SecureTransfer: A Transfer Learning Based Poison Attack Detection in ML Systems

Archa A T, K.Kartheeban
Computer Science and Engineering
Kalasalingam Academy of Research and Education
Krishnankoil, TamilNadu, India

***Abstract*—Critical systems are increasingly being integrated with machine learning (ML) models, which exposes them to a range of adversarial attacks.The vulnerability of machine learning systems to hostile attacks has drawn a lot of attention in recent years. When harmful input is added to the training set, it can lead to poison attacks, which can seriously impair model performance and threaten system security. Poison attacks pose a serious risk since they involve the injection of malicious data into the training set by adversaries, which influences the model's performance during inference. It's necessary to identify these poison attacks in order to preserve the reliability and security of machine learning systems. A novel method based on transfer learning is proposed to identify poisoning attacks in machine learning systems.The methodology for generating poison data is initially created and later implemented using transfer learning techniques. Here, the poisonous data is detected using the pre-trained VGG16 model. This method can also be used in distributed Machine learning systems with scattered data and computation across several nodes. Benchmark datasets are used to evaluate this strategy in order to prove the effectiveness of proposed method.Some real-time applications,advantages,limitations and future work are also discussed here.**

Keywords*—Poison attacks; machine learning security; transfer learning; generative adversarial networks; convolutional neural networks; VGG16

I. Introduction

Nowadays, machine learning techniques have been used across various domains such as healthcare, finance, and autonomous systems. However, the applications of machine learning models in real-world systems has also raised concerns about their vulnerability to adversarial attacks [5]. Among these attacks, poison attacks stand out as a particularly critical threat, where adversaries inject subtle but malicious perturbations into the training data to undermine the integrity and performance of the models. Detecting and mitigating poison attacks [20][23] in machine learning systems is a critical challenge that requires innovative solutions to safeguard the reliability and trustworthiness of deployed models. Traditional defense mechanisms, such as input sanitization [30] and robust training [10] have shown limited effectiveness against sophisticated poison attacks that exploit vulnerabilities in the training process.

Security is paramount in distributed settings [14] due to the decentralized nature of data and computation. In distributed systems, sensitive information is often spread across multiple nodes or devices.So, various security threats such as unauthorized access, data breaches, and malicious attacks affects the the confidentiality, integrity, and availability of data [3] and resources in distributed environments.So,it is essential for maintaining trust, protecting privacy, and upholding regulatory compliance. Moreover, the interconnected nature of distributed systems amplify the impact of security breaches, potentially leading to widespread disruption and financial loss. Therefore, implementing robust security measures is critical to safeguarding distributed settings against emerging threats and preserving the trust of users and stakeholders.

In this context, utilizing advanced techniques from the fields of GAN's and CNN's holds great promise for enhancing the security of distributed machine learning systems. GANs [10] are a type of deep learning models that is made of two neural networks, a generator and a discriminator, trained simultaneously. GANs is useful for generating realistic synthetic data, which can be leveraged to augment the training dataset and improve model robustness against poison attacks in federated learning Systems [17]. On the other hand, CNNs have emerged as a cornerstone in computer vision tasks, owing to their capacity to learn hierarchical representations of data automatically. CNNs excel at extracting discriminative features from images, making them wellsuited for detecting subtle patterns indicative of poison attacks. By combining the generative power of GANs [31] with the discriminative capabilities of CNNs, a comprehensive defense mechanism can be developed for poison attack detection in distributed machine learning systems.Poisoning attacks affect the wrong prediction of system.It is very crucial in health care,self driving vehicles and many other applications.So,in order to improve machine learning systems' ability to resist poison attacks.

This paper proposed an innovative approach that uses GANs to create threat model and VGG16 for transfer learning techniques to identify poisonous and nonpoisonous data. Some widely used datasets such as CIFAR10, CIFAR-100 are used to illustrate the effectiveness of this method in detecting poison attacks.

A. Research Motivation

The necessity for strong security measures has been highlighted by the incorporation of machine learning into critical systems. An adversarial approach known as "poisoning" can seriously impair model performance by contaminating the training set. It is frequently not possible for traditional defense measures to identify and prevent these highly trained attackers. Since transfer learning may make use of pre-trained model knowledge, it presents a viable path toward a more precise and efficient defense against poison attacks. The goal of this study is to investigate and validate the application of transfer

learning to improve ML system security against these kinds of attacks.

II. Related Works

Several studies have explored various techniques for defending against poison attacks [3] [15] in machine learning systems. Modern Deep learning techniques such as auto encoder [4] are also used to detect poisoning attacks. Early approaches focused on input sanitization and outlier detection, which proved insufficient against sophisticated adversaries [6]. Recent research has shifted towards more advanced defense mechanisms leveraging techniques such as robust training, model verification, and adversarial training.

Advanced advancements in adversarial attacks and defence strategies in vision applications were covered by [1]. This article discusses several kinds of adversarial attacks on realtime applications. This paper formulates many types of attacks, including white box, black box, and real-world attacks. This survey also mentions a few defensive techniques, including randomised smoothing, regularisation schemes for ReLU networks, ensemble generative cleaning with a feedback loop, and the usage of variational auto-encoders (VAEs). This study also discusses how detecting attacks in language models and vision is becoming a tedious tasks.

