electronics

Article

Federated Split Learning Model for Industry 5.0: A Data Poisoning Defense for Edge Computing

Firoz Khan [1], R. Lakshmana Kumar [2], Mustufa Haider Abidi [3,*], Seifedine Kadry [4], Hisham Alkhalefah [3] and Mohamed K. Aboudaif [3]

[1] Higher Colleges of Technology, Dubai P.O. Box 25026, United Arab Emirates
[2] Hindusthan College of Engineering and Technology, Coimbatore 641032, Tamil Nadu, India
[3] Industrial Engineering Department, College of Engineering, King Saud University, P.O. Box 800, Riyadh 11421, Saudi Arabia
[4] Department of Applied Data Science, Noroff University College, 0459 Oslo, Norway
* Correspondence: mabidi@ksu.edu.sa

Abstract: Industry 5.0 provides resource-efficient solutions compared to Industry 4.0. Edge Computing (EC) allows data analysis on edge devices. Artificial intelligence (AI) has become the focus of interest in recent years, particularly in industrial applications. The coordination of AI at the edge will significantly improve industry performance. This paper integrates AI and EC for Industry 5.0 to defend against data poisoning attacks. A hostile user or node injects fictitious training data to distort the learned model in a data poisoning attack. This research provides an effective data poisoning defense strategy to increase the learning model's performance. This paper developed a novel data poisoning defense federated split learning, DepoisoningFSL, for edge computing. First, a defense mechanism is proposed against data poisoning attacks. Second, the optimal parameters are determined for improving the performance of the federated split learning model. Finally, the performance of the proposed work is evaluated with a real-time dataset in terms of accuracy, correlation coefficient, mean absolute error, and root mean squared error. The experimental results show that DepoisoningFSL increases the performance accuracy.

Keywords: Industry 4.0; Industry 5.0; edge computing; data poisoning

Citation: Khan, F.; Kumar, R.L.; Abidi, M.H.; Kadry, S.; Alkhalefah, H.; Aboudaif, M.K. Federated Split Learning Model for Industry 5.0: A Data Poisoning Defense for Edge Computing. *Electronics* **2022**, *11*, 2393. https://doi.org/10.3390/electronics11152393

Academic Editor: Dimitra I. Kaklamani

Received: 6 July 2022
Accepted: 27 July 2022
Published: 31 July 2022

1. Introduction

Industry 5.0 is now being envisioned to combine the distinct innovation of professional knowledge with robust, smart, and precise machines. Several scientific innovators think that Industry 5.0 will restore the manufacturing industry's elements [1]. Industry 5.0 is expected to combine high-speed, precise machines with personal analytical and cognitive thinking. Another key addition of Industry 5.0 is mass personalization, in which customers can choose customized and tailored things based on their preferences and needs. Predictive analytics and working knowledge are used in Industry 5.0 to construct models that strive to make more precise and valid judgments. Most of the manufacturing process will be digitized in Industry 5.0, since real-time data from machines will be combined with highly trained people. Machine learning algorithms are facilitating these advancements in the industries [2–4].

The fast proliferation of the Internet of Things (IoT) and the availability of multiple cloud services has given rise to a new concept, known as edge computing (EC), which allows data processing at the network edge. Not just in the future Industry 5.0, but also in the transition to Industry 4.0, EC can provide substantial value. EC can address latency costs, battery life limits, reaction time demands, data security, and privacy [5]. EC reduces communication costs and ensures that applications run smoothly in remote locations. Additionally, EC can process data without sending them to the public cloud, reducing security concerns for Industry 5.0's major events. Data processing, cache coherency, computation offloading, transporting, and delivering requests are all things that EC can do. The edge must be designed effectively to assure protection, stability, and confidentiality with these network services. EC provides high throughput, data protection, and anonymity for Industry 5.0 applications and provides quality service to target consumers [6]. Furthermore, EC delivers real-time connectivity for next-generation Industrial 5.0 technologies, such as UAVs, driverless cars, and remote medical services [7].

EC enables Industry 5.0 to acquire and communicate data concerning their major industries using more accessible, standardized hardware and software components. Industries are attempting to access information from specific servers frequently to handle large amounts of data. One of the difficulties in evaluating all of these machines is that raw data are far too large to be evaluated effectively. By reducing the amount of data transferred to a centralized server, EC allows Industry 5.0 to filter data. Furthermore, in Industry 5.0, EC offers preventive analytics, which allows for the early identification of hardware failures and mitigation by empowering workers to make informed decisions.

Artificial intelligence (AI) is the technology used to conduct intelligent processes that humans can easily perform, such as observing, thinking, learning, and problem solving [8,9]. Because of the vast amount of data generated by the Internet over the last two decades, AI has gained traction worldwide. Microdata centers can integrate EC with AI at network edge devices. The edge device will become a cognitive edge as EC reduces delays and ensures fast data responses. AI will generate data forecasts at the edge, making it an intelligent edge. The application of AI in Industry 5.0 will provide a possible attack vector. The quality and the confidentiality of the data must be secured before being used for training in industry 5.0.

This paper considers the data poisoning problem in industry 5.0, for which a three-layer edge computing architecture is designed (shown in Figure 1). The edge computing servers use a machine learning algorithm to analyze the data generated by edge devices. A data poisoning attack is a harmful strategy that injects modified data into machine learning (ML) models as they are being built [10]. Creating reliable ML models is a big task with important real-world implications. This paper proposes a novel data poisoning defense federated split learning called DepoisoningFSL for efficient defense against data poisoning and finds the optimal parameter for federated split learning to improve performance. Each edge device keeps its own private information secret in this model and training takes place alone on the edge devices.

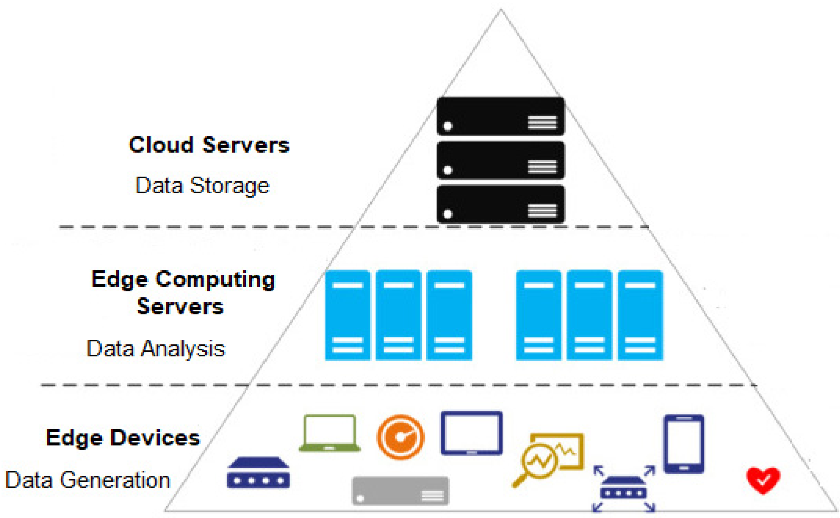


Figure 1. Three-Layer Edge Computing.

1.1. Related Work

While resolving ambiguity in the decision-making method, AI provides accurate and cost-effective approaches. Process automation has enabled faster decision-making due to improved algorithms' ability to handle complex data. The combination of edge computing and AI will cast doubt on various issues, allowing for more beneficial applications. Edge computing applications urgently require AI's tremendous processing skills to handle a variety of complex scenarios [6]. AI can suffer several security attacks, such as evasion [11], poison [12], and backdoor attacks [13], etc. This section describes related work on data poisoning attacks and federated learning (FL) for edge computing.

Tolpegin et al. [14] investigated appropriate data poisoning threats against FL systems. A malignant fraction of participants tries to poison the global model by transmitting model updates based on mislabeled information. According to their findings, poisons given late in the training phase are substantially more efficient than those inserted early. In an online setting, Seetharaman et al. [15] presented a defense solution to reduce the degradation caused by poisoned training sets on a learner's model. An influence function, a traditional methodology in robust statistics, is used in our suggested method.

