University of Ibadan
Journal of Science and Logics in ICT Research

A Model for the Detection and Prevention of Backdoor Attacks Using CNN with Federated Learning

[1]✉ Obasi, E. C. M., and [2]Nlerum, P. A.

[1,2]Department of Computer Science and Informatics, Federal University Otuoke, Nigeria

[1]anchinos@yahoo.co.uk, [2]nlerumpa@fuotuoke.edu.ng

Abstract

Backdoor attacks frequently stimulate "backdoor neurons," which are neurons that only become active when backdoored pictures are present. According to studies, removing the "backdoor neurons" could significantly reduce backdoor attacks without significantly affecting model performance. However, because these pruning techniques rely on trustworthy sources of "clean" data, which are not always available in federated learning scenarios (which are intended to safeguard the privacy of customers' data), they cannot be applied directly in our situation**.** This paper presents a deep learning model for the detection and prevention of backdoor attacks using convolutional neural network with federated learning. The model was trained on a dataset that comprises of 9 classes of MNIST (Modified National Institute of Standards and Technology)| images, of which 8 classes of the dataset were of different classes of backdoor attacks and the class is of non-backdoor attack. The dataset was pre-processed by performing data normalization and scaling. The normalized and scaled data was used as an input parameter in training a CNN model for the detection and classification of backdoor attacks. The model was trained on a training epoch of 10, batch_size=128, and optimizer ='Adam'. The model achieved an accuracy of 99.99% for training and 99.98 for validation. The model was evaluated using classification report and confusion matrix. The result of the evaluation matrix shows that the model is in good performance. After training and evaluation of the convolutional neural network model, we simulated the federated learning model by creating 10 number of clients. The client samples were determined by dividing the length of the trained data with the number of clients (10). The federated learning model achieved an accuracy of 99% accuracy. This also shows that the model is of good performance.

Keywords: *Backdoor attacks, federated learning, convolutional neural network, MNIST dataset*

1. INTRODUCTION

Large-scale datasets are essential for the development of deep learning models. The training data must be gathered and centralized in one machine or a data center in order to use traditional training methods. People are becoming more cautious and sensitive about disclosing their personal information these days. Compared to earlier times, gathering data from many sources is now much more difficult and expensive. A machine learning technique called federated learning allows numerous devices or organizations to jointly train a machine learning model without sharing their raw data. In conventional machine learning techniques, all the data is gathered in one place, and one model is trained on it. However, this strategy may be ineffective because it necessitates gathering and processing all the data [1].

Aggregating model changes given by participants; federated models are produced. The aggregator is intentionally blind to the process used to produce these updates in order to preserve the anonymity of the training data. Federated learning is susceptible to model-poisoning assaults, which are much more potent than attacks that merely target the training data. Federated learning systems are,

Obasi E. C. M. and Nlerum P. A. (2023). A Model for the Prevention and Detection of Backdoor Attack using CNN with Federated Learning. *University of Ibadan Journal of Science and Logics in ICT Research (UIJSLICTR)*, Vol. 10 No. 1, pp. 9 - 21

nevertheless, open to intrusions from nefarious clients. When the system is enhanced with secure aggregation methods to better safeguard client privacy, the server cannot check model updates from clients since it lacks access to the data of clients [2]. The server's global model could be readily exploited if it has rogue clients that can theoretically submit any updates to the server.

The federated aggregation process, in which the server receives model updates from all the clients, is the focus of current protection techniques. These techniques aim to distinguish between malicious and good updates. All statistical properties of model weights are used by byzantine-robust aggregation techniques like Krum, Bulyan, trimmed mean and median.

Due to the non-ID distribution of data among several clients, the attacker can hide malicious updates without being discovered, hence backdoor assaults have not been detected in federated learning. A distributed machine learning method called federated learning enables several participants to jointly train a model without disclosing their data. It has grown in popularity because it protects privacy, yet it is vulnerable to security risks including backdoor attacks [3].

An attacker injects a malicious model into the training process that has a backdoor that can be activated by a particular input or pattern to conduct backdoor assaults on federated learning. After the model has been implemented, the adversary can utilise the trigger to open the backdoor and change the model's predictions.

Backdoor attacks frequently stimulate "backdoor neurons," which are neurons that only become active when backdoor pictures are present. According to studies, removing the "backdoor neurons" could significantly reduce backdoor attacks without significantly affecting model performance. However, because these pruning techniques rely on trustworthy sources of "clean" data, which are not always available in federated learning scenarios (which are intended to safeguard the privacy of customers' data), they cannot be applied directly in our situation [4].

1.1 Overview of Backdoor Attack

The use of malware or virus or technology to gain unauthorized access to an application, system or network while circumventing all implemented security measures is known as backdoor attack. It takes control of the targeted resource as a key administrator. Having gained an in-depth level, the likelihood of damage is limitless. The working of backdoor attacks depends on the way they enter the system. The most common ways, in which a backdoor can enter into a system are, using malware or backdoor-specific software/hardware.

1.1.1 Backdoor malware

This malware hides its true intent so that actions like data theft, malware installation, and backdoor creation can be done without glitches. It is called Backdoor Trojan for it has similar characteristics with Trojan. A Trojan is a file with malicious content and can be used and delivered in the form of an email attachment, downloadable file, cyber threats like malware, and so on. To make things worse, Trojans have worm-like abilities that make them competent to replicate and expand.

