

DeSMP: Differential Privacy-exploited Stealthy Model Poisoning Attacks in Federated Learning

Md Tamjid Hossain, Shafkat Islam, Shahriar Badsha, Haoting Shen
University of Nevada, Reno, NV, USA
Email: {mdtamjidh, shafkat}@nevada.unr.edu, {sbadsha, hshen}@unr.edu

Abstract—Federated learning (FL) has become an emerging machine learning technique lately due to its efficacy in safeguarding the client’s confidential information. Nevertheless, despite the inherent and additional privacy-preserving mechanisms (e.g., differential privacy, secure multi-party computation, etc.), the FL models are still vulnerable to various privacy-violating and security-compromising attacks (e.g., data or model poisoning) due to their numerous attack vectors which in turn, make the models either ineffective or sub-optimal. Existing adversarial models focusing on untargeted model poisoning attacks are not enough stealthy and persistent at the same time because of their conflicting nature (large scale attacks are easier to detect and vice versa) and thus, remain an unsolved research problem in this adversarial learning paradigm. Considering this, in this paper, we analyze this adversarial learning process in an FL setting and show that a stealthy and persistent model poisoning attack can be conducted exploiting the differential noise. More specifically, we develop an unprecedented DP-exploited stealthy model poisoning (DeSMP) attack for FL models. Our empirical analysis on both the classification and regression tasks using two popular datasets reflects the effectiveness of the proposed DeSMP attack. Moreover, we develop a novel reinforcement learning (RL)-based defense strategy against such model poisoning attacks which can intelligently and dynamically select the privacy level of the FL models to minimize the DeSMP attack surface and facilitate the attack detection.

Index Terms—Privacy, Security, Differential Privacy (DP), Federated Learning (FL), Reinforcement Learning (RL)

I. INTRODUCTION

Federated Learning (FL), also known as collaborative learning, has caught a lot of attention from the research community since it has been first introduced back in 2016 by McMahan et al. [1]. It is mostly because of the inherent privacy protection that FL offers to its users. In the FL process, a model is trained on a diffuse network of edge nodes using their local data; rather than the traditional centralized training fashion. This provides a level of data privacy assurance to the users since the confidential data do not leave the edge nodes.

However, the process of FL can be vulnerable to differential attacks (e.g., membership inference attacks (MIA)) which aim to reveal the sensitive information of a node by analyzing the distributed model parameters [2] or gradients [3]. To alleviate this privacy issue, extensive research have been carried out lately, focusing on developing secure multi-party computation (SMPC) [4], trusted execution environments (TEEs) [5], cryptographic encryption [6]–[8], and differential privacy (DP)-based privacy-preservation techniques

[9]–[11] for FL. Among these, DP is considered a very promising technique to preserve the data privacy and prevent MIA [12]. Existing works along this research line include DP-based distributed SGD [13], local DP (LDP) [14],

Although DP is providing a level of privacy guarantee, an adversary can exploit the DP noise to inject false data into the original data and hide the attack identity exploiting the noise range [15]. In this paper, we investigate this vulnerability of DP-based applications and show that in a differentially private FL setting (we call it ‘DPFL’), a malicious actor can inject the false data either into the differentially private training data (i.e., data poisoning attack [16] or into the model parameters (i.e., model poisoning attack [11], [17]). More specifically, we demonstrate a stealthy model poisoning attack in the FL model exploiting the noise of the DP mechanism that (1) reduces the overall accuracy of the global federated model, and (2) deceives the traditional anomaly detection mechanisms by hiding the false data into the DP-noise. The results in this paper reveal a new backdoor for stealthy and untargeted model poisoning attacks in FL through the exploitation of the DP mechanism.

A. Motivations

Poisoning attacks in any machine learning (ML) setting can be broadly divided into two major categories: *targeted* and *untargeted* attacks [10]. Targeted poisoning attacks [11], [17] aim to change the outcome or behavior of the model on particular inputs while maintaining a good overall accuracy on all other inputs, thus makes the attack and defense processes more difficult. On the contrary, the untargeted model poisoning attacks [16], [18] have the power to make a model unusable and eventually leads to a denial-of-service attack [18]. For instance, an adversary may perform untargeted attacks on its competitor’s FL model with an intention to make the model unfeasible.

However, traditional untargeted poisoning attacks mainly utilize the hyperparameters of the targeted model to scale up the effectiveness of the malicious model [11]. To attain the goal of poisoning, the adversary may use explicit boosting that deforms the weights’ distribution, however, then it can be easily detected by the server through simple server-side model checking [19]. Hence, untargeted model poisoning attacks in a stealthy manner remain an open problem in FL [20]. Moreover, since an FL system usually consists of a huge number of clients and only a portion of clients are chosen for any particular round [21], the odds of impacting

the global model accuracy significantly by a single malicious contribution is very low. This leads us to the question- “*How can the adversary perform an untargeted model poisoning attacks in a stealthy but persistent fashion?*”. Motivated by this, in this paper, we investigate the DP mechanism as a tool to conduct such adversarial poisoning attacks in FL. In the rest of the paper, the ‘false data injection (FDI)’ attack and ‘model poisoning’ attack is mentioned interchangeably.

B. Contributions

In this paper, we show that the DP mechanism is creating a new attack avenue for stealthy false data injection (FDI) or model poisoning attacks in a DPFL environment. We name this attack model as ‘DP-exploited stealthy model poisoning’ (in short, DeSMP) attacks. Particularly, we make the following contributions:

- We demonstrate that DP, as a privacy-preserving tool, is opening a new backdoor for untargeted model poisoning attacks in the FL setting. Our proposed attack strategy (DeSMP) is stealthy and persistent in nature.
- To tackle the proposed DeSMP attack, we develop a reinforcement learning (RL)-based defense strategy. The proposed RL-based defense approach intelligently selects the differential privacy level for the clients’ model update. It also minimizes the attack vectors and facilitates attack disclosure.

Section II of this paper covers preliminaries of FL and a brief review of the related works while section III outlines the research problem and threat model. Section IV formulates the proposed DeSMP attack and defense model and their working principle. In section V, we analyze and evaluate the effectiveness of our proposed model. Finally, in section VI, we conclude the paper with some future research directions.

