IAES International Journal of Artificial Intelligence (IJ-AI)
Vol. 13, No. 1, March 2024, pp. 459~466
ISSN: 2252-8938, DOI: 10.11591/ijai.v13.i1.pp459-466

Defending against label-flipping attacks in federated learning systems using uniform manifold approximation and projection

Deepak Upreti[1], Hyunil Kim[1], Eunmok Yang[2], Changho Seo[1]
[1]Department of Convergence Science, Kongju National University, Chungcheongnam-do, Gongju-si, South Korea
[2]Department of Information Security, Cryptology, and Mathematics, Kookmin University, Seoul, South Korea

Article Info

Article history:

Received Dec 8, 2022
Revised Feb 1, 2023
Accepted Feb 9, 2023

Keywords:

Federated learning
K-mean
KPCA
Principal component analysis
UMAP

ABSTRACT

The user experience can be greatly improved by using learning models that have been trained using data from mobile devices and other internet of things (IoT) devices. Numerous efforts have been made to implement federated learning (FL) algorithms in order to facilitate the success of machine learning models. Researchers have been working on various privacy-preserving methodologies, such as deep neural networks (DNN), support vector machines (SVM), logistic regression, and gradient boosted decision trees, to support a wider range of machine learning models. The capacity for computing and storage has increased over time, emphasizing the growing significance of data mining in engineering. Artificial intelligence and machine learning have recently achieved remarkable progress. We carried out research on data poisoning attacks in the FL system and proposed defence technique using uniform manifold approximation and projection (UMAP). We compare the efficiency by using UMAP, principal component analysis (PCA), Kernel principal component analysis (KPCA) and k-mean clustering algorithm. We make clear in the paper that UMAP performs better than PCA, KPCA and k-mean, and gives excellent performance in detection and mitigating against data-poisoning attacks.

Corresponding Author:

Changho Seo
Department of Convergence Science, Kongju National University
Chungcheongnam-do, Gongju-si, 32588, South Korea
Email: chseo@kongju.ac.kr

1. INTRODUCTION

Data is usually sensitive to privacy and is very large, or both, which may make it impossible to access to the data centre and conduct training while using traditional methods [1]. In federated learning (FL), several clients collaborate to train a model under the control of a single server while maintaining decentralized access to the training data [2]. However, due to the sensitive nature of the data, there are risks and responsibilities associated with storing it in a centralized location [3], [4]. A new paradigm known as FL has enormous potential for improving edge computing capabilities in modern distributed networks [5]. FL has a lot of potential but also presents some special difficulties in real-world situations [2]. Systemic heterogeneity [6], communication effectiveness [2], [7] privacy issues [2], and more recently on fairness and resilience throughout the network of clients [7] have been the main topics of current studies. Massive, decentralized networks have stochastic heterogeneity of client's data as a defining feature [8]. Innovations in implementation of machine learning (ML) models have considerably improved acceptance of this technology in several real-world systems that transform practically all industries in recent years [9], in fields like computer vision [10], healthcare [11], finance [12], marketing [13], smart manufacturing [13], transportation [6], agriculture [14], education [6], and natural language processing (NLP) [6]. The vast amounts of structured, semi-structured, and unstructured data that are regularly produced by people, companies, and machines, such data gathering and storing big data can be costly and time-consuming [15]. The amount of data being produced is increasing exponentially, and this presents a challenge for organizations to manage and store the data effectively. Various sources and storing it in a centralized location such as a data center. The data is then pre-processed and used to train the artificial intelligence (AI) model using different algorithms like supervised, unsupervised or reinforcement learning. The desired activity, such as image recognition or NLP, is subsequently carried out using the trained model that has been set up.

In 2017, McMahan *et al.* used the term "Federated learning" [2]. FL has gained popularity as a machine learning (ML) techniqu [9] due to its inherent speed, more efficient, and offers more data privacy than classical machine learning [11]. However, if poisoning is not addressed quickly, it can seriously affect FL systems and result in inaccurate predictions [16]. Clients in FL learn models locally using local training data, and they subsequently send model changes to a central aggregator who integrates them into a global model [17]. The next training cycle then propagates the global model back to the clients. Since the training is carried out simultaneously for numerous clients, FL provides efficiency and scalability [2]. FL specifically enhances confidentiality by allowing users to save their training data locally [2]. Despite its advantages, FL has been shown to be vulnerable to attacks known as "poisoning" [9], in which the attacker changes the local models of a subset of federated clients in order to make harmful updates to the global model [16]. Poisoning attempts that solely aim to negatively impact the performance of the global model can be stopped by monitoring the performance of uploaded models [8]. The inference attack and the poisoning attack are the two basic categories of attacks in FL. Where poisoning attack is further divided into data poisoning and model poisoning [18]. In FL, data poisoning refers to a malicious client using corrupted data for the training model [19]. Model poisoning is when an attacker tries to change the local model's parameters, which then affect the global model's parameters.