Chen, Xiaolin, et al. [8] discussed a data poisoning framework based on Gan against anomaly detection.Here, the poisoning model is based on a generative adversarial network.Perturbations are added for poisoning some inputs. They also developed a a serverside algorithm based on a deep autoencoder in order to defend against such attacks. When the number of labelled datasets increases, its performance decreases slowly.

Psychogyios, Konstantinos [17] discussed generating images using GAN. Here, label flipping attacks are generated and tested based on an aggregation algorithm. This method is also examined in the FL system using secure aggregation methods. Accuracy issues still pose a major challenge in this area. This paper also suggested adding additional datasets and hyperparameters to improve accuracy.

The primary machine learning (ML) concerns for an AI system are the data, model, training, testing, and validation procedures. However, AI also uses a number of knowledge-based techniques, which presents particular security challenges [19] both in the testing and training stages. Although this assumption isn't always accurate, machine learning approaches operate under an assumption that their environment is benign. One of these security issues is the potential for training data manipulation and the exploitation of model sensitivity to reduce the effectiveness of ML classification and regression [27].

Convolutional Neural Networks are extensively used in the computer vision field for the detection of poisoning attacks due to their ability to extract toxic characteristics from images [1] Here, pre-trained CNN models are refined on poisoned data using transfer learning approaches, and the result is highly effective poisonous data detection. Tolpegin et al. [21] used the CIFAR-10 and Fashion MNIST datasets to investigate label flipping based attacks within a distributed system. They used these datasets to evaluate different labelflipping scenarios. In Federated Systems, these techniques yield better results.

TABLE I. Summarizing Existing Recent Surveys

Literature	Methods used	Challenges
Jonnalagadda et al.(2024)	CNN method of poisoning using MNIST	Data leakage issue
Lahe, A.D et al.(2023)	Different stages of ML pipelining and their vulnerabilities	Not efficient to detect attacks in real-time scenarios
Bovenzi et al.(2022)	Shallow autoencoder, deep autoencoder, and ensemble-based encoder for anomaly detection	Less effectiveness of countermeasures against Data Poisoning Attacks in real-time.
Anisetti et al.(2022)	Random Forest method	Label flipping degrading the performance of plain random forests
Raghavan et al.(2022)	Real-Time Poisoning attacks detection using Model Verification in deep computer vision	Method works on neural networks only
Altoub et al.(2022)	convex polytope method	Ensemble method can be used to improve transferability
Liu, I-Hsien, et al.(2022)	Data Washing and IDA Algorithm for detecting poisoned datasets	Not suitable for detecting poisoning attacks on non-DNN models

Table I summarizes the approaches used in a few recent articles along with the difficulties they faced. While these approaches have made significant strides in mitigating poison attacks, there remains a need for more robust and comprehensive defense mechanisms.

III. Poisoning Attack Model

A. Generation of Threats

In order to identify impure images, a threat model is simulated. Generative networks are used here to develop such threat model creation. It is created by injecting poisons into different types of labelled images with the help of Generative Adversarial Networks.

1) Training circumstance: Here, the primary goal is to classify the poisonous and nonpoisonous images using CIFAR10 datasets. A certain amount of data was trained by several clients. Assume a global CNN is trained in a distributed fashion, with each client having access to a subset of the entire dataset. Clients have access to images for every class, and each local data distribution roughly resembles the distribution of the whole dataset.

B. Attacker's Goal

The main goal of attacker is to add malicious behaviour to training datasets. Here, the attacker targets on specific labels which causes manipulation of global model's prediction.GAN is used to misclassify poisonous and nonpoisonous images.Attacker also focusses on degrading the accuracy of global model by adding poison to local datasets. Hence, it creates GAN generated images and assigns some labels to them.The model is trained locally utilising poisoned samples once the resultant images have been combined with each malicious node dataset.

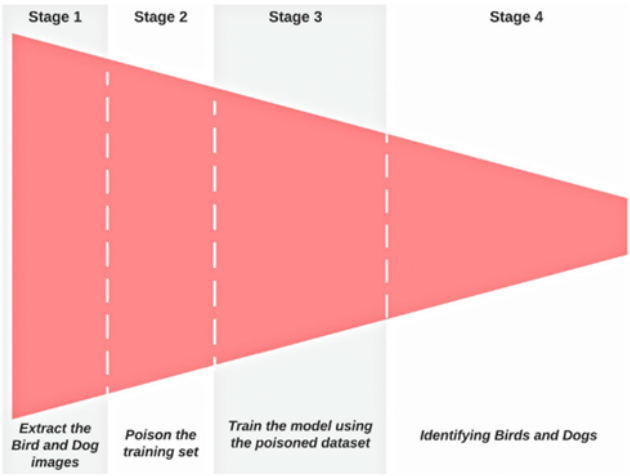

Fig. 1. Steps of image poisoning.

Algorithm: GAN-based poisoning tailored for binary classification tasks

Input: Clean Dataset: (D_{clean}) - Dataset of clean images.
Target Model: (M) - Binary classification model to be poisoned.
GAN Architecture Parameters: (θ_G, θ_D) - Parameters of the generator and discriminator networks.
Training Hyperparameters: (num_epochs,batch_size) - Number of epochs and batch size for GAN training.

Output:Poisoned Dataset: ($D_{poisoned}$) - Dataset containing both clean and poisoned images.

Procedure:

Initialize GAN:

a. Initialize the generator (G) and discriminator (D) with random weights using parameters (θ_G)and(θ_D).