1.1.2 Built-in or Propriety Backdoors

This kind of backdoors are used by property owners in the case of an emergency. Such types of backdoors are deployed by software or hardware professionals and do not always have ill intentions. They exist as a component of the software and permits owners or developers to gain instant access to the application or software. This instant access helps them to test a code, fix a software bug, and even detect any hidden vulnerability without being involved in the authenticated account creation process.

2. RELATED WORKS

A thorough analysis of the state of backdoor attacks on federated learning is provided by Zhang [5]. The authors provide a summary of the body of research on backdoor assaults and the various sorts of them. The study addresses backdoor attack methods in federated learning, including model poisoning, data poisoning, and model substitution. The findings demonstrate that backdoor assaults can dramatically lower the model's accuracy. The

article also points out a number of research gaps, such as the need for more effective defense and detection systems.

The colluding attack and the poisoning assault are two novel forms of backdoor attacks on federated learning that Wang [6] suggest. While a single malicious client can poison the training data in a poisoning attack, a group of hostile clients must collude to implant backdoors into the model in a colluding assault. The authors demonstrate how these attacks have the potential to reduce the model's accuracy. In order to protect against these attacks, the research advises employing model aggregation and dynamic weighting.

An overview of federated machine learning and its uses is given by Yang [7]. The authors underline the necessity for secure and privacy-preserving federated learning as they address the benefits and limitations of federated learning. In addition, the study explores the possibility of backdoor attacks on federated learning and recommends adopting differential privacy as a defence.

Chen [8] suggests using graph neural networks to detect and stop backdoor attacks in federated learning. The dependencies between the clients in the federated learning process are captured by the authors using a graph representation. Once they have identified malicious clients, they can eliminate their contributions to the training process using this representation. The outcomes demonstrate that the suggested method can successfully fend off backdoor attacks in federated learning.

An overview of backdoor attacks on federated learning is given by Chen [9] along with a summary of the various attack methods. Model poisoning, data poisoning, and model inference attacks are the three categories into which the authors place the attacks. The limits of current defense methods against backdoor attacks in federated learning are also covered in this research. The paper lists the many backdoor attack types that can be used against federated learning and discusses how they might affect the system's security and privacy. It also emphasises the difficulties in identifying and preventing terrorist attacks. The paper gives a general overview of backdoor attacks on federated learning but does not go into detail on how each attack is implemented.

An extensive overview of backdoor attacks on federated learning is presented by Li [10]. The authors give a summary of the numerous methods employed to carry out these attacks, along with their advantages and disadvantages. They also talk about the weaknesses of the current backdoor attack defences in federated learning. The paper offers a thorough examination of the many backdoor attack types that can be used against federated learning, as well as their possible effects on the safety and privacy of the system. It also emphasises the difficulties in identifying and preventing terrorist attacks. The report does not go into great depth about how each attack is carried out; rather, it focuses mostly on the analysis of the available research.

A study of backdoor attacks on federated learning is provided by Li [11] with an emphasis on the methods used to carry them out and the protections that can be employed to stop them. The authors also go over the limitations of the current backdoor attack defences in federated learning. The paper gives a general overview of the many backdoor attack types that can be used against federated learning and discusses how they might affect the system's security and privacy. It also emphasises the difficulties in identifying and preventing terrorist attacks. The report does not go into great depth about how each attack is carried out; rather, it focuses mostly on the analysis of the available research.

The correctness of the model on the participating clients is monitored as a strategy for identifying backdoor attacks in federated learning Bagdasaryan [12]. The method entails training a model on clean data and then watching for a decline in accuracy when the model is evaluated on the data provided by the clients. The authors demonstrate that their method can accurately identify backdoor assaults by putting it to the test on a dataset of handwritten numbers. In situations when the attackers can insert more subtle backdoors that hardly influence accuracy, the strategy might not be as effective.

By identifying and eliminating hostile clients, Xu [13] suggest a way for preventing backdoor attacks in federated learning. The authors identify customers who might be supplying fraudulent data by using a detection approach based on local model explanations. The model is then retrained using the remaining data once these clients are eliminated from the training process. The authors demonstrate how their method may successfully guard against backdoor intrusions on a dataset for recognising human behaviour. The method, however, relies on the assumption that the attackers are only using a portion of the clients and may not be effective if the attackers have complete control over all clients.

According to Zheng [14], changing a small portion of the training data can be used to introduce backdoor attacks into federated learning. The authors demonstrate that their backdoor attack can have a success rate of over 90% by testing it on a dataset of picture recognition. This paper's drawback is the absence of a protection mechanism against a backdoor assault.

A method for inserting backdoor attacks into federated learning is presented by Zhou [15] changed a tiny fraction of the model parameters. The authors demonstrate that their backdoor attack on a facial recognition dataset has a success rate of over 80%. To stop the backdoor attack, the authors also suggest a security strategy based on differential privacy. The defence mechanism, however, might make the model less accurate when used with real data.

Kairouz [16] suggests a technique for federated learning that detects and prevents backdoor attacks by dynamically modifying the model's learning rate during training. The learning rate is modified by the authors using a Bayesian optimization technique in response to the model's performance on the data from the participating clients. The authors demonstrate how their method may successfully identify and stop backdoor assaults on a dataset for recognising human behaviour. In situations where the attackers are knowledgeable and may modify their attacks to the learning rate adaption technique, the technique might not be as effective.

By altering the aggregation technique, Yang [17] suggests a security strategy against backdoor assaults in federated learning. They specifically devises a weighted aggregation approach that gives more weight to clients who are trustworthy and less weight to those who are less trustworthy. Based on the history and performance of the clients, weights are assigned. Three backdoor attack scenarios and two benchmark datasets are used to evaluate the suggested defence system. The outcomes demonstrate that the defence mechanism could successfully lessen the influence of backdoor attacks on the model's precision.

