Manipulating Supply Chain Demand Forecasting with Targeted Poisoning Attacks

Jian Chen, *Student Member, IEEE*, Yuan Gao, Jinyong Shan, Kai Peng, Chen Wang, *Senior Member, IEEE*, and Hongbo Jiang, *Senior Member, IEEE*

***Abstract*—Demand forecasting plays an essential role in supply chain management, as it provides an estimate of the goods that customers are expected to purchase in the foreseeable future. While machine learning techniques are widely used for building demand forecasting models, they also become more susceptible to data poisoning attacks. In this paper, we study the vulnerability of targeted poisoning attacks for linear regression demand forecasting models, where the attacker controls the behavior of forecasting models on a specific target sample without compromising the overall forecasting performance. We devise a gradient-optimization framework for targeted regression poisoning in white-box settings, and further design a regression value manipulation strategy for targeted poisoning in black-box settings. We also discuss some possible countermeasures to defend against our attacks. Extensive experiments are conducted on two real-world datasets with four linear regression models. The results demonstrate that our attacks are very effective, and can achieve a high prediction deviation with control of less than 1% of the training samples.**

***Index Terms*—Poisoning attack, demand forecasting, supply chain, linear regression.**

I. Introduction

Over the past few decades, demand forecasting (DF) has become a vital component in supply chain management [1]. It enables system operators to make better strategic decisions on rationally allocating manufactured resources and logistics scheduling for deliveries in the face of uncertainty and volatility [2]. Various supply chain systems leverage emerging DF technologies in improving the availability of logistics operations and the accurate demands for products. Consequently, precise DF directly impacts the reliability of the supply chain systems, and low accurate forecasting may cause unnecessary costs in logistics operations and unexpected demands.

With growing promising technologies into the DF side [3], how to improve the accuracy of DF models has always been vigorously pursue. Fortunately, various advanced machine learning algorithms have received wide attentions and become a vital tool to address DF issues in supply chains [4], [5]. They help system operators perform the DF task without being explicitly programmed and automatically learn the patterns from the historical samples.

Despite great convenience to obtain accurate DF models, software supply chains turn out to be vulnerable to data poisoning attacks [6]–[8], where attackers could corrupt the DF model by injecting deliberately crafted malicious code into the existing logistics information systems. These types of attacks may pose serious risks in various logistics software, which can lead to over-forecasts that demand unnecessary costs for maintenance and cause system operators to make unreasonable logistics scheduling. Moreover, regarding the supply chain management, autoregressive integrated moving average (ARIMA)-based systems have been applied to a wide range of time-series forecasting scenarios [9], [10]. The attacker can deliberately manipulate the current-moment information, which would then be collected by the ARIMA-based system and allowed to make false forecasting in the future.

In this paper, we study the vulnerability of targeted poisoning attacks for linear regression DF models in supply chains. As fundamental methods to improve the forecasting accuracy, regression-based techniques have been widely used in both academia and industry. For instance, regression-based models have revealed powerful prediction capability than other machine learning models and won the top place in load forecasting competition [11]. However, such regression-based techniques under strong data poisoning attacks models are not yet well understood.

The goal of our attacks is to confuse the behavior of linear regression DF models on a specific target testing sample without compromising the overall forecasting performance. Unlike existing indiscriminate poisoning attacks, our targeted attacks enhance the attacking ability, in a sense, in real-world DF scenarios. Note that the targeted samples are selected based on the attackers' own needs. For example, in pharmaceutical supply chains, the DF models can be manipulated to alter the sales of one specific medicine, leading to the lack of such an important medicine and threatening human lives. By controlling the feedback of consumers in agricultural supply chains, the DF models can also make wrong decisions on providing specific food. Thus, competitors will lose sales for this specific food, while suppliers supported by the attacker can make huge profit.

This work was supported in part by the National Natural Science Foundation of China under Grants 61872416, 62171189, 62002104 and 62071192; by the Fundamental Research Funds for the Central Universities of China under Grant 2019kfyXJJS017; by the Key Research and Development Program of Hubei Province under Grant 2020BAB120; and by the special fund for Wuhan Yellow Crane Talents (Excellent Young Scholar). *(Corresponding author: Chen Wang.)*

J. Chen, Y. Gao, K. Peng and C. Wang are with the Hubei Key Laboratory of Smart Internet Technology, School of Electronic Information and Communications, Huazhong University of Science and Technology, Wuhan 430074, China. Email: {jianchen, yuangao, pkhust, chenwang}@hust.edu.cn.

J. Shan is with Sudo Technology Co., LTD., Beijing, China. Email: shanjy@sudoprivacy.com.

H. Jiang with College of Computer Science and Electronic Engineering, Hunan University, Changsha 410082 China. Emails: hongbojiang2004@gmail.com.

We first consider white-box targeted poisoning attacks in supply chain DF, where we can obtain all the information about DF models and the training data to be attacked. Under this setting, we propose a gradient-optimization framework based on the formulated bi-level optimization problem. With the above framework, we can then optimize the poisons by maximizing the attacking loss on the target sample and minimizing the victim loss on the testing samples excluding those with Euclidean distance close to that of the target sample. By doing so, the objective function, the feature vectors, and the response variables of the attacker can be identified to influence the specific target sample maximally.

We further consider a more realistic scenario under black-box settings where we cannot obtain any knowledge of the victim DF models and the training data. Since true samples rarely have features out of the feasibility domain and the response variables generally play an important role in DF models, this observation motivates us to develop a simple yet effective black-box attack method. Specifically, we first obtain samples with similar distribution as the original training dataset and then only alter the response variables of samples with the closest Euclidean distance to the target sample. In this way, we can generate effective poisoned data to achieve black-box targeted poisoning attacks in supply chain DF.

We summarize our major contributions as follows:

- To the best of our knowledge, this is the first work on targeted poisoning attacks against linear regression DF models in the scenario of supply chains.
- We devise a gradient-optimization framework for targeted regression poisoning in white-box settings, and further design a regression value manipulation strategy for targeted poisoning in black-box settings.
- We evaluate our attacks extensively on two real-world datasets with four linear regression models. The experimental results illustrate that our attacks are very effective, and can achieve a high prediction deviation with control of less than 1% of the training samples.

The remainder of this paper is organized as follows. Section II formulates the problem. Section III provides the design details of our attacks, followed by the performance evaluation in Section IV. Section V briefly discusses some possible countermeasures, and Section VI reviews related works. Finally, Section VII concludes this paper. The code of our attacks has been released for reproducibility purposes[1].

II. Problem Formulation

In this section, we formally introduce the DF in supply chains and its applied models. On this basis, we present our threat model from the perspective of the goal, the knowledge, and the capability of the attacker.

A. Demand Forecasting Formulation

As a fundamental learning method, linear regression is widely used in many DF applications. Many advanced learning methods, such as random forest (RF), support vector machine (SVM), deep neural networks (DNN), can be regarded as linear expansions of linear regression and have been applied to training DF models. To further secure other advanced learning models against poisoning attacks, it is thus quite important to understand the vulnerabilities in linear regression. Thus, we focus on linear regression methods in supply chain DF and move the first step to explore the targeted poisoning attacks under a supervised setting.

Mathematically, the DF operator would get access to a training dataset $\mathcal{D}_{tr} = \{(\boldsymbol{x_i}, y_i)\}_{i=1}^{n}$ based on historical samples, where $\boldsymbol{x_i} \in [0,1]^d$ are scaled, d-dimensional *feature vectors* and $y_i \in [0,1]$ are *response variables*, for $i \in \{1,\ldots,n\}$. n determines how many historical samples the model operators desire to take into consideration for DF. Clearly, the DF model would have higher precision accuracy with more historical samples, yet bringing more training difficulty.

