to train the updated attack model Ω^{t+1}. Following this for q training rounds, a set of attack models is obtained $\{\Omega^1, \Omega^2, ..., \Omega^q\}$.



(a) Poisons the target model by GAN data via malicious clients.

(b) Trains shadow models and uses data from both to train attack model.

(c) Records timestamps and gets changes in membership probability.

Fig. 3: The proposed FL-TIA steps involve three phases: (a) data generation, (b) attack model training, and (c) inference.

Algorithm 1 Progressing attack model for q training rounds

Input: In a new round $t+1$, D_{gen}^{t+1}; $D_{gen}^{t+1} \subset D_{gen}$
Output: $\{\Omega^1, \Omega^2, ..., \Omega^q\}$: A set of attack models for q training rounds
Let $A = \{\}$
for q rounds, in each training round **do**
    POISON() $\leftarrow$ *Input*: $D_{gen}^{t+1} \cup D_{gen}^{t}$; *Return*: D_{target}^{t+1}
    $D_{attack}^{t+1} \leftarrow (D_{target}^{t+1} \cup D_{shadow})$
    ATTACK_TRAIN() $\leftarrow$ *Input*: D_{attack}^{t+1}; *Return*: Ω^{t+1}
    $A = A \cup \Omega^{t+1}$
end for
Return $A = \{\Omega^1, \Omega^2, ..., \Omega^q\}$

3) FL-TIA using the trained attack model: The updated attack models in each training round are used to infer the time interval when a new data element is added to the FL model training process. In between two training rounds t and $t+k$, we have two attack models Ω^t and Ω^{t+k} respectively. For a given data record r, the membership state probability can be obtained from the attack model Ω^t as p_t and Ω^{t+k} as p_{t+k}. For a threshold probability of P_K, if $p_t < P_K$ and $p_{t+k} > P_K$, an attacker can identify that the data record r has been entered into the FL model training during that period.

C. Attack Type 2: Detection of Significant Events over Time

Membership changes over time can hint to an attacker about specific changes happening to the original dataset. For example, suppose in the IDS system, the attacker has captured some data records of a Denial of Service (DoS) attack incident. Membership changing from non-member to a member in all these data at the same time interval can provide evidence of an ongoing attack on the victim IDS. However, detecting such changes can be difficult unless the attacker possesses enough data records of the same time interval. Therefore, we propose another method to identify such critical attack events over time and their recovery periods by observing overall trend changes.

1) Distance metric using cosine similarity: Suppose the attacker has two such target local models l_t and l_{t+k} at rounds t and $t+k$. The IDS should update these local models based on the latest data collected in their corresponding time windows. Similar to the attack type 1, here, the attacker can also generate a sample of GAN data D_{gen} and send this data through both models to get the model predictions as $l_t(D_{gen}) = y^t$ and $l_{t+k}(D_{gen}) = y^{t+k}$, where y^t and y^{t+k} are the output vectors obtained by the two models. The cosine similarity can be calculated as follows:

$$S_c(y^t, y^{t+k}) := cos(\theta) = \frac{y^t . y^{t+k}}{||y^t|| ||y^{t+k}||} \quad (1)$$

Higher values of $S_c(y^t, y^{t+k})$ imply the distance between the models is high, which can signify the nature of data used to train the local models in the two-time windows have significantly different properties. Thus, the TIA attacker can infer if a major incident in the IDS has occurred.

2) Reference shadow models for inferring attack direction: However, comparing only the two target models does not provide information on the attack direction. Determining which of the target models at t or $t+k$ has a higher ratio of attack data can show the direction of the attack. The attacker can infer that an attack event has occurred if new attack data is accumulating between t and $t+k$. If the ratio has decreased, a defence mechanism is launched. To obtain the direction, we use the shadow models as the intermediary reference in the cosine similarity. A shadow model ϕ_i is trained with a fixed ratio of $a_i : b_i$ of *attack:benign* data. Previously generated GAN dataset D_{gen} is sent through ϕ_i as $\phi_i(D_{gen}) = y^i$ to get its prediction y^i. If cosine similarity $S_c(y^t, y^i) > S_c(y^{t+k}, y^i)$, the attacker can infer that l_t has closer value of *attack:benign* ratio to $a_i : b_i$ than the other target model l_{t+k}. Finding the exact ratio of l_t then can be a search problem, where the attacker has to determine the shadow model ratio $a_i : b_i$ that results in maximum cosine similarity values between l_t and ϕ_i. For this, the attacker can use an algorithm like binary search, where they have to train a shadow model at each step.

