International Journal of Computer Sciences and Engineering
Vol.12, Issue.7, pp.53-59, July 2024
ISSN: 2347-2693 (Online)
Available online at: www.ijcseonline.org

ISSN: 2347-2693 (E)

Research Article

A Framework for Prevention of Backdoor Attacks in Federated Learning Using Differential Testing and Outlier Detection

C. B. Biragbara[1*], O.E. Taylor[2], D. Matthias[3]

[1,2,3]Computer Science/Science, River State University, Port Harcourt, Nigeria

*Corresponding Author: mayowacynthia@gmail.com,

Received: 20/May/2024; **Accepted**: 21/Jun/2024; **Published**: 31/Jul/2024. **DOI**: https://doi.org/10.26438/ijcse/v12i7.5359

Abstract: The integrity and security of federated learning systems, in which several users work together to train a single model while maintaining privacy, are seriously threatened by backdoor attacks. In this paper, we propose a preventive approach that includes differential testing and outlier detection mechanisms to identify and mitigate the risks associated with backdoor attacks. Federated learning aims for high model accuracy and performance, and the proposed preventive measures help maintain the integrity and reliability of the collaborative learning process. Differential testing is used to detect possible deviations or inconsistencies in the distribution of training data across multiple participants. By comparing the performance of models on different subsets of data, the presence of a backdoor attack can be identified. This differential testing framework acts as an early warning system, enabling the detection of introduced model biases or malicious attempts at data manipulation. Anomaly detection methods are also employed to find abnormalities or peculiar patterns that can point to the existence of a backdoor attack. When an outlier substantially deviates from the federated learning system's expected behavior, it is identified and marked for additional research. This approach improves the robustness of federated learning models against malicious participants and manipulated data. Object-Oriented Analysis and Design (OOAD) techniques were used to ensure a structured and methodical design process. Python programming language was used for model implementation and simulation. The suggested defense strategy, which makes use of a federated CNN model, successfully reduces the possibility of backdoor attacks in federated learning systems. Other benchmark systems were compared with the suggested model. The results of the proposed system are stronger than existing systems as it achieves an accuracy of 99.95% in training and 99.97% in testing. In conclusion, we have detected backdoor attacks with different counts using both outlier and differential tests and prevented backdoor attacks (Differential, Gaussian_Backdoored, Gradient_Backdoored, Integral, Julia_Backdoored, Metaphor_Backdoored, Non-Backdoored, Pixel_Distort, Relu_Backdoored) using federated learning.

Keywords: Framework, Prevention, Backdoor, Attacks, Learning and Outlier Detection)

1. Introduction

Deep learning models require large datasets to function well. Using standard training methods requires gathering and centralizing training data on a computer or at a data center.
These days, people are increasingly cautious and sensitive when it comes to sharing personal information. Obtaining data from multiple sources has become more expensive and challenging. With the help of federated learning, machine learning models can be cooperatively trained across several devices or organizations without requiring the sharing of raw data. Traditional machine learning techniques collect all of the data in one location and use it to train a single model. But because it demands the collection and processing of all data in one place, this method may prove to be inefficient [1].

Federated models are constructed by combining member-submitted updates to the model. The aggregator is not aware that these updates are being generated in order to protect the training data's privacy by design. Model poisoning attacks are more effective at affecting federated learning than poisoning attacks that are restricted to the training data. However, malicious clients are still able to attack federated learning technologies. Even with a secure aggregation protocol introduced to the system, Due to the server's lack of access to the data used to inform the clients' model modifications, the privacy of the clients will not be completely secured [2]. Malicious clients might theoretically send the server random updates. If there are insufficient security measures on the server to identify and handle these malicious updates.

These techniques aim to differentiate between benign and malicious updates. Byzantinerobust aggregation rules utilizing the model weights' statistical features include Krum, Bulyan, reduced mean, and median. However, because of the non-IID-based data dissemination across many clients, which offers attackers plenty of leeway to conceal malicious updates and stay undiscovered, Federated learning was not able to detect backdoor attacks. Federated learning is a distributed machine learning technique that allows several users to train a model concurrently without sharing data. Despite being more popular since it respects privacy, this method is not impervious to security risks like backdoor attacks [3]. The attacker incorporates a backdoor that is triggered by particular inputs or patterns into the malicious model's training process in order to initiate a backdoor attack on federated learning. After the model is put into use, the attacker can control the model's predictions and activate the backdoor by using the trigger. "Backdoor neurons" are neurons that only fire when an image containing a backdoor is present, and they are frequently activated by backdoor attacks. Studies have indicated that eliminating these "backdoor neurons" can effectively reduce backdoor attacks without having a major effect on model performance. But because these cleaning methods depend on a reliable supply of "clean" data, they are not immediately applicable in this context; federated learning situations, which are meant to safeguard consumer privacy, may not provide the same level of assurance [4]. Therefore, in order to lessen the effects of model inversion by deleting harmful data from the training dataset or retraining the model, we explicitly construct a backdoor (outlier) detection mechanism in this study that can detect the presence of backdoor features in federated learning. Additionally, the system is constructed using the Python programming language, and the robustness of the backdoor detection process is enhanced by including differential testing and outlier identification.