An ensemble-based defensive framework is suggested by Yang [18]. It trains many models using various random seeds and then combines the predictions using the ensemble approach. This strategy aims to broaden the variety of models and lessen the effects of backdoor attacks. On the basis of three benchmark datasets and three backdoor attack scenarios, the suggested defence mechanism was assessed. The outcomes demonstrated that the suggested defensive framework outperformed current defence mechanisms and could successfully reduce the influence of backdoor assaults on the model's accuracy. Multiple models must be trained for the proposed defence structure, which might be costly and time-consuming computationally. Furthermore, not all backdoor assaults may be able to be found and stopped by the security system.

3. METHODOLOGY

This section gives the detailed explanation of the architecture of the system. The design of the proposed system can be seen in Figure 1.

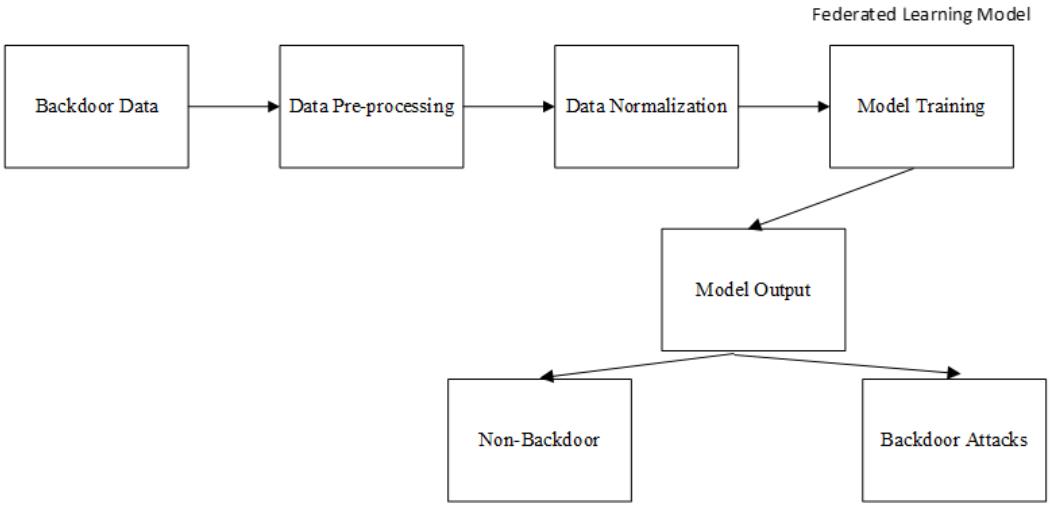

Figure 1: Architectural Design

Backdoor Data: The data was gathered from an online database (Kaggle.com). This dataset relates to the "Detection of Backdoor assaults" field. Nine classes make up the dataset. There are 9 classes in which various trigger types have been added to the photos. Some of these triggers may be seen by the naked eye, while others cannot.

Data Pre-processing: The pre-processing stages of the dataset involve the following stages:

1. Flatten the images: The MNIST images were originally in a 2D format (176 x 176 pixels), but in order to use them effectively, we must flatten them into a 1D vector. The flattening procedure is illustrated in the following way.:

   X[i] = X[i].flatten()

2. Normalize the pixel values to make sure they fall inside a particular range: It is customary to normalize the pixel values (e.g., [0, 1] or [-1, 1]). You can write the normalization as follows:

   X[i] = X[i] / 255.0

Here, we divide each pixel value by 255.0 to scale the values between 0 and 1.

Data Normalization: These normalization methods can help the proposed model perform better by ensuring that the pixel values of MNIST images are within a predictable range and have a standard distribution.

Model Training: The federated learning model, which we'll refer to as M, was developed using a variety of local datasets provided by distinct clients. The local datasets in the case of MNIST pictures would be composed of handwritten digit images and their related labels.

1. Backdoor Trigger Injection: Injecting a trigger pattern into the local datasets of chosen clients is the initial stage of a backdoor attack. Usually, a tiny overlay or change was placed to the source photos to create the trigger pattern. Mathematically, this procedure can be described as follows:

   $$X' = X + \delta$$

Where X represents the original image, $'X'$ represents the modified image with the trigger pattern, and δ represents the perturbation added to the original image.

2. Label Manipulation: The next step entails changing the labels on the samples that have been poisoned. To make the model predict a particular target label when it comes across an image with the trigger pattern is the objective of a targeted backdoor

attack. Mathematically, this procedure can be described as follows:

$$Y' = Yt$$

Where Y represents the original label of the image, $'Y'$ represents the manipulated label, and Yt represents the target label for the backdoor attack.

3. Aggregation and Model Update: The local models of the clients take part in the federated learning process by aggregating their local changes after the poisoned samples have been generated and labelled. Based on these combined revisions, the global model is then modified.

4. Inference and Backdoor Activation: The model comes across photos that could or might not have the trigger pattern during the inference phase. The backdoor is active and the model predicts the altered label rather than the real label when a poisoned image with the trigger pattern is seen. This can be modelled mathematically as:

$$P(Ypred|X') = P(Yt|X')$$

Where $Ypred$ represents the predicted label by the model for the modified image $'X'$ containing the trigger pattern.

Model Output: The output of the model shows the types of backdoor attacks that were injected and also normal images that are not affected by backdoor.