3) Detection of attack events: Training multiple shadow models for each target model could be computationally expensive for the attacker, especially if the number of local clients and local models is high. As an alternative, we develop a one-shot mapping where *attack:benign* data are compared using a fixed number of shadow models. Here, we train two types of shadow models: static and dynamic models. A small set of static models $\phi_s = \{\phi_{s1}, \phi_{s2}..., \phi_{sk}\}$ were used to train with

known fixed ratios attack data vs. benign GAN data. Another set of dynamic shadow models $\phi_d = \{\phi_{d1}, \phi_{d2}..., \phi_{dn}\}$ are trained using different other ratios of *attack:benign* GAN data. The cosine similarities were taken between each combination of pairs in ϕ_s and ϕ_d. This results in a mapping of coordinates between *attack:benign* ratio vs. cosine similarity for ϕ_s. These coordinates can be fit to functions $\psi_s = \{\psi_{s1}, \psi_{s2}, ..., \psi_{sk}\}$, for each static shadow model in ϕ_s. When a target model l_t is captured, the attacker can identify cosine similarity with each static shadow model ϕ_s and get the inverse function ψ_s^{-1} to retrieve the corresponding *attack:benign* ratio for the target model. The summary of the attack is shown in Algorithm 2.

Algorithm 2 Predicting events attack with shadow mapping

Input: Static shadow models $\phi_s = \{\phi_{s1}, \phi_{s2}, ..., \phi_{sk}\}$, dynamic shadow models $\psi_d = \{\psi_{d1}, \psi_{d1}, ..., \psi_{dn}\}$, sample GAN dataset D_{gen}, local target models l_t and l_{t+k}
Output: Change in *attack:benign* data ratio in target models $(\Delta a : \Delta b)$
function SHADOW_MAPPING()
    Let $\psi_s = \{\}$
    for each ϕ_i in ϕ_s **do**
        Let $M = \{\}$
        for each ϕ_j in ϕ_d **do**
            $y_i = \phi_i(D_{gen}); y_j = \phi_j(D_{gen})$
            $M = M \cup S_c(y_i, y_j)$
        end for
        ψ_i=FIT_FUNCTION(M)
        $\psi_s = \psi_s \cup \psi_i$
    end for
    Return: ψ_s
end function
Let $y_t = l_t(D_{gen}); y_{t+k} = l_{t+k}(D_{gen})$
Let $\theta_t = S_c(y_t, \phi_i); \theta_{t+k} = S_c(y_{t+k}, \phi_i); \phi_i \in \phi_s$
ψ_s =SHADOW_MAPPING()
$(\Delta a : \Delta b) = \psi_s^{-1}(\theta_{t+k}) - \psi_s^{-1}(\theta_t)$
Return: $(\Delta a : \Delta b)$

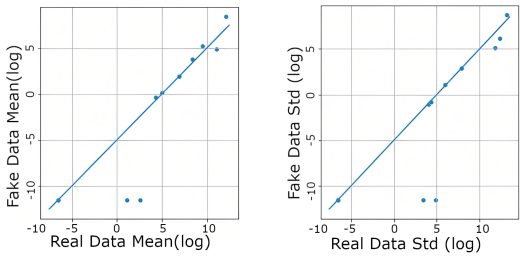
IV. EXPERIMENTS

A. Dataset and Experimental Setup

For our experiments, we used two datasets: NSL-KDD and the more recent 5G-Network Intrusion Detection Dataset (5G-NIDD) [14]. NSL-KDD is a popular benchmark dataset for IDS, which consists of 125,973 data records providing details on network traffic based on various network attack scenarios. In NSL-KDD, we pick the two most frequent types of traffic for our experiments, normal and DoS attacks. The 5G-NIDD dataset consists of 1,215,890 data records with 9 different types of network traffic, 8 representing different attack scenarios. For this dataset, we consider all these 9 network traffic types as output classes in the FL models. In the experiments, we only select small subsets of the original data records that maintain the original distribution of the datasets. All the experiments were implemented using Keras and Tensorflow Federated (TFF) framework. We set the configurations in TFF for both clients and the aggregator to automatically train the models with Stochastic Gradient Descent (SGD) optimization. For the datasets, we used 70% training vs. 30% test set. The training set was equally split among the clients. To run the experiments, we use a computing instance with Intel Xeon 2.20 GHz CPU, 26GB RAM, and an NVIDIA Tesla T4 GPU. We considered the target model a sequential Neural Network (NN) with a hidden layer of 512 dense units followed by a dropout layer with a 0.2 dropout rate. We use the similar NN model architectures for both datasets with except for input and output layers, which differ based on the shapes of input and output classes. We trained the target models with 50 clients and 20 initial rounds for each.