II. PRELIMINARIES AND LITERATURE REVIEW

Here, we discuss the basic mechanism of FL while pointing out some significant contrasting contributions between this work and existing notable research work in adversarial FL. Table I describes the major symbols used in this paper.

A. Mechanism of Federated Learning with DP

FL introduces a collaborative zone for training a model among a set of workers. Here, each participating node maintains a local model for its local training dataset. Additionally, FL incorporates a server that aggregates all the local models to form a global model [1]. Furthermore, to tackle MIA through analyzing the model weights, the FL server generally includes a privacy-preserving mechanism such as DP [2]. Here, DP adds the random Laplacian ($LAP(\frac{\Delta f}{\epsilon})$) or Gaussian noise ($\mathcal{N}(\theta = 0, \sigma^2 = \frac{2\ln(1.25/\delta) \cdot (\Delta f)^2}{\epsilon^2})$) to the model weights. Nonetheless, while deploying the DP mechanism, researchers [2], [21] have suggested using norm clipping or early stopping methods to compensate for the high level of random differential noise and prevent the model to be completely unusable. Once a pre-defined testing criterion (e.g., model accuracy is greater than a threshold

TABLE I: List of major symbols and their description

Symbols	Description	Symbols	Description
τ	Accuracy or loss threshold	θ	Mean
μ_a	Attack impact	$\mathcal{D}_{\mathcal{M}}$	Measurement data
γ	Attacker’s tolerance	ζ	Norm of model updates
b	Batch	k	Participating clients in each round
t	Communication round	f_a	PDF of attack distribution
\mathcal{P}_{DP}	DP parameters	f_0	PDF of benign Gaussian distribution
\mathcal{M}_G^S	Final global model	\mathcal{B}	Privacy budget
\mathcal{P}_{FL}	FL parameters	ϵ	Privacy loss
\mathcal{M}_G	Global model	δ	Privacy spent in each round
$\nabla \mathcal{L}$	Gradient descent	\mathcal{P}_{RL}	RL parameters
x	Input data	Δf or S	Sensitivity
\mathcal{D}_{KL}	Kullback-Leibler divergence	σ	Standard deviation
η	Learning rate	\mathcal{K}	Total clients
\mathcal{M}_L	Local Model	w	Weights
m_l	Attacker loss	f_l	Federated loss
R	Agent reward	S	Agent state
a	Action	α	Learning rate of RL agent
π^*	Optimal policy	χ	Discount factor
$Q^*(s, a)$	converged Q table	ψ	Reward balancing parameter

or privacy budget exceeds) is met, the server finalizes the global model and stops the training procedure; otherwise, the training process re-initiates.

B. Adversarial Federated Learning

Although the DP-based FL models do not expose the client’s training data to the rest of the world, there exist several attack vectors that an adversary can exploit to perform malicious modification or gain unauthorized access to confidential information. For instance, there could be some malicious clients who might inspect all messages received from the server and then, in the training phases, selectively poison the local models to reduce the efficiency of the global model [10]. Other examples of the adversarial FL include the targeted and untargeted model poisoning attacks [11], [18]. However, unlike the centralized ML schemes, the FL systems may employ a large number of untrusted devices which may facilitate the training-time attacks and inference-time attacks [10]. In this paper, we focus on one of the powerful attack classes which is an untargeted model poisoning attack [18]. The adversary can conduct this model poisoning attack either by directly manipulating a client’s model or through the widely known man-in-the-middle attack formation leveraging the network and system vulnerabilities [10].

C. Related Research Work

In this part, we discuss some notable prior research related to the untargeted model poisoning attacks and defenses in FL while outlining some contrasting points with ours.

1) Byzantine-robust Aggregation in Adversarial Setting:

Byzantine threat models [22] produce arbitrary outputs for any wrong inputs (either by an honest participant or a malicious actor). These arbitrary outputs can lead to converging the model to a sub-optimal model. Moreover, the Byzantine clients may need to have the white-box access or the non-Byzantine client updates to make their attack stealthy [10]. Nonetheless, to the best of our knowledge, none of the existing works explore the vulnerabilities of the DP-based applications in tailoring such stealthy attacks. In contrast, we demonstrate that the Byzantine clients or the server can conduct stealthy and persistent untargeted model poisoning attacks by hiding behind the DP mechanism. *In particular,*

we demonstrate the DP-exploited stealthy model poisoning (DeSMP) attacks in an untargeted manner for FL models.

2) *DP-assisted FL Frameworks in CPSs*: Another related line of research focus on developing novel FL frameworks for cyber-physical systems (CPSs) such as power IoT [23], internet of vehicles (IoV) [24], smart grids [25] etc. They pave the way for adopting FL into the CPS domain. Particularly, [25] shows that the FL models, coupled with edge computing, perform very efficiently in short-term load forecasting while significantly reducing the networking load compared to a centralized model. *Nevertheless, they do not cover the adversarial analysis of the FL systems for model update poisoning attacks in CPSs.* Since, in CPSs like smart grids, many mission-critical operations depend on the model accuracy, the DP-assisted poisoning attacks may create devastating consequences through the failure of physical layer devices. Therefore, it is non-trivial to investigate the attack surfaces of a DPFL model in CPSs. *In this context, we focus on the adversarial analysis of the DP-technique in the CPS domain, which will facilitate the future development of novel and effective defense strategies.*

3) *Attack Mitigation Strategies in Adversarial DP*: Although some recent works [15], [26] consider active attacks (e.g., FDI attacks, poisoning attacks, etc.) in DP-based CPSs (e.g., smart grids, transportation systems, etc.), they neither discuss the stealthy model poisoning attacks nor develop any defense strategies based on intelligent decision making for differential privacy level through RL. In particular, they discuss and successively solve the optimal FDI attack problems by developing defense mechanisms based on the anomaly detection schemes for the post-attack phases; instead of taking any initiative to reduce the attack surface beforehand.

