IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 14 July 2023, accepted 14 August 2023, date of publication 17 August 2023, date of current version 23 August 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3306333

RESEARCH ARTICLE

A Novel Framework for Smart Cyber Defence: A Deep-Dive Into Deep Learning Attacks and Defences

IRAM ARSHAD[1], SAEED HAMOOD ALSAMHI[2,3], YUANSONG QIAO[1], BRIAN LEE[1], AND YUHANG YE[1]

[1]Software Research Institute, Technological University of Shannon: Midlands Midwest, Athlone N37 HD68 Ireland
[2]Insight Centre for Data Analytics, University of Galway, Galway, H91 AEX4 Ireland
[3]Faculty of Engineering, IBB University, Ibb, Yemen

Corresponding author: Iram Arshad (i.arshad@research.ait.ie)

This work was supported by the Technological University of Shannon: Midland Midwest under the President Doctoral Scholarship. The work of Saeed Hamood Alsamhi was supported by the Science Foundation Ireland co-funded by the European Regional Development Fund under Grant SFI/12/RC/2289_P2.

ABSTRACT Deep learning techniques have been widely adopted for cyber defence applications such as malware detection and anomaly detection. The ever-changing nature of cyber threats has made cyber defence a constantly evolving field. Smart manufacturing is critical to the broader thrust towards Industry 4.0 and 5.0. Developing advanced technologies in smart manufacturing requires enabling a paradigm shift in manufacturing, while cyber-attacks significantly threaten smart manufacturing. For example, a cyber attack (e.g., backdoor) occurs during the model's training process. Cyber attack affects the models and impacts the resultant output to be misled. Therefore, this paper proposes a novel and comprehensive framework for smart cyber defence in deep learning security. The framework collectively incorporates a threat model, data, and model security. The proposed framework encompasses multiple layers, including privacy and protection of data and models. In addition to statistical and intelligent model techniques for maintaining data privacy and confidentiality, the proposed framework covers the structural perspective, i.e., policies and procedures for securing data. The study then offers different methods to make the models robust against attacks coupled with a threat model. Along with the model security, the threat model helps defend the smart systems against attacks by identifying potential or actual vulnerabilities and putting countermeasures and control in place. Moreover, based on our analysis, the study provides a taxonomy of the backdoor attacks and defences. In addition, the study provides a qualitative comparison of the existing backdoor attacks and defences. Finally, the study highlights the future directions for backdoor defences and provides a possible way for further research.

INDEX TERMS Backdoor attacks, cyber-attacks, deep learning, defences, security, smart cyber defence, smart manufacturing security.

I. INTRODUCTION

Recently, the most valuable resource is data collected from the smart devices in Smart Manufacturing (SM). Internet of things and cyber-physical systems are one of the fundamental pillars of the 4.0 and 5.0 industry revolution. These pillars can smart anything like manufacturing, cities, home, agriculture and so on. Substantial recent investment has been directed towards developing SM systems that can respond in real-time to changes in customer demands and the conditions in the supply chain and the factory itself. SM is a crucial component of the broader thrust towards Industry 4.0 and 5.0. Cyber attacks are significantly increased where hackers target various organizations, institutes, health sectors, industries, and individuals. The escalation of technology and leaning on digital systems have made it easier for cyber attackers to exploit vulnerabilities and attacks. Integrating digital technologies in SM industry systems brings potential new

The associate editor coordinating the review of this manuscript and approving it for publication was Derek Abbott.

security challenges [65]. Deep Learning (DL) algorithms play a crucial role in manufacturing intelligence to make better decisions, e.g., reducing energy consumption and improving product quality. DL models have been widely used employed to detect and prevent security threats in various applications. Intrusion detection systems, fraud detection, and abnormal system behavior are examples. However, recent studies show the variety of security threats against these DL models as mentioned in [1], and [2].

Adapting internet connectivity devices, collecting massive data, cleaning, preparing, and using the DL algorithm without considering security threats makes SM industries vulnerable. A well-known attack is a backdoor attack, where semantically consistent secret triggers, e.g., visible or invisible, which are secretly known to the attackers, can mislead the DL models into a wrong classification determined by the attacker at inference [3]. These backdoor attacks are difficult to detect because the attack effect remains dormant without the backdoor trigger. The attacks could bring disaster and causalities if the disrupted DL models are deployed in safety and critical applications without being diagnosed. For example, a self-driving system could be attacked to classify the sign of stopping as a "speed of 80km/hr" by adding a reflecting trigger,which could lead to a crash [4].