Doku and Rawat [16] examined a method for preventing data poisoning attacks in a federated learning environment. The concept of a facilitator, who is allocated to an end device, is presented. The facilitator's responsibility is to ensure that an end device's data has not been tampered with. It accomplishes this by using an SVM model for data validation. Zhang et al. [17] offered a generative adversarial network-based toxic data creation approach. This method mainly depends on continuously updated global design variables to regenerate samples of interested victims. Against the federated learning framework, a unique generative poisoning assault model is proposed. The suggested data generating method is used in this model to decrease attack constraints and make attacks practicable quickly.

Chen et al. [18] investigated data poison detection techniques in basic and semi-distributed machine learning contexts. In the simplest situation, the data poison detection technique uses a set of parameters to determine which sub-datasets are poisoned. The chance of discovering threats with various training loops is analyzed using a mathematical model. In the semi-scenario, the author offered an enhanced data poison detection system and the best resource allocation.

Edge federated learning, introduced by Ye et al. [19], divides modifying the learning algorithm, which is expected to be accomplished autonomously by smartphones. To boost training efficiencies and minimize global communication frequency, the outputs of portable devices are consolidated in the edge server. Lu et al. [20] presented a privacy-preserving asynchronous federated learning mechanism for edge network computing, allowing numerous edge nodes to accomplish more effective federated learning without disclosing their private information. Liu et al. [21] proposed a cooperative intrusion detection approach that offloads the training model to dispersed edge devices. The centralized server's resource consumption is reduced thanks to a dispersed federated-based method, ensuring security and privacy. The training models are stored and shared on blockchain to ensure the security of the aggregate model.

A deep learning model is used for the intrusion detection system [22]. Botnet detection [23,24]. Sriram et al. [25] suggested a network traffic-flow-based botnet detection using deep learning approaches. In fog computing, homomorphic techniques for data security are analyzed [26]. The work scheduling technique optimizes the makespan and resource consumption in the fog computing environment [27]. For the Internet of Things, a trust-aware routing framework [28] was developed.

1.2. Problem Definition

Let us consider that there are N edge nodes, each with its local dataset $D = \{\chi_1, \chi_2, \chi_3, \cdots, \chi_N\}$. The malicious node has normal data $D_{norm} = \left\{\left(x_i^n, c_i^n\right)\right\}_{i=1}^{n}$ and poisoning data $D_{pois} = \left\{\left(x_i^p, pc_i^p\right)\right\}_{i=1}^{p}$ where $x_i \in \mathcal{R}^d$ is the ith instance with the d features and $c_i \in \{0, 1\}$ and pc is the attacker flipped labels. The attacker uses D_{pois} to reduce the performance of prediction. The goal is to create a three-stage learning framework to improve the performance of prediction.

1.3. Contribution

The key objective of this research paper is to develop the defense mechanism against data poisoning attacks and improves the performance of federated split learning through optimal parameters.

The main contributions of this research work are as follows:

- A data poisoning defense mechanism for edge-computing-based FSL—DepoisoningFSL—is proposed;
- The optimal parameters are found for improving the performance of federated split learning;
- The performance of the proposed algorithm with several real-time datasets is evaluated.

The remaining parts of this paper are organized as follows. Section 2 discusses the preliminary concept related to this paper. Then, the proposed DepoisoningFSL is explained in Section 3, and the experimental result of the proposed work is analyzed in Section 4. Finally, Section 5 presents the conclusions deduced from this research work.

2. Preliminaries

This section describes the preliminary concept of data poisoning attacks and federated learning.

2.1. Federated Learning

FL is a participatory machine learning process that allows participating machines to change model parameters regularly while storing all training data locally [19]. Edge-based FL allows edge nodes to learn a globally integrated model cooperatively without submitting private local data to a central server. Consider there are N edge nodes, each with its local dataset $D = \{\chi_1, \chi_2, \chi_3, \cdots, \chi_N\}$. Here $\varkappa_n \cong |\chi_n|$ Figure 2 shows a federated learning model. At each round t, the server sends a global learning model $GM^t\wp$ to all edge nodes. Select subset S^t of n nodes from N edge nodes. The edge node i optimizes the global learning model $GM^t\wp$ to get the local model LM_i^{t+1} for $t+1$ round. The node i provides the updated model $UM_i^{t+1} = \left(LM_i^{t+1} - GM^t\wp\right)$ to the cloud server. The server updates the new model.

$$GM^{t+1}\wp = GM^t\wp + \frac{\mathfrak{H}}{n}\sum_{i\in S^t} UM_i^{t+1} \quad (1)$$

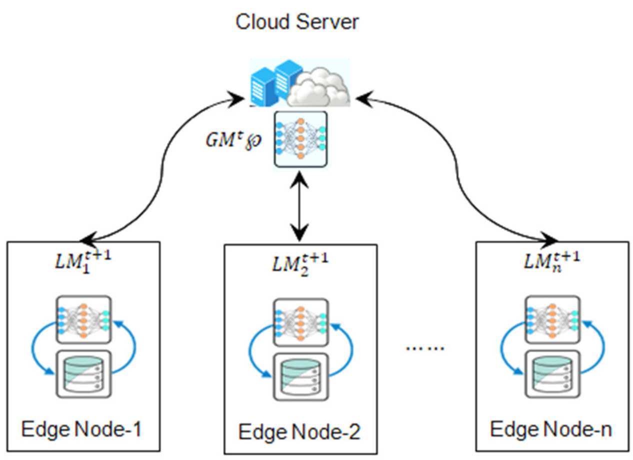


Figure 2. Federated Learning Model.

Here, $\mathfrak{H}$ is the learning rate and UM_i^{t+1} indicates updated model.

Each edge node gets a common shared model $\wp$ from the parameter server and trains it with its data. Edge nodes then upload the new weights or gradients to the parameter server, updating the global model. As a result, the following formula is used to calculate the total size (TS) of data samples from N edge nodes:

$$TS = \sum_{n=1}^{N} \varkappa_n = X \tag{2}$$

The edge node n's loss function (LF) with the dataset χ_n is

$$LF_n(\wp) \cong \frac{1}{\varkappa_n} \sum_{j \in \chi_n} fl_j(\wp) \tag{3}$$

where fl_j is the loss function of jth data. The global loss function can be computed as,

$$GLF(\wp) \cong \frac{\sum_{n=1}^{N} \varkappa_n LF_n(\wp)}{X} \tag{4}$$

The key advantages of FL are as follows:

- The amount of time spent training has been lowered. In addition, multiple devices are employed simultaneously to compute gradients, resulting in considerable speedups.
- The time it takes to make a decision is cut in half. Each device has its local copy of the model, allowing exceptionally fast predictions without relying on slow cloud requests.
- Privacy is protected. Uploading sensitive data to the cloud poses a significant privacy risk for applications, such as healthcare devices. In these situations, privacy breaches could mean the difference between life and death. As a result, keeping data local protects end users' privacy.
- It is less difficult to learn in a group setting. Instead of collecting a single large dataset to train a machine learning model, federated learning allows for a form of "crowdsourcing" that can make the data collection and labeling process considerably faster and easier.

2.2. Data Poisoning Attack

A data poisoning attack, also known as a causative attack, tries to modify a training set or model structure, resulting in the misclassification of subsequent input data linked with a particular label (a targeted attack) or the manipulation of data forecasts from all categories (an indiscriminate attack) [29]. The attacker aims to increase the error (loss) value in the dataset.