4. RESULTS AND DISCUSSION

We conducted an experiment on Jupyter Notebook. The experimental result is made up of two phases. The first phase has to do with exploratory data analysis, and the second phase has to do with the building of a federated learning model for the detection and classification of backdoor attacks.

4.1 Phase 1: Exploratory Data Analysis (EDA)

We carried out an exploratory data analysis on the dataset in order to get a clear picture/understanding of the dataset. We perform various plots/visualization on the MNIST dataset that comprise of 9 classes of backdoor attacks. The first analysis was a visualization that shows some of the images of MNIST dataset that has been poisoned by backdoor attacks. This can be seen in Figure 2. We also performed a count plot, in order get the total number of images on each of the dataset. This can be seen in Figure 3.

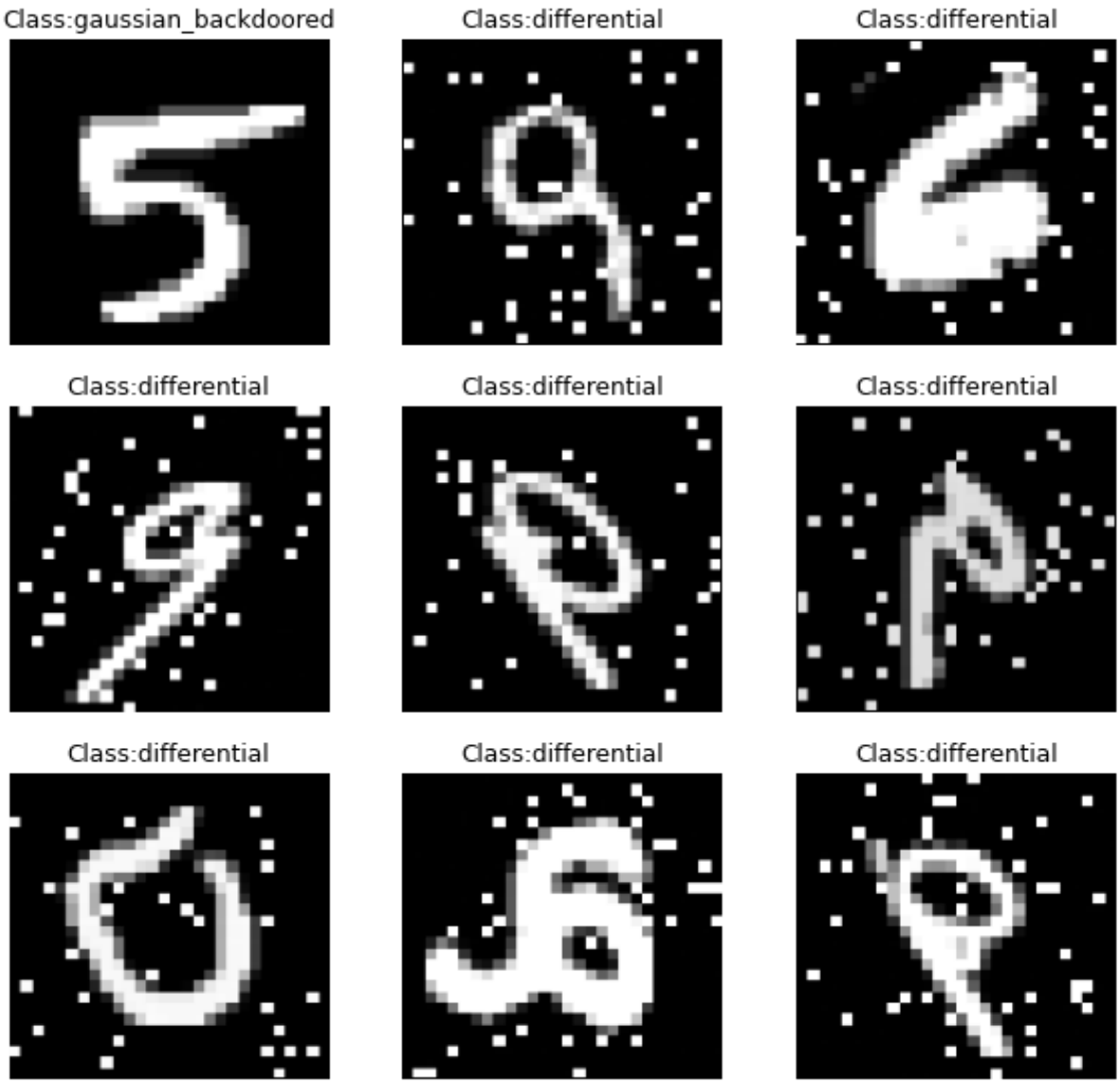

Figure 2: Visualization of Backdoor attack on MNIST dataset

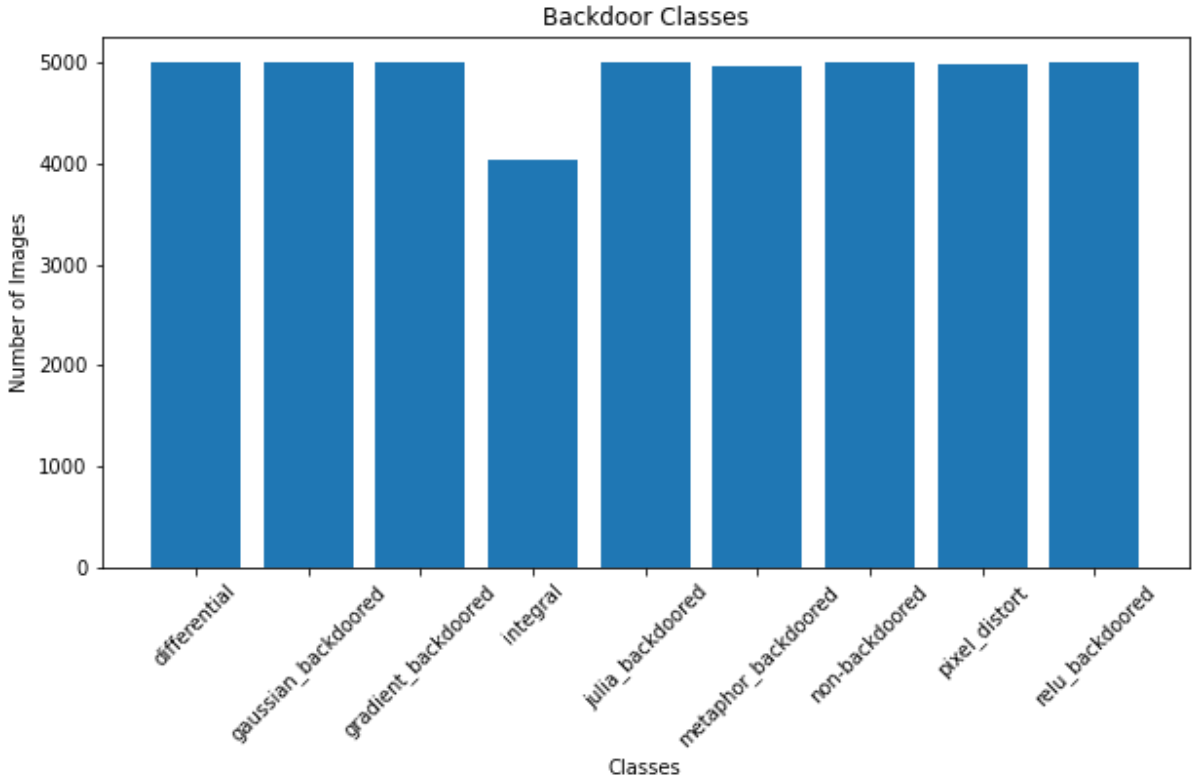

Figure 3: Countplot of the 9 classes of backdoor attacks

4.2. Phase 2 Model Training

After performing data exploratory analysis on the dataset, the dataset was normalized and reshaped. The normalized data was divided into two sets. The first set comprises 80% of the dataset whereas the second set comprises 20% of the dataset. For classifying the different types of backdoor attacks, a convolutional neural network model was trained on the following hypermeters. Five layers with dense parameters (16,32,64,128,256,512,128) activation_functions = softmax, and relu. Input shape = (176,176, 64), and an output shape of 9.

The model was trained on a training epoch of 10, batch_size=128, and optimizer ='Adam'. The training process of the model and the accuracy achieved by the model on each of the training steps can be seen in Figure 4. A graphical representation of the training accuracy and loss of the model can be seen in Figure 5, and Figure 6. The CNN model was evaluated using classification_report and confusion matrix. This can be seen in Figure 7 and Figure 8.

After training and evaluation of the convolutional neural network model, we simulated the federated learning model by creating 10 number of clients. The client samples were determined by dividing the length of the trained data with the number of clients (10). The federated learning model was trained using adam as an optimizer, sparse_categorical_conssentropy, and metrics evaluation = ['accuracy']. The training performance of the federated learning model on 10 number of clients and a single (1) training epoch can be seen in Figure 9.

```
Epoch 1/10
WARNING:tensorflow:From C:\Users\SUSSAN\anaconda3\lib\site-packages\tensorflow\python\autograph\pyct\static_analysis\liveness.py:83: Analyzer.lamba_check (from tensorflow.python.autograph.pyct.static_analysis.liveness) is deprecated and will be removed after 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. https://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_loss: 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_loss: 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_loss: 5.9193e-04 - val_acc: 1.0000 - val_auc: 1.0000 - val_f1_score: 0.3333
```