B. Membership Variation Detection over Time

As the next step, we trained a GAN to generate representative data for the attacker. We used 1,000 samples from the original datasets to train the generator and discriminator models. The generated results show a high similarity in the numeric features of actual vs. GAN data, as shown in Fig. 4 for NSL-KDD.



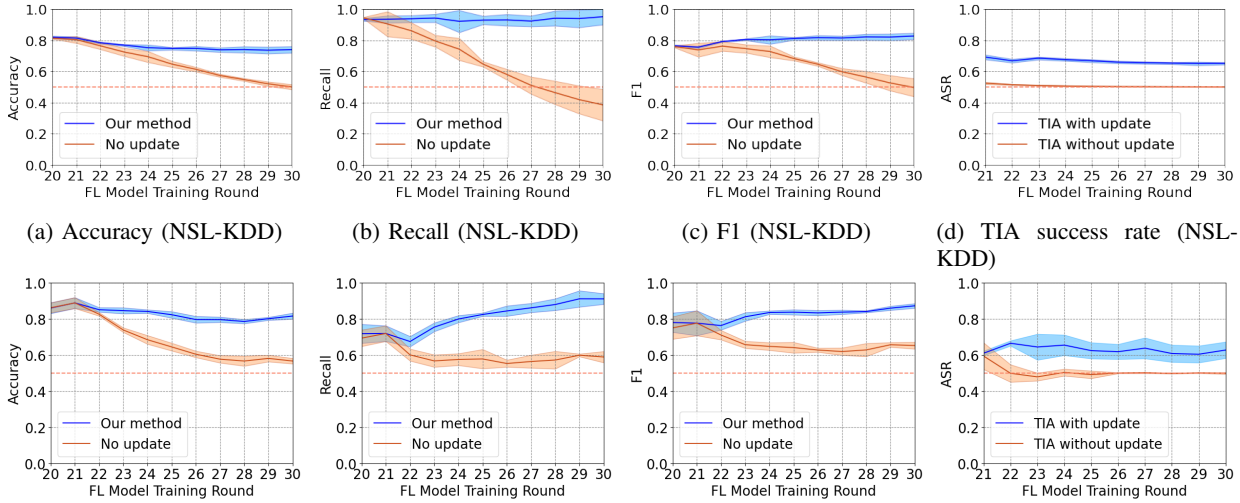
(a) Mean (b) Standard deviation

Fig. 4: GAN model training metrics for NSL-KDD dataset.

To create the attack model, we poison the target models using the generated labelled data with 20,000 records. We also used 10,000 GAN data records over 20 rounds to train five shadow models similar in architecture to the target model. The shadow models produce overall 50,000 shadow training outputs. Both target model and shadow model outputs are used to train the attack models, which are Multilayer Perceptron (MLP) classifiers with two hidden layers of 64 and 16 units, respectively. Results obtained for NSL-KDD in Table I show our combined approach of using both GAN and shadow models produces the best attack models for membership variation detection TIA.

TABLE I: Metrics for TIA attack model at round 21.

Metric	Only shadow models	Only GAN-based poisoning	**Both GAN and shadow models**
Test accuracy	66.66%	69.85%	**79.68%**
Precision	54.82%	58.18%	**64.97%**
Recall	64.16%	70.25%	**88.07%**
F1	59.12%	63.65%	**74.78%**



(a) Accuracy (NSL-KDD) (b) Recall (NSL-KDD) (c) F1 (NSL-KDD) (d) TIA success rate (NSL-KDD)

(e) Accuracy (5G-NIDD) (f) Recall (5G-NIDD) (g) F1 (h) TIA success rate (5G-NIDD)

Fig. 5: The accuracy, recall, F1 of attack models, and FL-TIA success rate with and without continuous updates.