2. Related Work

Malware that permits illegal access to a computer system, network, or application is known as a backdoor. Backdoors are made to get around standard authentication processes and give an attacker covert access to a system. They are frequently installed by hackers who have obtained access to a system through other techniques, like taking advantage of security flaws or tricking a user into downloading malware via phishing. Backdoors can be used to remotely take over a system, alter files, or steal confidential information. Additionally, they can be used to start more assaults on other systems or to build a botnet a collection of compromised computers that can be utilized in concerted attacks. Because backdoors are made to blend in with authentic system files, they are frequently hard to find. To hide them from view, they can also employ names and file extensions that look similar to those of real files. Furthermore, some backdoors are made to only function in specific ways, which makes it even harder to find them. It's critical to maintain your operating system, applications, and antivirus software updated with security patches if you want to defend against backdoors. It might also be helpful to routinely check your system for malware and questionable files, as this can assist find backdoors before an attack can take place. Even in cases where a backdoor has been built, strong authentication techniques like two-factor authentication can aid in preventing attackers from accessing a system [6]. All things considered, backdoors are a major risk to the safety of networks and computer systems. They can be used to initiate a range of attacks on both individuals and organizations, and they are challenging to detect. Strong security measures and constant attention to backdoor indicators can help thwart such assaults and reduce the likelihood of a successful incursion [7]. In machine learning, backdoor attacks include inserting hidden triggers into the model's training process, which leads to the model misclassifying inputs when the trigger is present. a description of the several backdoor attack types that you described, including the ReLU backdoor attack, the Gaussian backdoor attack, the gradient attack, the integral attack, the Julia attack, the metaphor assault, the non-backdoor attack, and the pixel distortion attack [8]. Federated learning is a machine learning technique that allows many parties to train a model simultaneously without sharing raw data. Instead, every participant trains a local model with its own data, which it then integrates to create a global model.

There are various benefits to this technique, such as less communication burden and enhanced privacy. Since its initial introduction by Google in 2016, federated learning has grown in prominence across a number of sectors, including telecommunications, banking, and healthcare. Federated learning, for instance, can be used in the healthcare sector to train models using patient data while maintaining patient privacy. Federated learning in finance can be used to identify fraud across several organizations without exchanging sensitive information [9]. Phases one through three of the federated learning process are startup, local training, and global aggregation. Local models are supplied to the parties during the initialization phase, while a global model is initialized with random weights. Using its own data, each party trains a local model in the local training phase using the existing global model. The local models are then merged into a new global model during the global aggregation phase, and the process is repeated until convergence is attained. Federated learning is not without its difficulties, though. Making sure the local data is diverse and of high quality is one of the biggest challenges. The data utilized to train the local model determines its quality. Maintaining security is another difficulty since malevolent actors can change the local model [10]. Federated learning is a machine learning approach that allows multiple clients to train models at the same time without sharing raw data. Federated learning can take many different shapes, each with unique features and uses. Several popular varieties include federated transfer learning, federated reinforcement learning, federated vertical learning, and federated horizontal learning [10]. Outlier detection is the process of identifying data points or observations that significantly deviate from the expected pattern or distribution of the remaining data. A number of things, including as measurement mistake, data entry problems, and extreme numbers in the underlying population, can result in outliers. To ensure that the results of data analysis are reliable and accurate, outliers must be located and removed. Numerous methods exist for detecting outliers, each with unique benefits and drawbacks. The Z-score, box plot, Mahalanobis distance, local outlier coefficient (LOF), and isolated forest approach are a few of the most widely utilized techniques [13].

This provides a comprehensive overview of the current state of backdoor attacks against Federated Learning. The authors present an overview of the corpus of studies on the many types of backdoor attacks. The paper discusses three techniques for carrying out backdoor attacks on Federated Learning: model replacement, data poisoning, and model poisoning. The findings show that model accuracy can be significantly decreased by backdoor attacks. The analysis also identifies a number of gaps in the literature, highlighting the need for more robust detection and protection mechanisms. [14]. An approach that uses dynamic model learning rate adjustments during training to identify and stop backdoor threats in federated learning. Based on how well the model performs on data from participating clients, the authors employ a Bayesian optimization strategy to modify the learning rate.