The malignant attack (e.g., backdoor) receives increased attention from the research community because these DL models are used in various safety and critical applications. Several literature surveys and review papers on the attack surface of Machine Learning (ML) and DL models have been published in [7], [8], [9], [10], [11], [12], and [13]. However, the unified security framework and threat model are generally not discussed. For example, the authors of [7] reviewed adversarial attacks on DL approaches in computer vision. Moreover, in study [8], the authors reviewed, summarized and discussed the adversarial generation method and countermeasures on DL approaches. In the study [9], the authors discussed and analyzed security threats and defences on ML. In studies [10] and [11], authors classified backdoor attacks based on attackers' capabilities and characteristics in general. Further, the authors of [12] reviewed the concept, cause, characteristics, and evaluation metrics and discussed the advantages and disadvantages of generating adversarial examples. Also, in study [13], the authors reviewed attacks on ML algorithms and illustrated them on the spam filters.

To the best of our knowledge, studies have yet to be done on smart cyber defence to protect data and DL model security altogether and provide a unified security framework for smart cybersecurity. Therefore, we provide a novel unified multi-layered a comprehensive framework for the security infrastructure. The proposed framework helps in protecting the data and models from the backdoor and other attacks. It consists of a collection of strategies, procedures and policies organizations can use to protect their systems from cyber-attacks. The framework encompasses a wide range of security measures, including data privacy and protection, and model protection. In addition, to further enhance security and protection, we also provide a threat model to analyze the potential security risk and vulnerabilities in the design and implementation of the models, ensuring the overall security of models to make them robust.

A. MOTIVATIONS AND CONTRIBUTIONS

Cybersecurity has become increasingly important in recent years due to the rising number of cyber-attacks and data breaches. It is critical to ensure data security and intelligent models in SM to protect the digital industry from potential cyber threats. ML and DL algorithms are used in various applications to detect patterns and anomalies in SM systems' vast amounts of data. By analyzing data, these algorithms can identify potential cyber threats in real-time and alert security teams, enabling them to take swift action and prevent harm to the system or data. However, these algorithms are susceptible to various types of attacks, making the security of these algorithms essential in SM to protect against general cyber threats and model attacks.

Smart cyber defence is significantly important in SM because these systems are often interconnected and rely on data to make decisions. A single vulnerability in the system could have far-reaching consequences. Therefore, a comprehensive smart defence framework is essential to ensure the security and integrity of SM systems. By implementing advanced cybersecurity solutions and practices, manufacturers can protect their operations, customers, and bottom line from the growing threat of cyber attacks.

In SM security, evaluating a system involves continually identifying the categories of attacks, assessing the system's resilience against those attacks, and strengthening the system against those categories of attacks. This study introduces a novel framework for smart cyber defence analysis of DL model security. The framework also provides a threat model to identify potential security risks and vulnerabilities in designing and implementing DL systems that aim to make models robust and secure. The summary of this research contribution is described as follows:

1) In order to identify the potential vulnerabilities of data and model attacks (e.g., backdoor) and offer to mitigate them, this paper introduces a novel framework for smart cyber defence of deep learning security. In the proposed framework, data is acquired, and subsequently technical measures are taken to shield it from threats.
2) The study categorizes the attacks based on specimen analysis. Different methods and properties used to generate the backdoor specimen are discussed in specimen analysis. Class-specific and class-agnostic, one-to-one (single trigger to the same label), and one-to-N (multiple triggers to different labels) trigger transparency, feature, and image space are among the properties and methods. Then, accesses the fully structured adversary threat model in terms of goals, capabilities, assumptions, attack/defence surface, and defence target.

3) The study highlights the future direction in smart cyber defence based on taxonomy, assessment, and qualitative analysis, which aid interested researchers in making additional contributions to secure SM systems and other applications.

B. PAPER STRUCTURE

The scope of this paper is to explore the implementation of smart cyber defence solutions for SM and suggest a security framework that prioritizes data, model privacy, and protection. Our research specifically examines backdoor attacks and defences, emphasizing the importance of robust DL models in safeguarding SM systems. This is depicted in the accompanying Figure 1.

The rest of the paper is organized as follows. In section II, the definition of the backdoor, abbreviations, and acronyms in the paper has been discussed. In section III, we introduce the proposed security infrastructure framework. Based on our analysis in section IV we present the taxonomy of backdoor attacks. In section V, we discuss the defence of backdoor attacks. In section VI, we provide possible future directions on backdoor defences. At the same time, the paper is concluded in VII.