In contrast, we analyze the correlation of the DP and FL parameters under adversarial settings; then, leveraging the correlation, we facilitate deployment of the desired level of privacy, utility, and security among the participating nodes in a DPFL system through RL. Following the adversarial analysis and our proposed RL-assisted defense strategy, the large-scale poisoning attacks can be detected and the attack surface can be minimized, i.e., the incentive of the attacker can be reduced, which in turn reduces attack motivations while assisting attack prevention. In short, we develop our RL-assisted defense strategy as a part of the design process (pre-attack phase) to prohibit the untargeted model poisoning attacks. To the best of our knowledge, this is the first work that addresses the DP-exploitation issue in FL setting and successively develops the RL-based defense strategy.

III. PROBLEM FORMULATION AND THREAT MODEL

Suppose, we have \mathcal{K} clients, among which k number of clients are selected in each communication round by the server. If the local model updates are $\{\Delta w^1, \Delta w^2, \dots, \Delta w^i\}$, then the global model update at $(t+1)$ communication round is: $\Delta w_{t+1} = \frac{1}{k} \sum_{i=1}^k \Delta w^i$ where the i^{th} local model update at $(t+1)$ round is: $\Delta w_{t+1}^i = w^i - \Delta w_t$. In an alternative fashion, the loss of the predication can also be calculated as $f_i(w) = \ell(x_i, y_i; w)$ where (x_i, y_i) is the examples set and

w represent the weights of global model. Now, according to the FederatedAveraging algorithm [1], the objective of the federated server is to minimize the following function: $\min_{w \in \mathbb{R}^d} f(w)$ where $f(w) = \frac{1}{k} \sum_{i=1}^k f_i(w)$. The server continues the process until the objective is met.

To introduce DP for preventing model privacy leakage while keeping the model usable, we need to (a) clip the local model updates using the median norm of the unclipped contributions (\mathcal{S}) so that the norm is limited and learning is progressing, and (b) add noise from a DP-preserving randomized mechanism (e.g., Laplace or Gaussian mechanism). Therefore, the new global model update with Gaussian noise at $(t+1)$ round becomes: $w_{t+1} = w_t + \frac{1}{k} (\sum_{i=1}^k clip(\Delta w_i, \mathcal{S}) + \mathcal{N}(0, \sigma^2 \mathcal{S}^2))$. Here, σ^2 is the variance and \mathcal{S} is the sensitivity of the dataset with respect to the aggregation operation. The value of \mathcal{S} needs to be selected in an optimal way so that the noise variance stays sufficient while the aggregated weight's distribution remains as close as possible to the original distribution. Following the related previous research [2], [13], we set $\mathcal{S} = median\{\Delta w^i\}_{i \in k}$. We draw the noise from a Gaussian distribution with mean ($\theta = 0$), variance σ^2 and PDF (probability density function) as:

$$f_0(x) = \frac{1}{\sqrt{2\pi}\sigma_x} e^{-\frac{(x-\theta)^2}{2\sigma_x^2}} \quad (1)$$

However, a malicious actor (if presents) may modify (increase or decrease) the randomized noise in such a fashion that would facilitates (a) *maximum damage*, and (b) *avoid detection*. To perform such stealthy but strong malicious modification, the adversary needs to craft a fake noise profile from either the same or at least, similar distribution function as (1). Earlier research on adversarial differential privacy [15], [27] present us with such optimal attack distribution (f_a^*) and impact (μ_a^*) as follows:

$$f_a^*(x) = \frac{1}{\sqrt{2\pi}\sigma_x} e^{-\frac{(x-\theta-\sqrt{2\gamma}\sigma_x)^2}{2\sigma_x^2}} \quad \text{and} \quad \mu_a^* = \theta + \sqrt{2\gamma}\sigma_x \quad (2)$$

Here, a high value of attacker's tolerance (γ) represents that the adversary does not care to be detected whereas a low value γ means the adversary wants to keep a low profile to avoid detection, and thus sacrifices the attack impact (μ_a). More specifically, the adversarial objective of stealthiness can be formulated as: $\mathcal{D}_{\mathcal{KL}}(f_a || f_0) \leq \gamma$, where $\mathcal{D}_{\mathcal{KL}}(f_a || f_0)$ is the Kullback-Leibler divergence between the PDF of attack distributions (f_a) and benign distribution (f_0) and indicates the classifier's ability to correctly identify the inputs. Moreover, it can be inferred from (2) that during a data or model poisoning attack, the optimal attack impact (μ_a^*) is shifting the benign mean from θ to $\theta + \sqrt{2\gamma}\sigma_x$. However, μ_a^* is equal to the actual mean (θ) when γ is zero. In short, it implies that when there is no attack or no DP mechanism, the results (in this case, the model weights) remain intact. The optimal attack distribution of (2), f_a^* has been obtained by solving the functional multi-criteria optimization problem of the attacker (i.e., maximum attack while minimum disclosure) and the defender (i.e., maximum

privacy with maximum utility). Therefore, if the adversary deviates from the strategy as given by (2), he could end up with even lower payoffs [15].

This observation on adversarial DP analysis motivates us to first raise the question- “*what would be the adversarial impact on the DPFL system if the adversary follows the optimal attack strategy, f_a^* ?*” and then, answer it through theoretical and empirical analysis. Moreover, this potential research problem motivates us to develop a novel and effective defense strategy against such attacks using RL-based intelligent differential privacy level selection.

A. Threat Model

Our proposed threat model has been depicted in Fig. 1. Here, we are considering a simplified smart grid data transmission architecture which consists of some edge devices (e.g., distribution energy resources (DERs), intelligent electronic devices (IEDs), phasor measurement units (PMUs), etc.), data aggregators (e.g., phasor data concentrators (PDCs)), and a central server. The adversary can mark his presence in- (1) the edge nodes (i.e. disguise as an edge device), (2) the communication pathway between the clients and the server, and (3) the server-side. In case of data poisoning attacks, it is convenient for the adversary to compromise some edge devices (i.e., position 1), manipulates local training data, and disguises them as honest edge nodes. However, for model poisoning attacks, the suitable positions for the attacker are positions 2 and 3 since from those positions, the adversary can directly manipulate the FL models through compromising the communication path, sieging the model parameters, and then injecting fake noise into the parameters.