The function of the linear regression can be described as $f(\boldsymbol{x}, \boldsymbol{\theta}) = \boldsymbol{w}^T\boldsymbol{x} + b$, where $w \in \mathbb{R}^d$ is the feature weights and $b \in \mathbb{R}$ is the bias. The DF model aims to find a function parameterized by $\boldsymbol{\theta} = (w, b) \in \mathbb{R}^{d+1}$. These parameters are selected to minimize the following loss function:

$$\mathcal{L}(\mathcal{D}_{tr}, \boldsymbol{\theta}) = \frac{1}{n}\sum_{i=1}^{n}\left(f(\boldsymbol{x_i}, \boldsymbol{\theta}) - y_i\right)^2 + \lambda\Omega(w)\,, \qquad (1)$$

where $\Omega(w)$ is the regularization term penalizing with different weight values of λ.

During the training phase of the linear regression DF model, the mapping from historical samples to future samples can be learnt. During the prediction phase, samples containing feature vectors are used to forecast future response variables.

B. Demand Forecasting Models

As discussed before, regularization techniques is used to improve the DF model prediction ability and reduce errors on unseen (testing) data for regression problems. Current linear regression methodologies are different with the choice of the regularization term $\Omega(w)$. Particularly, we consider the following four linear regression models in this paper:

1) **Ordinary Least Squares (OLS)**, which uses no regularization with $\Omega(w) = 0$;
2) **Least Absolute Shrinkage and Selection Operator (LASSO)**, which employs ℓ_1-norm regularization $\Omega(w) = \|w\|_1$;
3) **Ridge Regression (RR)**, which applies ℓ_2-norm regularization $\Omega(w) = \frac{1}{2}\|w\|_2^2$;
4) **Elastic-net Regression (ENR)**, which utilizes a combination of l_1-norm and l_2-norm regularization $\Omega(w) = \rho\|w\|_1 + (1-\rho)\frac{1}{2}\|w\|_2^2$, where $\rho \in (0,1)$ is a configurable parameter; we set it to 0.5 as previous works do.

The above four DF models select different regularization terms to prevent overfitting, and have their respective points of focus. In particular, OLS is a low bias model and is suitable to have its variance lowered by adding bias; LASSO performs well on feature selection and helps mitigate multicollinearity and model complexity; RR can also decreases the model complexity and obtains a lower variance by reducing the coefficients; while ENR provides a compromise between RR and LASSO with different coefficients.

[1] https://www.dropbox.com/s/i3ultlmajt6abu3/SCDF-TP-Code.zip?dl=0

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2022.3175958, IEEE Transactions on Industrial Informatics

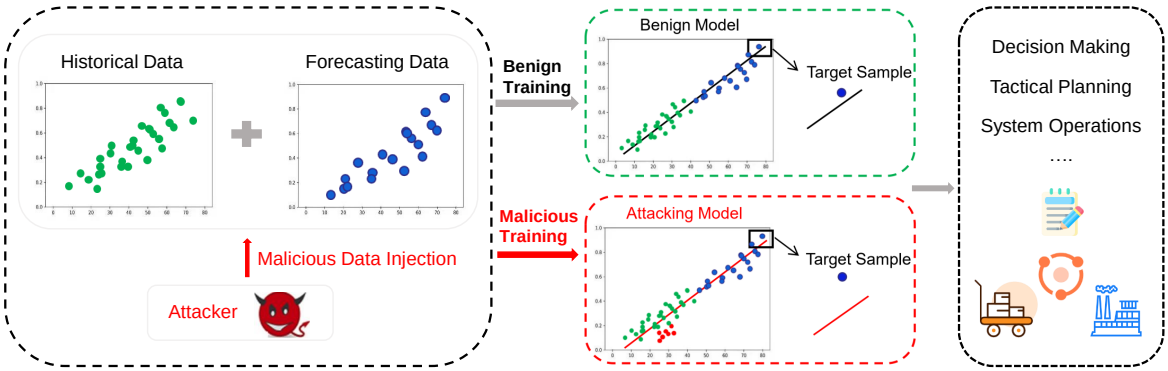


Fig. 1. Overview of our attacks over supply chain DF systems. In the training phase, the attacker tends to inject poisons into the training dataset to manipulate the decision boundary. The training data is the historical data of supply chain systems in the white-box attack, and becomes substitute samples with similar distribution as the historical dataset in the black-box attack. In the testing phase, the poisoned DF model will mislead the specific targeted input into another desired value without compromising the overall forecasting performance.

C. Adversarial Model

1) Adversary's Goal: The attacker's goal is to give rise to incorrect prediction on a particular sample without compromising overall DF performance (a.k.a. poisoning integrity attacks). This is different from poisoning availability attacks, where the goal of the attacker is to affect prediction results indiscriminately and cause the system unavailable [12].

2) Adversary's Knowledge: The attacker could obtain different levels of knowledge of the victim model in previous works. In this work, we assume that the attacker is under both white-box and black-box scenarios. In the white-box scenario, the attacker has the knowledge of the DF model consisting of the following four parts:

1) The training data $\mathcal{D}_{tr}$: The attacker obtains the whole clean training data for the DF model.
2) The feature vectors $\boldsymbol{x}$: The attacker obtains the feature vectors for each clean training data.
3) The regression learning algorithm $\mathcal{L}$: The attacker obtains the regression learning algorithm during training the DF model.
4) The trained parameters $\boldsymbol{\theta}$: The attacker obtains the parameters of the trained DF model.

In the black-box scenario, on the contrary, we assume the attacker obtains no knowledge of the above four parts, but can collects substitute samples with a similar distribution to the original training dataset.

3) Adversary's Capability: The attacker is capable to add n_p poisoned samples into the training samples. Here, n_p is the attacking number and the poisoning ratio can be computed by $n_p/(n_p+n_c)$ with total n_c clean training samples. Generally, the poisoning ratio is lower than 0.2 in previous poisoning availability attacks. In this work, we can manipulate the DF model trained on poisoned samples to alter arbitrary target examples at prediction time by only injecting maliciously-crafted poisons totaling just 1% of the training dataset size.

4) Adversary's Strategy: We aim to cause incorrect prediction on the target sample without corrupting overall DF performance. To do so, we formulate our attack as a bi-level optimization problem. The outer optimization problem amounts to craft the poisoning samples $\mathcal{D}_p$ to maximize the loss function $\mathcal{W}$ on a target sample d_t, while the inner optimization problem corresponds to train the regression model on poisoned samples and clean training samples without those whose Euclidean distance is close to that of the target sample. Thus our attack can be written as:

$$\arg\max_{\mathcal{D}_p} \mathcal{W}(d_t, \boldsymbol{\theta}_p^*), \tag{2}$$

$$\textbf{s.t. } \boldsymbol{\theta}_p^* \in \arg\min_{\boldsymbol{\theta}} \mathcal{L}\left((\mathcal{D}_{tr} - \mathcal{D}_d) \cup \mathcal{D}_p, \boldsymbol{\theta}\right), \tag{3}$$

where $\mathcal{D}_d$ are samples with Euclidean distance close to the target sample.