II. PRELIMINARIES

A. BACKDOOR FORMULATION

We can formally formulate the backdoor attack as follow. Given an input (x_i, y_i) belongs to D_c to a clean DL model F_{Θ_c}, which takes input and based on a decision function $z_c = f(x)$, outputs the final predicted label. Where z_c is the predicted label. A dataset D_c is inclusive for training, and D_t is a testing dataset. In the context of a backdoor, an adversary Adv aims to inject perturbations to a small number of inputs, as in 1.

$$x_i^a = x_i + \delta \tag{1}$$

Where δ is Adv trigger stamp on clean input x_i, the predicted label will always be Adv targeted class z_{adv}, where z_{adv} is given in 2. It is a backdoor model decision function with a high probability of being the same as per the Adv targeted label.

$$z_{adv} = F_{\Theta bd}(x_i^a) \tag{2}$$

The injection of the perturbations is added to the training dataset D_c that becomes poison training datasets as in 3.

$$D_{bd} = D_c U D_{adv} \tag{3}$$

The dataset mentioned in 3 is used to train the f(x), where the model learned to minimize the cross-entropy loss on the D_{bd} training dataset. In addition, when the model is deployed, and a new backdoor sample x_i^a is tested, the probability of the given input is high $f(x_i^a)$ so that the Adv targeted class will choose. In addition, the model will behave effectively for the benign inputs without performance detraction. The success of the backdoor attack model can also be evaluated.

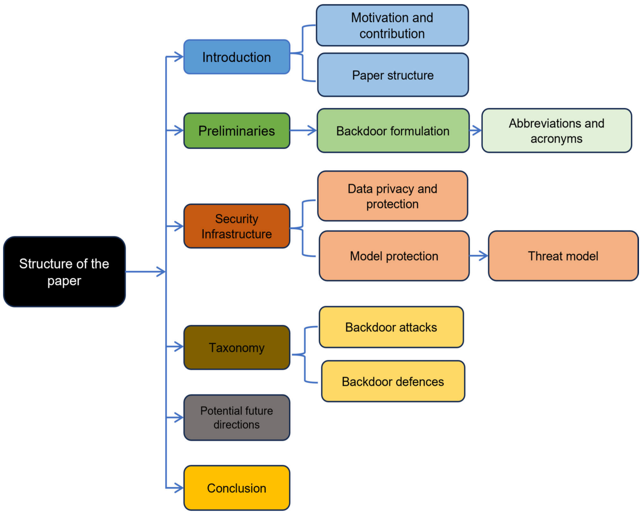


FIGURE 1. Structure of the paper.

We have observed that most of the backdoor models are generally evaluated based on Injection rate (IR) (i.e., the ratio of poison samples injected in the clean dataset during the training of the model), Clean Data Accuracy (CA) (i.e., the portion of the clean test samples that are correctly classified to the ground truth class), Poison Data Accuracy (PA) (i.e., the portion of the poison test samples that are correctly classified to the attackers decided label) and Attack Success Rate (ASR) (i.e., the portion of the benign samples stamp with the trigger successfully classify to the attackers targeted class.) as mentioned in research [3]. For a successful backdoor model, the model accuracy should be as similar as CA, and IR should be the smallest ratio of the total clean dataset as mentioned in research [14], [15]. In contrast, ASR should be high, which may be close to 100%. In Figure 2, colorblackby way of example, we illustrate a process of generating clean-label backdoor attacks.

1) ABBREVIATIONS AND ACRONYMS

To ease the readability, the study generally provides some terms used frequently in this paper. The terms are described in the table 1.

III. SECURITY INFRASTRUCTURE

Protecting DL models from cyber-attacks has greatly concerned practitioners and researchers. We briefly discuss a proposed comprehensive multi-layered data and model protection security framework that can potentially be used to discover the security insights from the data to the model; to build smart cyber security systems, e.g., predictive analysis, behavioral analysis, and automatic response. In order to make sure a secure data-driven intelligent decision, a comprehensive analysis is required to understand the potential security vulnerabilities. For this purpose, our proposed suggested framework takes into account both the security of the models from numerous attacks as well as protecting data. Further,