In the proposed setting, we assume that the adversary can manipulate the model updates regardless of the attack vectors (i.e., through man-in-the-middle or server-side attack formation). However, the adversary cannot directly change the models that are already on the server. He has white-box access (i.e., full knowledge of the global and local model parameters). The adversary might have partial knowledge of the training and testing data (i.e., distributions of the data); however, this is not a strict requirement in our threat model. In addition, we assume that the adversary has the knowledge of the imposed DP mechanism and privacy budget (ϵ). This assumption is particularly important and realistic as many researchers including Dwork et al. [28] emphasize the necessity of publishing the privacy budget in order to increase the trustworthiness of the system.

IV. MODELING DeSMP ATTACK AND DEFENSE IN DPFL

In this section, we first describe the methodology of our proposed system development from an algorithmic point of view, and then, we model the proposed DeSMP attack and RL-assisted defense strategy.

A. Development of DPFL Systems

As discussed in section II-A, in a DPFL system, the global model is first constructed by aggregating all the local models from the randomly selected clients, and then, DP-noise is

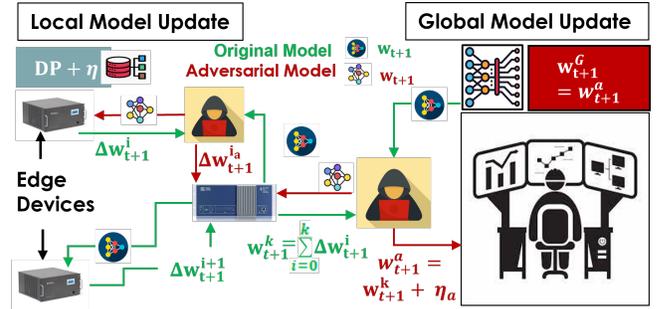


Fig. 1: Threat model: The adversary is exploiting DP to inject false data into the model weights by compromising either the communication path or acting as a server.

added into the model parameters to obfuscate the individual contribution of the clients. The working principle of a DPFL system with RL-based privacy selection is described through the pseudocodes of algorithm 1. The algorithm simply takes the measured data and the parameters of FL, RL, and DP as input. Then, through some intermediary functions (i.e., \mathcal{LM} : local model, \mathcal{RL} : reinforcement learning model, \mathcal{GM} : global model), the global model is computed. If the computed global model passes the accuracy-test (i.e., accuracy is more than a pre-defined threshold, τ), the global model is finalized and the DPFL process completes.

1) *Local Model (\mathcal{LM}):* The \mathcal{LM} function takes the measurement data and learning parameters as inputs. Each client shares a portion of data (i.e., mini-batch) and train the global model with their local data. Finally, the local model and norm updates are calculated and sent back to the server.

2) *Reinforcement Learning Model (\mathcal{RL}):* The purpose of the \mathcal{RL} function is to generate the optimal policy for determining the privacy budget (ϵ) considering the trade-off among the privacy, utility, and security in a DPFL system. The input of this function is the state of the system which comprises of (m_l, f_l, ϵ) . The function exploits the converged Q-table to determine the optimal action (or value of ϵ) at each state of the learning process.

3) *Global Model (\mathcal{GM}):* The sole purpose of \mathcal{GM} function is to produce the global model (\mathcal{M}_G) after each communication round through FederatedAveraging procedure [1] until the model finally converges around a pre-defined threshold value, τ . The function also checks if the privacy budget is expired on it. Another important task of this function is to clip the gradient to avoid over-fitting or gradient exploding and add Gaussian noise accordingly.

B. Modeling DeSMP Attack

To perform the proposed DeSMP attack, the adversary needs to choose the level of his stealthiness or attacker’s tolerance (γ). Here, the adversarial goal is to perform the attack so that the model is unusable and ineffective (i.e., converges to a bad-minimum or starts denial-of-service) and the attack is stealthy. For instance, in a classification problem, if the test inputs are $\{X_i\}_{i=1}^n$, output labels are $\{Y_i\}_{i=1}^n$, global weight vector is \mathcal{W}_G^t , global model is \mathcal{M}_G^{PS} , benign and attack

Algorithm 1: DP- and RL- assisted FL process

Inputs: $\mathcal{D}_{\mathcal{M}}^k, \mathcal{P}_{\mathcal{FL}}, \mathcal{P}_{\mathcal{DP}}, \mathcal{P}_{\mathcal{RL}}$ **Output:** Final Global FL model ($\mathcal{M}_{\mathcal{G}}^{\mathcal{PS}}$)**Function** $\mathcal{LM}(\mathcal{D}_{\mathcal{M}}^k, \mathcal{P}_{\mathcal{FL}})$:

```
 $w^k \leftarrow w_t \leftarrow \mathcal{P}_{\mathcal{FL}};$ 
for local epoch  $i = 1, 2, \dots, n$  do
  for batch  $b_i^k \in \mathcal{D}_{\mathcal{M}}^k$  do
     $w^k \leftarrow w^k - \eta \nabla \mathcal{L}(w^k, b_i^k)$ 
  end
end
return
 $\mathcal{M}_{\mathcal{L}}^k \leftarrow (\Delta w_{t+1}^k, \zeta^k) \leftarrow (w^k - w_t, \|\Delta w_{t+1}^k\|_2);$ 
```

End Function**Function** $\mathcal{RL}(\mathcal{M}_{\mathcal{L}}^k, \mathcal{P}_{\mathcal{RL}})$:

```
 $S_t = (m_l, f_l, \varepsilon_0) \leftarrow (\mathcal{M}_{\mathcal{L}}^k, \mathcal{P}_{\mathcal{RL}});$ 
Choose action using epsilon-greedy policy;
Observe  $R_{t+1}, S_{t+1};$ 
 $\pi^*(s) = \arg \max_{\pi} Q^*(s, a);$ 
return  $a \leftarrow \pi^*(s);$ 
```

End Function**Function** $\mathcal{GM}(\mathcal{M}_{\mathcal{L}}^k, \varepsilon, \mathcal{P}_{\mathcal{DP}})$:

```
 $(\Delta w_{t+1}^k, \zeta^k) \leftarrow \mathcal{M}_{\mathcal{L}}^k;$ 
 $(\delta, \Delta f, \mathcal{B}) \leftarrow \mathcal{P}_{\mathcal{DP}}, n \leftarrow \text{count}(k);$ 
 $\sigma \leftarrow \{\varepsilon, \delta, \Delta f\}, S^k = \text{median}\{\zeta^k\}_{k \in \mathcal{K}};$ 
if  $\delta > \mathcal{B}$  then return  $w_t$ ;
else
 $w_{t+1}^k \leftarrow w_t^k + \frac{1}{n} (\sum_{k=1}^{\mathcal{K}} \frac{\Delta w_{t+1}^k}{\max(1, \frac{\zeta^k}{S})} + \mathcal{N}(0, \sigma^2 S^2));$ 
return  $\mathcal{M}_{\mathcal{G}} \leftarrow w_{t+1}^k;$ 
```