The authors give an example of how their technique could effectively detect and thwart backdoor attacks on datasets that are used to analyze human behavior. Nevertheless, this tactic might not work in circumstances where attackers are proficient and could alter their attacks to benefit from the learning rate adjustment technique [15]. A security architecture that uses ensemble methods to integrate the predictions of several models trained with various random numbers. This tactic seeks to increase the range of models and lessen the impact of backdoor attacks. The proposed defense mechanism was evaluated using three benchmark datasets and three backdoor attack scenarios. The results showed that the proposed defense framework works better than the existing defenses and can effectively reduce the impact of backdoor assaults on model accuracy. The recommended defense structure requires the training of multiple models, which could be computationally and time-consuming. Moreover, the security system might not be able to identify and prevent every type of backdoor attack [15].

3. Methodology

Object-Oriented Design (OOD) is the methodology that is employed. OOD is a class-based, object-oriented technology. Software is organized as a collection of discrete objects that combine behaviors (processes) and data structures that are based on the actual components the system interacts with.

The work was deemed most appropriate for a seamless development methodology, enhanced validation, and heightened coherence across analysis, design, and execution. Using object-oriented programming and visual modeling in the analysis and design of an application, system, or company, OOD is a technological methodology that manages stakeholder communication throughout the software development process and the quality of the final output. This process involves defining the issue, figuring out what the needs of the users are, and creating a model.

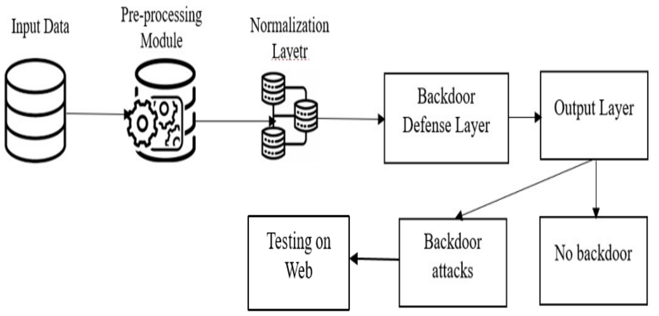


Figure 1: Architecture of the Proposed System

The information was extracted from the Kaggle.com web database. The "Detection of Backdoor assaults" field is covered by this dataset. The dataset consists of nine classes. A variety of trigger types have been applied to the images in nine distinct classes. While some of these triggers are visible to the unaided eye, others are not. The following steps are included in the dataset's pre-processing stages: The MNIST images were initially in a 2D format (176 x 176 pixels), but we need to flatten them into a 1D vector to use them efficiently. The following provides an illustration of the flattening process:

$$X[i] = X[i].\text{flatten}() \quad (1)$$

The pixel values are normalized to ensure that they are within a specific range: Normalizing the pixel values is standard procedure (e.g., [0, 1] or [-1, 1]). The normalization can be expressed as follows:

$$X[i] = X[i] / 255.0 \quad (2)$$

To scale the values between 0 and 1, we divide each pixel value by 255.0 in this case. By ensuring that the MNIST picture pixel values fall into a regular distribution, these normalization strategies can enhance the performance of the proposed model and are within a predictable range.We'll call this federated learning model M. It was created with a range of local datasets from different clients. In the case of MNIST photographs, the handwritten digit images and their corresponding labels would make up the local datasets.Backdoor Trigger Injection: A backdoor attack starts with inserting a trigger pattern into the chosen customers' local datasets. Typically, the trigger pattern was created by making a small overlay or modification to the original photographs. This process can be explained mathematically as follows:

$$X' = X + \delta \quad (3)$$

Where δ is the perturbation applied to the original image, X is the original image, and ′X′ is the altered picture featuring the pattern of triggers.

Label Manipulation: The poisoned samples have to have their labels changed as the following stage. The purpose of a targeted backdoor attack is to train the model to anticipate a specific target label each time it comes across an image that has the trigger pattern in it. The following is a mathematical explanation of this process:

$$Y' = Yt \qquad (4)$$

Where Y stands for the image's original label, ′Y′ for the altered label, and Yt for the backdoor attack's target label.

1. Aggregation and Model Update: Once the poisoned samples are generated and labeled, by aggregating their local changes, the customers' local models take part in the federated learning process.After then, the global model is adjusted based on these combined modifications.