FL, also known as collaborative learning, is a decentralized methodology for training machine learning models. Unlike traditional approaches that involve transferring data from client devices to global servers, FL leverages the raw data available on edge devices to train the model locally. This approach enhances data privacy by eliminating the need for data transmission to central servers. FL resolves the data privacy problems by insisting that users only disclose the model parameters—not the actual data [8]. Thus, a global data model is created, where decisions are made using the parameters that each participant submits to a server. However, FL is threatened in a number of ways, including data poisoning [20]. Data poisoning seeks to degrade the final learning model's quality, resulting in misclassification. The data features or labels are intentionally changed in this attack scenario by adverse or hacked clients [21], [22]. The easiest and most effective method to flip the label of each training data point is the label-flipping attack, which will be addressed in the next section. Therefore, FL needs a defense mechanism to provide resilience in order to stop such data poisoning attempts. When there are I clients, each client i possesses its own dataset D_i= (x_1, x_k), and each data point x_k comprises a set of features f_i and a corresponding class label c_i∈ C, where C represents the set of all possible class values. Then, to compute the loss function L, federated averaging runs over $\{w_t^i\}_{i=1}^I$where $\{w_t^i = \nabla l_i$ (w_t) and finally, the updated global model parameter is calculated as,

$$w_{t+1} \leftarrow w_t - \eta\, \nabla(l w_t)$$

where w_t represents the previous global model parameter, η denotes the learning rate ∇l(w_t) is determined by,

$$\nabla l(w_t) = \sum_{k=1}^{K} \frac{n_i}{n} w_t^i$$

where n_i denotes the size of each individual dataset D_i and K represents the number of benign clients selected. However, to ensure the integrity of the aggregated global model, an additional identification method is required to identify and address malicious clients.

The label flipping attack in machine learning [23] is widely recognized as one of the most prevalent data poisoning attacks. This malicious technique involves flipping or altering the labels of training data, resulting in a degradation of the model's classification performance [24], [25]. An instance of this attack could be witnessed in the Canadian Institute for Advanced Research (CIFAR-10), where the label-flipping attack intentionally misclassifies airplanes as birds [22]. The absence of a centralized curator for data analysis poses a significant risk of data poisoning to the FL system [26].

We initiated an investigation into the feasibility of label flipping attacks on poison FL systems. Our approach relies on utilizing dimensionality reduction algorithms to identify malicious updates based on specific characteristics exhibited by malicious parameters [27]. However, given the large number of parameters in deep neural networks (DNN), manually checking for harmful parameter updates can be challenging [27]. To address this, we propose an automated strategy using uniform manifold approximation and projection (UMAP) for dimension reduction to locate and filter out parameters sent by malicious updates, as demonstrated in our study [11]. Within the FL system, the aggregator possesses the capability to detect and identify participants involved in malicious activities [16]. After being recognized, the aggregator has the option to put these participants on a blacklist or decline their contributions for future rounds [11].

2. METHOD AND STRATEGY

We know that label-flipping attacks on FL systems are quite effective and that a dimensionality method might be used as a defense mechanism based on prior research. We have developed a better method that combines dimensionality reduction with clustering strategies in order to advance previous research. For dimensionality reduction, we employed the UMAP technique [28]. UMAP proves to be a robust nonlinear dimensionality reduction method when compared to other techniques like t-distributed stochastic neighbor embedding (t-SNE) [29] and other clustering methods, UMAP is a strong nonlinear dimensionality reduction technique. UMAP assumes that the data points are evenly spread out across the manifold, which is determined by the Riemannian metric on the manifold [28]. Additionally, UMAP provides faster processing speeds compared to alternative techniques for reducing dimensionality, including principal component analysis (PCA) and t-SNE [29].

The experiments conducted involved varying the rate of malicious client presence (m) within the range of 10 to 30%. The results indicate that as the malicious percentage (m) increases, there is a decline in the test accuracy of the global model. Even a small increase in m leads to decreases in both the global model's test accuracy and the source-class recall, with the latter experiencing a more significant drop. For instance, in the scenario with a malicious percentage of 40%, the global model's test accuracy for CIFAR-10 decreases to 76% compared to 78% in the non-poisoned model case. Similarly, the source-class recall drops to approximately 0%. These findings demonstrate that even a small fraction of manipulated participants can have a significant impact on the accuracy of the global model. The vulnerability to label-poisoning attacks varies between the CIFAR-10 and Fashion Modified National Institute of Standards and Technology (Fashion-MNIST) datasets, with Fashion-MNIST being more sensitive compared to CIFAR-10 [23]. Surprisingly, the experiment shows that the attacker doesn't always require knowledge about the most susceptible source or target category. Furthermore, the effectiveness of the attack is not solely dependent on the misclassification rate of the non-poisoned model. Ultimately, removing malicious participation can help converge the system towards high-utility outcomes.