End Function**while** $\text{TestAccuracy}(\mathcal{M}_{\mathcal{G}}) < \tau$ **do**

```
if  $\mathcal{D}_{\mathcal{M}}$  is available then
   $\mathcal{M}_{\mathcal{L}}^k = \mathcal{LM}(\mathcal{D}_{\mathcal{M}}^k, \mathcal{P}_{\mathcal{FL}})$ 
   $\varepsilon = \mathcal{RL}(\mathcal{M}_{\mathcal{L}}^k, \mathcal{P}_{\mathcal{RL}})$ 
   $\mathcal{M}_{\mathcal{G}} = \mathcal{GM}(\mathcal{M}_{\mathcal{L}}^k, \varepsilon, \mathcal{P}_{\mathcal{DP}})$ 
else
  wait for  $\mathcal{D}_{\mathcal{M}}^k$  to be available
end
```

end**return** $\mathcal{M}_{\mathcal{G}}^{\mathcal{PS}} = \mathcal{M}_{\mathcal{G}}$

distributions are f_0 and f_a respectively, then the adversarial objective is-

$$\mathcal{A}(W_{\mathcal{G}}^t) = \max_{f_a} \sum_{i=1}^n [\mathcal{M}_{\mathcal{G}}^{\mathcal{PS}}(X_i) - Y_i] \quad (3)$$
$$\text{s.t. } \mathcal{D}_{\mathcal{KL}}(f_a \| f_0) \leq \gamma$$

It means the adversary wants to maximize the number of misclassification ($[\mathcal{M}_{\mathcal{G}}^{\mathcal{PS}}(X_i) - Y_i]$) while keeping $\mathcal{D}_{\mathcal{KL}}(f_a \| f_0)$ divergence value below his tolerance level (γ). To achieve this goal, the adversary carefully selects the tolerance value (γ) and draws noise from the optimal attack distribution, f_a^* as represented by (2). In other words, the adversary replaces the benign Gaussian noise mechanism, $\mathcal{N}(\theta, \sigma^2 S^2)$ by malicious noise adding mechanism

$\mathcal{N}_a(\theta + \sqrt{2\gamma}\sigma, \sigma^2 S^2)$ following (2). Here, θ represents the mean value or location parameter of the Gaussian distribution while $\sigma^2 S^2$ indicates the scaling factor of the same distribution. By controlling the value of the tolerance level (γ), the adversary can control the attack impact level (μ_a) and shift the mean value further from the actual value (i.e., θ to $\theta + \sqrt{2\gamma}\sigma$). In short, increasing/decreasing the value of γ increases/decreases the level of noise and vice versa. Nevertheless, since the attack distribution (f_a^*) follows the same statistical properties of a benign distribution (f_0), the adversarial noise (\mathcal{N}_a) as well as the poisonous weights will not be very different statistically from other weights. More specifically, unless the adversary chooses a very large γ , the proposed DeSMP attack will achieve stealthiness while remaining persistent. We empirically observe and evaluate the proposed DeSMP attack on the FL models in section V.

C. Modeling RL-assisted Defense Strategy:

RL [30] is an adaptive ML algorithm that can facilitate conventional mechanisms with intelligence without the need for any supervision. Distinguishable attributes of RL is a feedback loop (or trial and error) based on the search for optimal action set and delayed rewards. These attributes motivate researchers in deploying RL in divergent sectors, i.e., mmWave communications, smart grid, IoV [31], etc.

The addition of DP during the training process will enable the adversary in launching stealthy FDI or poisoning attacks. Moreover, DP will cause degradation in federated accuracy which is difficult to understand and balance the trade-off between privacy, and model performance, both theoretically and empirically [32]. On top of this, the FDI attack vector extends the requirement for a trade-off among three different parameters, e.g., privacy, utility, and security. Therefore, selecting the privacy loss (ε) level optimally is a crucial requirement in a DPFL system considering the privacy, utility, and security aspects. Our proposed RL-based model assists this optimal privacy policy selection process. Moreover, it defends the learning process from the DeSMP attacks by reducing the incentive of the adversary, which in turn reduces attack motivations while assisting attack prevention. In short, in this paper, $S = (m_l, f_l, \varepsilon)$.

Action Space: We assume that the agent makes a decision in an event-driven manner. By observing the federated environment's current state, the agent makes one of the decisions as described in the action set (A). We can define the action-space as, $A = \{\text{increase, decrease, static}\}$. To fine grain the agent's action making process, we assume that the agent can increase or decrease privacy loss (ε) by multiple steps (alternatively, a single unit or double unit at any state).

Reward Function: Reward motivates an agent to make decision towards the learning objectives. For defense against DeSMP attack, the objective for the agent is to minimize the maximum attack accuracy as well as maximize the federated accuracy. We assume that the maximum and minimum thresholds are set and regulated by the DPFL system

designer. We define the reward function for the agent as in equation (4),

$$\beta_1 = \psi_1 \frac{f_l^{max}}{f_l} + \psi_2 \frac{m_l^{max}}{m_l} + \psi_3 \frac{1}{\varepsilon} \quad (4)$$

where f_l^{max} and m_l^{max} denotes the maximum value of FDI attack loss and federated loss whereas ψ_1 , ψ_2 , and ψ_3 denotes the balancing parameters.

Here, we use epsilon-greedy policy [33] for determining the trade-off between exploration and exploitation. We set the initial exploration probability at 1.0, and gradually reduce the exploration probability over episodes until it matches with the minimum exploration probability (which we assume 0.05 in this paper).

V. EXPERIMENTAL ANALYSIS

We simulate an FL environment in order to test our proposed algorithm. Moreover, for comprehensive evaluation of our proposed DeSMP attack model, we focus on the persistence, effectiveness, and stealthiness of the proposed attack under different scenarios for two well-known dataset.