2. Inference and Backdoor Activation: During the inference stage, the model sees images that could or might not have the trigger pattern. The backdoor is active and the model predicts the altered label rather than the genuine label when an image that has been compromised and has the trigger pattern on it is examined. A mathematical model of this would be:

$$P(Ypred\ X') = P(Yt\ X') \qquad (5)$$

Where Ypred is the label that the model predicts to be assigned to the altered image ′X′ that has the trigger pattern on it.

Outlier detection is the act of locating and annotating data items that significantly deviate from normal or expected behavior. For the purpose of outlier detection, a variety of mathematical expressions and methods can be applied. Here are a handful that are frequently used:

1. Z-Score: The Z-score algorithm identifies outliers by measuring how many standard deviations a data point is away from the mean. A data point is deemed to be an outlier if its Z-score is higher than a certain level, usually 2 or 3.

Z-score = $(x - \mu) / \sigma$ where x is the data point, μ is the mean of the data, and σ is the standard deviation of the data. The improved Z-score algorithm is a stable version of the Z-score method that is less affected by extreme values. It uses the median and median absolute deviation (MAD) in place of the mean and standard deviation.The updated Z-score is 0.6745 * (x - median) / MAD.

The interquartile range (IQR), or the range between the 25th and 75th percentiles of the data, is used by the boxplot method to identify outliers. Outliers are defined as data points that are above the upper whisker (Q3 + 1.5 * IQR) or below the lower whisker (Q1 - 1.5 * IQR). Lower whisker = Q1 - 1.5 * IQR Upper whisker = Q3 + 1.5 * IQR where Q1 is the 25th percentile, Q3 is the 75th percentile, and IQR = Q3 - Q1. One method for identifying backdoor threats in machine learning models is differential testing. It contrasts how well the model performs or behaves with data that is clean with data that might include backdoor triggers. Generally speaking, the process of doing differential testing involves training a machine learning model with a clean dataset that reflects the system's typical behavior. Make a dataset including possible triggers for backdoors. Certain patterns or modifications in the input data may serve as these triggers, causing the backdoor behavior to be activated. Evaluate the model's performance using a different set of pristine data. Assess performance indicators that are relevant to the work, such as accuracy, precision, recall, or other metrics. Evaluate your model's performance using a dataset that might include backdoor triggers. Examine how your model behaves and performs on this dataset in comparison to clean data. Examine how the behavior of the data with the backdoor trigger and the clean data differs. Keep an eye out for notable variances in performance indicators, like: B. A sharp decline in accuracy or a modification to the model's decision boundaries. These differences could be a sign that a backdoor assault is active. The backdoor protection module's component design breaks down the interconnections between its subcomponents and demonstrates how to stop backdoor assaults on federated learning. Here is a thorough breakdown of the component design: When a hostile party introduces tainted data samples into a federated learning system, it is known as a data poisoning attack. These contaminated data samples are deliberately designed to influence a model's behavior when it is trained using combined data from several subjects. Data poisoning aims to create a backdoor or cause a model to misclassify something when it encounters particular input patterns. In a federated learning system, model poisoning attacks modify a global model that is shared by several participants. During training, a hostile participant tries to insert a backdoor by changing the model's architecture or parameters. This can be accomplished by submitting a maliciously modified model update that has been carefully designed. Once in place, the poisoned model responds to most inputs normally before exhibiting the intended malicious behavior in response to specific triggers. Through the manipulation of model parameters during the aggregation step, attacks using parameter poisoning aim to compromise the federated learning process's integrity. In federated learning, users train locally on the data and submit only modified model parameters to be aggregated to a central server.

Algorithm 3.1: Back door attack on Federated Learning

```
1. Initialize the global model parameters:
   i.   θ_global = initial_model_parameters()
2. Repeat until convergence:
3. For each participating client c:
   i. Receive the global model parameters θ_global from the server
   ii. Update the local dataset D_c with any new labeled data
4. Train the local model on the local dataset:
   i. θ_c = local_training(θ_global, D_c)
5. Check for backdoor patterns in the local model:
   i. backdoor_detected = is_backdoor_pattern_present(θ_c)
6. If backdoor_detected:
   i. Clean the local model by removing the backdoor pattern:
   i θ_c = remove_backdoor_pattern(θ_c)
7. Send the cleaned local model parameters θ_c to the server
8. Aggregate the cleaned local models on the server:
   i. θ_global = aggregate_models([θ_c1, θ_c2, ..., θ_cn])
```

9. Send the updated global model parameters θ_global to all clients
2. Terminate when convergence criteria are met (0.5).