III. Attack Methodology

In this section, we first present details of the proposed optimization-based framework under the white-box setting for targeted poisoning attacks in the supply chains employing gradient-descent algorithm, and further design a regression value manipulation strategy for black-box targeted poisoning attack, where the attacker obtains no knowledge of DF model parameters and training data (c.f. Fig. 1).

A. White-box Targeted Poisoning Attacks

We first introduce our targeted poisoning attacks under the white-box setting (dubbed WhiTP). Most previous DF models use complex models like neural networks and it is not appropriate to solve our bi-level optimization problem. Nevertheless, the attacker can still craft poisoned feature vectors iteratively by using gradients with respect to each iterations' feature values. Since the model parameters and the training data are both known to the attacker under the white-box setting, it is possible to find the attacking data via solving the optimization problem.

Let us denote $\boldsymbol{x_p}$ as the feature vectors of the original poisoning sample and y_p as its corresponding response variable. To initialize the poisoning samples, according to the previous works on poisoning attacks for classification problems [13], [14], where they randomly select a subset of the training samples and flip their labels, we follow a similar method on regression problems. Specifically, we first randomly select

(a) WhiTP. (b) BlkTP.

Fig. 2. Schematics for the proposed attacks. In WhiTP, the attacker has the access to the historical data and randomly selects part of the historical data as the initial poisoned data. The attacker then optimizes poisons via solving the bi-level optimization problem and alters the feature vectors (X) and the response variables (Y) of the poisoned data simultaneously. In BlkTP, the attacker can obtain the substitute historical data via black-box queries and then randomly selects part of the substitute historical data as the initial poisoned data. After that, the attacker keeps X of the initial poisoned data fixed and only optimizes Y by finding the largest value in a feasible domain.

a subset of the training samples, and then set the response value y_p of each poisoned sample to $round(1-y_p)$, where *round* sets the response variables to 0 or 1. In WhiTP, we optimize multiple poisoned samples at the same time in each iteration. After the initialization of poisoned samples, we then optimize WhiTP through optimization formula as illustrated in Algorithm 1 (c.f. Fig. 2(a)). The feature vectors $\boldsymbol{x_p}$ can be updated by the direction of the gradient $\nabla_{\boldsymbol{x}_p}\mathcal{W}$ of the outer objective $\mathcal{W}$ with fixed step size α in each iteration. We will obtain the optimized poisoned samples when the objective function $\mathcal{W}$ has almost remained unchanged.

B. Gradient Computation

As a gradient-descent algorithm for WhiTP, the most challenging problem is to compute the required gradient $\nabla_{\boldsymbol{x}_p}\mathcal{W}(d_t,\boldsymbol{\theta})$. It is noted that $\mathcal{W}$ depends directly on $\boldsymbol{\theta}$, instead of $\boldsymbol{x}_p$. With the guidance of the chain rule, we can then compute $\nabla_{\boldsymbol{x}_p}\mathcal{W}(d_t,\boldsymbol{\theta})$ by:

$$\nabla_{\boldsymbol{x}_p}\mathcal{W} = \nabla_{\boldsymbol{x}_p}\boldsymbol{\theta}(\boldsymbol{x}_p)^T \cdot \nabla_{\boldsymbol{\theta}}\mathcal{W} \tag{4}$$

Clearly, it is easy to compute $\nabla_{\boldsymbol{\theta}}\mathcal{W}$ because it is the derivative of $\mathcal{W}$ with respect to the model parameters. However, it is still difficult to compute $\nabla_{\boldsymbol{x}_p}\boldsymbol{\theta}(\boldsymbol{x}_p)^T$ with its implicit dependency of $\boldsymbol{\theta}$ on $\boldsymbol{x}_p$. To tackle this issue, motivated by [15], [16], we use Karush-Kuhn-Tucker (KKT) conditions to replace the inner optimization problem with its equilibrium conditions, such that:

$$\nabla_{\boldsymbol{\theta}}\mathcal{L}((\mathcal{D}_{tr}-\mathcal{D}_d)\cup\boldsymbol{x}_p,\boldsymbol{\theta}) = 0, \tag{5}$$

It is guaranteed that such conditions are also valid when updating $\boldsymbol{x}_p$:

$$\nabla_{\boldsymbol{x}_p}(\nabla_{\boldsymbol{\theta}}\mathcal{L}((\mathcal{D}_{tr}-\mathcal{D}_d)\cup\boldsymbol{x}_p,\boldsymbol{\theta})) = 0, \tag{6}$$

In the above formula, we can easily observe that the inner optimization function $\mathcal{L}$ relies directly on $\boldsymbol{x}_p$ and indirectly upon the model parameters $\boldsymbol{\theta}$. Thus, we can use the chain rule to differentiate it, obtaining the following linear system:

$$\nabla_{\boldsymbol{x}_p}\nabla_{\boldsymbol{\theta}}\mathcal{L} + \nabla_{\boldsymbol{x}_p}\boldsymbol{\theta}(\boldsymbol{x}_p)^T \cdot \nabla^2_{\boldsymbol{\theta}}\mathcal{L} = 0, \tag{7}$$

After that, we can solve $\nabla_{\boldsymbol{x}_p}\boldsymbol{\theta}(\boldsymbol{x}_p)^T$ by:

$$\nabla_{\boldsymbol{x}_p}\boldsymbol{\theta}(\boldsymbol{x}_p)^T = \begin{bmatrix} \frac{\partial \boldsymbol{w}}{\partial \boldsymbol{x}_p}^T & \frac{\partial b}{\partial \boldsymbol{x}_p}^T \end{bmatrix} = -\nabla_{\boldsymbol{x}_p}\nabla_{\boldsymbol{\theta}}\left(\nabla^2_{\boldsymbol{\theta}}\right)^{-1}, \tag{8}$$

Also, the aforementioned derivative equals to:

$$\nabla_{\boldsymbol{x}_p}\boldsymbol{\theta}(\boldsymbol{x}_p)^T = -\frac{1}{n}\begin{bmatrix} \mathbf{M} & \boldsymbol{w} \end{bmatrix}\begin{bmatrix} \Sigma+\lambda\boldsymbol{g} & \boldsymbol{\mu} \\ \boldsymbol{\mu}^T & 1 \end{bmatrix}^{-1}, \tag{9}$$

Algorithm 1: White-box Targeted Poisoning Attack

Input: Training data D_{tr}, target data d_t, regression learning algorithm $\mathcal{L}$, loss function $\mathcal{W}$, a small positive constant ϵ, and n_p poisoning samples $\mathcal{D}_p^{(0)} = (\boldsymbol{x}_p^{(j)}, y_p^{(j)})_{j=1}^{n_p}$.

