Defending against Label-Flipping Attacks in Federated Learning Systems with UMAP

Deepak Upreti  
Kongju National University

Hyunil Kim  
Kongju National University

Eunmok Yang  
Kookmin University

Changho Seo (chseo@kongju.ac.kr)  
Kongju National University

Research Article

Keywords: Federated Learning, Uniform Manifold Approximation and Projection (UMAP), PCA, K-mean

Posted Date: August 25th, 2022

DOI: https://doi.org/10.21203/rs.3.rs-1984301/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License. Read Full License

Additional Declarations: No competing interests reported.
Defending against Label-Flipping Attacks in Federated Learning Systems with UMAP

Deepak Upreti¹, Hyunil Kim¹†, Eunmok Yang²† and Changho Seo¹*†

¹*Department of Convergence Science, Kongju National University, Chungcheongnam-do, Gongju-si, 32588, South Korea.
²*Department of Information Security, Cryptology, and Mathematics, Kookmin University, Seoul, 02707, South Korea.

*Corresponding author(s). E-mail(s): chseo@kongju.ac.kr; Contributing authors: deepak@smail.kongju.ac.kr; hyunil89@kongju.ac.kr; emyang@kookmin.ac.kr; †These authors contributed equally to this work.

Abstract

The importance of computing and storage capacity has increased over time, and the importance of data mining in industrial engineering has become more apparent. Recently, artificial intelligence and machine learning have made significant advancements in industrial engineering. Federated learning is a machine learning technique that aims to solve the problem of distributed computing systems and their applications of data storage while ensuring data privacy. Tolpegin et al. conducted research on data poisoning attacks in a federated learning system, which we have extended to an analysis of the efficiency of Tolpegin’s proposed defense technique. We have subsequently compared the efficiency using uniform manifold approximation and projection (UMAP), principal component analysis (PCA), kernel PCA (KPCA), and K-means clustering algorithms. This study confirms that UMAP performs better than PCA, KPCA, and K-means, and provides excellent performance in mitigating data-poisoning attacks.

Keywords: Federated Learning, Uniform Manifold Approximation and Projection (UMAP), PCA, K-mean
1 Introduction

Federated learning (FL) is a machine learning (ML) approach that seeks to overcome a challenge of distributed learning, that of data storage while maintaining data privacy [1]. It represents numerous clients (such as mobile devices and other IOT devices). However, the FL system faces a serious threat of data poisoning which, if not addressed as soon as possible, could lead to system failure [2]. Although federated learning has increasingly been applied in mobile environments, some analytical problems occur, including trusted worker selection for model training [3]. Different attack methods have been found in federated learning for model performance and data privacy, such as data poisoning, model poisoning, model inversion, membership inference, and generative adversarial network (GAN) reconstruction attacks. Existing poisoning attacks affect the training data set; for example, in data poisoning attacks, malicious data are inserted into the training data set, while model poisoning attacks involve manipulating the ML model to make a wrong outcome by providing a specific input. In poisoning attacks, malicious workers intentionally manipulate a small percentage of training data or insert malicious data into the training data sets to increase the probability of misclassification [3].

The transmission of gradients and partial parameters can lead to indirect privacy leaks and incomplete protection of the system. [4], [5]. A small portion of the gradient can reveal information regarding local data [6]. In federated learning systems, several attack methods, such as data poisoning, model poisoning, model inversion, membership inference, and GAN reconstruction attacks have been observed. In federated learning, sensitive information about a device can be revealed, although local data are unavailable.

Using a label flipping attack, an attacker can use training data labels that can poison the data [1]. The model worked smoothly and was undisturbed. Model poisoning attacks aim to cause large errors in the model, thereby resulting in a malfunctioning federated learning system. Data poisoning may be performed by any participant. Data poisoning is becoming more popular because it is simple to perform. An earlier study in this area studied a poisoning attack on a support vector machine (SVM) [7]. This attack was classified as data poisoning. In addition, various researchers have examined data poisoning attacks and defenses, which are more closely tied to machine learning systems than the FL system [8]. Other researchers have examined the filtering of spam [9], the detection of malware and network problems [10], [11], computer vision [12], and recommender systems [13], among other types of data poisoning assaults. Defenses against optimal poisoning attacks include recognizing malicious cases and eliminating them from the data set for training [14], or the solution of a robust optimization method [15], [16].