4. Results and Discussion

Following an exploratory data analysis, the dataset underwent reshaping and normalization. Two sets of the normalized data were created. Eighty percent of the dataset is made up of the first set, and twenty percent is made up of the second set. Using the following hypermeters, a convolutional neural network model was trained to classify the various kinds of backdoor attacks: Five layers: activiton_functions = softmax, relu; dense parameters: (16, 32, 64, 128, 256, 512, 128). (176, 64) is the input shape, while 9 is the output shape. The model was trained with batch_size=128 and optimizer='Adam' across a training period of ten. Figure 6 illustrates the model's training procedure as well as the accuracy the model attained on each training phase. The training accuracy and loss of the model are graphically represented in Figures 7 and 8. Confusion matrix and classification_report were used to assess the CNN model. This is depicted in Figures 9 and 10. Ten clients were created in order to replicate the federated learning model once the convolutional neural network model had been trained and evaluated. The customer samples were computed by dividing the total number of clients (10) by the duration of the trained data. The sparse categorical_cons entropy, metrics evaluation = ['accuracy'], and Adam optimizer were utilized to train the federated learning model. The federated learning model's training results across ten client numbers. Table 1 displays the first five training steps of the Federated Learning. Additionally, Figure 10 displays the accuracy for each customer number

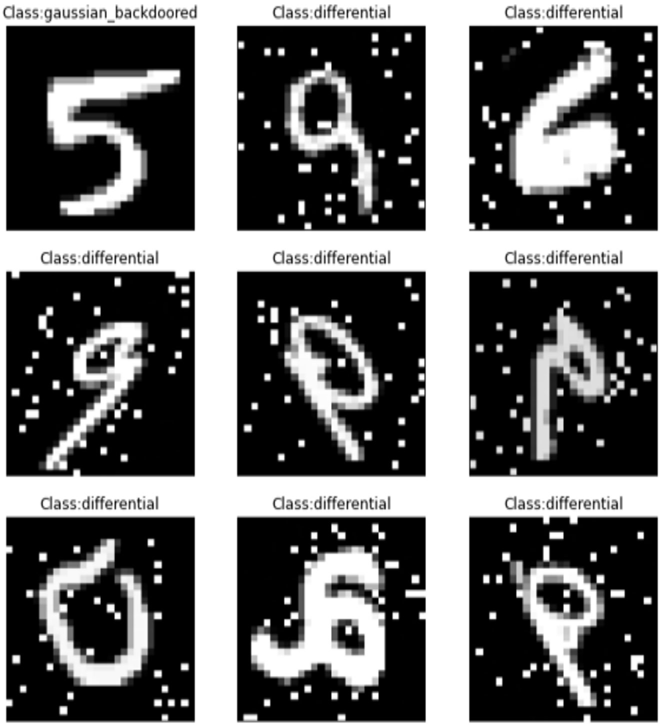


Figure 4: Visualization of Backdoor attack on MNIST dataset

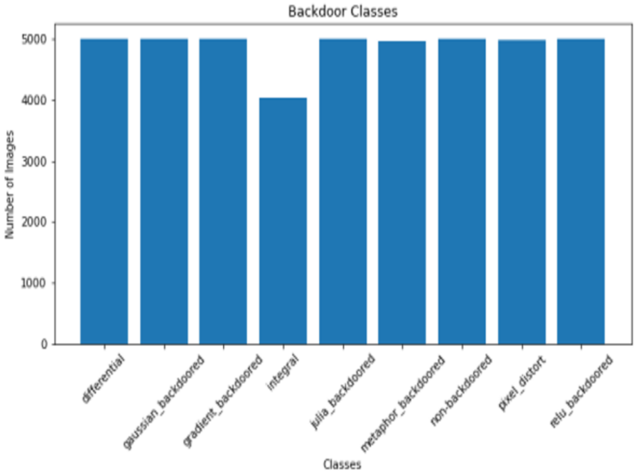
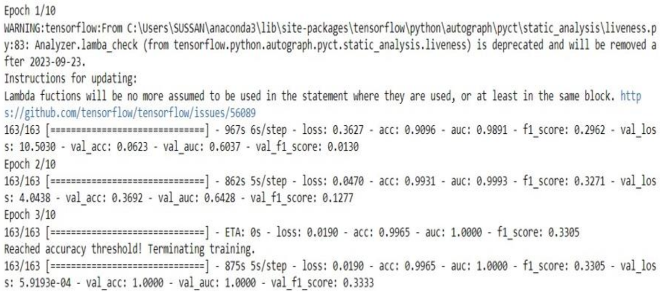