A. Dataset Description and Experimental Setup:

We utilize the benchmark dataset MNIST (with Non-I.I.D. distributions) [29], Individual household electric power consumption dataset [34] to evaluate our proposed DeSMP attack. For MNIST, we have used 10,000 test images to evaluate the performance of the attack model whereas In all of the experiments using these two datasets, following the standard FL setup, each selected participants use the SGD (stochastic gradient descent) optimizer to train their local model for internal epoch with local learning rate (η). All of the experiments are done on a server with Intel(R) Core(TM) i7-9700F CPU @ 3.00GHz, 4 NVIDIA GeForce RTX 2060 GPUs with 16 GB RAM each, and Windows 10 (64-bit) OS, with Python 3.8.8 and PyTorch 1.5.1.

B. Deployment and Evaluation of DPFL Model:

To simulate the DPFL environment, we follow some notable prior works [2], [11], [18] and select the value of some major parameters according to the Table II. Moreover, for simplicity, we conduct the experiments with a neural network of three layers. For classification problem (i.e., MNIST), the *Log_Softmax* activation function has been used on top of the *ReLU* function whereas in regression problems, only *ReLU*

TABLE II: Parameters for FL simulation

Parameters/ Dataset	\mathcal{K}	k	b_k^i	\mathcal{B}	ε	i	η
MNIST	100	30	32	0.001	0.1-20	10	0.01
Consumption	100	30	7	0.001	0.1-20	10	0.1

function has been used. To add DP-generated noise into the model weights, we modify the *FederatedAveraging* [1] procedure according to the Algorithm 1. For each experiments, when the privacy budget (\mathcal{B}) exceeds, the learning stops and the server finalizes the global model.

For MNIST, the training (Tr) and validation (Val) loss of three random clients (C1, C2, and C3) in an arbitrary communication round has been depicted in Fig. 2(a). It can be inferred that, in each incremental epoch, the training and validation loss is decreasing. Also, from Fig. 2(b), we can see that the DPFL algorithm converges after a few communication rounds. The final accuracy value after round 30 is around 0.95. Another important thing to notice is that the privacy budget (\mathcal{B}) is spent very quickly if the ε is small and the model can not converge properly. Therefore, it is significantly important to select the privacy loss and budget level (i.e., ε and \mathcal{B}) intelligently and in an optimal way so that the model possesses the desired level of privacy and utility. Likewise, for Power consumption dataset, we verify the DPFL approach and find similar results. The cost of applying DP (i.e., the ‘privacy cost’) over the global model loss varies with ε . Thus, more privacy leads to more loss for both the classification and regression problems.

C. Implementation and Evaluation of DeSMP Model:

To demonstrate the proposed DeSMP attack, we replace the benign noise addition mechanism (\mathcal{N}) of DP-technique with the adversarial noise addition scheme (\mathcal{N}_a). More specifically, to simulate the behavior of the actual adversary, instead of drawing noise from the benign Gaussian distribution (f_0), now we draw noise from attack distribution (f_a^*). We can see the impact of such model poisoning action through the DP-FDI curves of Fig. 2(b). Due to the addition of malicious noise, the overall accuracy has been decreased. However, the degree of model accuracy largely depends on the attacker’s tolerance level (γ). If the attacker chooses to perform more devastating attacks without paying much attention towards

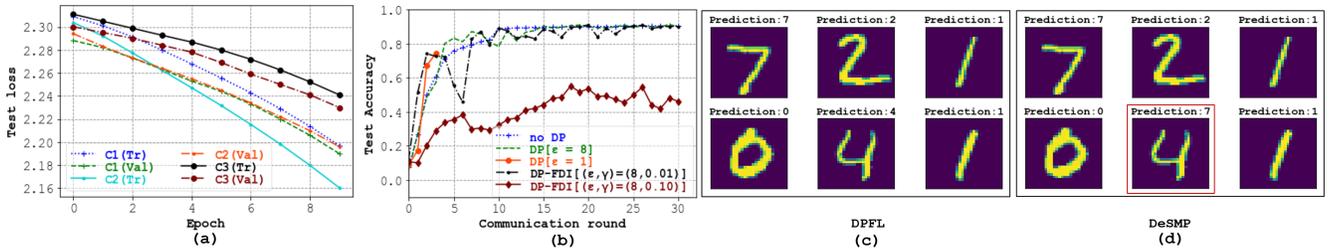


Fig. 2: Evaluation of DeSMP attack model on MNIST dataset [29]: (a) training vs validation loss for three random clients (b) test accuracy for non-DP, DP, and FDI-DP data with varying privacy loss (ε) and attacker’s tolerance (γ) (c) DPFL model prediction (d) generating incorrect prediction due to DeSMP attack

achieving the stealthiness, he would select a large γ (i.e., $\gamma = 0.10$) and in the process, be able to reduce the accuracy largely. In opposite, selecting a small γ (i.e., $\gamma = 0.01$) would give him less payoff in terms of attack impact (μ_a).

In Fig. 2(c) and (d), We can further observe the impact of our proposed approach. Fig. 2(c) reflects the outcome of the proposed DPFL model on some randomly selected MNIST image samples whereas 2(d) depicts the adversarial outcomes through our proposed (DeSMP) model for the same samples. Due to the stealthy adversarial noise with $\gamma = 0.10$, only the image of digit ‘4’ has been predicted wrongly as digit ‘7’ while the other digits are predicted correctly. Since we are considering the untargeted model poisoning attack, the adversarial action through the proposed DeSMP model may alter the image label differently each time. However, as the overall accuracy does not degrade too much with a low γ , the malicious action becomes stealthy and goes unnoticed by the anomaly detectors.