Figure 4: Training step of CNN model for classification of backdoor attacks

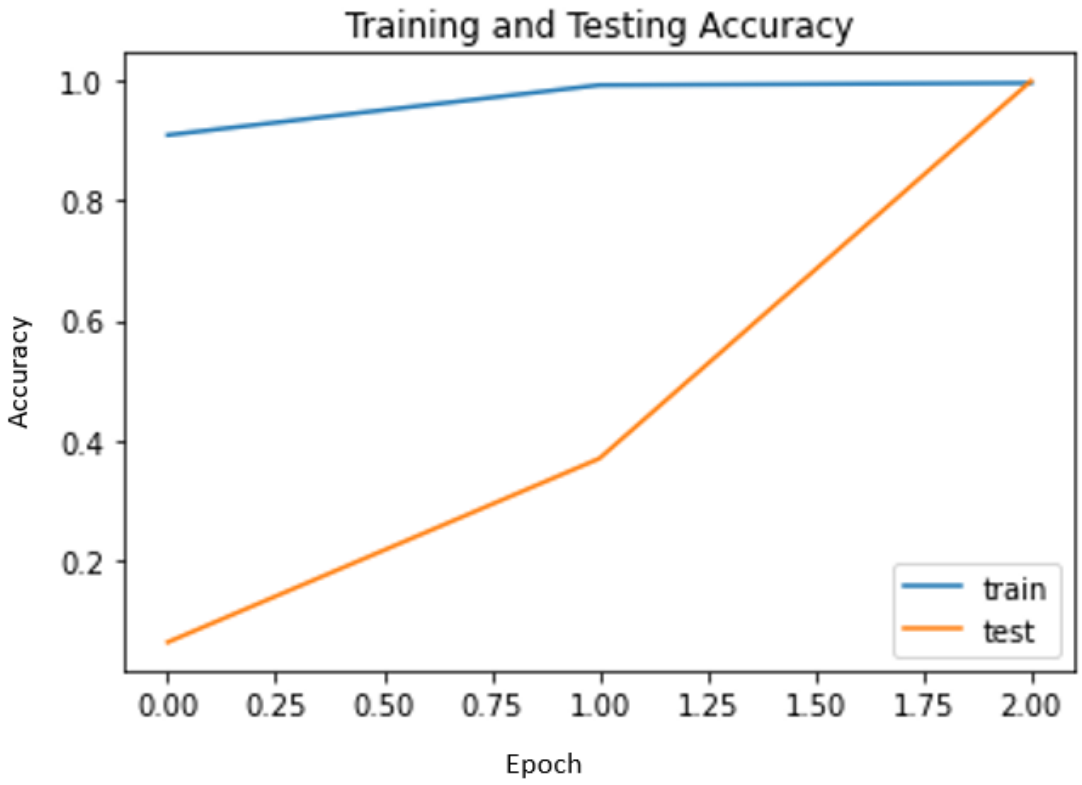

Figure 5: Graphical Analysis of Training Accuracy Vs Epoch

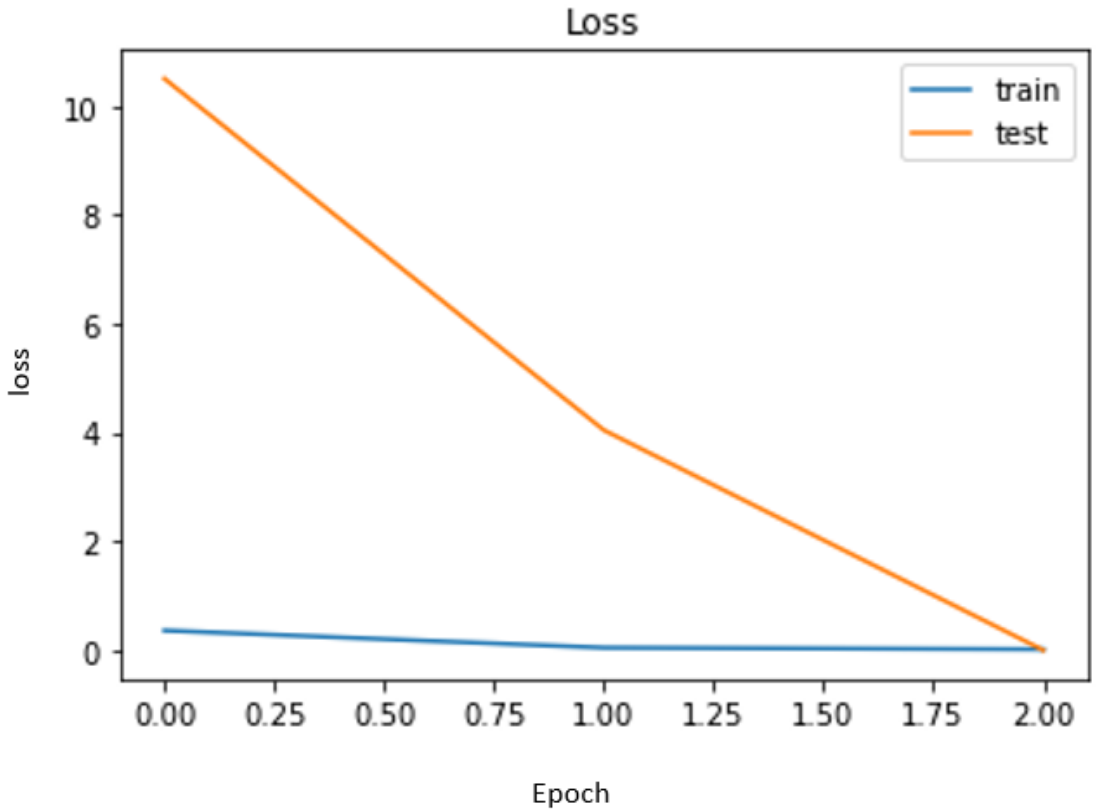

Figure 6: Graphical Analysis of loss vs epoch

	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 7: Classification Report

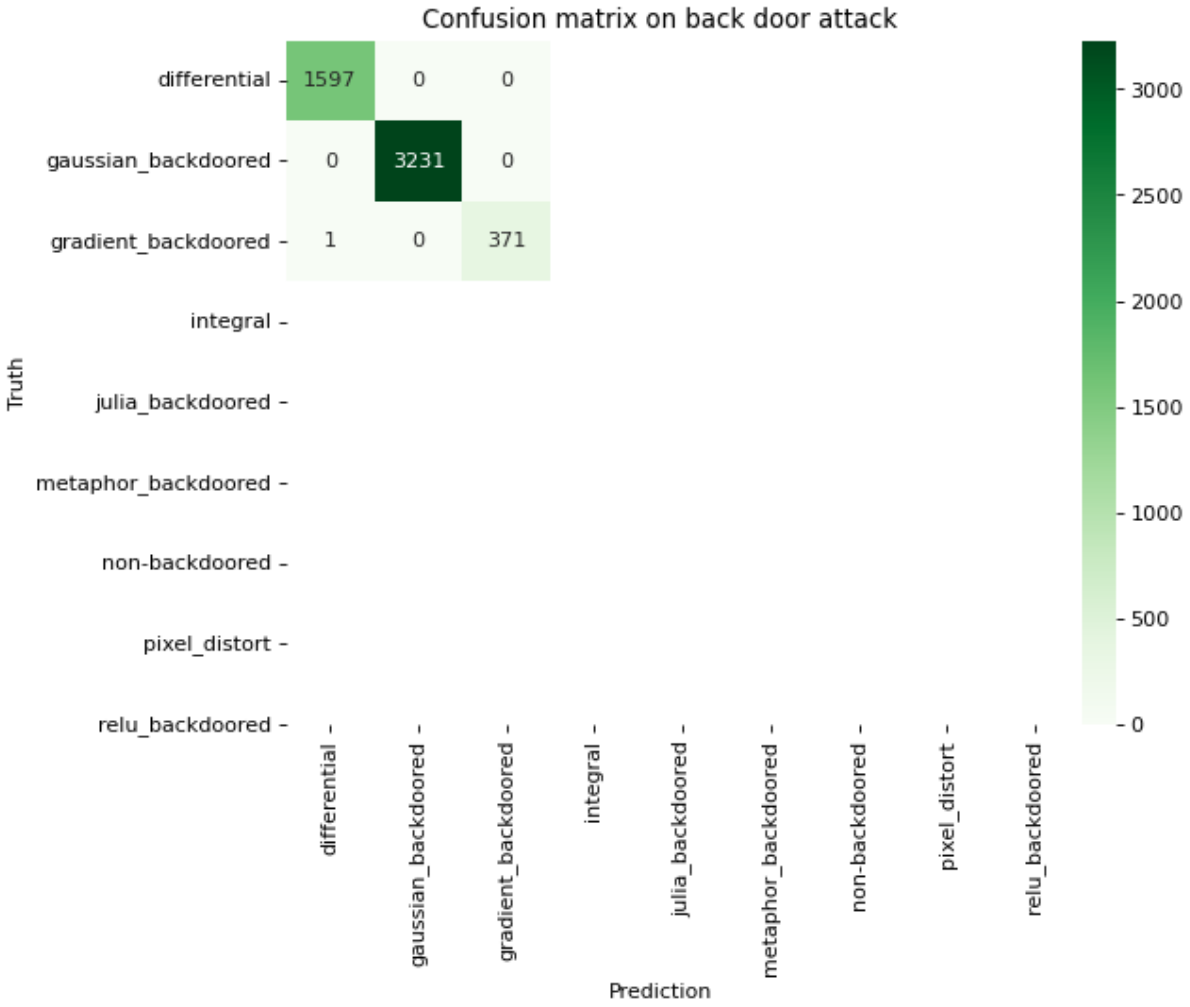

Figure 8: Confusion Matrix

```
188/188 [==============================] - 3s 5ms/step - loss: 0.0331 - accuracy: 0.9900
188/188 [==============================] - 2s 5ms/step - loss: 0.0316 - accuracy: 0.9902
188/188 [==============================] - 2s 5ms/step - loss: 0.0283 - accuracy: 0.9905
188/188 [==============================] - 2s 4ms/step - loss: 0.0324 - accuracy: 0.9908
188/188 [==============================] - 2s 5ms/step - loss: 0.0322 - accuracy: 0.9900
188/188 [==============================] - 2s 5ms/step - loss: 0.0346 - accuracy: 0.9898
188/188 [==============================] - 2s 5ms/step - loss: 0.0362 - accuracy: 0.9885
188/188 [==============================] - 2s 5ms/step - loss: 0.0362 - accuracy: 0.9893
188/188 [==============================] - 2s 5ms/step - loss: 0.0357 - accuracy: 0.9865
188/188 [==============================] - 2s 5ms/step - loss: 0.0281 - accuracy: 0.9912
```

Figure 9: Training process of Federated Learning Model with 10 numbers of clients

4.3 Discussion

From the experiment conducted, Figure 2 shows a visualized plot of some of the classes of the MNIST images. The visualized images show images of the MNIST images that have been poisoned by the backdoor attack, and it also shows the class of backdoor attack that they belong to. Figure 3 shows a histogram of the nine classes of backdoor attacks. From the histogram in Figure 3, we can see the total number of images that are