Initialize: $i \leftarrow 0$ (iteration number),
$\boldsymbol{\theta}^{(i)} \leftarrow \arg\min_{\boldsymbol{\theta}} \mathcal{L}((\mathcal{D}_{tr}-\mathcal{D}_d)\cup\mathcal{D}_p^{(i)},\boldsymbol{\theta})$,
Calculate n_d samples close to the Euclidean distance of the target sample: $\mathcal{D}_d$=$\min_{n_d}$(ED($\mathcal{D}_{tr}$,d_t)).

while $|w^{(i)} - w^{(i-1)}| < \epsilon$ **do**
  for j=1,...,n_p **do**
    $\boldsymbol{x}_{p(i+1)}^{(j)} \leftarrow \boldsymbol{x}_{p(i)}^{(j)} - \alpha\nabla_{\boldsymbol{x}_{p(i)}^{(j)}}\mathcal{W}(d_t,\boldsymbol{\theta}^{(i+1)})$;
    $\boldsymbol{\theta}^{i+1} \leftarrow \arg\min_{\boldsymbol{\theta}} \mathcal{L}((\mathcal{D}_{tr}-\mathcal{D}_d)\cup\mathcal{D}_p^{(i+1)},\boldsymbol{\theta})$;
    $w^{(i+1)} \leftarrow \mathcal{W}(d_t,\boldsymbol{\theta}^{(i+1)})$;
  $i \leftarrow i+1$;
  $\mathcal{D}_p \leftarrow \mathcal{D}_p^{(i)}$;

Output A set of poisoned samples $(\{\boldsymbol{x}_p^{(j)}, y_p^{(j)}\}_{j=1}^{n_p})$

where $\boldsymbol{\Sigma} = \frac{1}{n}\sum_i \boldsymbol{x}_i\boldsymbol{x}_i^T$, $\boldsymbol{\mu} = \frac{1}{n}\sum_i \boldsymbol{x}_i$, and $\mathbf{M} = \boldsymbol{x}_c\boldsymbol{w}^T + (f(\boldsymbol{x}_c)-y_c)\mathbb{I}_d$. For OLS and LASSO, the term $\mathbf{g}$ is zero; for RR, $\mathbf{g}$ is the identity matrix $\mathbb{I}_d$; and for ENR, $\mathbf{g}$ is the identity matrix $(1-\alpha)\mathbb{I}_d$.

Until now, we have calculated $\nabla_{\boldsymbol{x}_p}\boldsymbol{\theta}(\boldsymbol{x}_p)^T$. For the second part of $\nabla_{\boldsymbol{x}_p}\mathcal{W}$, we can compute $\nabla_{\boldsymbol{\theta}}\mathcal{W}$ simply by:

$$\nabla_{\boldsymbol{\theta}}\mathcal{W}_t = \begin{bmatrix} \nabla_{\boldsymbol{w}}\mathcal{W}_t \\ \nabla_{\boldsymbol{b}}\mathcal{W}_t \end{bmatrix} = \begin{bmatrix} (f(\boldsymbol{x}_t)-y_t)\boldsymbol{x}_t \\ f(\boldsymbol{x}_t)-y_t \end{bmatrix} \tag{10}$$

Finally, we can understand how to optimize the response variable of each poisoned sample along with its feature vectors during our gradient-based optimization process.

C. Black-box Targeted Poisoning Attacks

WhiTP alters the feature vectors and the response variables simultaneously through solving the bi-level optimization problem. However, the attacker needs to have knowledge of the victim model, which may be unrealistic in real-world settings. Since true data rarely has feature vectors around the corners of the feasibility domain and optimizing the response variables in regression plays an important role in making the attacks more effective, we can thus generate poisoned data by only manipulating the response variables. Motivated by this insight, we further design a black-box targeted poisoning attack called BlkTP as presented in Algorithm 2 (c.f. Fig. 2(b)).

Algorithm 2: Black-box Targeted Poisoning Attack

Input: Substitute training data $D_{tr} = (\boldsymbol{x_i}, y_i)_{i=1}^{k}$, number of poisoned samples n_p, and target data d_t.

Initialize Calculate the feasible domain of the response variable: $[y_{\min}, y_{\max}] \leftarrow y_{\max} = \max(\{y_i\}_{i=1}^{k})$, $y_{\min} = \min(\{y_i\}_{i=1}^{k})$.

Poisoned samples: $(\boldsymbol{x}_p^{(j)}, y_p^{(j)})_{j=1}^{n_p} \leftarrow n_p$ samples in D_{tr} with lowest values of $ED(D_{tr}, d_t)$.

for *j=1,...,n_p* **do**

  $y_p^{(j)} = y_{\max}$;

  fix $\boldsymbol{x}_p^{(j)}$

Output A set of poisoned samples $(\{\boldsymbol{x}_p^{(j)}, y_p^{(j)}\}_{j=1}^{n_p})$

In BlkTP, we assume the attacker does not obtain the training dataset, the regression learning algorithm, and the parameters of the victim DF model, but obtain dataset $\mathcal{D}'_{tr}$ with a similar distribution as the historical dataset. To do so, we assume that we can estimate the mean and covariance of the historical dataset via sufficient number of black-box queries. Then we can simply generate $\mathcal{D}'_{tr}$ by sampling from a multivariate normal distribution. It is important to note that the degree of similarity between the historical dataset and $\mathcal{D}'_{tr}$ has some impact on the effectiveness of BlkTP. Generally, the effectiveness of BlkTP would decrease if the distribution of $\mathcal{D}'_{tr}$ is far away from the that of the historical dataset due to insufficient black-box queries. Owing to the continuity of the response variables, slightly altering their values would have a great impact on the predictions of the linear regression models; thus optimizing the response variables can be more important in making BlkTP more effective. We then only manipulate the response variables instead of the feature vectors in BlkTP.

To manipulate the response variables of the poisoned samples, we first find the most influential samples that can be disturbed. To this end, we search initial samples from $\mathcal{D}'_{tr}$ whose Euclidean distance is close to the target sample, and then alter their response variable values. Meanwhile, BlkTP has to maintain the performance on other samples, so we limit the altered values to a feasibility domain which is defined as the minimum and maximum of all the samples in $\mathcal{D}'_{tr}$. To maximize the effectiveness of BlkTP, we calculate the distance between the response variable values of these initial samples and the minimum (or maximum) value, and then select the larger one as our altered values. Therefore, BlkTP requires no knowledge on the training process than that of WhiTP, but may be slightly less effective.

IV. Performance Evaluation

A. Experiment Setup

1) Datasets: In our experiments, we adopt the following two public regression datasets in supply chains.

Bike Sharing Demand (BSD). This dataset is used to predict the total number of public bikes rented at each hour in the Seoul bike sharing system as a function of predictor variables such as temperature, weather and season. This dataset contains total of $8,761$ samples and 17 features. For preprocessing, all the categorical features are manipulated to one-hot encoding and all the numerical features are normalized.

GEFCom2012. This dataset comes from the 2012 Global Energy Forecasting Competition [17], where the goal is to predict the load data using the temperature information which varies from $2004/1/1$ to $2008/6/30$ hourly. GEFCom2012 contains 4.5 years of load forecasting data for 21 districts including 20 different districts and one district represents the sum of the other 20 districts. Also, the temperature data is gathered from a US utility for 11 stations.

2) Evaluation Metrics: To evaluate the performance of our attacks, we utilize the following two standard metrics:

MAPE (Mean Absolute Percentage Error) rate [18] evaluates the percentage errors of n_t target samples between the poisoned and unpoisoned models, which can be defined as:

$$MAPE = \frac{100\%}{n_t} \times \sum_{i=1}^{n_t} \left| \frac{y_{ep}^i - y_{ec}^i}{y_{ec}^i} \right|, \tag{11}$$

where n_t is the total number of target samples. $y_{ep}^i = y_p^i - y_g^i$ is the prediction error of unpoisoned model for i target sample, where y_p^i is the prediction value of unpoisoned model for i target sample and y_g^i is the groundtruth response variable of i target sample. $y_{ec}^i = y_{pp}^i - y_g^i$ is the prediction error of poisoned model for i target sample, where y_{pp}^i is the prediction value of poisoned model for i target sample. A larger MAPE value indicates that our attack causes successful forecasts.