Algorithm 1: Enhancing the resilience of FL through UMAP-based model updates.

```
updating the model_function (X, n, d, min-dist, n-epochs $w_t$)
$v \leftarrow \emptyset$
each round t = 1, 2, ….do
    $w_t \leftarrow previous\_global\_model\_parameters$
    ## executed for each client locally
    for each client i= 1, 2,…I do
        $w_t^i \leftarrow \nabla \mathrm{l}(w_t)$
    u ← $\{w_t^i\}_{i=1}^{I}$
    u’ ← standardize(u)
    u’’ ← UMAP (u’, n, d, min-dist, n-epochs
    Plot (u’’)
```

Algorithm 1 demonstrates the updates made to our robust model using FL with UMAP. UMAP is a recently introduced technique for manifold learning. Its purpose is to accurately capture local structures while effectively integrating global structure [28]. UMAP is generally a straightforward algorithm that requires several parameters. The parameters required for the algorithm consist of the following: the neighborhood size (n) used for approximating local metrics, the desired dimension (d) of the reduced space, min-dist (a parameter governing the layout), and the number of epochs (n epochs) that determine the level of optimization applied. To construct the fuzzy simplicial set local to a given point x, the algorithm involves identifying its n nearest neighbors, establishing the appropriate normalized distance on the manifold, and then converting the finite metric space into a simplicial set using the FinSing functor. In this case, the conversion is accomplished through an exponential function of the negative distance. To ensure a consistent cardinality for the fuzzy set of 1-simplices, we utilize a smoothed version of the k-nearest neighbors-distance algorithm. Rather than directly utilizing the distance to the nth nearest neighbor for normalization, we opt for log2 (n) based on empirical experiments. In spectral embedding, we analyze the weighted graph formed by the 1-skeleton of the overall fuzzy topological representation. We then apply traditional spectral techniques to the symmetric normalized Laplacian of the graph. Algorithm 2 outlines the procedure for conducting the optimization process, which involves the utilization of stochastic gradient descent [28].

```
Algorithm 2: UMAP (X, n, d, min-dist, n-epochs)
Input:
X: the data set to have its dimension reduced
n: the neighborhood size to use for local metric approximation
d: the the dimension of the target reduced space
min-dist: an algorithmic parameter controlling the layout
n-epochs: controlling the amount of optimization work to perform
#construct the relevant weighted graph
for all x ∈ X do
    fs-set[x] ← Local Fuzzy Simplicial Set (X, x, n)
top ← $U_{x \in X}$
# perform optimizations of the graph layout
Y ← Spectral Embedding (top-rep, d)
Y ← optimize Embedding (top-rep, Y, min-dist, n-epochs)
RETURN Y
```

3. EVALUATION, ANALYSIS AND RESULTS

We employed the PyTorch library to implement our attack and defense strategies within the FL environment. By default, our setup consisted of 100 participants (N), a central aggregator and one participant per round, k=5. In our experiment, we focused on targeted data poisoning in FL, specifically aiming to flip labels. We flipped a label l to l', where l belongs to the set of possible labels (L), and l' is a different label from l within L. L represents the total count of categories in the classification task. To distribute the entire training dataset equally among participants, we assumed that each participant randomly received a different portion of the training data. We conducted FL tests for a total of R = 200 rounds.

For our experiments, we utilized the CIFAR-10 and Fashion-MNIST datasets. CIFAR-10 is a commonly used image collection for training machine learning and computer vision algorithms. It consists of 10 classes representing airplanes, cars, birds, cats, deer, dogs, frogs, horses, ships, and trucks [30]. Fashion-MNIST, on the other hand, contains 60,000 training items and 10,000 testing items. It encompasses 10 different classes representing T-shirt/top, trouser, pullover, dress, coat, sandal, shirt, sneaker, bag, and ankle boot [30]. For the Fashion-MNIST experiment, we employed a configuration that consisted of two convolutional layers incorporating batch normalization, two max-pooling layers, and a single fully connected layer.