Train GAN:

b. For (t = 1,2, ...,num epochs):
   i. Sample a batch of noise vectors $(z_1, z_2, \ldots, z_m)$ from a prior distribution.
   ii. Generate fake images $(x'_1, x'_2, \ldots, x'_m)$ using the generator: $(x'_i = G(z_i; \theta_G))$.
   iii. Sample a batch of real images $(x_1, x_2, \ldots, x_m)$from the clean dataset.
   iv. Update the discriminator (D) by minimizing the adversarial loss:

$$\left[\min_{\theta_D} \frac{1}{m}\sum_{i=1}^{m}\left[\log D(x_i) + \log\left(1 - D(G(z_i))\right)\right]\right]$$

   v. Update the generator (G) by maximizing the adversarial loss:

$$\left[\max_{\theta_G} \frac{1}{m}\sum_{i=1}^{m} \log\left(1 - D(G(z_i))\right)\right]$$

Generate Poisoned Images:

c. If the generator is trained successfully:
   i. Sample noise vectors $(z'_1, z'_2, \ldots, z'_n)$from the prior distribution.
   ii. Generate poisoned images $(x'_1, x'_2, \ldots, x'_n)$ using the generator: $(x'_i = G(z'_i; \theta_G))$.

d. Else:
   i. Print error message indicating unsuccessful training.

Inject Poisoned Images:

e. If poisoned images are generated successfully:
   i. Integrate the poisoned images (x'1, x'2, ..., x'n) into the clean dataset:

$$(Dpoisoned = Dclean \cup x'_1, x'_2, \ldots, x'_n).$$

f. Else:
   i. Print error message indicating unsuccessful poisoning.

Fig. 2. Attack-Poison GAN algorithm.

C. The Capability of the Adversary

The global aggregation mechanism used by the server is unknown to the attacker.It can only influence learning by means of poisoned model updates. Here, the datasets of the benign customers are unknown to the attacker. But we assume that they are aware that all of the classes accessible in the federated system are also contain the poisoned dataset. Adversary can only increase the compromised client's private collection by adding more generated images, rather than altering it. Finally, it is assumed that the attacker cannot directly access or modify the weights of the local model or affect the local training method.

IV. Proposed Methods

In this paper, Generative Adversarial Networks and transfer learning approaches are used inorder to detect poisoning attacks in machine learning systems. Fig. 1 illustrates the stages of poisoning image.

A. Generating Poisons

To generate poisons in the training data, the GAN approach is employed as an improved technique. By training a generative model to produce images that are perceptually similar to real data, but can trick the classifier into making false predictions, poisoning with a GAN is accomplished. It first retrieves the global parameters from the parameter server in order to update the local model. It then employs a GAN, which consists of a discriminator and a generator, to generate samples of target labels through local training. The generator's goal is to mislead the discriminator into assuming that the generated samples are obtained from the target; the discriminator's role is to identify if the samples are fake and to classify the genuine samples as precisely as possible.The downloaded model acts as the discriminator, while the attacker defines the generator. Once the current round of GAN training is finished, the attacker will intentionally mislabeled the samples that the generator generates. The binary cross entropy loss function and the Adam optimizer are used in the compilation of the generators. Fig. 2 describes The Attack-Poison GAN algorithm.

In the context of Generative Adversarial Networks (GANs), both the generator and discriminator have specific loss functions that drive their training process. These loss functions are fundamental in guiding each network to improve its performance in the adversarial setup. The objective of the generator in a GAN is to generate synthetic data that resembles the real data well enough to fool the discriminator. The loss function for the generator typically aims to minimize the discrepancy between the generated data distribution and the real data distribution.Binary CrossEntropy Loss and Minimax Loss are used as loss function: Minimax Loss reflects the original adversarial nature of GANs where the generator aims to minimize the probability that the discriminator correctly classifies generated data as fake.

Binary Cross-Entropy Loss can be expressed as:

$$\mathcal{L}\text{gen} = -E_{z\sim p(z)}[\log D(G(z))] \tag{1}$$

where
z is the random noise vector input to the generator
G(z) is the generated output
D(G(z)) is the discriminator's output probability on the generated data.

Minimax Loss can be expressed as

$$\mathcal{L}\text{gen} = \log(1 - D(G(z))) \tag{2}$$

where
$\mathcal{L}_{\text{gen}}$ represents the generator loss function
log is the logarithm function.
D is the discriminator network.
G(z) is the generated output from the generator G.
z is the random noise vector input to the generator G.

B. Convolutional Layers [22]

An image's features are extracted and learned using a convolutional layer [2]. An image passes through or slides through a convolutional filter or kernel based on its size or stride.

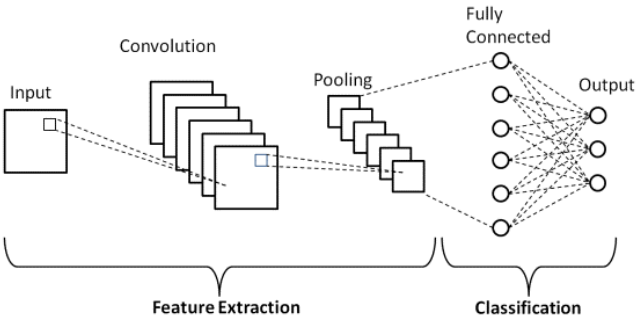

Fig. 3. CNN Architecture[28].