To address these critical challenges, this study has been proposed. This study offers a protection strategy for FL systems. In this study, we have investigated the effectiveness of Tolpegin’s data-poisoning defensive technique in FL systems. Then, we have presented an enhanced defensive method that exploits uniform manifold approximation and projection (UMAP). The remainder of
this paper is organized as follows. Section 2 describes federated learning and
label flipping attacks. An analysis of the defense strategies has been docu-
mented in Section 3. Finally, in Section 4, we compare the findings of the
investigation into the effectiveness of various protection strategies.

2 Preliminaries

2.1 Federated Learning

As machine learning (ML) prediction is based on the data presented and as
the amount of data increases, data analysis for prediction accuracy becomes a
difficult task. This is because the privacy of the data is also a concern when
they are collected for the model. Federated learning (FL) is an ML technique
that trains an algorithm without exchanging data by using numerous decen-
tralized edge devices or clients [17]. Therefore, FL addresses the data privacy
issues by requiring users to share only their model parameters rather than the
raw data. As a result, a global data model is formed in which all participants
submit their parameters to a server, and decisions are made using this informa-

However, there are many threats to FL in various aspects, including data poi-
soning [1]. Data poisoning aims to lower the quality of the final learning model,
causing misclassification, regression, etc. In this attack scenario, malicious or
compromised clients intentionally change the data features or labels. In partic-
ular, the label-flipping attack described in the next section is the simplest and
most powerful attack that flips the label of each training data point. Therefor-

2.2 Label Flipping Attacks in FL

One of the most popular data-poisoning attacks in ML is the label-flipping
attack [18], [19]. Label flipping is a type of data poisoning in which the labels
of the training data are flipped or modified, causing the model’s classification
performance to degrade [20]. For example, in CIFAR-10, a label-flipping attack
may mislabel the airplane class as the bird class [1]. FL lacks a centralized cura-
tor for data analysis, because of which it is at risk of data poisoning [1]. Data
poisoning can be implemented by a compromised participant who appears as
an FL system participant. These could have been updated by an attacker or contained malicious intent [1]. They may provide training data updates that are mislabeled or contain poisonous samples. The updates provided by malicious participants are not filtered in the absence of a central server; hence, the malicious global model continues to be trained. Data poisoning is untraceable using conventional methods; however, dimensionality reduction and clustering algorithms can detect it [1].

3 Analysis of Defense Strategies

This study extends the works of Tolpegin et al. [1] and Li et al. [2] related to FL systems. We have learnt how successful label-flipping attacks on the FL system can be from their research as well as the possibility of using a dimensionality algorithm to protect against it. We have enhanced previous research by applying a dimensionality reduction approach to clustering and comparing it. Instead of principal component analysis, we used uniform manifold approximation and projection (UMAP) for dimension reduction [21]. UMAP is a powerful nonlinear dimensionality reduction method compared to t-SNE [22]. It assumes that the data are uniformly distributed on the manifold, which is defined by the Riemannian metric on the manifold. In addition, UMAP is relatively faster than any other dimensionality reduction methods, including PCA and t-SNE [21]. For our investigation, we used the same architecture and dataset, CIFAR-10 and Fashion-MNIST, as previously used in Tolpegin et al. to compare the experimental results using several dimensionality reduction methods.

3.1 FL System Setup

In this experiment, we implemented data poisoning in the FL for label flipping. Here, we used the PyTorch library in Python programming to implement our attack and defense on FL [23]. By default, we assigned $N = 50$ participants, one central aggregator, and one participant in each round $k = 5$. In addition, we implemented targeted data poisoning in the FL for label flipping. Specifically, we flipped a label $l$ to $l'$, where $l \in L$ and $l' \in L, l' \neq l$ where $L$ is the number of classes in the classification problem. Specifically, we assumed that all participants would receive an equal distribution of the entire training data set randomly, with each participant receiving a different portion of the training data. FL tests were performed for $R = 200$ rounds.