To tackle CIFAR-10, we utilized a network architecture comprising six convolutional layers with batch normalization, followed by two fully connected dense layers and a softmax layer. To simulate attacks, we assigned a total of N participants, out of which a certain percentage (m%) were designated as malicious clients. At the beginning of each experiment, N * m participants were randomly selected as malicious, while the remaining participants were considered honest. To guarantee the accuracy of the experiment, we considered the potential impact of selecting participants with malicious intent. We conducted each experiment multiple times (ten times in total) and determined the final result by calculating the average outcome. In the CIFAR-10 dataset, we performed tests where we replaced dogs with cats, airplanes with birds, and automobiles with trucks. Similarly, for the Fashion-MNIST dataset, we conducted experiments by substituting shirts with t-shirts, trousers with dresses, and coats with shirts.

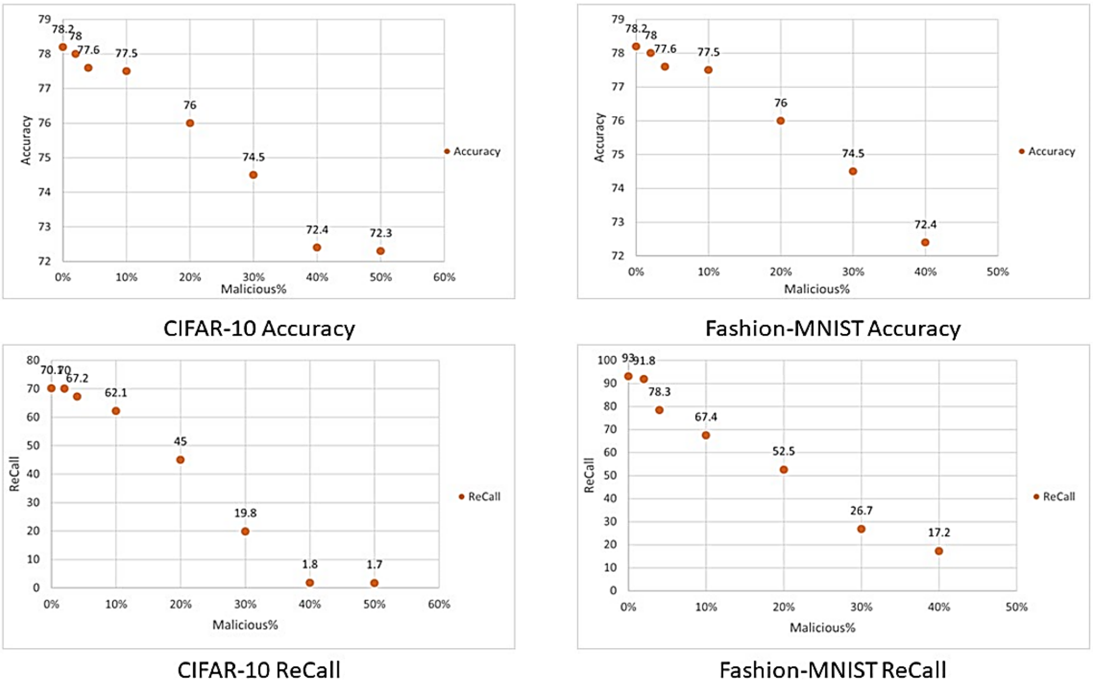
Our study aimed to apply the label flipping technique to both the CIFAR-10 and Fashion-MNIST datasets. We presented the outcomes in a table, considering four distinct scenarios. In the first scenario, we utilized PCA as the method to reduce dimensionality. In the second scenario, we employed kernel principal component analysis (KPCA) as the defense algorithm. The third scenario involved combining KPCA with k-means as a defense strategy. Lastly, in the fourth scenario, we used UMAP as a defense mechanism against adversarial attacks on the database. The experimental data we gathered demonstrated the effectiveness of employing PCA, KPCA, KPCA+K-means, and UMAP in distinguishing between malicious and genuine updates delivered by adversaries.

4. EXPERIMENTAL RESULTS

Our UMAP approach outperforms PCA and KPCA in terms of accuracy. Figure 1 presents the accuracy and recall metrics corresponding to the CIFAR-10 and Fashion-MNIST datasets. The CIFAR-10 and Fashion-MNIST datasets exhibit accuracy levels between 72% and 79%, while the presence of malicious values varies from 0% to 50%. The figure indicates that as the number of malicious users increases, the accuracy decreases. Initially, the decline is consistent, but once the percentage of malicious individuals exceeds 10%, the decline becomes more prominent for both datasets. When the malicious rate reaches 40%

to 50%, the accuracy remains at a minimal level. Similarly, when the FL system surpasses 10%, the recall results from the source show a significant decline, reaching zero.