Let us consider the dataset $= \{(x_i, C_i)\}_{i=1}^{n}$, $x_i \in \mathcal{R}^d$ is the ith instance with the d features and $C_i \in \{0, 1\}$ (binary class label). The classifier can be learned using the objective function

$$\min_{f \in \mathcal{H}} \sum_{i=1}^{n} L(y_i, f(x_i)) + \left(\frac{c}{2} \|\daleth\|^2\right) \tag{5}$$

where $f(x_i) = \daleth^T x_i + b$ is the decision function discovered by solving the optimization problem on the dataset D, and c is a constant value. $\mathcal{H}$ indicates hypothesis and L points to the loss function.

2.3. Threat and Adversary Model

Consider the scenario that a portion of FL users is hostile or under the direction of a malicious opponent. The symbol m% denotes the percentage of malicious users/edges between all users/edges. Malicious users/edges can be injected into the system by inserting adversary-controlled devices, corrupting the devices of m percent of the benign users/edges, or encouraging m% of the benign users/edges to poison the global model for a set number of FL rounds. Each malicious user/edge has different behavior or mechanisms to add poison data. The aggregator is trustworthy and uncorrupted.

The attacker's goal is to tamper with the learned characteristics so that the final global model contains big mistakes for specific classes. As a result, the adversary is launching a targeted poisoning attempt. On the other hand, untargeted attacks look for indiscriminately high global model faults across all classes. Targeted attacks offer the advantage of reducing the chance of the poisoning attack being detected by decreasing the impact on non-targeted groups.

The following limitations are applied to a realistic threat model. Each malevolent person can change the training data Di on their device, but they cannot access or modify the dataset or model the learning process of other users. The attack is not particular to the optimization or loss function being used. It necessitates the corruption of training data, but the learning process is unaffected.

3. Proposed Methodology

According to a recent study, attackers can introduce specially processed false information into the incremental dataset to compromise the training information's validity [30]. The data poisoning attack reduces the performance of the prediction accuracy. This section explains the defense mechanism against data poisoning attacks.

3.1. Defense against Data Poisoning

In a data poisoning attack, an attacker perturbs the inputs and modifies the labels to build a false correlation among them. When a machine learning algorithm is taught using poisoned data, it learns an inaccurate feature association among inputs and labels, making it less accurate with clean inputs. Outlier sanitization [10] defenses are data poisoning mechanisms that detect and reject items that deviate dramatically from prior training data. Outlier sanitization aims to find the best threshold for determining how far a data point must be from the "local group" to be designated an outlier. If this threshold is set too high, the model will be poisoned; nevertheless, it will be under-fitted and ineffective if it is set too low. However, outlier sanitization has other problems besides maintaining the threshold; poisoned data points can be carefully positioned near the threshold to alter the decision boundary across several iterations slowly.

This section explains a novel data poisoning defense federated split learning called DepoisoningFSL. Consider the dataset $D = \{X_i, C_i\}$, $i = 1, 2, .., n$, where n is the total number of instances. X_i represents ith data instance with d features and C_i indicates the label of ith instance $C_i \in \{0, 1\}$. The dataset D contains normal (D_{norm}) and poisoned data (D_{pois}).

$$D = D_{norm} \cup D_{pois} \tag{6}$$

The D_{pois} data is generated at the edge node. The attacker randomly selects the number of instances and inverts the original class label to generate D_{pois} data to reduce the accuracy and increase the error rate of the prediction.

The main objective of this defense mechanism is to increase the accuracy and reduce the error rate. This paper proposes three-stage learning models to defend against data poisoning attacks. First, the algorithm explains the defense mechanism.

In Algorithm 1, three learning models, k-nearest neighbor (KNN), linear regression (LR), and random forest (RF), are used to predict the correct labels. In the first stage, the original dataset D is trained using KNN, LR, and RF algorithms and, based on these learning algorithms, predict the labels of D (P_{knn1}, P_{lr1}, P_{rf1}). Then, update D, generate a new training model, and predict the labels for the updated dataset (P_{knn2}, P_{lr2}, P_{rf2}). In the next stage, the voting algorithm is used between (P_{knn1}, P_{lr1}, P_{rf1} P_{knn2}, P_{lr2}, and P_{rf2}) these predicted labels (Step16 and 17). In the last stage, the minLoss function is used to predict the final class label (Step18).

Algorithm 1: Three Stage Learning Model

Input: $D = D_{norm} \cup D_{pois}$
Output: $D_{correct}$
Step01: M_{knn} = Train(D) using K-Nearest Neighbor algorithm
Step02: P_{knn1} = Predict_Label(D)using M_{knn}
Step03: U_{D1} = Update D based on P_{knn1}
Step04: M_{lr1} = Train(D) using Linear regression algorithm
Step05: P_{lr1} = Predict_Label (D) using M_{lr}
Step06: M_{lr2} = Train(U_{D1}) using Linear regression algorithm
Step07: P_{lr2} =Predict_Label (U_{D1}) using M_{lr2}
Step08: U_{D2} = Update U_{D1} based on P_{lr2}
Step09: M_{rf1} = Train(D) using Random Forest algorithm
Step10: P_{rf1} = Predict_Label (D) using M_{rf1}
Step11: M_{rf2} = Train (U_{D2}) using Random Forest algorithm
Step12: P_{rf2} = Predict_Label(U_{D2}) using M_{rf2}
Step13: U_{D3} = Update U_{D2} based on P_{rf2}
Step14: M_{knn2} = Train(U_{D3}) using K-Nearest Neighbor algorithm
Step15: P_{knn2} = Predict_Label(U_{D3}) using M_{knn2}
Step16: P_1 = Voting (P_{knn1},P_{lr1},P_{rf1})
Step17: P_2 = Voting(P_{knn2}, P_{lr2}, P_{rf2})
Step18: D_{corr} = minLoss(P_1,P_2)

3.2. Finding Optimized Parameters

Parameter optimization provides significant issues in an FL environment and is a key open research field. The amount of communication is related to machine learning models. This section looked into a communication-efficient hyper-parameter optimization method and a local one that allows us to optimize hyper parameters. Algorithm 2 explains the parameter optimization for FL.

Algorithm 2: Parameter Optimization

Step01: GmL = 1; LmL = 1; LM = null;
Step02: for each round r in R do
Step03: for each edge node in EN do
Step04: Collect $D = D_{norm} \cup D_{pois}$
Step05: apply three stage learning model
Step06: Get LmL
Step07: Update GmL based on LmL
Step08: Add LmL into LM
Step09: end for
Step10: end for

In this algorithm, GmL and LmL indicate global minimum loss and local minimum loss respectively. The GmL is updated based on LmL.

4. Performance Evaluation

In this section, the performance of the proposed work is analyzed. In addition, the proposed work was implemented using Java and the experiments were performed on Windows 10 64-bit OS. The following section describes the evaluation metrics, datasets, and results comparison.

4.1. Evaluation Metrics

The following metrics are used to evaluate the proposed method: accuracy (*ACC*), correlation coefficient (*CC*), mean absolute error (*MAE*), and root mean squared error (*RMSE*).

Accuracy (ACC) is defined as,

$$ACC = \frac{\mathcal{CP}}{\mathcal{TI}} \tag{7}$$

where $\mathcal{CP}$ represents the count of correctly predicted instances and $\mathcal{TI}$ indicates the total number of instances.

The Correlation Coefficient (CC) is computed as,

$$CC = \frac{\wp}{\sqrt{\mathcal{A} * \mathcal{P}}} \tag{8}$$

Here, the variables $\wp$, $\mathcal{A}$, and $\mathcal{P}$ can be computed as follows

$$\wp = \frac{SPC - SC * SP}{CW - WUI}, \mathcal{A} = \frac{SSC - SC * SC}{CW - WUI} \text{ and } \mathcal{P} = \frac{SSP - SP * SP}{CW - WUI}$$

where SPC = sum of predicted class, SC = sum of class values, SP = sum of predicted values, SSC = sum of squared class values, SSP = sum of squared predicted values, CW = weight of class, and WUI = weight of unclassified instances.