3.2 Deep Neural Network Setup

In our experiments, the CIFAR-10 and Fashion-MNIST data sets were used. CIFAR-10 image collection is often used to train ML and computer vision algorithms. The 10 classes represent airplanes, cars, birds, cats, deer, dogs, frogs, horses, ships, and trucks [2]. Fashion-MNIST is a data set comprising a training set of 60,000 items and a testing set of 10,000 items. The 10 different
classes represent T-shirt/top, trouser, pullover, dressing, coat, sandal, shirt, sneaker, bag, and ankle boot [24]. For comparison with the same experimental setup, we used the same network architecture described in [1] and [2]. In Fashion-MNIST, we used two convolutional layers with batch normalization, two max-pooling layers, and one fully connected layer. In CIFAR-10, we used six convolutional layers with batch normalization, two fully connected dense layers, and one layer running the softmax function [2].

3.3 Data Poisoning Attack Setup

In this attack, we assigned the number of participants $N$ of which $m\%$ were malicious clients. We randomly selected $N \times m$ from participants $p$ where the rest were honest participants at the beginning of each experiment. To improve the precision of the experiment, we dealt with the consequences of the random selection of malicious participants. Each experiment was repeated ten times, and the average outcome was used as the result. For CIFAR-10, we tested with replacement of dog with cat, airplane with bird, and automobile with truck; for Fashion-MNIST, we experimented with the replacement of shirt with t-shirt, trouser with dress, and coat with shirt.

3.4 Our Defense Strategy

We investigated the defense strategy of Tolpegin et al. before presenting our own defense strategy. We begin by examining the possibility of using label flipping attacks on poison FL systems. The principle underlying the defense strategy is that dimensionality reduction algorithms can detect malicious updates because modifications of parameters have distinctive properties [1]. However, deep neural networks have many parameters, and it may be difficult to manually check for every harmful parameter update. An automated strategy that can be used is PCA, KPCA and UMAP to locate and filter the parameters sent by the malicious updates, as demonstrated in the study [2].

Algorithm 1 Robust Model Updates in FL with UMAP

function Model_Update(X, n, d, min-dist, n-epochs \( w_t \))
1: \( v \leftarrow \phi \)
2: each round \( t = 1, 2, \ldots \) do
3: \( w_t \leftarrow \text{previous global model parameters} \)
4: \#\# executed for each client locally
5: for each client \( i=1,2,\ldots, I \) do
6: \( w_i^t \leftarrow \nabla l(w_t) \)
7: \( u \leftarrow \{w_i^t\}_{i=1}^I \)
8: \( u' \leftarrow \text{standardize}(u) \)
9: \( u'' \leftarrow \text{UMAP}(u', n, d, \text{min-dist}, n - \text{epochs}) \)
10: plot \( (u'') \)
The rate of malicious client presence (m) used ranged from 2% to 50%. As per the outcomes, the test accuracy of the global model declines as m increases. Additionally, even with a slight increase in m, when compared to a non-poisoned model, the global-model test accuracy decreases, but in this situation, the source-class recall of the model shows a much more significant drop. The global model test accuracy for CIFAR-10 falls to 76% in the poisoned situation when m is 40%, compared with 78.3% in the non-poisoned model case. Like this, the source-class recall decreased to approximately 0%. The source class declined by approximately 10% when m was 10%. This demonstrates that an adversary can still reduce the accuracy of the global model even if they shift only a small fraction of the participants. As a result, an FL system can be dramatically affected by even a small number of participants under the control of an adversary. The CIFAR-10 and Fashion-MNIST data sets were vulnerable to label-poisoning attacks, although the level of vulnerability varied. The results of the experiment demonstrated that Fashion-MNIST was more sensitive than CIFAR-10. During the experiment, we found that an attacker does not necessarily need to know the most vulnerable source or target class. Because the performance of the non-poisoned model’s misclassification does not necessarily correlate with the attack effectiveness, as we can see from the section above, a high-utility convergence is finally possible if malicious participation is removed. This result is consistent with that of Tolpegin’s study. The outcome demonstrated above shows that Tolpegin’s suggested technique reduces the impact of label-flipping attacks on FL systems.