Figure 5. Countplot of the 9 classes of backdoor attacks

Table 1: Training process



```
Epoch 1/10
WARNING:tensorflow:From C:\Users\SUSSAN\anaconda3\lib\site-packages\tensorflow\python\autograph\pyct\static_analysis\liveness.p
y:83: Analyzer.lamba_check (from tensorflow.python.autograph.pyct.static_analysis.liveness) is deprecated and will be removed a
fter 2023-09-23.
Instructions for updating:
Lambda fuctions will be no more assumed to be used in the statement where they are used, or at least in the same block. http
s://github.com/tensorflow/tensorflow/issues/56089
163/163 [==============================] - 967s 6s/step - loss: 0.3627 - acc: 0.9096 - auc: 0.9891 - f1_score: 0.2962 - val_los
s: 10.5030 - val_acc: 0.0623 - val_auc: 0.6037 - val_f1_score: 0.0130
Epoch 2/10
163/163 [==============================] - 862s 5s/step - loss: 0.0470 - acc: 0.9931 - auc: 0.9993 - f1_score: 0.3271 - val_los
s: 4.0438 - val_acc: 0.3692 - val_auc: 0.6428 - val_f1_score: 0.1277
Epoch 3/10
163/163 [==============================] - ETA: 0s - loss: 0.0190 - acc: 0.9965 - auc: 1.0000 - f1_score: 0.3305
Reached accuracy threshold! Terminating training.
163/163 [==============================] - 875s 5s/step - loss: 0.0190 - acc: 0.9965 - auc: 1.0000 - f1_score: 0.3305 - val_los
s: 5.9193e-04 - val_acc: 1.0000 - val_auc: 1.0000 - val_f1_score: 0.3333
```

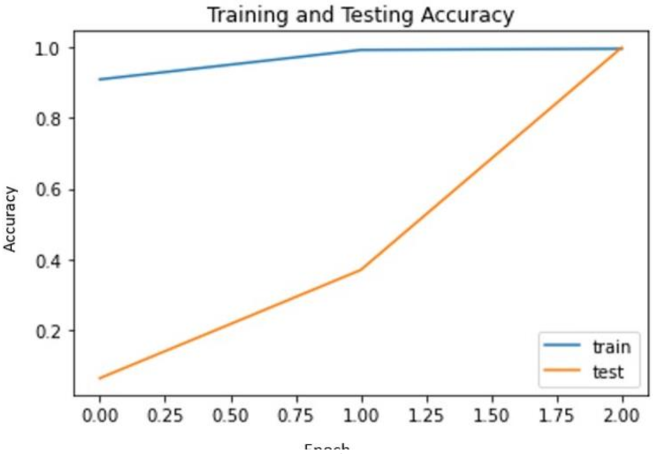


Figure 6: Graphical Analysis of Training Accuracy Vs Epoch

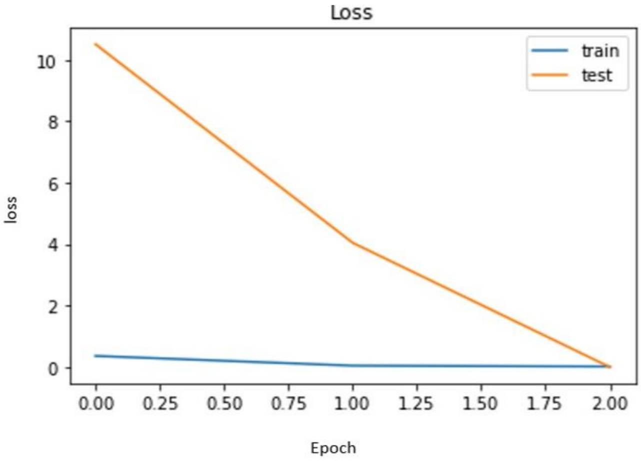
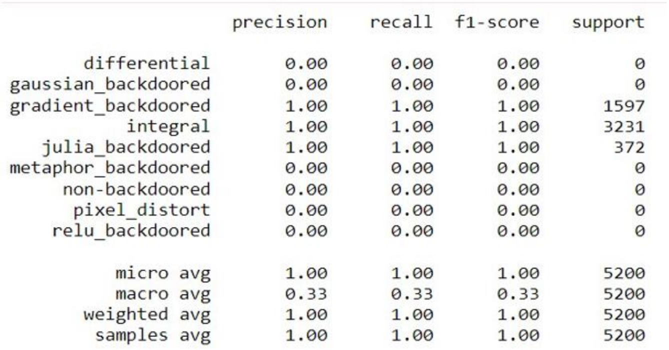