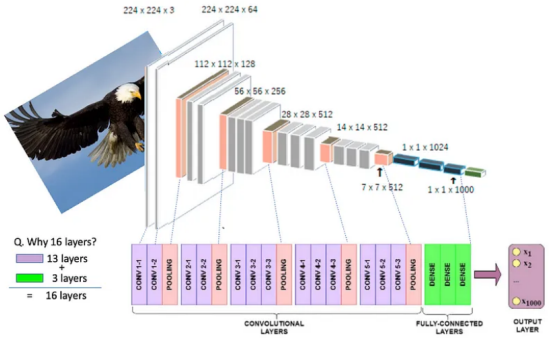

Fig. 4. VGG16 Architecture[9].

Using the kernel's sliding movement as a feature detector, the convolutional layer maps out the features in the image. The convolutional layer maps features from the image using the kernel, a feature detector, and its sliding action. Translational invariance is present in these convolutional filters.Multiple feature learning is possible with more filters. RGB colour images have three channels as well as a depth component. The network breaks these features with the help of the convolution layer. Deeper convolutional layers improve feature detection from a low to a high level, which helps in image detection.

C. Pooling Layer

By reducing the dimensionality of the image without losing its features, the pooling layer compresses the picture layer's dimensions. As a result, overfitting is minimised.Maxpooling, which includes extracting the largest value from a kernel window, is a technique used by CNNs [18].

D. Fully Connected Layer

In the fully linked layer, all neurons are linked to all other neurons. This layer is in control of classification and prediction. The neural network's weights are then modified using back propagation in accordance with the results of a comparison between these predictions and the labels. In the model, a pooling layer is placed after each convolutional layer. Two completely linked layers that come after these layers and meet the output predictions. Three convolutional layers and two fully linked layers constitute the model. Three input channels and sixteen output channels make up the first convolutional layer. The second convolutional layer has 16 input channels and 32 output channels, while the third convolutional layer has 32 input channels and 64 output channels.The 3x3 kernel size is the default. There are 500 output channels and 4*4*64 input channels in the first completely connected layer. The second completely connected layer receives these 500 output channels, which are divided into 10 output channels apiece. Add 0.25 dropout to lessen overfitting.The ADAM optimizer is used, and the learning rate is set to 0.0001. The flattening method and the Relu activation function aid in avoiding the vanishing gradient problem. The CNN workflow is shown in Fig. 3.

E. Transfer Learning with VGG16

When setting up the models for dogs and birds classification using transfer learning with VGG16 [9], we begin by leveraging the pre-trained VGG16 [29] model.This pre-trained model is highly effective at extracting meaningful features from images. By utilizing VGG16 [13] as a base, it has gained knowledge from learning to recognize a wide range of objects and features in images. To ensure that the features learned by VGG16 are preserved and effectively utilized for our classification task, we freeze the convolutional layers of the model. Freezing these layers prevents their weights from being updated during training, preserving the representations learned from ImageNet. This step is crucial for preventing the model from overfitting to the relatively small CIFAR-10 dataset and allows us to leverage the generalization power of the pre-trained model. In the next step,custom classifier layers are build on top of VGG16's frozen convolutional basis. These extra layers are in control of modifying the high-level characteristics that VGG16 extracted for the particular purpose of differentiating between dog and bird images. To transform the 3D feature maps the convolutional layers produced into a 1D feature vector, Flatten layer is added. One or more Dense layers process the flattened representation by applying nonlinear transformations to further process the features. The basic architecture of VGG16 is shown in Fig. 4.

In order to avoid overfitting, dropout layers are frequently placed in between Dense layers and randomly remove a portion of the input units during training. The final layer of the model is an output layer with softmax activation, which produces class probabilities for the categories of interest—in this case, dogs and birds. By compiling the model with appropriate loss and optimization functions,prepare it for training on the dataset. This typically involves using categorical crossentropy loss as the loss function and the Adam optimizer for gradient descent. Before training the model, it's beneficial to inspect the architecture of the model using the summary method.This method gives you information about the model's general structure and the number of trainable parameters. This step helps ensure that the model is configured correctly and ready for training.

Federated systems [24][25] are vulnerable to attacks due to their attacking nature. Malicious clients frequently appear with the intention of interfering with the federated system's training process by directly or indirectly altering the model's weights using data.In such scenario, a malevolent client might add new data or modify the already existing data to suit their needs.At such times, adversaries damage machine learning systems by inserting fake data points or altering already-existing data. One

or more opposing nodes within the federated framework may seek to disrupt the federated process in order to carry out model performance collapse or pattern injection. In order to predict this, various experiments helps to look into the effects of the dataset generation based on GAN [8] for the synthesis of attacks known as data poisoning, which can lead to the degradation of a FL model [26]. In order to increase accuracy of detection, the proposed transfer learning method is used. Assume that one attacker is in control of every wild node and can process all of their datasets simultaneously. First, targeted label attack model is generated in which a GAN is trained on a single class, poisoned bird and dog images in this case using CIFAR-10 datasets, and then generate samples of that class. After that the created samples are given the label "Clean" and sent to the malicious clients. As a result, both the benign dataset and the contaminated samples make up the enhanced dataset that each malicious client possesses. The attacks are detected using VGG16.