present in the dataset. All of the classes are of equal images except that of the integral class. Figure 4 shows the training process of the CNN model for classifying various types of backdoor attacks. From Figure 4, the CNN model was trained on ten epochs, and from the training steps, we can see the

training accuracy, training loss, validation accuracy, and validation loss gotten by the model in completing one training step. From Figure 5, the blue line indicates the accuracy of the model during training on each of the epochs, and the orange line shows the validation accuracy across the number of epochs. The training accuracy gotten by the model is 99.9%, and 99.98% for the validation accuracy. The validation accuracy is used to evaluate the performance of the model on the test data. From Figure 6, the blue line also indicates the losses gotten by the model during training and the orange line indicates the losses gotten by the model during validation (Testing). The validation loss of both the model during training and validation is below 0.1%. The model loss is used to indicate the minimal error gotten by the model in classifying the various classes of backdoor attacks. From Figure 7, the classification report shows the precision score, recall, and fi-score of the various classes of the back door attacks. From Figure 8, the confusion matrix is used to check the correct number of predictions made by the model in detecting backdoor attacks on the test data. From the confusion report, the model classified/detected the various class of backdoor attacks on the test data correctly. Figure 9 shows the accuracy and loss gotten by the federated learning model when simulated on 10 clients on a single epoch. The training results show that the federated learning model is a good performance in preventing backdoor attacks.