Likewise, the impacts of adversarial action exploiting the DP noise for Power consumption dataset have been illustrated in Fig. (3). It can be observed that the DeSMP attack is also increasing the loss with respect to the increase in ε and γ value. However, for the regression problem, if the raw training data across all the clients are similar and identically distributed, then the attack requires adding more noise (i.e., small ε and large γ) in order to achieve the desired level of attack impact. For instance, we can see from Fig. 3(a) that even after adding the DP mechanism with different ε , the model converges after a sufficient number of communication rounds. It is also desirable since the privacy preserves and the utility remains satisfactory. However, in the presence of an adversary, the loss starts to increase. This phenomena can be observed in Fig. 3(b)-(d). Moreover, as ε starts to decrease (i.e., privacy increases) from 8.0 to 1.0, the attacker obtains more attack opportunities. From Fig. 3(b), it can be inferred that the shifting from $(\varepsilon, \gamma) = (8, 0.01)$ to $(\varepsilon, \gamma) = (8, 0.10)$ is increasing the loss by 7 times (i.e., 0.01 to 0.07) whereas in Fig. 3(c), shifting from $(\varepsilon, \gamma) = (4, 0.01)$ to $(\varepsilon, \gamma) = (4, 0.10)$ is increasing the loss by more than 20 times (i.e., increasing from 0.01 to more than 0.20). Moreover, comparing the red FDI-DP curveS of Fig. 3(b)

and (c), it can be perceived that decreasing ε by half (from 8.0 to 4.0) is increasing the test loss by almost 3 times (0.07 to 0.20) when tolerance level is relatively high ($\gamma = 0.10$).

Therefore, the attack impact (μ_a) increases significantly with the increment of the attacker’s tolerance level, γ , and the model turns to a sub-optimal model. Eventually, through the DeSMP attack, at a very low ε and high γ , the DPFL model becomes unusable and initiates denial-of-service. Nevertheless, the proposed DeSMP model can also be tailored to conduct more devastating attacks while maintaining stealthiness through hyper-parameter tuning and selectively choosing the FL-parameters that are mentioned in Table II.

D. Implementing RL-assisted Privacy Selection

Fig. 4 illustrates the accumulated reward of the defending agent for learning rate ($\alpha = 0.1$) and discount factor ($\chi = 1$) for two distinct datasets (MNIST and Power consumption). The trend in the figure illustrates that the agent learns optimal policy over episodes, and it converges after sufficient episodes are executed. Since we define the reward function such that it takes care of federated loss (f_i), attacker loss (m_i), and privacy loss (ε), this convergence finds the optimal trade-off policy for the privacy, security, and utility of the system. Since the agent outputs an action (or ε) for each state, we can calculate the standard value of federated loss (f_i^s) for that state. Therefore, if the practical or real-time observed federated loss (f_i^p) differs from the expected (or standard) one, we can infer whether the attack is launched or not. Specifically, if the f_i^s is less than f_i^p , we can infer that the large scale (large γ) FDI attack is launched; otherwise, the system is not compromised or the degree of FDI attack scale (low γ) is very low.

VI. CONCLUSION AND FUTURE WORKS

Federated learning (FL) can be vulnerable to privacy-violating and security-compromising attacks despite having privacy-preserving tools like DP. Model update poisoning is one of such attacks. However, stealthy and persistent model poisoning attacks are difficult to achieve. Motivated by this, in this paper, we analyze the adversarial learning process in an FL setting and show that a stealthy and persistent model

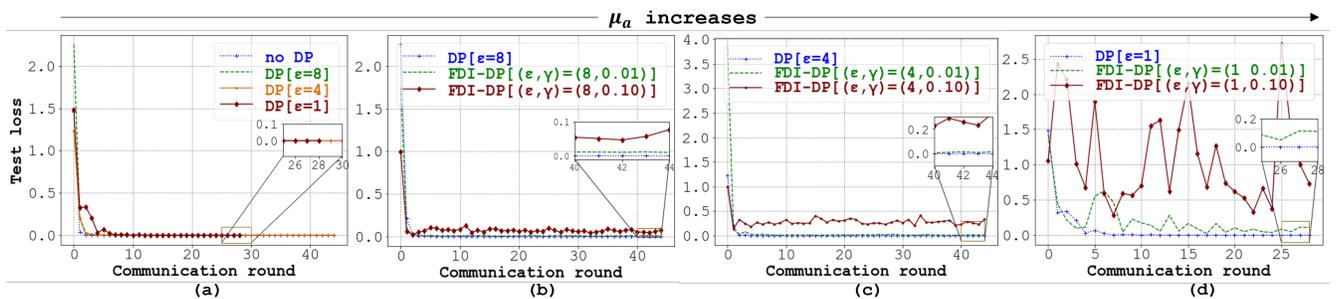


Fig. 3: Evaluation of DeSMP attack model on Individual household electric power consumption dataset [34]: (a) test loss converges even when DP is applied (b) test loss increases as the attacker’s tolerance (γ) increases. (c) more privacy (i.e., small ε) leads to more attack opportunity (d) high privacy and high attacker’s tolerance initiates denial-of-service.

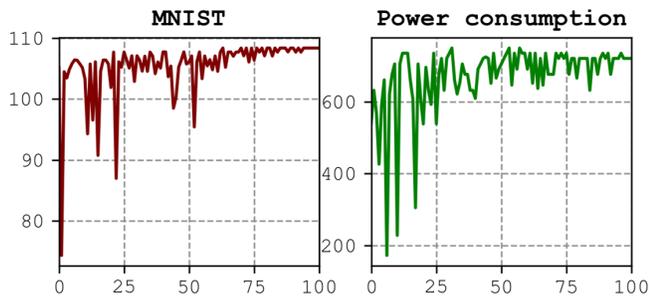


Fig. 4: No. of episodes vs. accumulated rewards for RL assisted privacy selection agent

poisoning attack can be conducted exploiting the differential noise. More specifically, we develop an unprecedented DP-exploited stealthy model poisoning (DeSMP) attack for FL models. Our empirical analysis on both the classification and regression tasks using two popular datasets reflects the effectiveness of the proposed DeSMP attack. Moreover, we develop a reinforcement learning (RL)-based novel defense strategy against such poisoning attacks which can intelligently and dynamically select the privacy policy of the FL models to minimize the DeSMP attack surface, optimize privacy, security, and utility, and facilitate attack detection.

In the future, we will extend our defense model for a collaborative multi-agent setting where the team of clients can exploit the learned policy for collaboratively provisioning privacy during the training phase. Although we focus on the untargeted model poisoning attacks in a DPFL system in this paper, it would be also interesting to investigate the adversarial impact in targeted model poisoning with our proposed DeSMP attack model. We leave it for our future works on adversarial federated learning.