RMSE (Root Mean Squared Error) rate measures the percentage errors of testing samples excluding the target sample, which can be defined as the absolute percentage errors of the MSE between the poisoned and unpoisoned models. It can be formulated as:

$$RMSE = 100\% \times \left| \frac{\sqrt{\frac{1}{n_T}(y_{pp}^i - y_g^i)^2} - \sqrt{\frac{1}{n_T}(y_p^i - y_g^i)^2}}{\sqrt{\frac{1}{n_T}(y_p^i - y_g^i)^2}} \right|, \tag{12}$$

where n_T is the total number of testing samples, y_{pp}^i is the prediction value of poisoned model for i testing sample, y_p^i is the prediction value of unpoisoned model for i testing sample, and y_g^i is the groundtruth response variable of i testing sample. A smaller RMSE value indicates less impact of our attacks to the model's overall performance.

To evaluate the performance of our attacks on existing defensive methods, we utilize the following standard metric:

MSE (Mean Squared Error) rate measures the percentage errors between the groundtruth response variable and the prediction response variable on the n_t target samples.

$$MSE = \frac{1}{n_t} \times \sum_{i=1}^{n_t} (y_{cp}^i - y_g^i)^2, \tag{13}$$

where n_t is the total number of target samples, y_{cp}^i is the prediction value of the defensed model for ith target sample and y_g^i is the groundtruth response variable of the ith target sample. A larger MSE value indicates our attacks are more resilient to the defensive methods.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2022.3175958, IEEE Transactions on Industrial Informatics

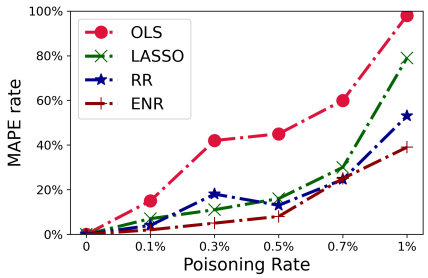


Fig. 3. MAPE rate of WhiTP on four linear regression models with different poisoning rates in BSD dataset.

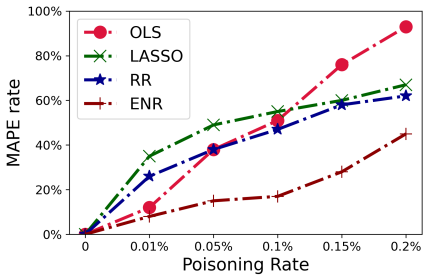


Fig. 4. MAPE rate of WhiTP on four linear regression models with different poisoning rates in GEFCom2012 dataset.

3) Experimental Settings: The standard cross-validation method are used to split these two datasets into 70% for training and 30% for testing. In all experiments, we run the experiments on NVIDIA Geforce GTX 1060 6GB and run 5 times to report the averaged results. We run WhiTP and BlkTP with 50 different targets selected from the testing dataset for BSD to collect performance statistics. To meet time and resource constraints, we report the averaged results for 10 different targets for the GEFCom2012 dataset. We set up four linear regression models, namely, OLS, LASSO, RR, and ENR. Note that these targeted samples are randomly selected in our experimental evaluation. We then train these regressors and measure MAPE and RMSE on the testing datasets.

TABLE I
THE RMSE RATE OF WHITP WITH DIFFERENT POISONING RATES FOR DIFFERENT LINEAR REGRESSION MODELS ON TWO DATASETS.

		Poisoning rate(%)		
Datasets	**Victim models**	0.1	0.2	0.3
BSD	OLS	0.12±0.05	0.13±0.05	0.51±0.30
	LASSO	0.15±0.07	0.23±0.05	1.50±0.50
	RR	0.09±0.02	0.18±0.06	0.72±0.20
	ENR	0.02±0.01	0.08±0.04	0.25±0.07
GEFCom2012	OLS	0.70±0.20	0.80±0.30	1.40±0.40
	LASSO	1.10±0.50	2.00±0.80	2.50±0.60
	RR	0.85±0.50	1.50±0.90	2.30±1.00
	ENR	0.02±0.01	0.08±0.05	0.13±0.05

B. Effectiveness of WhiTP

We first evaluate the effectiveness of WhiTP by comparing the MAPE rate and the RMSE rate of the different linear regression models. Here, we discuss the impact of the poisoning rate for a single target sample on these two datasets. The averaged results for 50 different targets for BSD dataset and 10 different targets (due to time and resource constraints) for GEFCom2012 dataset are illustrated in Figs. 3 and 4.

From Fig. 3, we can observe that the MAPE rate increases as the poisoning rate increases to 1% and can reach 70% on average when the poisoning rate is set to 1%. It is worth noting that the OLS model can outperform the other three models in most cases and the MAPE rate can reach 100% when only 30 poisons (nearly 1% of the training dataset) are injected into the training dataset. This is largely because the OLS model does not have any regularization to generalize well on unseen data and can be easier to be attacked. Furthermore, the MAPE rate of the ENR model is relatively low, because the ENR model is robust than the other three forecasting models and has a strong capability to fit training data with probable regularization so that the attacker can be harder to manipulate this model.

In Fig. 4, we can see that the MAPE rate on four models has similar trend and the averaged MAPE rate on four models can reach 70% when the poisoning rate is set to only 0.2%. Generally, the MAPE rate on four models increases as the poisoning rate increases. It is important to note that GEFCom2012 dataset requires fewer poisons added by the attacker to reach a similar performance than that of BSD dataset.

Furthermore, we present the RMSE rate over the number of poisoned samples of WhiTP against four linear regression models on the bike sharing demand dataset and the GEFCom2012 dataset in Table I. For the bike sharing demand dataset, we can see that the RMSE rate increases as the poisons increase and the RMSE rate keeps relatively low on average. With a lower RMSE rate, more effective as WhiTP becomes. It is noticed that the RMSE rate on the ENR model is the lowest among these four models. The results in Figs. 3 and 4 show that WhiTP can achieve good attacking performance.

C. Impact of Number of Deleting Samples in WhiTP

In WhiTP, we require to delete n_d training samples with the closest Euclidean distance to the target sample. So next, we evaluate the impact of the number of deleting samples in WhiTP and the results are shown in Fig. 5. In our settings, we set the poisoning rate to 1% for BSD dataset and 0.2% for GEFCom2012 dataset. We can observe in Fig. 5 that the MAPE rate will first increase and then slightly decrease as the number of deleting samples increases. WhiTP obtains the highest averaged MAPE rate when the number of deleting samples is 10, which is just what we set by default in our experiments.

D. Effectiveness of BlkTP

We then investigate the effectiveness of BlkTP by comparing the MAPE rate and the RMSE rate on the different linear regression models. Here, we discuss the impact of the number of poisoned samples for a single target sample on these two datasets. We can see in Fig. 6 which shows the MAPE rate over the poisoning rate of BlkTP against four linear

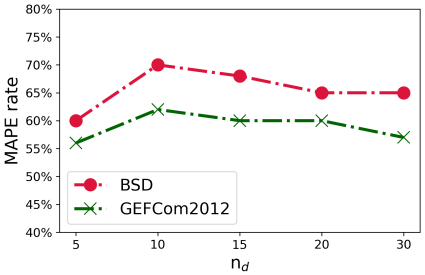


Fig. 5. The MAPE rate of WhiTP on four linear regression models with different deleting samples in BSD and GEFCom2012.

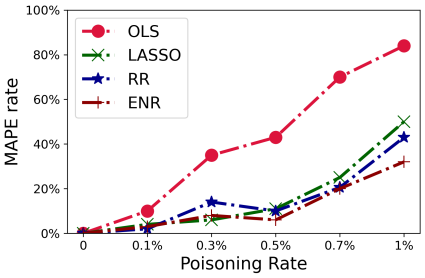


Fig. 6. The MAPE rate of BlkTP on four linear regression models with different poisoning rates in Bike Sharing Demand Dataset.