Table 1 presents the recall of the source and the overall accuracy of a global model using different defense strategies on CIFAR-10. These strategies include PCA, KPCA, KPCA + Kmean, and UMAP. Similarly, Table 2 provides the corresponding results for Fashion-MNIST. The tables provide clear evidence that using PCA results in a negligible reduction in accuracy until the number of malicious participants reaches 10%. However, beyond 30%, there is a slight decline in accuracy. In contrast, when KPCA is used instead of PCA, both tables indicate that the accuracy decline is less pronounced. With KPCA, the accuracy remains unaffected until the fraction of malicious participants exceeds 20%, while with PCA, the accuracy starts to decrease once the percentage surpasses 4% for both CIFAR-10 and Fashion-MNIST. Likewise, when considering the combination of KPCA and K-mean (KPCA+K-mean), the accuracy experiences a significant drop when the malicious percentage reaches 40%.



CIFAR-10 Accuracy

Fashion-MNIST Accuracy

CIFAR-10 ReCall

Fashion-MNIST ReCall

Figure 1. Accuracy and recall of global model against CIFAR-10 & Fashion-MNIST attacks with m% adversaries

Table 1. The source recall and global model accuracy can be evaluated using different defense strategies, including PCA, KPCA, KPCA combined with K-means, and UMAP in CIFAR-10

	PCA			KPCA			KPCA + K-mean			UMAP	
Malicious	Accuracy	Recall	Malicious	Accuracy	ReCall	Malicious	Accuracy	ReCall	Malicious	Accuracy	ReCall
0%	78.2	69.9	0%	78.4	71	0%	78.4	71	0%	78.4	73
2%	78.1	69.9	2%	78.3	70	2%	78.3	70	2%	78.3	72
4%	78	69.8	4%	78.2	69.8	4%	78.2	69	4%	78.2	71
10%	77.5	67	10%	78	66	10%	78	67	10%	78.1	69
20%	77.4	62	20%	77.8	64	20%	77.8	64	20%	78	66
30%	76.8	53.1	30%	77.3	59	30%	77.3	60	30%	77.9	62
40%	76.1	44.1	40%	76.8	52	40%	76.8	53	40%	77.8	55
50%	76	29	50%	75.5	43	50%	75.7	45	50%	77.7	47

Table 2. The source recall and global model accuracy can be evaluated using different defense strategies, including PCA, KPCA, KPCA combined with K-means, and UMAP in Fashion MNIST

	PCA			KPCA			KPCA + K-mean			UMAP	
Malicious	Accuracy	Recall	Malicious	Accuracy	ReCall	Malicious	Accuracy	ReCall	Malicious	Accuracy	ReCall
0%	91.3	88	0%	91.3	88	0%	91.3	88	0%	91.3	88
2%	91.2	87.8	2%	91.2	87.8	2%	91.2	87.8	2%	91.2	87.8
4%	91	87.6	4%	91	87.6	4%	91	87.6	4%	91	87.6
10%	90.6	79	10%	90.7	80	10%	90.9	80	10%	90.9	80.5
20%	89.9	71	20%	90.2	72	20%	90.3	73	20%	90.7	74.1
30%	89.1	65	30%	89.8	68	30%	90	69	30%	90.5	70.5
40%	88.7	57	40%	89	63	40%	89.1	64	40%	89.8	65.2
50%	88.6	58	50%	88.7	58	50%	88.8	59	50%	89.5	60.3

In Tables 1 and 2, the rightmost column displays the results obtained using UMAP, which demonstrates superior performance compared to other dimension reduction techniques. UMAP is a powerful tool with several advantages over t-SNE and similar methods. It is fast and scalable, capable of projecting high-dimensional datasets, such as the 784-dimensional, 70,000-point MNIST dataset, in under 3 minutes, while the implementation of scikit-t-SNE takes longer [29]. As the number of data points increases, UMAP proves to be more time-efficient compared to t-SNE. Additionally, UMAP effectively preserves the overall structure of the data due to its strong theoretical foundations, which allow it to balance local and global structures effectively. As a result, UMAP is highly effective for visualizing high-dimensional data, thanks to its speed, preservation of global structure, and easily interpretable parameters. It is clear that UMAP outperforms other methods in protecting against data poisoning attacks. Therefore, UMAP significantly enhances the clustering results in FL and makes it more resilient against data poisoning attacks.