Mean Absolute Error (MAE) is defined as,

$$MAE = \frac{\sum_{i=1}^{\mathcal{TI}} |\mathfrak{p}_i - \mathfrak{a}_i|}{\mathcal{TI}} \tag{9}$$

Root Mean Squared Error ($RMSE$) is defined as,

$$RMSE = \frac{\sum_{i=1}^{\mathcal{TI}} (\mathfrak{p}_i - \mathfrak{a}_i)^2}{\mathcal{TI}} \tag{10}$$

Here, $\mathfrak{p}_i$ and $\mathfrak{a}_i$ are represented as predicted class label and actual class label for ith instances, respectively.

4.2. Datasets

The experiments are based on four real-time datasets, heart disease, diabetes [31], IoT_Weather, and IoT_GPS_Tracker [32]. Table 1 shows the detailed dataset information.

Table 1. Dataset Summary.

Dataset	No of Instances	No of Attributes
Heart	800	13
Diabetes	768	8
IoT_Weather	59,260	6
IoT_GPS_Tracker	58,960	5

4.3. Result Comparison

The proposed DepoisoningFSL is compared with K-nearest-neighbor-based semi-supervised defense (KSSD) [33] with different data poisoning rates: 0%, 5%, 10%, 15%, 20%, and 25%.

If the attacker increases the poisoning rate, then the performance is degraded. If the entire dataset class label is changed (poison rate = 100%), then the result will be inverted.

4.3.1. Results for Heart Dataset

Table 2 shows the evaluation metrics for the heart dataset with different data poison rates. Using the heart dataset, 93.6% accuracy was achieved for no-poison data. When increasing the percentage of poison rate, the accuracy and correlation coefficients are decreased and the *MAE* and *RMSE* are increased.

Table 2. *ACC*, *CC*, *MAE*, and *RMSE* Comparison for Heart Disease dataset with different data poison rate.

Poison Rate (%)	*ACC*		*CC*		*MAE*		*RMSE*	
	KSSD	**DFSL**	**KSSD**	**DFSL**	**KSSD**	**DFSL**	**KSSD**	**DFSL**
0	94.1	93.625	0.974	0.979	0.0125	0.01	0.111	0.1
5	90.04	91	0.875	0.971	0.0709	0.013	0.245	0.117
10	87.75	90.375	0.817	0.972	0.104	0.013	0.293	0.117
15	83.12	89.75	0.764	0.962	0.129	0.018	0.326	0.137
20	79.25	80.87	0.688	0.94	0.169	0.016	0.372	0.127
25	75.75	80	0.595	0.93	0.202	0.022	0.411	0.15

Figures 3 and 4 show a comparison of *ACC*, *CC*, *MAE*, and *RMSE* for the proposed depoisoning FSL and KSSD. From that result, the proposed work increases the accuracy and reduces the error compared to the KSSD approach.

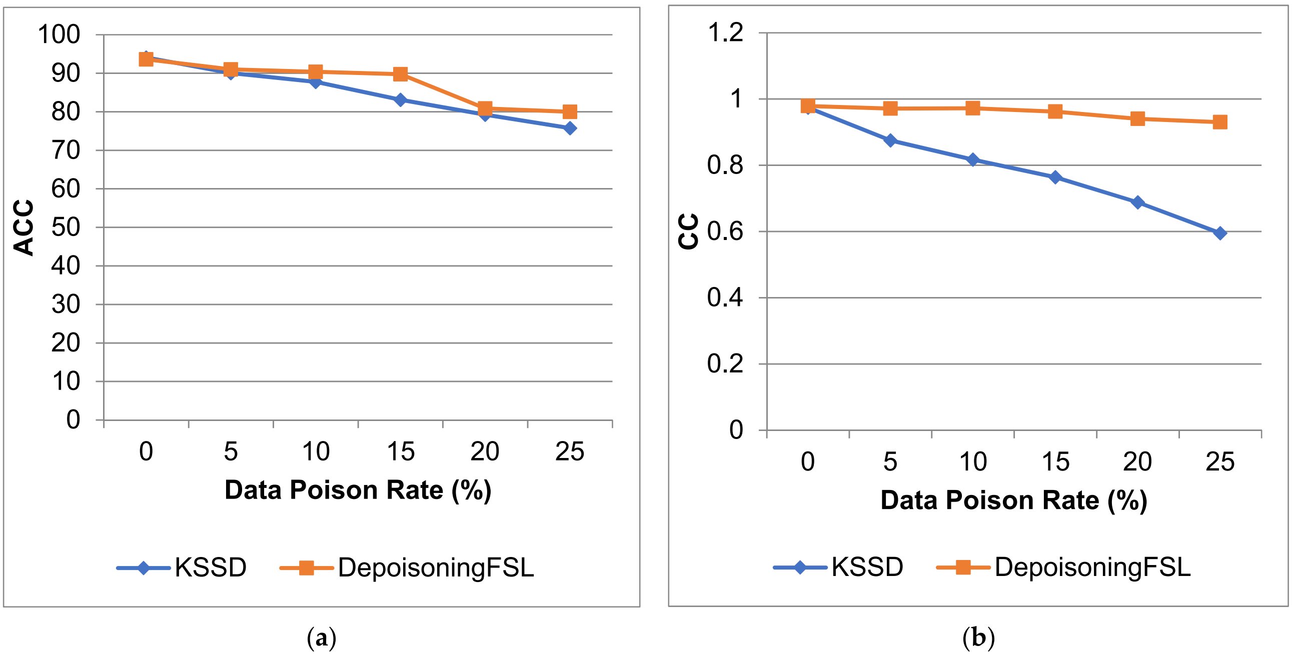


Figure 3. Heart Disease Dataset (**a**) A ccuracy; (**b**) Correlation Coefficient.

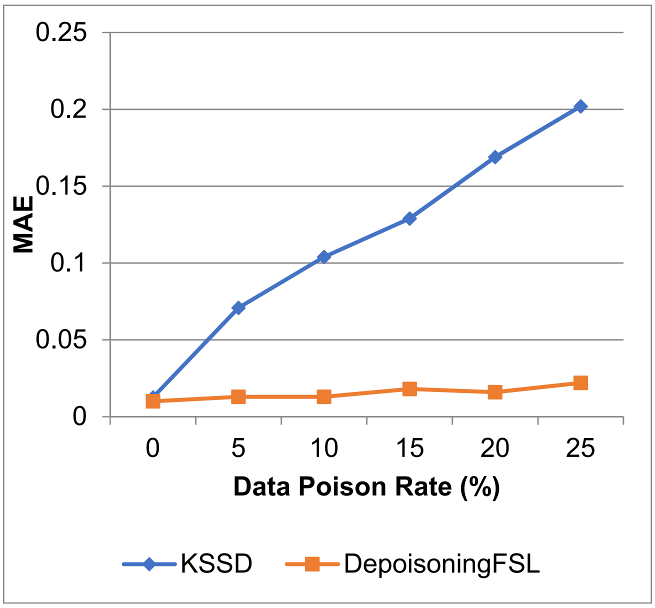
For 25% poison data, the *ACC* will increase by 5.61%, there is a 56.3% increase for *CC*, an 89.1% decrease in *MAE*, and a 63.5% decrease in *RMSE* compared to the KSSD approach.