3.5 Defense to Label-Flipping Attacks with UMAP [21]

\begin{algorithm}
\textbf{Algorithm 2} UMAP \((X,n,d,\text{min-dist},n\text{-epochs})\)
\begin{algorithmic}
\State $X$: the data set to have its dimension reduced
\State $n$: the neighborhood size to use for local metric approximation
\State $d$: the dimension of the target reduced space
\State $\text{min-dist}$: an algorithmic parameter controlling the layout
\State $n\text{-epochs}$: controlling the amount of optimization work to perform
\State \#	ext{ construct the relevant weighted graph}
\ForAll{$x \in X$}
\State $\text{fs-set}[x] \leftarrow \text{LocalFuzzySimplicialSet} \left(X, x, n\right)$
\EndFor
\State $\text{top} \leftarrow U_{x \in X}$
\State \#	ext{ perform optimization of the graph layout}
\State $Y \leftarrow \text{Spectral Embedding} \left(\text{top} - \text{rep}, d\right)$
\State $Y \leftarrow \text{optimize Embedding} \left(\text{top} - \text{rep}, Y, \text{min - dist, n - epochs}\right)$
\State \textbf{RETURN} $Y$
\end{algorithmic}
\end{algorithm}

Algorithm 1 illustrates our robust model updates in FL with UMAP, while Algorithm 2 illustrates UMAP. In general, UMAP is a simple algorithm.
Probabilistic t-conorm has proven to be the most successful method for executing a fuzzy union over a local fuzzy simplicial set. In Algorithm 2 in [21], we enter the data set X to reduce its dimensions i.e., the neighborhood size n to use for local metric approximation, the dimension d of the target reduced space, min-dist. an algorithmic parameter controlling the layout, and n epochs that control the amount of optimization work performed. In addition, the fuzzy simplicial set local to a given point x can be constructed by locating its n closest neighbors, generating the appropriate normalized distance on the manifold, and then converting the finite metric space to a simplicial set via functor FinSing, which translates into an exponential of the negative distance in this case. We used a smoothed version of the k-nearest neighbors-distance algorithm to fix the cardinality of the fuzzy set of 1-simplices to a fixed value instead of directly using the distance to the nth nearest neighbor as the normalization. Based on empirical experiments, log2 (n) was chosen for this purpose. In addition, spectral embedding was performed by considering the 1-skeleton of the global fuzzy topological representation as a weighted graph and using standard spectral methods on the symmetric normalized Laplacian. Finally, the optimization process was performed using stochastic gradient descent, as described in Algorithm 5 [21].

Fig. 1 The test accuracy for global model and source recall in FL obtained by m adversaries under various defense strategies while attacking the CIFAR-10 and Fashion-MNIST.
3.6 Experiments Setup

In our experiment, an adversary used the label flipping technique to target CIFAR-10 and Fashion-MNIST. The output is displayed for the five cases tested as a bar graph in figure 2. In the first case of the data set, no dimensionality reduction method was employed. In the second setting, PCA was used as the defense algorithm. In the third setting, KPCA was used as a defense strategy. In the fourth setting, KPCA and K-mean were used. UMAP was used in the final case to defend against adversarial database attacks. The same parameters were used to test the accuracy of the global models, and thus each includes five related graphs. The two-dimensional graph demonstrates the success of the algorithm when employing PCA, KPCA, KPCA + K-mean, and UMAP to distinguish between malicious and honest updates delivered by adversaries. The blue color represents malicious updates, whereas yellow represents updates given by honest individuals. Graphs were created using the output of the proposed algorithm.

Fig. 2 The clustering results of the proposed algorithm in classifying malicious local update with benign update (CIFAR-10)