Figures 2 and 3 exhibit the performance of the proposed defensive algorithm and compare it with a previously suggested method [30]. The results of the PCA defense method are shown in the top row, with varying levels of malicious percentages (m) at 10%, 20%, 25%, and 30%. Similarly, the second row illustrates the implementation of the KPCA defense method, while the third row demonstrates the utilization of the UMAP method. In Figures 2 and 3, malicious updates are represented by the color red, while updates from honest participants are depicted in green. The figures clearly highlight the distinctions between malicious and honest updates and demonstrate the effectiveness of the different algorithms in differentiating between them. Comparing the three rows, it becomes apparent that UMAP outperforms PCA and KPCA in effectively distinguishing between malicious and honest updates. Thus, incorporating UMAP into the defense algorithm yields superior results and represents a more favorable approach.

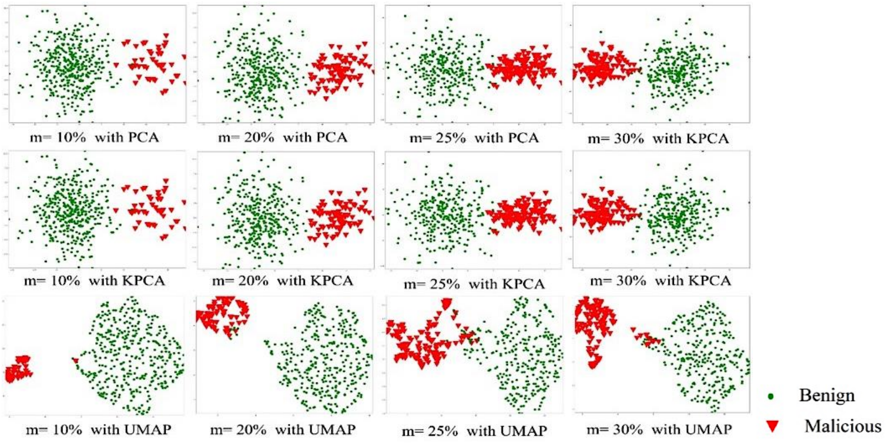


Figure 2. The proposed algorithm's clustering outcomes in distinguishing between malicious local updates and benign updates (CIFAR-10)

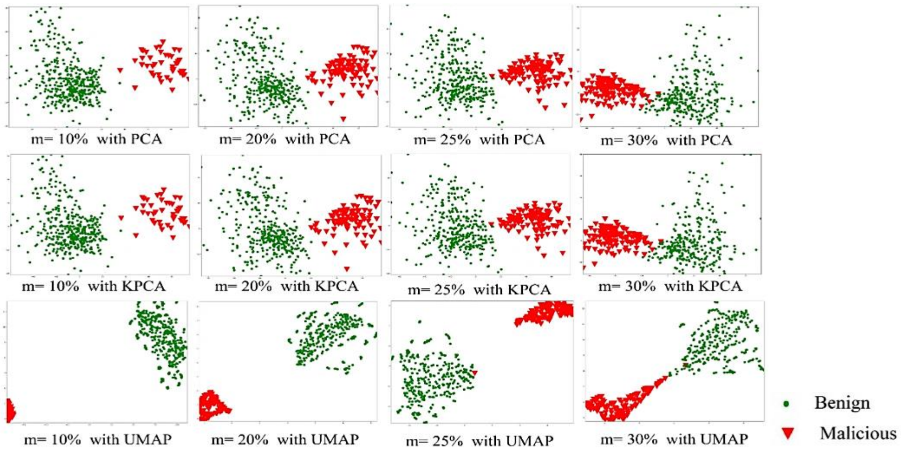


Figure 3. The proposed algorithm's clustering outcomes in distinguishing between malicious local updates and benign updates (Fashion-MNIST)

5. CONCLUSION

This research paper focuses on identifying and mitigating data poisoning threats in FL through the application of dimensionality reduction techniques. Among various techniques considered, including PCA, KPCA, and KPCA+K-mean, UMAP emerges as the most effective choice. The proposed protection technique leverages dimensionality reduction algorithms, such as UMAP, to detect malicious attempts in the FL system. By analyzing the specific characteristics of malicious parameters, UMAP can automatically identify and filter out harmful parameter updates, alleviating the need for manual inspection, especially in DNNs with numerous parameters. The study demonstrates that the FL system's aggregator can employ UMAP to locate and isolate malicious participants. Once identified, the aggregator has the option to blacklist these participants or reject their updates in subsequent rounds. The experimental results validate the effectiveness of UMAP in mitigating poisoning attacks within the FL system. Looking ahead, the researchers aim to extend their investigations by conducting experimental research on backdoor attacks. Additionally, they plan to explore and analyze the fools gold algorithm as a means to further minimize vulnerabilities in FL systems.

ACKNOWLEDGEMENTS

This work was supported by the Institute of Information and Communications Technology Planning and Evaluation (IITP) Grant by the Korean Government through MSIT (Robust AI and Distributed Attack Detection for Edge AI Security) under Grant 2021-0-00511.