4.3.2. Results for Diabetes Dataset

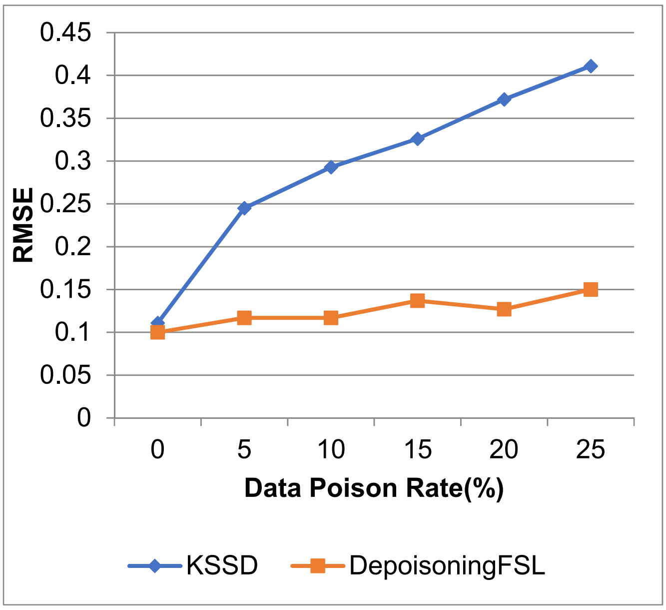
Table 3 shows the evaluation metrics for the diabetes dataset with different data poison rate.

Using the diabetes dataset, 84.11% accuracy was achieved for no-poison data. When increasing the percentage of poison rate, the accuracy and correlation coefficients are decreased.

Figures 5 and 6 show the comparison of *ACC*, *CC*, *MAE*, and *RMSE* for the diabetes dataset. From that result, the proposed work increases the accuracy and reduces the error compared to the KSSD approach.



(**a**)



(**b**)

Figure 4. Heart Disease Dataset (**a**) *MAE*; (**b**) *RMSE*.

Table 3. *ACC*, *CC*, *MAE*, and *RMSE* comparison for diabetes dataset with different data poison rates.

Poison Rate (%)	*ACC*		*CC*		*MAE*		*RMSE*	
	KSSD	**DFSL**	**KSSD**	**DFSL**	**KSSD**	**DFSL**	**KSSD**	**DFSL**
0	73.82	84.11	0.554	0.596	0.136	0.119	0.369	0.346
5	69.14	82.03	0.548	0.587	0.132	0.126	0.364	0.355
10	65.75	79.82	0.53	0.475	0.152	0.146	0.39	0.382
15	63.93	77.21	0.488	0.494	0.128	0.133	0.359	0.364
20	61.19	75.91	0.342	0.364	0.205	0.128	0.453	0.357
25	60.54	84.11	0.311	0.596	0.21	0.119	0.459	0.346

For 10% poison data, *ACC* increases by 21.39%, *CC* increases by 10.37%, *MAE* decreases by 3.9%, and *RMSE* decreases by 2.05% compared to KSSD. For 25% poison data, we observe an 18.92% increase in *ACC*, a 10.28% increase in *CC*, a decrease of 41.42% for *MAE* and 23.42% for *RMSE*.

4.3.3. Results for IoT_Weather Dataset

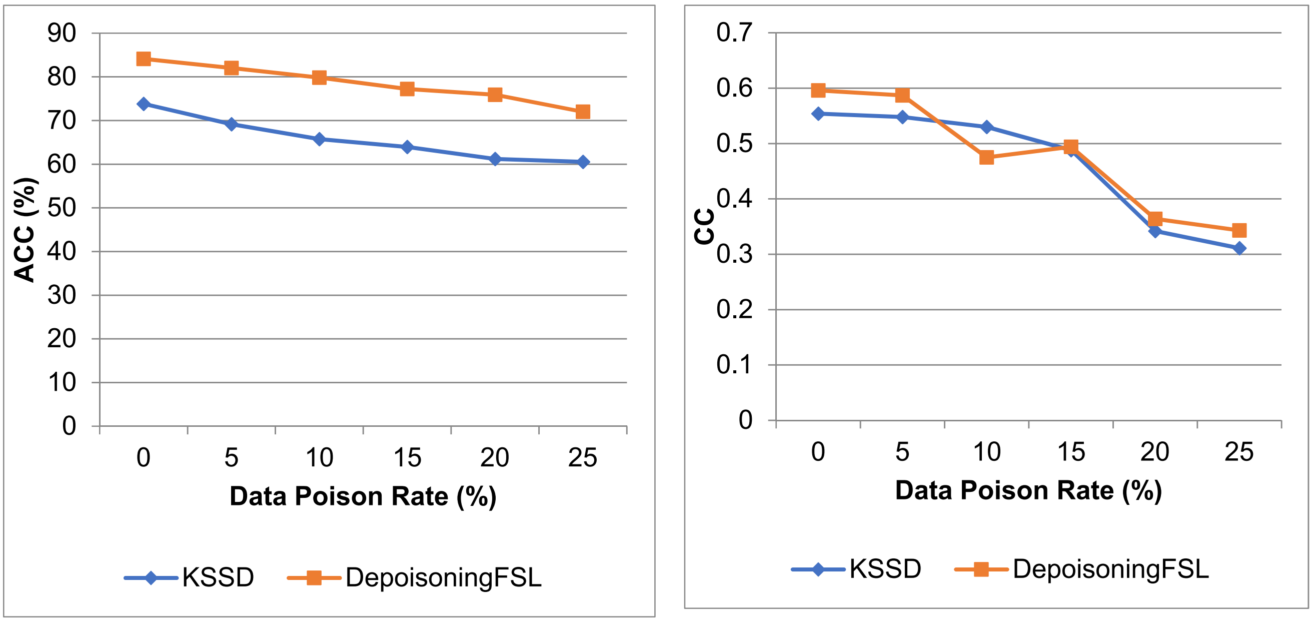
Table 4 shows the evaluation metrics for IoT_Weather dataset with different data poison rates.

Table 4. *ACC*, *CC*, *MAE*, and *RMSE* comparison for IoT_Weather dataset with different data poison rates.

Poison Rate (%)	*ACC*		*CC*		*MAE*		*RMSE*	
	KSSD	**DFSL**	**KSSD**	**DFSL**	**KSSD**	**DFSL**	**KSSD**	**DFSL**
0	95.5	98.8	0.906	0.971	0.019	0.014	0.137	0.118
5	93.12	96.1	0.873	0.922	0.0605	0.037	0.243	0.192
10	89.1	93.4	0.8	0.808	0.093	0.088	0.304	0.296
15	85.7	89.9	0.711	0.828	0.1315	0.079	0.361	0.281

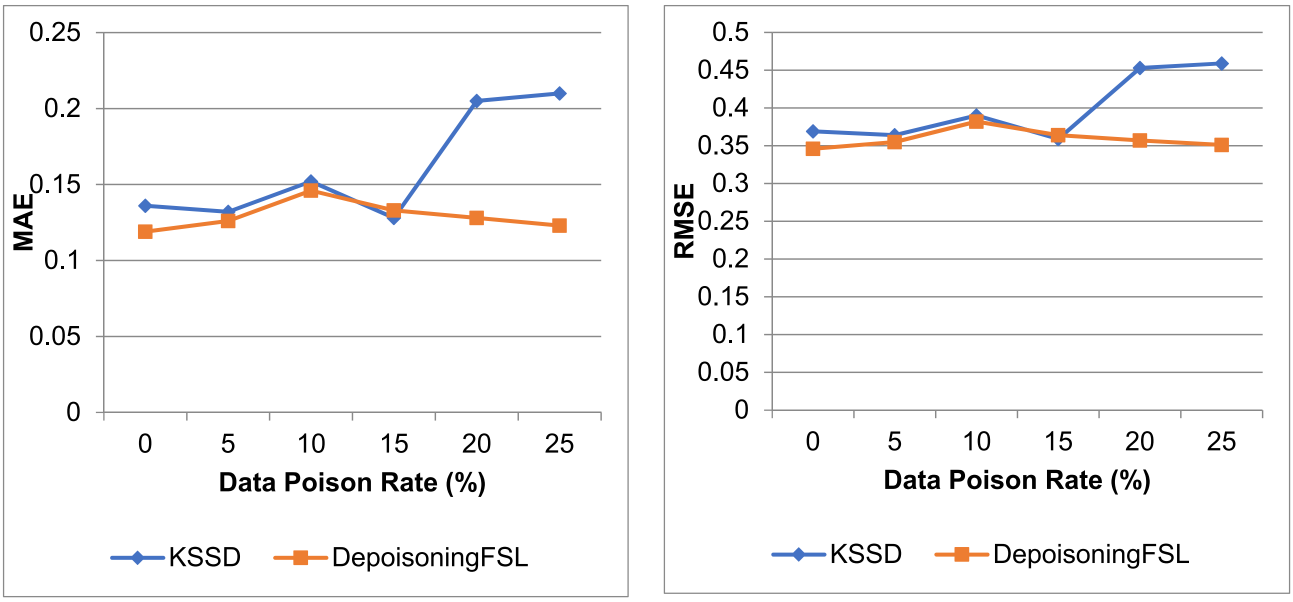
Table 4. *Cont.*

Poison Rate (%)	*ACC*		*CC*		*MAE*		*RMSE*	
	KSSD	**DFSL**	**KSSD**	**DFSL**	**KSSD**	**DFSL**	**KSSD**	**DFSL**
20	82.8	87.9	0.649	0.726	0.152	0.118	0.387	0.343
25	78.5	85.0	0.542	0.692	0.19	0.136	0.434	0.368