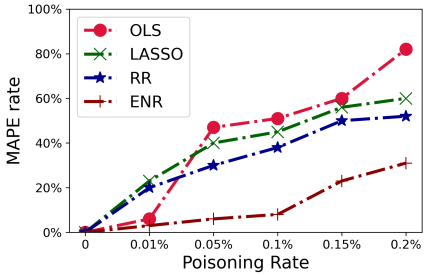


Fig. 7. The MAPE rate of BlkTP on four linear regression models with different poisoning rates in GEFCom2012.

TABLE II
THE RMSE RATE OF BLKTP WITH DIFFERENT POISONING RATES FOR DIFFERENT LINEAR REGRESSION MODELS ON TWO DATASETS.

		Poisoning rate(%)		
Datasets	**Victim models**	0.1	0.2	0.3
BSD	OLS	0.05±0.03	0.06±0.03	0.60±0.20
	LASSO	0.25±0.20	0.43±0.30	1.00±0.50
	RR	0.10±0.10	0.32±0.10	0.70±0.40
	ENR	0.02±0.02	0.08±0.05	0.18±0.10
GEFCom2012	OLS	0.80±0.30	1.60±0.80	2.40±0.70
	LASSO	1.30±0.40	2.20±0.80	2.90±0.50
	RR	0.96±0.50	2.30±1.00	3.10±1.00
	ENR	0.01±0.01	0.10±0.05	0.16±0.06

regression models on BSD dataset. Generally, the MAPE rate of BlkTP has slightly degraded on four linear regression models compared to that of WhiTP. This is not surprising, as BlkTP requires much less information about the training data and forecasting models. Similar to what we observed for WhiTP, the MAPE rate also increases as the poisoning rate increases and the MAPE rate can reach 55% on average when the poisoning rate is set to 1%. Also, the OLS model obtains the highest MAPE rate among these four forecasting models in most cases. The other three forecasting models have similar performance when the poisoning rate alters.

Fig. 7 reveals the MAPE rate over the different poisoning rates of BlkTP against four linear regression models on GEFCom2012 dataset. We can also observe that the OLS model can reach the highest MAPE rate among these four forecasting models and the ENR model obtains the lowest MAPE rate in most cases. However, the averaged MAPE rate can reach 65% when the poisoning rate is only 0.2%, which demonstrates the effectiveness of BlkTP. We notice that the response variable plays an important role in optimizing BlkTP and makes BlkTP more effective in regression models.

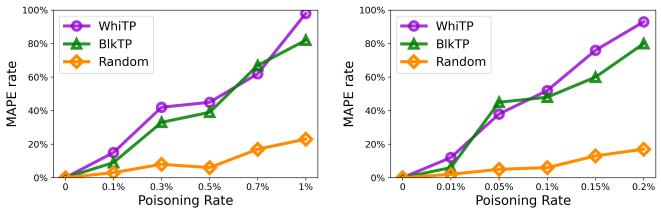
Furthermore, Table II presents the RMSE rate over the number of poisoned samples of BlkTP against four linear regression models. Similar to the cases in WhiTP, the RMSE rate of both datasets are kept lower than 1% on average in most cases. In general, we find that the RMSE rate increases as the poisoning rate increases and the RMSE rate becomes lower as the MAPE rate decreases on different forecasting models. Overall, the experimental results confirm that our framework is very effective at poisoning different linear regression models.

E. Comparisons with Different Poisoning Methods

In this section, we evaluate the effectiveness of different poisoning methods, including WhiTP, BlkTP and poison training data with random Gaussian noises (also called Random in our experiment). To maximize the attacking effectiveness of Random on the target sample and also maintain the performance on other samples, we add random Gaussian noises $n \sim \mathcal{N}(0, 0.01)$ for BSD and GEFCom2012 datasets. The MAPE rate for both datasets of these three poisoned methods has been reported in Fig. 8. We use training data with poisoned data generated by different poisoned methods to train the OLS model. The experimental results show that the MAPE rate of WhiTP and BlkTP are largely higher than that of Random when poisoning rate changes, which indicate the strong attacking effectiveness of WhiTP and BlkTP.

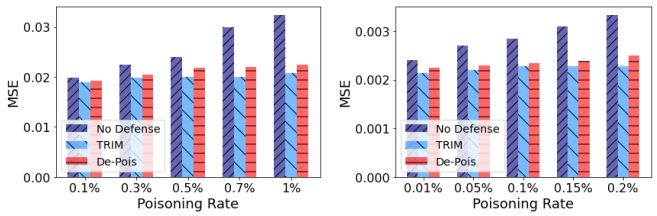
F. Evaluation on Run-time Overhead

Lastly, we evaluate the run-time overhead of both WhiTP and BlkTP. The averaged results on four linear regression models for generating $1,000$ poisons are shown in Table III. For BSD dataset, we can observe that WhiTP takes 1.33×10^2 seconds for generating $1,000$ poisons, while BlkTP needs much less time than WhiTP and it only needs 1.00×10^{-3} seconds to generate poisons. For GEFCom2012 dataset, WhiTP takes 1.09×10^4 seconds for generating $1,000$ poisons and BlkTP still requires less time and only takes 2.00×10^{-3} seconds. Note that WhiTP takes more time on optimizing the poisons and BlkTP saves time by only altering the value of corresponding response variables. As expected, BlkTP is



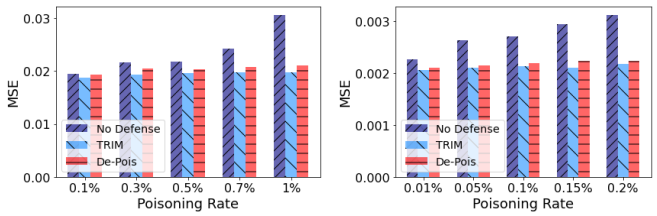
(a) BSD (b) GEFCom2012

Fig. 8. Comparison with different poisoned methods.



(a) BSD (b) GEFCom2012

Fig. 9. MSE of WhiTP against defenses on two datasets.



(a) BSD (b) GEFCom2012

Fig. 10. MSE of BlkTP against defenses on two datasets.

TABLE III
POISON TIME OVERHEAD (S).

Dataset	BSD		GEFCom2012	
Methods	WhiTP	BlkTP	WhiTP	BlkTP
Time	1.33×10^2	1.00×10^{-3}	1.09×10^4	2.00×10^{-3}

much faster on both datasets. It is also noticed that the runtime overhead of both WhiTP and BlkTP can be affected by the dataset size where WhiTP spends a longer time on GEFCom2012 dataset. These experiment results on WhiTP and BlkTP reveal tradeoffs between effectiveness and running time overhead, with WhiTP being more effective than BlkTP, at the expense of higher computational overhead.

G. Performance against Existing Defenses

In this section, we first introduce the existing two state-of-the-art defenses against data poisoning attacks on demand forecasting models. The first defense is TRIM [16], which iteratively trains a regression model with a subset of poisoned data and calculates the error on these samples. TRIM then regards samples with the smallest error in each iteration as clean samples. The other one is De-Pois [19], which is an attack-agnostic defense method by mimicking the target model using cGAN [20] and distinguishing poisoned samples by setting a detection boundary for the discriminator in a conditional version of WGAN-GP [21]. We average the results for four linear regression models used in our experiments and 50 different target samples.