4 Comparison and Analysis

Figure 1 illustrates the bar charts for two data sets, CIFAR-10 (a) and Fashion-MNIST (b), against the targeted data poisoning attack on test accuracy and
source recall, where the percentages of malicious participants in both experiments ranged from 0% to 50%. As a result, our approach using UMAP is the most accurate when compared with previous results [1], [2]. The accuracy of the CIFAR-10 model ranged from 68 to 80 and the source recall values varied from 0 to 70. The model accuracy values of Fashion-MNIST varied from 70 to 92, and the recall values varied from 0 to 90. The graph compares five different states, from the left-most to the right-most, in each column. The graph in the first case indicates that, as the number of malicious users increases, the accuracy decreases. The drop was initially constant; however, when the percentage of malicious individuals exceeded 10%, accuracy declined considerably. The accuracy does not change and remains at a minimal level when the malicious rate is 40%–50%. When the FL system exceeds 10%, the source recall results demonstrate the same outcome as observed with the model accuracy, and drops dramatically beyond 10% and reaches zero. In the second case, the graph represents a situation in which PCA is used to protect against data poisoning. It clearly shows that a minimal drop in accuracy until m reaches 30%. On exceeding 30%, there is a slight decrease in the accuracy. In the third case, KPCA is used [2] instead of PCA. The graph illustrates that there is less decline in accuracy compared to the case where PCA was employed. When KPCA was applied, the accuracy did not decrease until the fraction of malicious participants exceeded 4%, while when PCA was applied, the accuracy began to decline as the percentage exceeded 2%. The fourth column in figure
shows the case in which KPCA with K-means was used [2], and the result is similar to that of the third column, with a slight increase in accuracy. However, the K-mean methods is a more time-consuming method; therefore, it is not the best choice. Finally, the rightmost column in figure 1 is the result of the use of UMAP, showing the best result compared to other dimension-reduction techniques. UMAP is an extremely strong tool with several benefits over t-SNE and other dimensionality reduction techniques. UMAP is quick and scalable in terms of the data set size and dimensionality; for instance, it can project the 784-dimensional, 70,000-point MNIST data set in less than 3 min, while scikit-t-SNE learn’s implementation takes 45 min [22]. UMAP tends to maintain the global structure of the data better because of its strong theoretical foundations, which allow the algorithm to better balance stressing local versus global structures. Therefore, UMAP is a more effective tool for displaying high-dimensional data because of its enhanced speed, greater preservation of the global structure, and more intelligible parameters. It is observed that UMAP is the best method for protecting against data poisoning attacks. As a result, we can see that UMAP significantly improves the clustering results in FL against data poisoning attacks. Figures 2 and 3 show the ability of the proposed defensive algorithm and compare it to that proposed previously [1] and [2]. The results of the PCA defense method are shown in the first row, with m = 2%, 4%, 10%, and 20%. Similarly, in the second row, the KPCA defense method is implemented. In the third row, the UMAP method is used. From the figure, it can be determined that malicious updates differ from honest updates and the extent to which the different algorithms can differentiate between them. Comparing the three rows, it can be seen that UMAP differentiates malicious and honest updates better than PCA and KPCA can. Thus, using UMAP in the defense algorithm provides a better course of action.

5 Conclusion

In this paper, we summarize how to detect and avoid data-poisoning threats in FL using dimensionality reduction. The PCA, KPCA, and UMAP algorithms can assist FL systems in detecting malicious attempts. However, this technique is insufficient for countering the threats presented by data-poisoning attacks. This paper proposes the use of UMAP instead of PCA and KPCA. The results of the simulations carried out suggest that using UMAP instead of PCA or KPCA is more effective in defending an FL system from poisoning attacks. In the future, our aim is to conduct experimental research on alternate defensive approaches, as well as to explore and analyze additional dimensionality reduction algorithms to minimize FL system vulnerabilities.

Statements and Declarations

• Ethical Approval and Consent to participate
  Not Applicable
• Consent for publication
  Not Applicable
• Human and Animal Ethics
  Not Applicable
• Human and Animal Ethics
  Not Applicable
• Availability of supporting data
  Not Applicable
• Competing interests
  The authors declare that they have no Competing interests
• Funding
  Not Applicable
• Authors’ contributions
  Conceptualization: Deepak Upreti, Hyunil Kim;
  Methodology: Deepak Upreti;
  Formal analysis and investigation: Deepak Upreti, Hyunil Kim, Eunmok Yang;
  Writing - original draft preparation: Deepak Upreti;
  Writing - review and editing: Hyunil Kim, Changho Seo;
  Funding acquisition: Changho Seo; Resources: Eunmok Yang;
  Supervision: Changho Seo.
• Acknowledgments
  This work was supported by the Institute of Information and Communications Technology Planning and Evaluation (IITP) Grant by the Korean Government through MSIT (Development of User identity certification and management technology for self-sovereign identity applications.) under Grant 2021-0-00565.

References