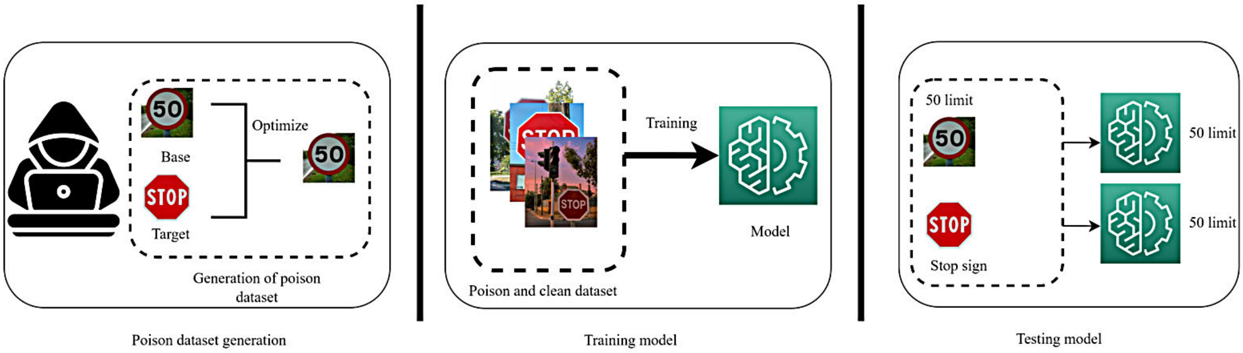


FIGURE 2. An example of generating clean label backdoor attack. 1) Poison dataset generation: The adversary can generate the poison instances close to the base instances in pixel space but looks like target instances in image space. 2) Training: Poison images are mixed with benign and included in the training dataset, thus, affecting the decision boundary. 3) Inference: The clean images of the target class will be recognized as a base class at inference time.

TABLE 1. A summary of the definition of terms.

Terms	Abbreviation	Description
Adversary	Adv	It is exchangeable with the attacker who wants to backdoor the model.
End-user	Eu	The end-user is exchangeable with a defender as in most cases the defender act as the end-user of the DL models.
Clean input	(x_i, y_i)	The input image and its corresponding label. It is interchangeable with benign input, clean sample, and clean instance.
Trigger input	δ	It is an input containing visible or invisible triggers that the attacker chooses to fire the backdoor. This term can be interchangeable with poison input, backdoor input, trigger sample, and trigger instance.
Target class	z_{adv}	A class refers to the attacker's target class. This can be interchangeable with the target label.
Source class	SC	A class referred to as the attackers' source class is chosen as an input to stamp the trigger. It can be interchangeable with the source label.
Digital attack	DIA	Imperceptible perturbation on digital images through modifications of pixels in the digital images. This can be interchangeable with triggerless or imperceptible attacks.
Physical attack	PHA	Perceptible perturbations on the digital images through stamping some shapes on digital images (e.g., flower, shape, triangle).
Latent representation	LR	Latent feature or representation refers to a low representation of high dimensional data of the input. In the number of countermeasures, latent representation is exploited to detect the backdoor behaviour.
Pixel space	PS	It is a high-dimensional space where images will have different possible combinations. Pixel space is used to inject backdoor triggers.
Image-space	IS	It is a low-dimensional space that is understandable for humans. Image space is used to ensure that poison images are perceptible to the original image.
Specimen	$\overline{\mathbb{B}}$	While generating a backdoor attack based on provided methods and properties. A specimen consists of a particular method and property for a targeted use case.
Clean dataset	D_c	A Clean dataset that is used to train models.
Poison dataset	D_t	A Poison dataset is used to inject during the training.
Mixed dataset	D_{bd}	A backdoor dataset that is a combination of poison and benign datasets
Clean predicted label	z_c	A Clean predicted label that is the final output based on the decision function.
Decision function	$f(x)$	A decision function is used to make the predictions on the model.
Poison instance	x_i^a	A poison instance are used during the inference time to test the backdoor.
Trigger	δ	An adversarial trigger that stamps on the input.
Poison model	$\mathrm{F}_{\Theta bd}$	A poison or backdoor model that disrupts the model.

a proposed threat model for deep learning could exploit the model flaws. In the threat model, looking from the lens of the attackers' perspective is one of the ways to focus on their perspective, goals, and capabilities.

In the proposed unified framework, the study considers several aspects of cyber security while protecting the data, and the models. The first stage is to ensure data privacy and protection because it is paramount in the digital world. The second stage is the model protection and the threat model analysis that is desired to build a smart cyber security system. The proposed unified framework could be more efficient and intelligent in providing two-tier security of models. In Figure 3, we illustrate a proposed novel framework for smart cyber defence for providing security. Further, the protection of models leads toward a threat model in Figure 4 for exploring model vulnerabilities regarding goals, capabilities, assumptions, and attack surfaces. In the following sections (III-A,III-B), we briefly discuss the working procedure of the proposed framework.