REFERENCES

[1] L. Li, Y. Fan, M. Tse, and K. Y. Lin, "A review of applications in federated learning," *Computers and Industrial Engineering*, vol. 149, p. 106854, Nov. 2020, doi: 10.1016/j.cie.2020.106854.

[2] H. Brendan McMahan, E. Moore, D. Ramage, S. Hampson, and B. Agüera y Arcas, "Communication-efficient learning of deep networks from decentralized data," *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, AISTATS 2017*, 2017.

[3] C. Fung, C. J. M. Yoon, and I. Beschastnikh, "Mitigating sybils in federated learning poisoning," 2018, [Online]. Available: http://arxiv.org/abs/1808.04866.

[4] H. Xiao, H. Xiao, and C. Eckert, "Adversarial label flips attack on support vector machines," *Frontiers in Artificial Intelligence and Applications*, vol. 242, pp. 870–875, 2012, doi: 10.3233/978-1-61499-098-7-870.

[5] A. Paudice, L. Muñoz-González, A. Gyorgy, and E. C. Lupu, "Detection of adversarial training examples in poisoning attacks through anomaly detection," 2018, [Online]. Available: http://arxiv.org/abs/1802.03041.

[6] B. Yu, W. Mao, Y. Lv, C. Zhang, and Y. Xie, "A survey on federated learning in data mining," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 12, no. 1, Dec. 2022, doi: 10.1002/widm.1443.

[7] J. Kang, Z. Xiong, D. Niyato, Y. Zou, Y. Zhang, and M. Guizani, "Reliable Federated Learning for Mobile Networks," *IEEE Wireless Communications*, vol. 27, no. 2, pp. 72–80, Apr. 2020, doi: 10.1109/MWC.001.1900119.

[8] W. Wei *et al.*, "A framework for evaluating gradient leakage attacks in federated learning," 2020, [Online]. Available: http://arxiv.org/abs/2004.10397.

[9] T. Zhang, L. Gao, C. He, M. Zhang, B. Krishnamachari, and A. S. Avestimehr, "Federated Learning for the Internet of Things: Applications, Challenges, and Opportunities," *IEEE Internet of Things Magazine*, vol. 5, no. 1, pp. 24–29, Mar. 2022, doi: 10.1109/iotm.004.2100182.

[10] Y. Liu *et al.*, "Fedvision: An online visual object detection platform powered by federated learning," *AAAI 2020 - 34th AAAI Conference on Artificial Intelligence*, vol. 34, no. 08, pp. 13172–13179, Apr. 2020, doi: 10.1609/aaai.v34i08.7021.

[11] 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 - IEEE Symposium on Security and Privacy*, May 2018, vol. 2018-May, pp. 19–35, doi: 10.1109/SP.2018.00057.

[12] N. Truong, K. Sun, S. Wang, F. Guitton, and Y. K. Guo, "Privacy preservation in federated learning: An insightful survey from the GDPR perspective," *Computers and Security*, vol. 110, p. 102402, Nov. 2021, doi: 10.1016/j.cose.2021.102402.

[13] A. Katal, M. Wazid, and R. H. Goudar, "Big data: Issues, challenges, tools and Good practices," in *2013 6th International Conference on Contemporary Computing, IC3 2013*, Aug. 2013, pp. 404–409, doi: 10.1109/IC3.2013.6612229.

[14] P. Kumar, G. P. Gupta, and R. Tripathi, "PEFL: Deep privacy-encoding-based federated learning framework for smart agriculture," *IEEE Micro*, vol. 42, no. 1, pp. 33–40, Jan. 2022, doi: 10.1109/MM.2021.3112476.

[15] L. Zhou, S. Pan, J. Wang, and A. V. Vasilakos, "Machine learning on big data: Opportunities and challenges," *Neurocomputing*, vol. 237, pp. 350–361, May 2017, doi: 10.1016/j.neucom.2017.01.026.

[16] V. Tolpegin, S. Truex, M. E. Gursoy, and L. Liu, "Data poisoning attacks against federated learning systems," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 12308 LNCS, Springer International Publishing, 2020, pp. 480–501.

[17] S. Chen *et al.*, "Automated poisoning attacks and defenses in malware detection systems: An adversarial machine learning approach," *Computers and Security*, vol. 73, pp. 326–344, Mar. 2018, doi: 10.1016/j.cose.2017.11.007.

[18] J. Steinhardt, P. W. Koh, and P. Liang, "Certified defenses for data poisoning attacks," *Advances in Neural Information Processing Systems*, vol. 2017-December, pp. 3518–3530, 2017.