REFERENCES

- [1] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial Intelligence and Statistics*. PMLR, 2017, pp. 1273–1282.
- [2] R. C. Geyer, T. Klein, and M. Nabi, "Differentially private federated learning: A client level perspective," *arXiv preprint arXiv:1712.07557*, 2017.
- [3] L. Zhu and S. Han, "Deep leakage from gradients," in *Federated learning*. Springer, 2020, pp. 17–31.
- [4] Y. Li, Y. Zhou, A. Jolfaei, D. Yu, G. Xu, and X. Zheng, "Privacy-preserving federated learning framework based on chained secure multi-party computing," *IEEE Internet of Things Journal*, 2020.
- [5] F. Mo and H. Haddadi, "Efficient and private federated learning using tee," in *EuroSys*, 2019.
- [6] C. Zhang, S. Li, J. Xia, W. Wang, F. Yan, and Y. Liu, "Batchcrypt: Efficient homomorphic encryption for cross-silo federated learning," in *2020 {USENIX} Annual Technical Conference ({USENIX} {ATC} 20)*, 2020, pp. 493–506.
- [7] F. Sadique, I. Astaburuaga, R. Kaul, S. Sengupta, S. Badsha, J. Schnebly, A. Cassell, J. Springer, N. Latourrette, and S. M. Dasalu, "Cybersecurity information exchange with privacy (cybex-p) and tahoe—a cyberthreat language," *arXiv preprint arXiv:2106.01632*, 2021.
- [8] F. Sadique, K. Bakhshaliyev, J. Springer, and S. Sengupta, "A system architecture of cybersecurity information exchange with privacy (cybex-p)," in *2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC)*. IEEE, 2019, pp. 0493–0498.
- [9] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Theory of cryptography conference*. Springer, 2006, pp. 265–284.
- [10] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings *et al.*, "Advances and open problems in federated learning," *arXiv preprint arXiv:1912.04977*, 2019.
- [11] A. N. Bhagoji, S. Chakraborty, P. Mittal, and S. Calo, "Analyzing federated learning through an adversarial lens," in *International Conference on Machine Learning*. PMLR, 2019, pp. 634–643.
- [12] 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.
- [13] 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.
- [14] S. Wang, L. Huang, Y. Nie, X. Zhang, P. Wang, H. Xu, and W. Yang, "Local differential private data aggregation for discrete distribution estimation," *IEEE Transactions on Parallel and Distributed Systems*, vol. 30, no. 9, pp. 2046–2059, 2019.
- [15] J. Giraldo, A. Cardenas, M. Kantarcioglu, and J. Katz, "Adversarial classification under differential privacy," in *Network and Distributed Systems Security (NDSS) Symposium 2020*, 2020.
- [16] B. Biggio, B. Nelson, and P. Laskov, "Poisoning attacks against support vector machines," *arXiv preprint arXiv:1206.6389*, 2012.
- [17] E. Bagdasaryan, A. Veit, Y. Hua, D. Estrin, and V. Shmatikov, "How to backdoor federated learning," in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2020, pp. 2938–2948.
- [18] M. Fang, X. Cao, J. Jia, and N. Gong, "Local model poisoning attacks to byzantine-robust federated learning," in *29th {USENIX} Security Symposium ({USENIX} Security 20)*, 2020, pp. 1605–1622.
- [19] K. Pillutla, S. M. Kakade, and Z. Harchaoui, "Robust aggregation for federated learning," *arXiv preprint arXiv:1912.13445*, 2019.
- [20] X. Zhou, M. Xu, Y. Wu, and N. Zheng, "Deep model poisoning attack on federated learning," *Future Internet*, vol. 13, no. 3, p. 73, 2021.
- [21] Z. Sun, P. Kairouz, A. T. Suresh, and H. B. McMahan, "Can you really backdoor federated learning?" *arXiv preprint arXiv:1911.07963*, 2019.
- [22] R. Guerraoui, S. Rouault *et al.*, "The hidden vulnerability of distributed learning in byzantium," in *International Conference on Machine Learning*. PMLR, 2018, pp. 3521–3530.
- [23] H. Cao, S. Liu, R. Zhao, and X. Xiong, "Ifed: A novel federated learning framework for local differential privacy in power internet of things," *International Journal of Distributed Sensor Networks*, vol. 16, no. 5, p. 1550147720919698, 2020.
- [24] Y. Zhao, J. Zhao, M. Yang, T. Wang, N. Wang, L. Lyu, D. Niyato, and K.-Y. Lam, "Local differential privacy based federated learning for internet of things," *IEEE Internet of Things Journal*, 2020.
- [25] A. Taïk and S. Cherkaoui, "Electrical load forecasting using edge computing and federated learning," in *ICC 2020-2020 IEEE International Conference on Communications (ICC)*. IEEE, 2020, pp. 1–6.
- [26] F. Farokhi and P. M. Esfahani, "Security versus privacy," in *2018 IEEE Conference on Decision and Control (CDC)*. IEEE, 2018, pp. 7101–7106.
- [27] M. T. Hossain, S. Badsha, and H. Shen, "Privacy, security and utility analysis of differentially private CPES data." submitted for publication, 2021.
- [28] C. Dwork, N. Kohli, and D. Mulligan, "Differential privacy in practice: Expose your epsilons!" *Journal of Privacy and Confidentiality*, vol. 9, no. 2, 2019.
- [29] L. Deng, "The mnist database of handwritten digit images for machine learning research [best of the web]," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 141–142, 2012.
- [30] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.
- [31] S. Islam, S. Badsha, and S. Sengupta, "Context-aware fine-grained task scheduling at vehicular edges: An extreme reinforcement learning based dynamic approach," in *2021 IEEE 22nd International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, 2021, pp. 31–40.
- [32] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE Signal Processing Magazine*, vol. 37, no. 3, pp. 50–60, 2020.

- [33] M. Wunder, M. L. Littman, and M. Babes, "Classes of multiagent q-learning dynamics with epsilon-greedy exploration," in *ICML*, 2010.
- [34] G. Hebrail and A. Berard, "Individual household electric power consumption data set," *É. d. France, Ed., ed: UCI Machine Learning Repository*, 2012.