1) Maintaining Accuracy of Attack Models over FL Model Training Rounds: In our attack, we introduce the approach of continuous progress of the attack model to maintain its accuracy. To compare our approach with the existing works, we continue to train attack models for further 10 FL rounds from Algorithm 1 using 20,000 GAN data records for NSL-KDD and 10,000 records for 5G-NIDD datasets, respectively, for each round. We also evaluate the metrics over the FL rounds with the attack model without further training, as presented in related work [10]. We tested the accuracy of the attack models with 5,000 poisoned data records for both datasets. From the results in Fig. 5 we observe that the attack model continuously updated with our approach maintains the initial accuracy, recall, and F1 metrics in Fig. 5 while the metrics for the attack model without any updates are reducing over the rounds.

2) Time Inference Attack via Round Prediction: To evaluate the FL-TIA for membership variation detection, we designed an experiment summarised in the Algorithm 3. Here for each round t, we get a sample GAN dataset D^t_{gen} where initial membership state $s = 0$ for all elements in D^t_{gen}. We send this data through the continuously progressing attack model, Ω^t, for round t. The membership probability p_0 for the round t in $D^t_{gen} \leftarrow (y^t_{pr_i}, y^t_{gen_i})$ can be obtained as:

$$p_0 = \frac{\sum_{i=1}^{|D^t_{gen}|}(\Omega^t((y^t_{pr_i}, y^t_{gen_i})) = 0)}{|D^t_{gen}|} \quad (2)$$

where $y^u_{pr_i}$ is the predicted value, $y^u_{gen_i}$ is the actual value of the i^{th} data record in D^t_{gen} and s is the membership state of either in or out. Then we poison the FL model with the same dataset over another round $t+k$ such that $s = 1$ for all values.

For each round, we use sample GAN datasets with 5,000 records as testing data across two rounds to evaluate the FL-TIA. We get the round predictions for the interval between each new attack model and the attack model in FL round 20, based on Algorithm 3. From this, we obtain the Attack Success Rate (ASR) for both with and without the continuously updated attack models. Here, we defined ASR as an average value for a success rate of the correct membership predictions between the two rounds. The results are in Fig. 5d and Fig. 5h They show FL-TIA has a significantly higher success rate with our approach of using continuously updating attack models.

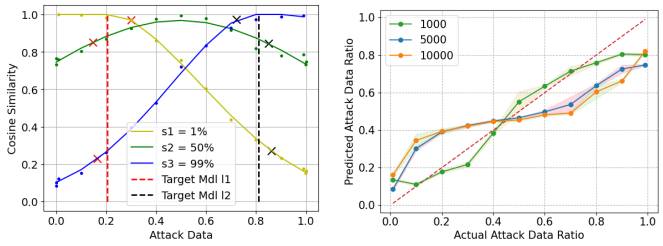
Algorithm 3 Experimenting FL-TIA through round prediction

1: **Input:** Representative dataset $D^t_{gen} \leftarrow (y^t_{pr}, y^t_{gen})$, attack model Ω^t for round t, attack model Ω^{t+k} for round $t+k$
2: **Output:** The Attack Success Rate ASR
3: Membership state $s = 0 \forall$ elements in D^t_{gen}
4: Get $p_0 = \frac{\sum_{i=1}^{|D^t_{gen}|}(\Omega^t((y^t_{pr_i}, y^t_{gen_i})) = 0)}{|D^t_{gen}|}$
5: At round $t+k$; POISON()$\leftarrow$ *Input*: D^t_{gen}; *Return*: D^{t+k}_{gen} where membership state $s = 1 \forall$ elements in D_{v1}
6: Get $p_1 = \frac{\sum_{i=1}^{|D^{t+k}_{gen}|}(\Omega^{t+k}((y^{t+k}_{pr_i}, y^{t+k}_{gen_i})) = 1)}{|D^{t+k}_{gen}|}$
7: **Return** $ASR = \frac{p_1 + p_0}{2}$

C. DoS attack Events Time Inference in IDS using Cosine Similarity and Shadow Model Mapping

For detecting DoS attacks, we follow the Algorithm 2 to create a shadow model mapping. We used 3 static shadow models $s1$, $s2$, $s3$ with respectively 1:99, 50:50, and 99:1 of *attack:benign* data ratio. The cosine similarity of each shadow model is compared with a set of 15 dynamic shadow models with varying attack data ratios from 0% to 100% of GAN-generated DoS attack data derived from NSL-KDD dataset.