Figure 7: Graphical Analysis of loss Vs Epoch



```
                    precision    recall  f1-score   support

       differential      0.00      0.00      0.00         0
gaussian_backdoored      0.00      0.00      0.00         0
gradient_backdoored      1.00      1.00      1.00      1597
           integral      1.00      1.00      1.00      3231
    julia_backdoored      1.00      1.00      1.00       372
metaphor_backdoored      0.00      0.00      0.00         0
       non-backdoored      0.00      0.00      0.00         0
        pixel_distort      0.00      0.00      0.00         0
     relu_backdoored      0.00      0.00      0.00         0

           micro avg      1.00      1.00      1.00      5200
           macro avg      0.33      0.33      0.33      5200
        weighted avg      1.00      1.00      1.00      5200
         samples avg      1.00      1.00      1.00      5200
```

Figure 8: Classification Report of the model

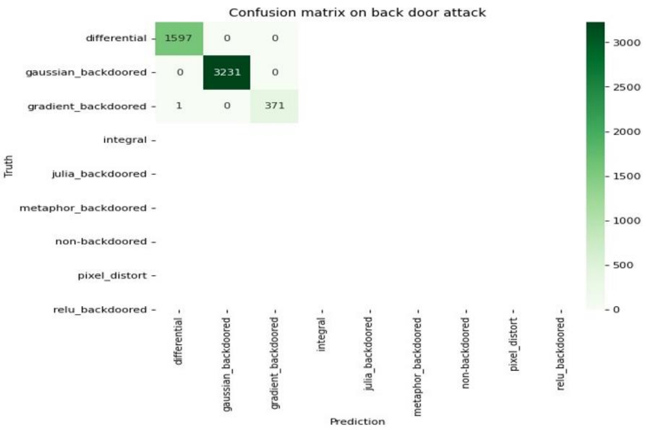


Figure 9: Confussion matrix of the model

Table 2: Global Test Performance of Backdoor Attacks In Federated Learning

Number of Clients	Accuracy (%)
Client 0	92
Client 1	96
Client 2	97
Client 3	98
Client 4	98
Client 5	98
Client 6	98
Client 7	99.
Client 8	99
Client 9	99

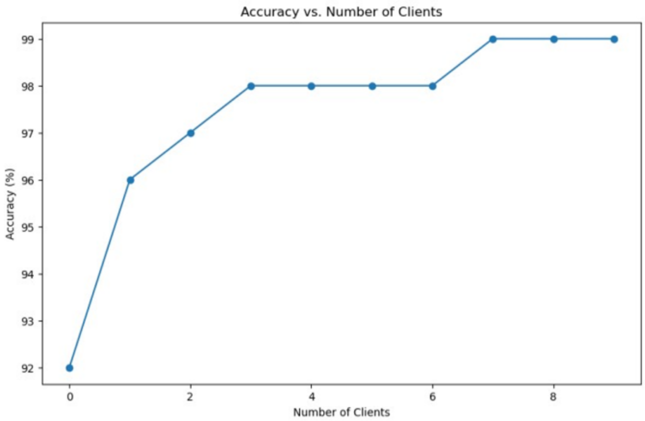
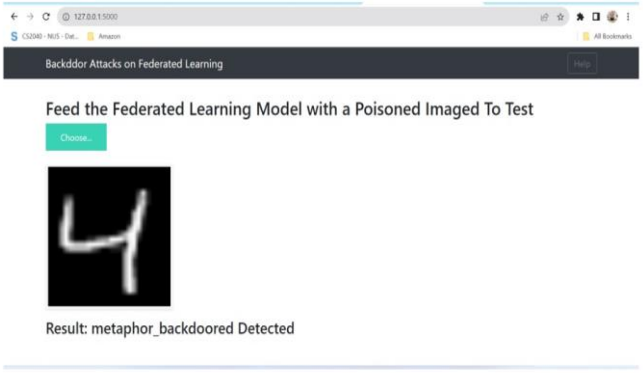