(**a**) (**b**)

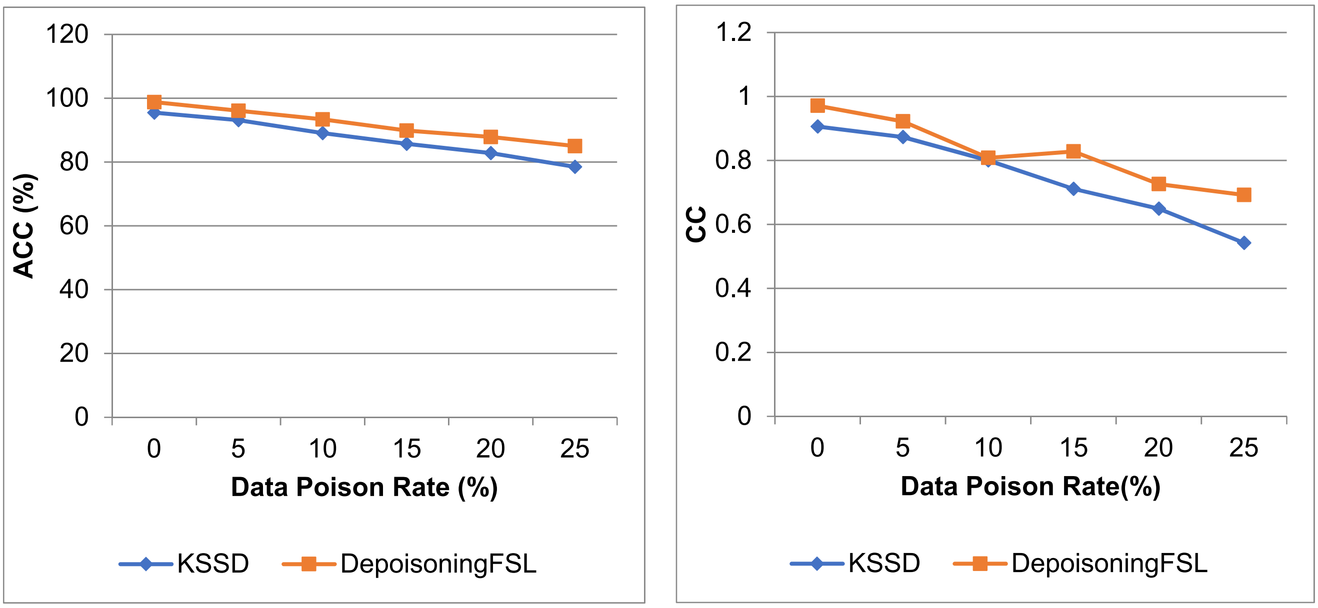
Figure 5. Diabetes dataset (**a**) accuracy; (**b**) correlation coefficient.



(**a**) (**b**)

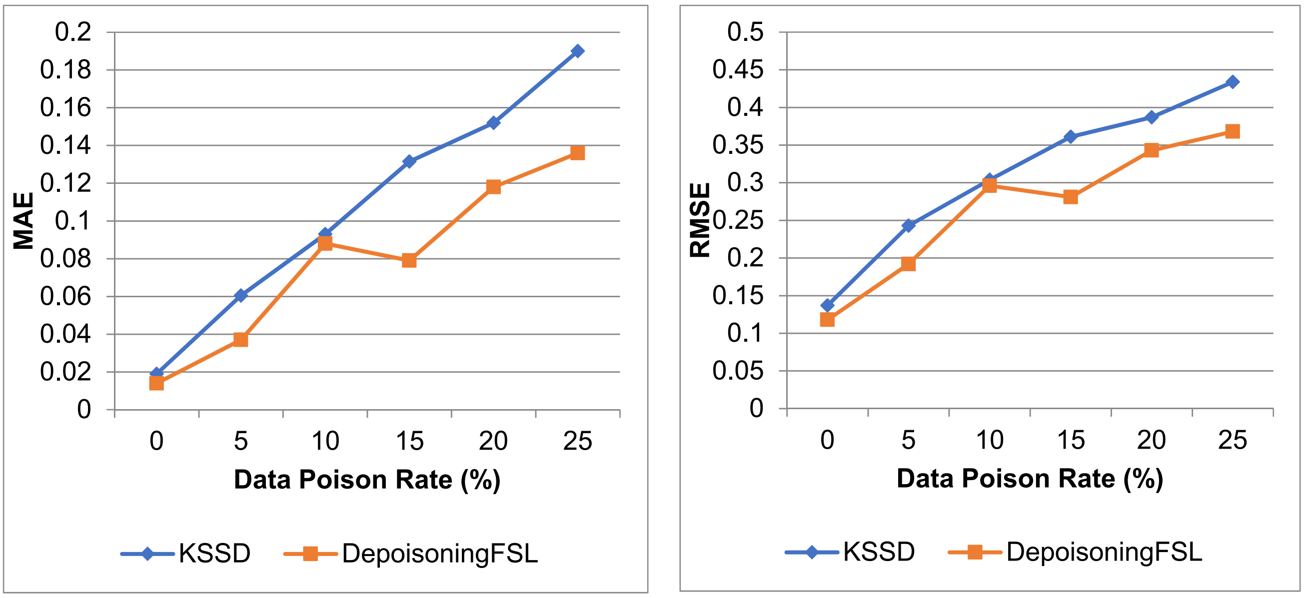
Figure 6. Diabetes dataset (**a**) *MAE*; (**b**) *RMSE*.

Figures 7 and 8 show the comparison of *ACC*, *CC*, *MAE*, and *RMSE* for the IoT_Weather dataset. From that result, the proposed work increases the accuracy and reduces the error compared to the KSSD approach.



(**a**) (**b**)

Figure 7. IoT_Weather dataset (**a**) accuracy; (**b**) correlation coefficient.



(**a**) (**b**)

Figure 8. IoT_Weather dataset (**a**) *MAE*; (**b**) *RMSE*.

4.3.4. Results for IoT_GPS_Tracker Dataset

Table 5 shows the evaluation metrics for the IoT_Weather dataset with different data poison rates.