We run the experiment for two scenarios of 10,000 and 1,000 overall data for the shadow models. Then the obtained values were averaged and fit a Gaussian distribution curve for each static shadow model as shown in Fig. 6a.



(a) Shadow models map (b) Actual vs. predicted data ratio

Fig. 6: Shadow models map and predicted cosine similarities for NSL-KDD.

We simulate two example target models, $l1$ and $l2$, which are two local IDS FL model updates with actual attack data percentages of 20% and 80%, respectively. By comparing cosine similarity of $l1$ with the three static shadow models, we observe that cosine similarity points for $l1$ as $\theta(s1, l1) > \theta(s2, l1) > \theta(s3, l1)$ and $\theta(s3, l2) > \theta(s2, l2) > \theta(s1, l2)$ for $l2$. From this, we can infer a reversal of the trend in training data. This signals that a DoS attack occurred between $l1$ and $l2$. Also, we derive the x value of $l1$ and $l2$ by obtaining the average value of the three points in each $l1$ and $l2$ that intersects the x axis. We see the inferred attack data ratio is very close to the actual ratio in Fig. 6a.

Furthermore, we perform another experiment by obtaining multiple target models with different NSL-KDD attack data ratios, ranging from 1% to 99%. It is shown in Fig. 6b. Three scenarios were considered with 1,000, 5,000, and 10,000 total datasets for the target models. We observe the predicted values are closer to the actual data ratios in the target models. Their Mean Squared Error (MSE) values are obtained as 0.0060, 0.0204, and 0.0294 for 1,000, 5,000, and 10,000, respectively. Here, we can conclude that lower window sizes of datasets increase the risk of the attack. Therefore, high vulnerability can be expected for IDS running in resource-limited IoT environments. Our method can also clearly identify the extremes where either high or low attack data is present.

V. Discussion

A. Defence Techniques

The TIA is done by the attack models used for the membership inference in FL training rounds. Therefore, if the attack models have lesser accuracy, the ASR of the round prediction will be reduced. Adding perturbations to the training data, model updates via Differential Privacy (DP) [15], [16] may provide resilience against the TIA. We tested this by adding DP with varying privacy budget ϵ. The FL model accuracy and average ASR at round 20 are shown in Table II.

Adding noise via DP has reduced ASR to 49.96% for NSL-KDD when $\epsilon = 0.5$, providing a better privacy guarantee.

TABLE II: DP against Membership Variation Detection.

DP status	No DP	$\epsilon = 4.0$	$\epsilon = 2.0$	$\epsilon = 1.0$	$\epsilon = 0.5$
FL Acc	95.72%	95.15%	94.64%	94.03%	84.75%
ASR	67.48%	67.24%	58.69%	57.08%	49.96%

However, it has reduced the model accuracy by about 10% as well. Therefore, it has a trade-off over accuracy when using DP-based defence. Considering defence against event prediction, other approaches, such as secure aggregation, can be used since it masks and protects individual model updates from eavesdroppers, which we extend as a future direction.

B. Comparison with Related Works

We compared the accuracy metrics of our attack model in membership variation with the study in [10], as shown in Fig. 5. Results show our attack with continuous updates has outperformed and maintained the accuracy of the attack model over time. Furthermore, to the best of our knowledge, none of the state-of-the-art works on FL consider TIA based on the significant event detection of IDS we present in the second attack. We also compare our findings with the other related inference attack types in Table III.

TABLE III: Summary contribution of our work.

Characteristics	Ref [4]	Ref [8]	Ref [9]	Ref [10]	Ref [5]	**Our work**
Time-based inference on FL-based IDS (Novel privacy attacks on FL)	-	-	-	-	-	✓
GAN to generate datasets for inference	-	-	-	✓	-	✓
Training shadow models for more data	-	✓	-	✓	-	✓
Continuously evolving attack model	-	-	-	-	-	✓
Experiment FL attack change over time	-	✓	-	-	✓	✓
Defence techniques and limitations	✓	✓	✓	-	✓	✓

VI. Conclusion

In this paper, we propose two time-based privacy attacks on FL to detect membership variation and to identify attack events of an IDS use case. We introduce a continuous updating attack model that has better accuracy of detecting membership variation detection over the existing approaches that show degradation in the performance through time. In event detection, we use the lightweight technique of cosine similarity to infer the changes for new events. Our experimental results demonstrate an event like a DoS attack occurrence in a target organisation can easily be inferred with only local FL models captured over time. We also show defences like DP may result in FL model utility trade-off with privacy. Therefore, we investigate multiparty computation-based techniques to address this trade-off in future work.