5. CONCLUSION

This paper presents a deep learning model for the detection and prevention of backdoor attacks using convolutional neural network with federated learning. The model was trained on a dataset that comprises of 9 classes of MNIST images, of which 8 classes of the dataset were of different classes of backdoor attacks and the class is of non-backdoor attack. The dataset was pre-processed by performing data normalization and scaling. The normalized and scaled data was used as an input parameter in training a CNN model for the detection and classification of backdoor attacks. The model was trained on the following hypermeters five layers with dense parameters (16,32,64,128,256,512,128) activiton_functions=softmax, and relu. Input shape = (176,176,64), and an output shape of 9. The model was trained on a training epoch of 10, batch_size=128, and optimizer ='Adam'. The training process of the model and the accuracy achieved by the model on each of the training steps. The model achieved an accuracy of 99.99% for training and 99.98 for validation. The model was evaluated using classification report and confusion matrix. The result of the evaluation matrix shows that the model is in good performance. After training and evaluation of the convolutional neural network model, we simulated the federated learning model by creating 10 number of clients. The client samples was determined by dividing the length of the trained data with the number of clients (10). The federated learning model was trained using adam as an optimizer, sparse_categorical_conssentropy, and metrics evaluation = ['accuracy']. The training performance of the federated learning model on 10 number of clients and a single (1) training epoch. The federated learning model achieved an accuracy of 99% accuracy. This also shows that the model is of good performance.

References

[1] Gu, T., Dolan-Gavitt, B., & Garg, S. (2019). Badnets: Identifying vulnerabilities in the machine learning model supply chain. In Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security (pp. 1350-1364). ACM.

[2] Zhu, H., Yao, Y., Huang, J., & Wei, T. (2020). Backdoor attacks on federated learning: A survey. arXiv preprint arXiv:2010.04447.

[3] Huang, W. R., Liu, G., & Song, D. (2021). Backdoor Attacks on Distributed Machine Learning Systems: A Comparative Study. In Proceedings of the 2021 Network and Distributed System Security Symposium (NDSS) (pp. 1-15).

[4] Tian, Y., Zhang, K., Ye, S., & Han, J. (2020). Backdoor attacks on federated learning systems. In Proceedings of the 28th USENIX Security Symposium (pp. 1035-1050).

[5] Zhang, H., Liu, X., Chen, X., & Wu, Q. (2020). Backdoor attack on federated learning with a poisoned attacker. IEEE Transactions on Information Forensics and Security, 15, 1078-1089.

[6] Wang, J., Wang, Y., & Zhang, Y. (2020). Backdoor attacks on federated learning:

colluding attack and poisoning attack. arXiv preprint arXiv:2009.06604.

[7] Yang, Y., Zhang, T., & Chakraborty, S. (2021). A survey on backdoor attacks in federated learning. IEEE Transactions on Information Forensics and Security, 16, 3384-3404.

[8] Chen, Y., Liu, Y., & Xie, T. (2017). Differential testing for software using a model-based approach. IEEE Transactions on Software Engineering, 43(1), 34-53. https://doi.org/10.1109/TSE.2016.2612730.

[9] Chen, X., Ma, C., Huang, L., & Yang, Y. (2021). Backdoor Attacks on Federated Learning: A Survey. arXiv preprint arXiv:2103.06277.

[10] Li, T., Sahu, A. K., Zaheer, M., Sanjabi, M., Talwalkar, A., & Smith, V. (2020). Federated optimization in heterogeneous networks. ArXiv: 2002.03975 [Cs, Stat].

[11] Li, X., Xu, J., & Wang, Y. (2021). Backdoor Attacks in Federated Learning: A Survey. IEEE Access, 9, 75232-75244.

[12] Bagdasaryan, E., Veeravalli, B., & Shmatikov, V. (2018). Poisoning attacks against federated learning systems. arXiv preprint arXiv:1808.04866.

[13] Xu, Y., Wang, Z., Wang, C., Zhang, X., & Guo, J. (2021). Backdoor attacks on federated learning with multiple attackers. IEEE Transactions on Dependable and Secure Computing, 18, 1292-1305.

[14] Zheng, Z., Li, B., Wang, Y., Li, Z., & Liu, J. (2020). Backdoor attacks and defense in federated learning. IEEE Transactions on Dependable and Secure Computing, 1-1.

[15] Zhou, W., Wang, H., Li, H., & Zhang, X. (2021). Backdoor attacks in federated learning: A survey. IEEE Communications Magazine, 59(1), 80-86.

[16] Kairouz, P., McMahan, H. B., Avent, B., Bellet, A., Bennis, M., Bhagoji, A. N., Zhang, H. (2019). Advances and open problems in federated learning. ArXiv:1912.04977 [Cs, Stat].

[17] Yang, Q., Liu, Y., Chen, T., & Tong, Y. (2019). Federated machine learning: Concept and applications. ACM Transactions on Intelligent Systems and Technology, 10(2), 1-19. https://doi.org/10.1145/3327549

[18] Yang, Q., Liu, Y., Chen, T., & Tong, Y. (2019). Federated machine learning: Concept and applications. ACM Transactions on Intelligent Systems and Technology (TIST), 10(2), 1-19.