Figure 10: Global Performance of Backdoor Attacks n Federated Learning



Feed the Federated Learning Model with a Poisoned Imaged To Test

Result: metaphor_backdoored Detected

Figure 11: Result of the System when a poisoned image of 4 is posted

6. Conclusion and Future Scope

The system for accurately detecting backdoor attacks in federated learning was created in this dissertation. This was accomplished by employing the MNIST picture dataset, which includes nine classes of backdoor attacks in federated learning, to train a model. The model was retrained after malicious data was eliminated from the training dataset. Through careful examination and calculated action, this has been accomplished with success, yielding a system that strengthens the security of the entire model by detecting and eliminating the risks brought about by model inversion. by combining techniques for outlier detection with differential testing. The implementation of this two-pronged strategy has greatly enhanced the system's capacity to identify and fight possible backdoor threats. The deployment guarantees a more tenacious and trustworthy defense against malevolent manipulations. A model for federated learning backdoor attack detection and prevention was constructed using the Python programming language. The system's output was contrasted with that of other current systems. With an accuracy result of 99.97%, the findings here demonstrate that the suggested method performed better than the current system. It is advised to perform extensive testing and validation processes in order to find any indications of hacked models or backdoors. It can be more difficult for attackers to introduce subtle manipulations into the learning process in a more transparent and secure federated learning environment by implementing differential privacy safeguards and incorporating techniques like model explainability.

Conflict of Interest

The authors declare that they do not have any conflict of interest.

Funding Source
None

Authors' Contributions

Author-1 conceived the study. Author-2 developed the protocol and analyzed the data. Author-3 wrote the first draft. All authors reviewed and approved the final manuscript.

Acknowledgements
We acknowledge the support of our colleagues and reviewers.

References

[1] Bagdasaryan, E., Veit, A., Hua, Y.,Estrin, D., & Shmatikov, V. How to back door federated earning. In International Conference on Learning Representations (ICLR), **2020.**

[2] Bhagoji, A.N., Chakraborty, S., Mittal, P., Calo, S., & Feamster, N. Analyzing federated learning through an adversarial lens. arXiv preprinted Xiv:1811.12470. **2018.**

[3] Bonawitz, K., Eichner, H., Grieskamp, W., Huba, D., Ingerman, A., Ivanov, V.,...Raykova, M. Towards federated learning at scale: System design. arXiv preprint arXiv:1902.01046. **2019.**

[4] Gu, X., Chen, Y., Wang, Q., & Kong, W. Back door attacks and defenses in federated learning: State-of-the art, taxonomy, and future directions. IEEE Wireless Communications, Vol.**30**, Issue.**2**, pp.**114-121, 2022.**

[5] Hong,Y.,& Kim, M., Automated differential testing of web applications using mutation analysis. Journal of Systems and Software, 157, 110405, **2019.** https://doi.org/10.1016/j.jss.2019.

[6] Kairouz, P., Mc Mahan, H.B., Avent, B., Bellet, A., Bennis, M., Bhagoji, A.N.,... Zhang, H. Advances and open problems in federated learning, **2019.**

[7] Li, X., Xu, J., & Wang,Y. Back door attacks in federated learning: A survey. IEEE Access, 9, pp.**75232-75244, 2021.**

[8] Melis, L., Song, C., De Cristofaro, E., & Shmatikov, V., Exploiting unintended feature leakage in collaborative learning. In Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, 281-1295, **2019.**

[9] Tian, F. Q., Wang, S. L., & Liew, A. W. C., Towards practical watermark for deep neural networks in federated learning. arXiv preprint arXiv:2105.03167, **2021.**

[10] Wang, Z., Huang, Y., Song, M., Wu, L., Xue, F., & Ren, K., Poisoning-assisted property inference attack against federated learning. IEEE Transactions on Dependable and Secure Computing, **2022.**

[11] Yang, Z., Li, J., & Luo, X., Federated learning: Challenges and future directions. Journal of Parallel and Distributed Computing, 144, pp.**1-24, 2020.**

[12] Yang, Q., Liu, Y., Chen, T., & Tong, Y., Federated machine learning: Concept and applications. ACM Transactions on Intelligent Systems and Technology, Vol.**10**, Issue.**2**, 119, **2019.**

[13] Zhou, W., Wang, H., Li, H., & Zhang, X., Backdoor attacks in federated learning: A survey. IEEE Communications Magazine, Vol.**59**, Issue.**1**, pp.**80-86, 2020.**

AUTHORS PROFILE

C.B. Biragbara earned her B.Sc in Computer Science from Rivers State University of Science and Technology.

Dr. O.E. Taylor obtained his B.Sc, M.Sc and Ph.D degrees all in Computer Science from the Rivers State University of Science and Technology, University of Ibadan and University of Port Harcourt, Nigeria respectively. He is currently an Associate Professor in the Department of Computer science, Rivers State University, Port Harcourt, Nigeria. He is a chartered member of the Computer Professionals (Registration Council) of Nigeria and Nigeria Computer Society. His research focuses on intelligent systems, smart systems, context-aware systems, machines learning algorithms and pervasive systems. He has over sixty academic publications and more fifteen years of teaching and research experience.