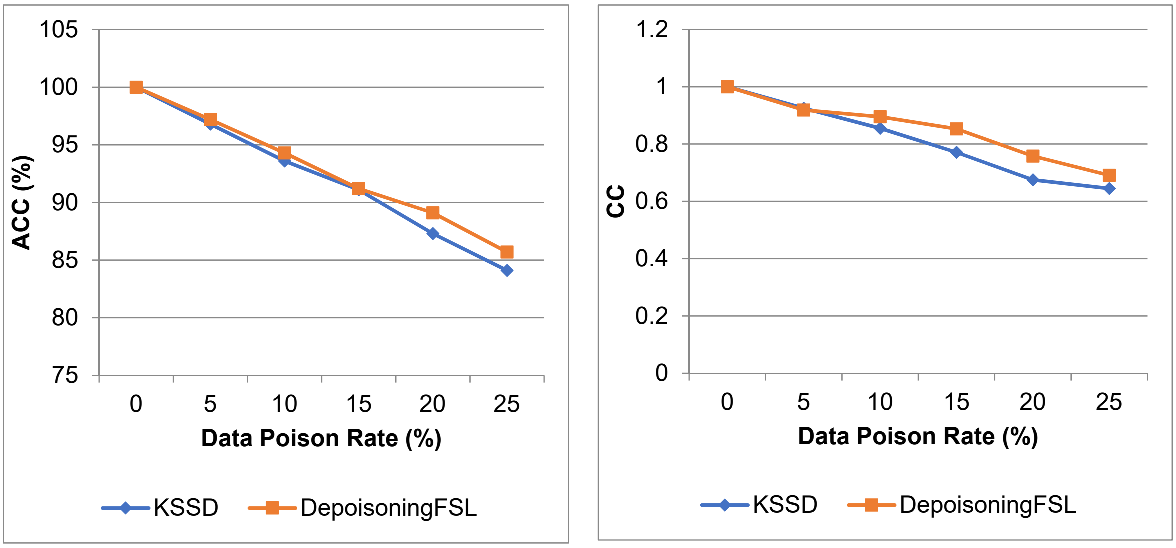
Table 5. *ACC*, *CC*, *MAE*, and *RMSE* comparison for IoT_GPS_Tracker dataset with different data poison rates.

Poison Rate (%)	*ACC*		*CC*		*MAE*		*RMSE*	
	KSSD	**DFSL**	**KSSD**	**DFSL**	**KSSD**	**DFSL**	**KSSD**	**DFSL**
0	100	100	1	1.0	0	0	0	0

Table 5. *Cont.*

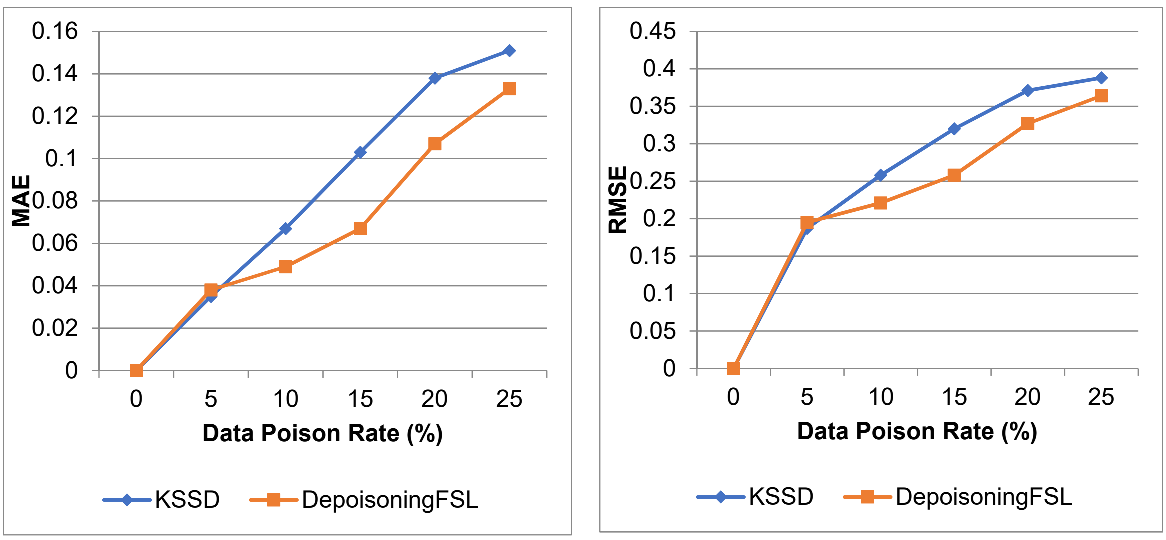
Poison Rate (%)	*ACC*		*CC*		*MAE*		*RMSE*	
	KSSD	**DFSL**	**KSSD**	**DFSL**	**KSSD**	**DFSL**	**KSSD**	**DFSL**
5	96.8	97.2	0.925	0.919	0.035	0.038	0.187	0.195
10	93.6	94.3	0.855	0.895	0.067	0.049	0.258	0.221
15	91.1	91.2	0.771	0.853	0.103	0.067	0.32	0.258
20	87.3	89.1	0.675	0.758	0.138	0.107	0.371	0.327
25	84.1	85.7	0.645	0.691	0.151	0.133	0.388	0.364

Figures 9 and 10 show the comparison of *ACC*, *CC*, *MAE*, and *RMSE* for the IoT_GPS_Tracker dataset. From that result, the proposed work increases the accuracy and reduces the error compared to the KSSD approach.



(**a**) (**b**)

Figure 9. IoT_GPS_Tracker dataset (**a**) accuracy; (**b**) correlation coefficient.



(**a**) (**b**)

Figure 10. IoT_GPS_Tracker dataset (**a**) *MAE*; (**b**) *RMSE*.

5. Conclusions

Industry 5.0 is an approach that aims to consistently balance the working environment and effectiveness of people and machines. Industry 5.0, which is supported by various new applications and supporting technology, is projected to boost manufacturing output and consumer satisfaction. This study presents a method for dealing with data poisoning in a federated learning network for Industry 5.0. The DepoisoningFSL is built for the industrial 5.0 edge computing architecture, composed of three levels (Cloud, Edge Server, and Edge Node). This paper proposes a three-stage learning-model-based defense mechanism for data poisoning attacks. The learning model KNN, linear regression, and random forest are used for prediction. The real-time dataset was used to analyze the performance of the proposed work. The evaluation result shows that the proposed method has higher accuracy and lower error than the KSSD method. The proposed model approximately increases accuracy by 5%, (heart), 14% (diabetes), 7% (IoT Weather), and 1.7% (IoT_GPS_Tracker) for the 25% data poison rate. Instead of federated learning, future research will look into comparable defensive techniques for different learning models.

Author Contributions: Conceptualization, F.K., R.L.K. and M.H.A.; methodology, F.K., M.H.A. and S.K.; software, F.K. and R.L.K.; formal analysis, F.K., M.K.A. and H.A.; investigation, F.K., R.L.K. and M.H.A.; resources, F.K. and H.A.; writing—original draft preparation, F.K. and R.L.K.; writing—review and editing, M.H.A., H.A., M.K.A. and S.K.; project administration, F.K., R.L.K. and H.A.; funding acquisition, H.A. All authors have read and agreed to the published version of the manuscript.