V. Experimental Evaluation

A. Dataset

CIFAR-10 [11] [16]and CIFAR-100 image datasets are used for experimental evaluation. CIFAR-10 images has over 60,000 colour, low-resolution images in a 32 by 32 format. The photos are separated into ten sections, with roughly 6000 images in each class. Here, GAN is used to create poisoned images of dogs and birds. Subsequently, the contaminated dataset used as a training set for the development of an image classification algorithm. Then,VGG16 is used to extract the characteristics from the manipulated images. Train the model using its features, then assess the model's performance.The dataset undergone preprocessing procedures to get it ready for training and evaluation. Normalization: By dividing each pixel value by 255.0, the image pixel values were scaled to the range [0, 1]. This ensures that the input data falls within a similar numerical range, which can help improve the convergence of optimization algorithms during training. Data Augmentation: To boost the models' capacity for generalisation and to broaden the dataset. This techniques were applied to the training images.The steps contain rotation, width shift, height shift, and horizontal flipping. Data augmentation helps prevent overfitting by providing the model with variations of the training data. Resizing: The poisoned images generated by the GANs were resized to match the dimensions of the CIFAR-10 images, which are 32x32 pixels in size. This resizing step ensures that the poisoned images are compatible with the input size expected by the classification models.

B. Experimental Procedures

Here,both CIFAR-10 datasets and CIFAR-100 datasets are for the proposed work. There are several key steps to build and evaluate models for classifying and detecting poisonous images incorporating transfer learning with VGG16 and addressing the presence of poisoned images. Firstly, it loads the CIFAR-10 dataset, a collection of 60,000 labeled images in 10 classes, and normalizes the pixel values to a range between 0 and 1. This dataset serves as the foundation for training and evaluating the models. Next,by utilizing Generative Adversarial Networks (GANs) to generate poisoned images. It defines and compiles two GAN architectures, one for generating images of dogs and the other for birds. These GANs are trained to produce synthetic images that may potentially disrupt the performance of the subsequent classification models. After the generation of the poisoned images, the algorithm uses VGG16, a pre-trained convolutional neural network (CNN) known for its efficiency in image identification applications, to build up the models for the classification of dogs and birds via transfer learning. The convolutional layers of the VGG16 model are frozen to preserve their learnt features, and the model is loaded without its top classification layers.

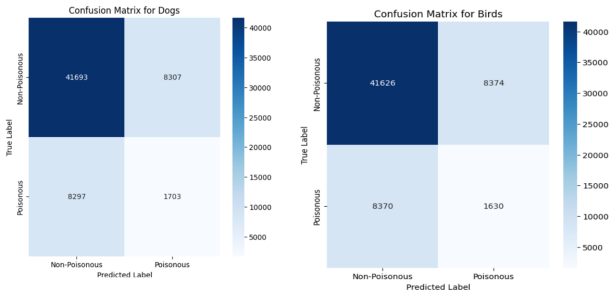

Fig. 5. Confusion matrix using CIFAR-10 datasets.

After that, further layers of custom classifiers are added to the model to modify it for the particular goal of classifying among dogs and birds. The combined dataset—which is divided into training and validation sets—contains both original and polluted images. This data is used to train the models. To enhance model generalisation, data augmentation methods including rotation, width/height shift, and horizontal flip are utilised in addition to the training set. Finally,model is able to detect poisoned images by predicting labels for all images, including both original and poisoned ones. It calculates the percentage of poisoned images correctly identified by each model, shedding light on their robustness in the face of adversarial attacks. This comprehensive evaluation process ensures a thorough understanding of the models' performance and their resilience to potential threats posed by poisoned data.The models' performance is assessed on the validation sets, and the training process is monitored over several epochs. To assess the models' effectiveness,the confusion matrix is shown in Fig. 5. Here,there are two classes such as class0 and class1. Class0 represents images of dogs and the images of bird comes under class1. The model effectiveness can be evaluated by plotting the training and validation loss curves as well as the accuracy curves are plotted which are shown in Fig. 6 and Fig. 7.

C. Experimental Results

The summary of the performance of a classification model on a set of test data is described on the classification report which is shown in Fig. 8.

Inorder to effectively improve F1 score, CIFAR-100 image datasets are also used.CIFAR-100 offers a wider range of classes and images, making it appropriate for a wider range of challenging and complex recognition tasks. The CIFAR-100 dataset is a collection of 60,000 32x32 color images in 100 classes, with 600 images per class. It serves as a

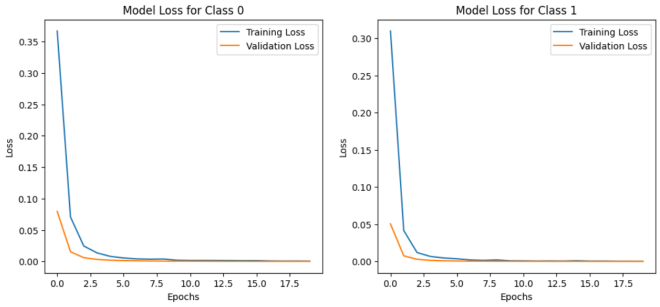

Fig. 6. Model loss curve using CIFAR-10 datasets.