[19] C. Wu, X. Yang, S. Zhu, and P. Mitra, "Mitigating backdoor attacks in federated learning," 2020, [Online]. Available: http://arxiv.org/abs/2011.01767.

[20] H. Xiao, B. Biggio, G. Brown, G. Fumera, C. Eckert, and F. Roli, "Is feature selection secure against training data poisoning?," *32nd International Conference on Machine Learning, ICML 2015*, vol. 2, pp. 1689–1698, 2015.

[21] H. Zhang, N. Cheng, Y. Zhang, and Z. Li, "Label flipping attacks against Naive Bayes on spam filtering systems," *Applied Intelligence*, vol. 51, no. 7, pp. 4503–4514, 2021, doi: 10.1007/s10489-020-02086-4.

[22] P. Parameshwarappa, Z. Chen, and A. Gangopadhyay, "Analyzing atack strategies against rule-based Intrusion Detection Systems," Jan. 2018, doi: 10.1145/3170521.3170522.

[23] H. Xiao, K. Rasul, and R. Vollgraf, “Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms,” 2017, [Online]. Available: http://arxiv.org/abs/1708.07747.

[24] H. Xiao, B. Biggio, B. Nelson, H. Xiao, C. Eckert, and F. Roli, “Support vector machines under adversarial label contamination,” *Neurocomputing*, vol. 160, pp. 53–62, Jul. 2015, doi: 10.1016/j.neucom.2014.08.081.

[25] C. Zhang, Y. Xie, H. Bai, B. Yu, W. Li, and Y. Gao, “A survey on federated learning,” *Knowledge-Based Systems*, vol. 216, p. 106775, Mar. 2021, doi: 10.1016/j.knosys.2021.106775.

[26] A. Paudice, L. Muñoz-González, and E. C. Lupu, “Label sanitization against label flipping poisoning attacks,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 11329 LNAI, Springer International Publishing, 2019, pp. 5–15.

[27] C. Xie, K. Huang, P. Y. Chen, and B. Li, “DBA: Distributed backdoor attacks against federated learning,” *8th International Conference on Learning Representations, ICLR 2020*, pp. 1–19, 2020.

[28] L. McInnes, J. Healy, N. Saul, and L. Großberger, “UMAP: Uniform manifold approximation and projection,” *Journal of Open Source Software*, vol. 3, no. 29, p. 861, Sep. 2018, doi: 10.21105/joss.00861.

[29] L. Van Der Maaten and G. Hinton, “Visualizing data using t-SNE,” *Journal of Machine Learning Research*, vol. 9, no. 11, pp. 2579–2625, 2008.

[30] D. Li, W. E. Wong, W. Wang, Y. Yao, and M. Chau, “Detection and mitigation of label-flipping attacks in federated learning systems with KPCA and K-means,” in *Proceedings - 2021 8th International Conference on Dependable Systems and Their Applications, DSA 2021*, Aug. 2021, pp. 551–559, doi: 10.1109/DSA52907.2021.00081.

BIOGRAPHIES OF AUTHORS

Deepak Upreti received his BCA (Bachelor’s in computer science) from the Bangalore University, Bangalore, India, and M.Sc. (Computer Science) from the REVA University, Bangalore. Currently, he is a Ph.D. Student at the Kongju National University. He can be contacted at email: deepak@smail.kongju.ac.kr.

Dr. Hyunil Kim received a Ph.D. degree in information security from Kongju National University, South Korea, in 2019, and a Postdoctoral Researcher from Daegu Gyeongbuk Institute of Science & Technology (DGIST), Daegu, South Korea, in 2022. He is currently the Research Professor with Kongju National University, South Korea. His research interests include cryptographic applications, privacy-preserving deep learning, and private and secure federated learning. He can be contacted at email: hyunil89@kongju.ac.kr.

Dr. Eunmok Yang receives his master’s degree in Computer Engineering, Kongju University, Received Ph.D. in Mathematics in 2016. In 2016, he worked at UbiTech Research Center. He worked as a researcher at Soongsil University's Industry-University Cooperation Foundation. Since 2020, he has been working as a research professor at Kookmin University's Department of Financial Information Security. His research interests include communication network design, intrusion detection, data mining, machine learning, and security. He can be contacted at email: emyang@kongju.ac.kr.

Dr. Changho Seo received the B.S., M.S., and Ph.D. degrees in mathematics from Korea University, Seoul, South Korea, in 1990, 1992, and 1996, respectively. He is currently a Full Professor with the Department of Convergence Science, Kongju National University, South Korea. His research interests include cryptography, information security, data privacy, and system security. He can be contacted at email: chseo@kongju.ac.kr.