We report the MSE of both WhiTP and BlkTP against these two defenses on BSD and GEFCom2012 in Figs. 9 and 10. As seen in Fig. 9, TRIM and De-Pois have decreased 50% MSE on average and they are not always effective at defending against WhiTP. As the poisoning rate continues increasing, the MSE of TRIM and De-Pois also have increased. The results suggest that while both TRIM and De-Pois reach similar MSE on 50 different target samples, the defense capability of TRIM is slightly superior to De-Pois for both datasets. Compared to the scenario without defense, TRIM and De-Pois can both reduce the MSE of 50 different target samples by 30% when the poisoning rate is 1% on BSD and 0.2% on GEFCom2012. Generally, TRIM performs well than De-Pois at most cases. Furthermore, BlkTP is easier to be defended compared to WhiTP and this is largely because the poisoned samples generated by BlkTP are generally far away from the clean data.

V. POSSIBLE COUNTERMEASURES

Since our goal is to explore the adversarial capability of targeted poisoning attacks on different linear regression DF models and the existing defenses for mitigating the poisoning attacks in regression settings are few involved. We now briefly discuss possible countermeasures to defend against our proposed attacks in this section.

For WhiTP, we manipulate the feature vectors and the response variables simultaneously as discussed before. These generated poisoned samples are generally far away from the original data, and we can thus leverage data sanitization, which can defend against data poisoning attacks by identifying and removing poisoned data using distance similarity strategies. However, data sanitization techniques are still not robust enough against WhiTP because it can also generate poisoned data that are similar to the original data distribution. Therefore, we could apply other strategies (e.g., certified defensive) to data sanitization for mitigating WhiTP and the defensive assumptions on which they rely also need to be considered in practice.

For BlkTP, we only manipulate the response variables and the generated poisoned data are not distinguished from the original dataset. In this way, existing data sanitization methods are not appropriate for defending it. Alternatively, defensive methods that can recover the prediction accuracy of linear regression models are intuitively effective based on the fact that BlkTP aims at decreasing the prediction accuracy on target sample. Thus, a possible countermeasure is to make robust regression learning models and periodically minimize the training loss of the regression learning models. However, the computation overhead for training these models should also be taken into consideration.

VI. Related Work

In this section, we provide a brief existing literature review on both DF and poisoning attacks in the supply chains.

A. DF in Supply Chains

A wide range of DF models has been used to provide better accuracy and to avoid negative consequences in various supply chain areas [22], [23]. Statistical methods like exponential smoothing [24] apply the averaged weight of past observations to make predictions of electric energy consumptions, and the Theta methods [25] can adaptively provide predictions from a combination of the historical data and prediction function. Furthermore, for time-series regression models, the sum of squared errors is often minimized by forecasters using the historical observations. Traditional methods such as extrapolative methods, or autoregressive moving average methods, are used to predict the future demand based on the past pattern with other variables, and can be applied as benchmark models [26]. Step-wise regression method and shrinkage estimation approaches can also be considered to facilitate the predicting.

More recently, machine learning models have received wide attentions for DF. DNN [27] and SVM [28] methods have a strong ability to automatically learn the historical pattern to generate predictive models and outperform traditional methods under uncertain conditions. Especially in DNN, neurons in different layers are connected to find the underlying correlations between the inputs and the outputs, while for time-series regression models generally make continuous-valued functions that can predict the response variable values based on predictor variables. For example, Liu et al. [29] explore the combination of a grey neural network model and a stacked autoencoder to predict logistics demand of transportation disruption. Merkuryeva et al. [30] apply multiple linear regressions for DF in the pharmaceutical area. In other studies, hybrid models [31] are proposed to make robust and accurate forecasts with different levels of volatilities and quantitative information can be extracted out of each algorithm.

B. Data Poisoning Attacks in Supply Chains

Machine-learning-based DF models are vulnerable to a range of security vulnerabilities. We focus on data poisoning attacks here which have been studied in different supply chain scenarios, including industry areas [32] and service areas [33]. Especially, in the load forecasting area, data poisoning attacks have been widely used for failing the load forecasting models to produce accurate load forecasts. For example, data poisoning attacks on state estimation have been explored in [34], where the attackers deliberately craft estimation errors on state variables (e.g., voltage magnitudes and phase angles). Liang et al. [8] explore data poisoning attacks in the load forecasting field and attack the load forecasting models even with anomaly detection. Furthermore, in transportation DF areas, Wu et al. [33] analyze the vulnerability of traffic DF to data poisoning attacks so that the model cannot provide useful guidance for resource scheduling. Though this kind of attacks has been extensively explored in DF supply chains areas, few works have studied poisoning attacks on a single target sample, which is the focus in this paper.

VII. Conclusion

In this paper, we have presented the first study on targeted poisoning attacks in supply chain DF. To control the prediction behavior of forecasting models on a specific target sample without compromising the overall forecasting performance, we formulated our WhiTP attack as a bi-level optimization problem and proposed a gradient-optimization method to solve it. We further design a regression value manipulation strategy for black-box targeted poisoning attack without any knowledge of the training process. Experiment results on two public datasets with four different regression models validate that our attacks are very effective. We believe our efforts may deepen the understanding about the vulnerability in the supply chain DF domain and inspire the design of more efficient detection methods in the near future.

References

[1] M. Seyedan and F. Mafakheri, "Predictive big data analytics for supply chain demand forecasting: methods, applications, and research opportunities," *Journal of Big Data*, vol. 7, no. 1, pp. 1–22, 2020.

[2] Y. Du, Y. Li, C. Duan, H. B. Gooi, and L. Jiang, "Adjustable uncertainty set constrained unit commitment with operation risk reduced through demand response," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 2, pp. 1154–1165, 2020.

[3] A. T. Eseye and M. Lehtonen, "Short-term forecasting of heat demand of buildings for efficient and optimal energy management based on integrated machine learning models," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 12, pp. 7743–7755, 2020.

[4] A. Aamer, L. Eka Yani, and I. Alan Priyatna, "Data analytics in the supply chain management: Review of machine learning applications in demand forecasting," *Operations and Supply Chain Management: An International Journal*, vol. 14, no. 1, pp. 1–13, 2020.

[5] Z. Dou, Y. Sun, Y. Zhang, T. Wang, C. Wu, and S. Fan, "Regional manufacturing industry demand forecasting: A deep learning approach," *Applied Sciences*, vol. 11, no. 13, p. 6199, 2021.

[6] S. Sridhar and M. Govindarasu, "Model-based attack detection and mitigation for automatic generation control," *IEEE Transactions on Smart Grid*, vol. 5, no. 2, pp. 580–591, 2014.

[7] M. Yue, "An integrated anomaly detection method for load forecasting data under cyberattacks," in *Proceedings of IEEE Power & Energy Society General Meeting*, 2017, pp. 1–5.

[8] Y. Liang, D. He, and D. Chen, "Poisoning attack on load forecasting," in *Proceedings of IEEE Innovative Smart Grid Technologies-Asia*, 2019, pp. 1230–1235.

[9] P. Chen, T. Pedersen, B. Bak-Jensen, and Z. Chen, "Arima-based time series model of stochastic wind power generation," *IEEE transactions on power systems*, vol. 25, no. 2, pp. 667–676, 2009.

[10] X. Liu, Z. Lin, and Z. Feng, "Short-term offshore wind speed forecast by seasonal arima-a comparison against gru and lstm," *Energy*, vol. 227, p. 120492, 2021.