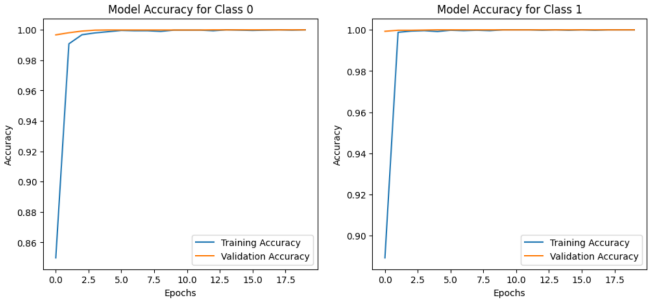

Fig. 7. Accuracy curve using CIFAR-10 datasets.

benchmark dataset for image classification tasks, particularly for multiclass classification.Initially load such datasets,then normalized and resize, and select a subset of classes (select classes 0 (apple) and (aquarium fish) for poisoning. Creates a Generative Adversarial Network to generate poisoned images for selected classes,trains two separate models (one for each class) using transfer learning, evaluates the models' performance by plotting the training loss and accuracy which is shown in Fig. 9 and Fig. 10. It also generates confusion matrices shown in Fig. 11, and assessing its performance on both original and poisoned data using classification report for both classes which is shown in Fig. 12.

Classification Report for Class 0:

	precision	recall	f1-score	support
0.0	0.34	0.34	0.34	5000
1.0	0.67	0.67	0.67	10000
accuracy			0.56	15000
macro avg	0.50	0.50	0.50	15000
weighted avg	0.56	0.56	0.56	15000

Classification Report for Class 1:

	precision	recall	f1-score	support
0.0	0.34	0.34	0.34	5000
1.0	0.67	0.67	0.67	10000
accuracy			0.56	15000
macro avg	0.50	0.50	0.50	15000
weighted avg	0.56	0.56	0.56	15000

Fig. 8. Classification report using CIFAR-10 datasets.

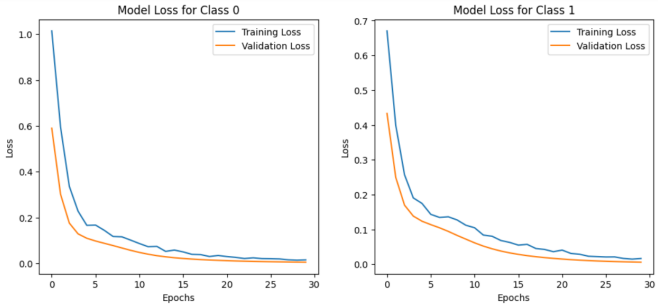

Fig. 9. Loss curve using CIFAR-100 datasets.

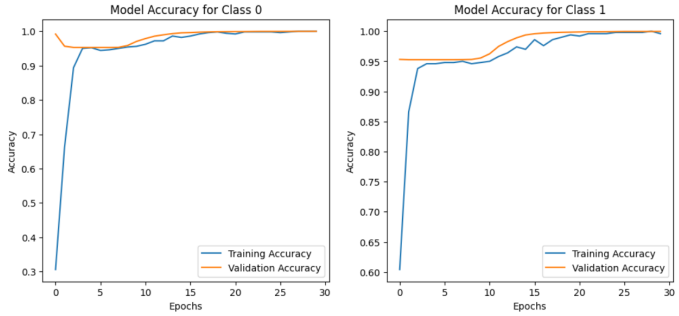

Fig. 10. Accuracy curve using CIFAR-100 datasets.

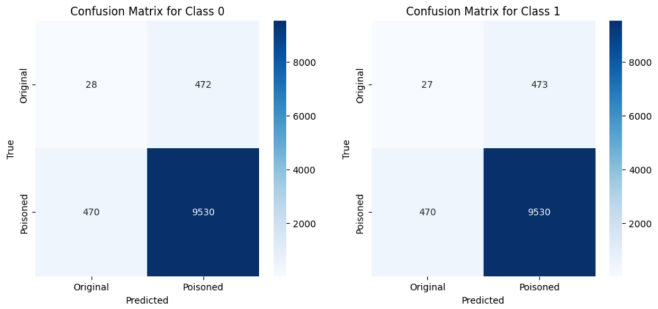

Fig. 11. Confusion matrix using CIFAR100 datasets.

Classification Report for Class 0:

	precision	recall	f1-score	support
0.0	0.05	0.05	0.05	500
1.0	0.95	0.95	0.95	10000
accuracy			0.91	10500
macro avg	0.50	0.50	0.50	10500
weighted avg	0.91	0.91	0.91	10500

Classification Report for Class 1:

	precision	recall	f1-score	support
0.0	0.05	0.04	0.05	500
1.0	0.95	0.96	0.95	10000
accuracy			0.91	10500
macro avg	0.50	0.50	0.50	10500
weighted avg	0.91	0.91	0.91	10500

Fig. 12. Classification report using CIFAR100 datasets.

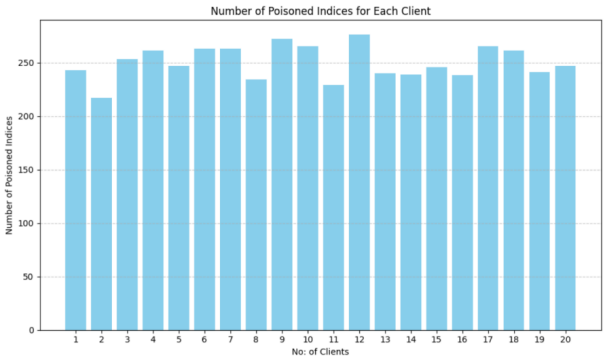

Fig. 13. No. of Poisoned indices in multiple clients.

Federated settings can be implemented using Keras having 20 clients and the no.of poisoned points can be detected which is shown in the Fig. 13. This novel approach can be applied in distributed ML systems also.