Acknowledgment

This work is partly supported by European Union in SPATIAL (Grant No: 101021808) and Science Foundation Ireland under CONNECT phase 2 (Grant no. 13/RC/2077_P2) projects.

REFERENCES

[1] A. Keshavarzi and W. van den Hoek, "Edge Intelligence—on the Challenging Road to a Trillion Smart Connected IoT Devices," *IEEE Design & Test*, vol. 36, no. 2, pp. 41–64, 2019.

[2] X. Yin, Y. Zhu, and J. Hu, "A comprehensive survey of privacy-preserving federated learning: A taxonomy, review, and future directions," *ACM Computing Surveys (CSUR)*, vol. 54, no. 6, pp. 1–36, 2021.

[3] V. Mothukuri, R. M. Parizi, S. Pouriyeh, Y. Huang, A. Dehghantanha, and G. Srivastava, "A Survey on Security and Privacy of Federated Learning," *Future Generation Computer Systems*, vol. 115, 2021.

[4] H. Hu, Z. Salcic, L. Sun, G. Dobbie, and X. Zhang, "Source Inference Attacks in Federated Learning," in *2021 IEEE International Conference on Data Mining (ICDM)*. IEEE, 2021, pp. 1102–1107.

[5] J. Gao, B. Hou, X. Guo, Z. Liu, Y. Zhang, K. Chen, and J. Li, "Secure Aggregation is Insecure: Category Inference Attack on Federated Learning," *IEEE Transactions on Dependable and Secure Computing*, 2021.

[6] Z. Wang, Y. Huang, M. Song, L. Wu, F. Xue, and K. Ren, "Poisoning-assisted Property Inference Attack against Federated Learning," *IEEE Transactions on Dependable and Secure Computing*, 2022.

[7] L. Melis, C. Song, E. De Cristofaro, and V. Shmatikov, "Exploiting Unintended Feature Leakage in Collaborative Learning," in *2019 IEEE symposium on security and privacy (SP)*. IEEE, 2019, pp. 691–706.

[8] A. Pustozerova and R. Mayer, "Information Leaks in Federated Learning," in *Proceedings of the Network and Distributed System Security Symposium*, vol. 10, 2020.

[9] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership Inference Attacks against Machine Learning Models," in *2017 IEEE symposium on security and privacy (SP)*. IEEE, 2017, pp. 3–18.

[10] J. Zhang, J. Zhang, J. Chen, and S. Yu, "GAN Enhanced Membership Inference: A Passive Local Attack in Federated Learning," in *IEEE International Conference on Communications (ICC)*. IEEE, 2020.

[11] C. Sandeepa, C. Moremada, N. Dissanayaka, T. Gamage, and M. Liyanage, "An emergency situation detection system for ambient assisted living," in *2020 IEEE International Conference on Communications Workshops (ICC Workshops)*. IEEE, 2020, pp. 1–6.

[12] F. Meneghello, M. Calore, D. Zucchetto, M. Polese, and A. Zanella, "IoT: Internet of Threats? A Survey of Practical Security Vulnerabilities in Real IoT Devices," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 8182–8201, 2019.

[13] G. Somani, M. S. Gaur, D. Sanghi, M. Conti, M. Rajarajan, and R. Buyya, "Combating DDoS Attacks in the Cloud: Requirements, Trends, and Future Directions," *IEEE Cloud Computing*, 2017.

[14] S. Samarakoon, Y. Siriwardhana, P. Porambage, M. Liyanage, S.-Y. Chang, J. Kim, J. Kim, and M. Ylianttila, "5g-nidd: A comprehensive network intrusion detection dataset generated over 5g wireless network," *arXiv preprint arXiv:2212.01298*, 2022.

[15] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep Learning with Differential Privacy," in *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, 2016, pp. 308–318.

[16] K. Wei, J. Li, M. Ding, C. Ma, H. H. Yang, F. Farokhi, S. Jin, T. Q. Quek, and H. V. Poor, "Federated Learning with Differential Privacy: Algorithms and Performance Analysis," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 3454–3469, 2020.