[11] N. Charlton and C. Singleton, "A refined parametric model for short term load forecasting," *International Journal of Forecasting*, vol. 30, no. 2, pp. 364–368, 2014.

[12] C. Wang, J. Chen, Y. Yang, X. Ma, and J. Liu, "Poisoning attacks and countermeasures in intelligent networks: status quo and prospects," *Digital Communications and Networks*, vol. 8, pp. 230–239, 2022.

[13] S. Mei and X. Zhu, "Using machine teaching to identify optimal training-set attacks on machine learners," in *Proceedings of AAAI*, 2015.

[14] H. Xiao, B. Biggio, G. Brown, G. Fumera, C. Eckert, and F. Roli, "Is feature selection secure against training data poisoning?" in *Proceedings of ICML*, 2015, pp. 1689–1698.

[15] L. Muñoz-González, B. Biggio, A. Demontis, A. Paudice, V. Wongrassamee, E. C. Lupu, and F. Roli, "Towards poisoning of deep learning algorithms with back-gradient optimization," in *Proceedings of AISec*, 2017, pp. 27–38.

[16] M. Jagielski, A. Oprea, B. Biggio, C. Liu, C. Nita-Rotaru, and B. Li, "Manipulating machine learning: Poisoning attacks and countermeasures for regression learning," in *Proceedings of IEEE S&P*, 2018, pp. 19–35.

[17] T. Hong, P. Pinson, and S. Fan, "Global energy forecasting competition 2012," pp. 357–363, 2014.

[18] J. Luo, T. Hong, and S.-C. Fang, "Benchmarking robustness of load forecasting models under data integrity attacks," *International Journal of Forecasting*, vol. 34, no. 1, pp. 89–104, 2018.

[19] J. Chen, X. Zhang, R. Zhang, C. Wang, and L. Liu, "De-pois: An attack-agnostic defense against data poisoning attacks," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 3412–3425, 2021.

[20] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *CoRR arXiv: 1411.1784*, 2014.

[21] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of wasserstein gans," *Proceedings of NeurIPS*, vol. 30, 2017.

[22] T. Ahmad and H. Chen, "Utility companies strategy for short-term energy demand forecasting using machine learning based models," *Sustainable Cities and Society*, vol. 39, pp. 401–417, 2018.

[23] X. Gao and G. M. Lee, "Moment-based rental prediction for bicycle-sharing transportation systems using a hybrid genetic algorithm and machine learning," *Computers & Industrial Engineering*, vol. 128, pp. 60–69, 2019.

[24] E. M. de Oliveira and F. L. C. Oliveira, "Forecasting mid-long term electric energy consumption through bagging arima and exponential smoothing methods," *Energy*, vol. 144, pp. 776–788, 2018.

[25] E. Spiliotis, V. Assimakopoulos, and S. Makridakis, "Generalizing the theta method for automatic forecasting," *European Journal of Operational Research*, vol. 284, no. 2, pp. 550–558, 2020.

[26] S. Makridakis, E. Spiliotis, and V. Assimakopoulos, "Statistical and machine learning forecasting methods: Concerns and ways forward," *PloS One*, vol. 13, no. 3, p. e0194889, 2018.

[27] A. Husna, S. H. Amin, and B. Shah, "Demand forecasting in supply chain management using different deep learning methods," in *Demand Forecasting and Order Planning in Supply Chains and Humanitarian Logistics*. IGI Global, 2021, pp. 140–170.

[28] Y. K. Semero, J. Zhang, and D. Zheng, "Emd–pso–anfis-based hybrid approach for short-term load forecasting in microgrids," *IET Generation, Transmission & Distribution*, vol. 14, no. 3, pp. 470–475, 2020.

[29] C. Liu, T. Shu, S. Chen, S. Wang, K. K. Lai, and L. Gan, "An improved grey neural network model for predicting transportation disruptions," *Expert Systems with Applications*, vol. 45, pp. 331–340, 2016.

[30] G. Merkuryeva, A. Valberga, and A. Smirnov, "Demand forecasting in pharmaceutical supply chains: A case study," *Procedia Computer Science*, vol. 149, pp. 3–10, 2019.

[31] M. Abolghasemi, E. Beh, G. Tarr, and R. Gerlach, "Demand forecasting in supply chain: The impact of demand volatility in the presence of promotion," *Computers & Industrial Engineering*, vol. 142, p. 106380, 2020.

[32] M. Billah, A. Anwar, Z. Rahman, and S. M. Galib, "Bi-level poisoning attack model and countermeasure for appliance consumption data of smart homes," *Energies*, vol. 14, no. 13, p. 3887, 2021.

[33] Y. Wu, W. Yu, Y. Cui, and C. Lu, "Data integrity attacks against traffic modeling and forecasting in m2m communications," in *Proceedings of ICC*, 2020, pp. 1–6.

[34] J. Tian, B. Wang, Z. Wang, K. Cao, J. Li, and M. Ozay, "Joint adversarial example and false data injection attacks for state estimation in power systems," *IEEE Transactions on Cybernetics*, to appear. DOI: 10.1109/TCYB.2021.3125345.

Jian Chen (S'20) received B.S. degree from Hubei University of Technology in 2014 and the M.S. degree from Huazhong University of Science and Technology in 2018. He is currently working toward the Ph.D. degree in School of Electronic Information and Communications, Huazhong University of Science and Technology, China. His recent research interests focus on machine learning and data privacy. He is a student member of IEEE.

Yuan Gao received the B.E. degree from China University of Geosciences, Wuhan, in 2022. She is currently pursuing the M.S. degree in Electronics and Information Engineering at Huazhong University of Science and Technology, China. Her research interests focus on data poisoning attacks and adversarial learning.

Jinyong Shan received the B.S. and M.S. degrees in mathematics from Hubei University, Wuhan, China, in 2009 and 2012, respectively and the Ph.D. degree in information security from Institute of Information Engineering, CAS, Beijing, China, in 2015. He is currently a senior researcher with Sudo Technology Co.,LTD., Beijing, China. His research interests include cryptography applications, blockchain and privacy computing.

Kai Peng received the B.S., M.S., and Ph.D. degrees from Huazhong University of Science and Technology, China, in 1999, 2002, and 2006, respectively. He is now the faculty of Huazhong University of Science and Technology as a full professor. His current research interests are in the areas of wireless networking and big data processing.

Chen Wang (S'10-M'13-SM'19) received the B.S. and Ph.D. degrees from the Department of Automation, Wuhan University, China, in 2008 and 2013, respectively. From 2013 to 2017, he was a postdoctoral research fellow in the Networked and Communication Systems Research Lab, Huazhong University of Science and Technology, China. Thereafter, he joined the faculty of Huazhong University of Science and Technology where he is currently an associate professor. His research interests are in the broad areas of wireless networking, Internet of Things, and mobile computing, with a recent focus on privacy issues in wireless and mobile systems. He is a senior member of IEEE and ACM.

Hongbo Jiang (M'09-SM'15) received the Ph.D. degree from Case Western Reserve University in 2008. He is currently a Full Professor with the College of Computer Science and Electronic Engineering, Hunan University. He ever was a Professor with the Huazhong University of Science and Technology. His research concerns computer networking, especially algorithms and protocols for wireless and mobile networks. He was the Editor of IEEE/ACM Transactions on Networking, the Associate Editor for IEEE Transactions on Mobile Computing, and the Associate Technical Editor for IEEE Communications Magazine. He is an elected Fellow of IET, Fellow of BCS.