VI. Discussion

The results indicate that the proposed method using transfer learning and a pre-trained VGG16 model is effective in detecting poisoned images in the CIFAR-10 and CIFAR-100 dataset. Both models (for classes 0 and 1) showed high training and validation accuracy, along with strong precision, recall, and F1 scores.

VII. Real World Applications

The proposed framework for poison attack detection in distributed machine learning systems offers tangible benefits across diverse sectors, as corroborated by existing research findings. In the cybersecurity domain, where data integrity is paramount, advanced detection mechanisms are imperative. By integrating the framework into intrusion detection systems, organizations can fortify their capabilities against malicious activities.

Autonomous vehicles, reliant on machine learning algorithms for safe navigation, stand to gain significantly from the framework's implementation. The potential risks posed by poison attacks targeting distributed learning systems within autonomous vehicles. By deploying the framework, automotive manufacturers and transportation authorities can augment their vehicles' defenses against adversarial manipulation, bolstering passenger safety and public trust, as evidenced by studies conducted by [7].

In healthcare, where accurate diagnoses are critical, safeguarding distributed machine learning systems is essential. Poison attacks on these systems can compromise patient confidentiality and introduce diagnostic errors. By adopting the framework, healthcare providers can fortify their defenses against adversarial threats, ensuring the integrity of medical data and the reliability of clinical decision-making processes. Financial institutions face significant risks from poison attacks targeting distributed machine learning systems used in fraud detection and algorithmic trading. The author in [8], underscore the potential impact of adversarial manipulation on financial markets and investor confidence. By integrating the framework into their security protocols, financial institutions can mitigate these risks, protect customer assets, and uphold the integrity of financial transactions.

The main advantages of using this method include:

1) Enhanced detection accuracy: SecureTransfer improves the accuracy of poison attack detection by using pre-trained models that recognise tiny abnormalities in the training set.

2) Scalability: SecureTransfer is a scalable solution for diverse machine learning applications because it utilises transfer learning, which enables it to be applied across several datasets and domains without requiring a significant amount of retraining.

3) Efficiency: By using pre-trained models, the method saves time and resources by reducing the computational overhead often involved with anomaly recognition.

4) Robustness: By offering an extra line of defence against complex poison attacks, SecureTransfer strengthens the resilience of machine learning systems and guarantees dependable model performance.

5) Adaptability: The technique can be applied in a variety of contexts since it can be tailored to various data kinds and attack circumstances.

VIII. Limitations and Future Work

While SecureTransfer demonstrates promising results, it is essential to acknowledge certain limitations. The reliance on pre-trained models may not always guarantee optimal performance, especially when the target task significantly deviates from the original training context. Additionally, the approach's effectiveness may vary based on the nature and sophistication of the poison attack, necessitating ongoing research to refine the model.

Furthermore, the flexibility and adaptability of transfer learning enable the integration of additional defense mechanisms to mitigate the impact of poison attacks [12] on ML systems. Future research directions may explore the combination of transfer learning with other anomaly detection techniques and investigate the robustness of the proposed method against sophisticated poisoning strategies.The findings and methodologies can inspire further studies into the application of transfer learning for other types of adversarial attacks. Future research could explore the integration of SecureTransfer with other ML security techniques to develop comprehensive, multilayered defense strategies.It will also focus on improving the scalability and applicability of SecureTransfer to a broader range of ML tasks.

IX. Conclusion

In conclusion, the use of transfer learning methods for poison attack detection in machine learning systems presents a promising approach to enhance the security and robustness of ML models. Through the integration of pre-trained models such as VGG16, the proposed method leverages the knowledge learned from large-scale datasets to detect anomalies introduced by poisoned data.The experimental results demonstrate the effectiveness of the transfer learning-based

approach in accurately identifying poisoned instances across different datasets and scenarios. By combining the feature extraction capabilities of pre-trained models, the proposed method achieves high detection accuracy while maintaining computational efficiency.

Overall, the findings suggest that transfer learning-based approaches hold significant potential for enhancing the security and reliability of ML systems in real-world applications,paving the way for more resilient defense mechanisms against adversarial attacks.

REFERENCES

[1] N. Akhtar, et al., "Advances in adversarial attacks and defenses in computer vision: A survey," *IEEE Access*, vol. 9, pp. 155161-155196, 2021.

[2] S. Albelwi and A. Mahmood, "A framework for designing the architectures of deep convolutional neural networks," Computer Science and Engineering Department, University of Bridgeport, Bridgeport, CT 06604, USA.

[3] M. Altoub, et al., "An ontological knowledge base of poisoning attacks on deep neural networks," *Applied Sciences*, vol. 12, no. 21, p. 11053, 2022, doi: 10.3390/app122111053.

[4] E. M. Anass, C. Gouenou, and B. Reda, "Poisoning-attack detection using an auto-encoder for deep learning models," in *International Conference on Digital Forensics and Cyber Crime*, S. Jahankhani, G. R. S. Murthy, and A. Abdoli, Eds. Cham: Springer Nature Switzerland, 2022, pp. 123-136, doi: 10.1007/978-3-030-95409-4_9.

[5] M. Anisetti, et al., "On the Robustness of Ensemble-Based Machine Learning Against Data Poisoning," *arXiv preprint arXiv:2209.14013*, 2022.

[6] T. Bai, et al., "Recent advances in adversarial training for adversarial robustness," *arXiv preprint arXiv:2102.01356*, 2021.