Funding: The authors extend their appreciation to King Saud University for funding this work through Researchers Supporting Project number (RSP2022R499), King Saud University, Riyadh, Saudi Arabia.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: All the data and dataset links are available in the manuscript.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Nahavandi, S. Industry 5.0—A human-centric solution. *Sustainability* **2019**, *11*, 4371. [CrossRef]
2. Abidi, M.H.; Alkhalefah, H.; Umer, U.; Mohammed, M.K. Blockchain-based secure information sharing for supply chain management: Optimization assisted data sanitization process. *Int. J. Intell. Syst.* **2020**, *36*, 260–290. [CrossRef]
3. Abidi, M.H.; Umer, U.; Mohammed, M.K.; Aboudaif, M.K.; Alkhalefah, H. Automated Maintenance Data Classification Using Recurrent Neural Network: Enhancement by Spotted Hyena-Based Whale Optimization. *Mathematics* **2020**, *8*, 2008. [CrossRef]
4. Ch, R.; Gadekallu, T.R.; Abidi, M.H.; Al-Ahmari, A. Computational System to Classify Cyber Crime Offenses using Machine Learning. *Sustainability* **2020**, *12*, 4087. [CrossRef]
5. Shi, W.; Cao, J.; Zhang, Q.; Li, Y.; Xu, L. Edge Computing: Vision and Challenges. *IEEE Internet Things J.* **2016**, *3*, 637–646. [CrossRef]
6. Pham, Q.-V.; Fang, F.; Ha, V.N.; Piran, J.; Le, M.; Le, L.B.; Hwang, W.-J.; Ding, Z. A Survey of Multi-Access Edge Computing in 5G and Beyond: Fundamentals, Technology Integration, and State-of-the-Art. *IEEE Access* **2020**, *8*, 116974–117017. [CrossRef]
7. Abdirad, M.; Krishnan, K.; Gupta, D. A two-stage metaheuristic algorithm for the dynamic vehicle routing problem in Industry 4.0 approach. *J. Manag. Anal.* **2020**, *8*, 69–83. [CrossRef]
8. Abidi, M.H.; Alkhalefah, H.; Umer, U. Fuzzy harmony search based optimal control strategy for wireless cyber physical system with industry 4.0. *J. Intell. Manuf.* **2021**, *33*, 1795–1812. [CrossRef]
9. Abidi, M.H.; Mohammed, M.K.; Alkhalefah, H. Predictive Maintenance Planning for Industry 4.0 Using Machine Learning for Sustainable Manufacturing. *Sustainability* **2022**, *14*, 3387. [CrossRef]
10. Steinhardt, J.; Koh, P.W.; Liang, P. Certified defenses for data poisoning attacks. In Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA, 4–9 December 2017; pp. 3517–3529.
11. Sidi, L.; Nadler, A.; Shabtai, A. Maskdga: An evasion attack against dga classifiers and adversarial defenses. *IEEE Access* **2020**, *8*, 161580–161592. [CrossRef]
12. Sun, G.; Cong, Y.; Dong, J.; Wang, Q.; Lyu, L.; Liu, J. Data Poisoning Attacks on Federated Machine Learning. *IEEE Internet Things J.* **2021**, *9*, 11365–11375. [CrossRef]

13. Chen, B.; Carvalho, W.; Baracaldo, N.; Ludwig, H.; Edwards, B.; Lee, T.; Molloy, I.; Srivastava, B. Detecting Backdoor Attacks on Deep Neural Networks by Activation Clustering. *arXiv* **2018**, arXiv:1811.03728. [CrossRef]
14. Tolpegin, V.; Truex, S.; Gursoy, M.E.; Liu, L. Data poisoning attacks against federated learning systems. In Proceedings of the European Symposium on Research in Computer Security 2020, Darmstadt, Germany, 4–8 October 2020; pp. 480–501.
15. Seetharaman, S.; Malaviya, S.; Vasu, R.; Shukla, M.; Lodha, S. Influence Based Defense Against Data Poisoning Attacks in Online Learning. *arXiv* **2021**, arXiv:2104.13230. [CrossRef]
16. Doku, R.; Rawat, D.B. Mitigating Data Poisoning Attacks On a Federated Learning-Edge Computing Network. In Proceedings of the 2021 IEEE 18th Annual Consumer Communications & Networking Conference (CCNC), Las Vegas, NV, USA, 9–12 January 2021; pp. 1–6.
17. Zhang, J.; Chen, B.; Cheng, X.; Binh, H.T.T.; Yu, S. PoisonGAN: Generative Poisoning Attacks Against Federated Learning in Edge Computing Systems. *IEEE Internet Things J.* **2020**, *8*, 3310–3322. [CrossRef]
18. Chen, Y.; Mao, Y.; Liang, H.; Yu, S.; Wei, Y.; Leng, S. Data Poison Detection Schemes for Distributed Machine Learning. *IEEE Access* **2020**, *8*, 7442–7454. [CrossRef]
19. Ye, Y.; Li, S.; Liu, F.; Tang, Y.; Hu, W. EdgeFed: Optimized Federated Learning Based on Edge Computing. *IEEE Access* **2020**, *8*, 209191–209198. [CrossRef]
20. Lu, X.; Liao, Y.; Lio, P.; Hui, P. Privacy-Preserving Asynchronous Federated Learning Mechanism for Edge Network Computing. *IEEE Access* **2020**, *8*, 48970–48981. [CrossRef]
21. Liu, H.; Zhang, S.; Zhang, P.; Zhou, X.; Shao, X.; Pu, G.; Zhang, Y. Blockchain and Federated Learning for Collaborative Intrusion Detection in Vehicular Edge Computing. *IEEE Trans. Veh. Technol.* **2021**, *70*, 6073–6084. [CrossRef]
22. Vinayakumar, R.; Alazab, M.; Soman, K.P.; Poornachandran, P.; Al-Nemrat, A.; Venkatraman, S. Deep Learning Approach for Intelligent Intrusion Detection System. *IEEE Access* **2019**, *7*, 41525–41550. [CrossRef]
23. Vinayakumar, R.; Alazab, M.; Srinivasan, S.; Pham, Q.-V.; Padannayil, S.K.; Simran, K. A Visualized Botnet Detection System Based Deep Learning for the Internet of Things Networks of Smart Cities. *IEEE Trans. Ind. Appl.* **2020**, *56*, 4436–4456. [CrossRef]
24. Ravi, V.; Alazab, M.; Srinivasan, S.; Arunachalam, A.; Soman, K.P. Adversarial Defense: DGA-Based Botnets and DNS Homographs Detection Through Integrated Deep Learning. *IEEE Trans. Eng. Manag.* **2021**, *early access*. [CrossRef]
25. Sriram, S.; Vinayakumar, R.; Alazab, M.; KP, S. Network Flow based IoT Botnet Attack Detection using Deep Learning. In Proceedings of the IEEE INFOCOM 2020—IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), Toronto, ON, Canada, 6–9 July 2020; pp. 189–194.
26. Murugesan, A.; Saminathan, B.; Al-Turjman, F.; Kumar, R.L. Analysis on homomorphic technique for data security in fog computing. *Trans. Emerg. Telecommun. Technol.* **2021**, *32*, e3990. [CrossRef]
27. Vijayalakshmi, R.; Vasudevan, V.; Kadry, S.; Kumar, R.L. Optimization of makespan and resource utilization in the fog computing environment through task scheduling algorithm. *Int. J. Wavelets Multiresolution Inf. Process.* **2020**, *18*, 1941025. [CrossRef]
28. Sankar, S.; Somula, R.; Kumar, R.L.; Srinivasan, P.; Jayanthi, M.A. Trust-aware routing framework for internet of things. *Int. J. Knowl. Syst. Sci. (IJKSS)* **2021**, *12*, 48–59. [CrossRef]
29. Wang, Y.; Mianjy, P.; Arora, R. Robust Learning for Data Poisoning Attacks. In Proceedings of the Virtual Mode Only, Proceedings of Machine Learning Research 2021, Virtual, 13 December 2021; pp. 10859–10869.
30. Jagielski, M.; Oprea, A.; Biggio, B.; Liu, C.; Nita-Rotaru, C.; Li, B. Manipulating Machine Learning: Poisoning Attacks and Countermeasures for Regression Learning. In Proceedings of the 2018 IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA, 20–24 May 2018; pp. 19–35.
31. UCL Database. Machine Learning Databases. 2021. Available online: https://archive.ics.uci.edu/ml/datasets.phphttps://archive.ics.uci.edu/ml/datasets.php (accessed on 23 March 2022).
32. UNSW Database. Machine Learning Datasets. 2021. Available online: https://research.unsw.edu.au/projects/toniot-datasetshttps://research.unsw.edu.au/projects/toniot-datasets (accessed on 24 March 2022).
33. Paudice, A.; Muñoz-González, L.; Lupu, E.C. Label Sanitization Against Label Flipping Poisoning Attacks. In Proceedings of the ECML PKDD 2018 Workshops, Dublin, Ireland, 10–14 September 2018; pp. 5–15.