[7] G. Bovenzi, et al., "Data poisoning attacks against autoencoder-based anomaly detection models: A robustness analysis," in *Proceedings of the ICC 2022-IEEE International Conference on Communications*. IEEE, 2022.

[8] X. Chen, et al., "A GAN-based data poisoning framework against anomaly detection in vertical federated learning," *arXiv preprint arXiv:2401.08984*, 2024.

[9] Dataman AI, "Transfer learning for image classification 7: Fine-tune the transfer learning model," Medium, Jan. 15, 2023. [Online].

[10] I. Goodfellow, et al., "Generative adversarial networks," *Commun. ACM*, vol. 63, no. 11, pp. 139-144, 2020.

[11] R. Jha, J. Hayase, and S. Oh, "Label poisoning is all you need," in *Advances in Neural Information Processing Systems 36*, 2023, pp. 71029-71052.

[12] A. D. Lahe and G. Singh, "A survey on security threats to machine learning systems at different stages of its pipeline," *Int. J. Inf. Technol. Comput. Sci.*, vol. 15, no. 2, Article e0203, 2023, doi: 10.5815/ijitcs.2023.02.03.

[13] I.-H. Liu, J.-H. Wu, I.-C. Chang, and W.-C. Chen, "A robust countermeasures for poisoning attacks on deep neural networks of computer interaction systems," *Applied Sciences*, vol. 12, no. 15, p. 7753, 2022, doi: 10.3390/app12157753.

[14] C. Ma, et al., "Trusted AI in multi-agent systems: An overview of privacy and security for distributed learning," *arXiv preprint arXiv:2202.09027*, 2022.

[15] A. Mehra, et al., "How robust are randomized smoothing based defenses to data poisoning?," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

[16] T. Pang, et al., "Accumulative poisoning attacks on real-time data," in *Advances in Neural Information Processing Systems 34*, 2021, pp. 2899-2912.

[17] K. Psychogyios, et al., "GAN-Driven Data Poisoning Attacks and Their Mitigation in Federated Learning Systems," *Electronics*, vol. 12, no. 8, p. 1805, 2023.

[18] V. Raghavan, T. Mazzuchi, and S. Sarkani, "An improved real-time detection of data poisoning attacks in Deep Learning Vision systems," *Discover Artif. Intell.*, vol. 2, no. 1, 2022.

[19] Z. Tian, et al., "A comprehensive survey on poisoning attacks and countermeasures in machine learning," *ACM Comput. Surv.*, vol. 55, no. 8, pp. 1-35, 2022.

[20] R. Tomsett, K. Chan, and S. Chakraborty, "Model poisoning attacks against distributed machine learning systems," in *Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications*, vol. 11006. SPIE, 2019.

[21] V. Tolpegin, et al., "Data poisoning attacks against federated learning systems," in *Computer Security–ESORICS 2020: 25th European Symposium on Research in Computer Security, ESORICS 2020, Guildford, UK, September 14–18, 2020, Proceedings, Part I*, vol. 25. Springer International Publishing, 2020.

[22] UpGrad, "Basic CNN Architecture: Explained with Simple and Practical Example," upGrad, Mar. 25, 2021. [Online]. Available: https://www.upgrad.com/blog/basic-cnn-architecture/. [Accessed: Jun. 19, 2024].

[23] C. Wang, et al., "Poisoning attacks and countermeasures in intelligent networks: Status quo and prospects," *Digit. Commun. Netw.*, vol. 8, no. 2, pp. 225-234, 2022.

[24] F. Wibawa, F. O. Catak, M. Kuzlu, S. Sarp, and U. Cali, "Homomorphic encryption and federated learning based privacy-preserving CNN training: Covid-19 detection use-case," in *Proceedings of the 2022 European Interdisciplinary Cybersecurity Conference*, 2022, pp. 85-90.

[25] G. Xia, et al., "Poisoning Attacks in Federated Learning: A Survey," *IEEE Access*, vol. 11, pp. 10708-10722, 2023.

[26] J. Zhang, et al., "Defending poisoning attacks in federated learning via adversarial training method," in *Frontiers in Cyber Security: Third International Conference, FCS 2020, Tianjin, China, November 15–17, 2020, Proceedings 3*, S. Jahankhani, G. R. S. Murthy, and A. Abdoli, Eds. Springer Singapore, 2020.

[27] S. Zhang, H. Gao, and Q. Rao, "Defense against adversarial attacks by reconstructing images," *IEEE Trans. Image Process.*, vol. 30, pp. 6117-6129, 2021.

[28] Shimja, M., and K. Kartheeban. "Empowering diagnosis: an astonishing deep transfer learning approach with fine tuning for precise lung disease classification from CXR images." *Automatika*, vol. 65, no. 1, pp. 192-205, 2024.

[29] Kartheeban, K. "Beyond the Norm: A Modified VGG-16 Model for COVID-19 Detection." *International Journal of Advanced Computer Science & Applications*, vol. 14, no. 11, 2023.

[30] Archa, A. T., and K. Kartheeban. "Real Time Poisoning Attacks and Privacy Strategies on Machine Learning Systems." In *2024 3rd International Conference on Sentiment Analysis and Deep Learning (ICSADL)*, IEEE Computer Society, 2024.

[31] Jonnalagadda, A., et al. "Modelling Data Poisoning Attacks Against Convolutional Neural Networks." *Journal of Information & Knowledge Management*, vol. 23, no. 2, 2024, pp. 2450022.