A. DATA PRIVACY AND PROTECTION

In today's digital era, it is essential to safeguard people's personal information from unauthorized access, and this is referred to as data privacy. The protection of personal data ensures that individuals maintain their rights over it. Therefore, once the dataset has been collected from numerous sources [5] and [6], this layer is responsible for providing privacy to the data. High-quality data is needed to achieve highly accurate predictions on the predictive models. The data collection process requires cleansing of data and handling

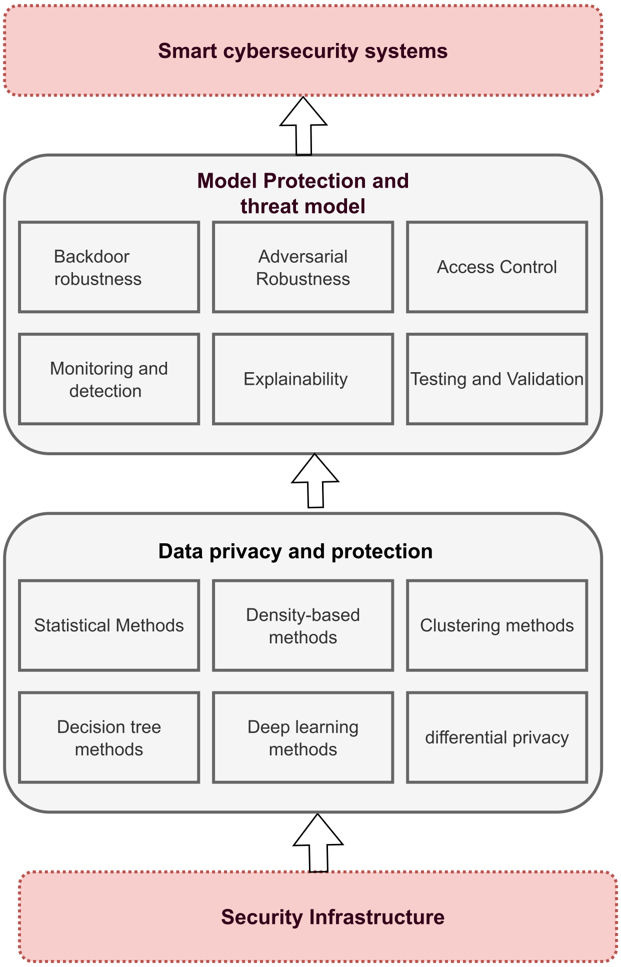


FIGURE 3. Proposed Framework for smart cyber security system.

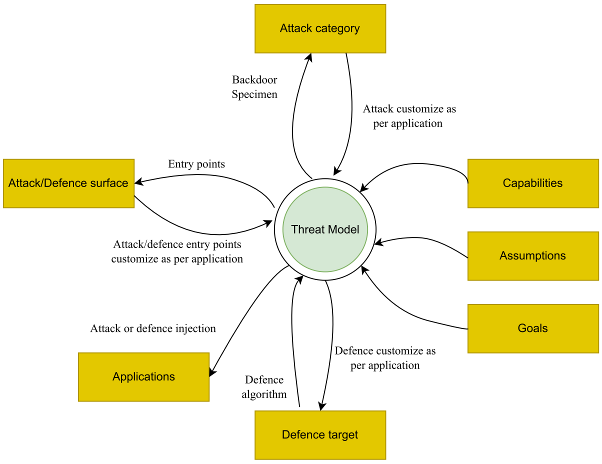

Attack category
Backdoor Specimen
Attack customize as per application
Attack/Defence surface
Entry points
Capabilities
Threat Model
Attack/defence entry points customize as per application
Assumptions
Goals
Attack or defence injection
Defence customize as per application
Applications
Defence algorithm
Defence target

FIGURE 4. A linking threat model of our proposed framework for analyzing the potential security risk of models.

missing or corrupted values. However, beyond a solid understanding of the data preparation process, privacy the data is also needed. Several anomaly detection techniques can be used to enhance privacy, where we can identify the unusual or abnormal data points within the dataset.

We can use different statistical model techniques to detect the anomalies based on the data distribution. For example, z-score methods identify anomalies as data points significantly different from the mean of the dataset. Using density-based methods, we can identify anomalies as data points in the dataset's low-density regions. For example, the local outlier factor method is a technique that can be used to detect anomalies. Furthermore, we can assign a score to each data point based on its relative density compared to the surrounding data points. We can also use clustering techniques where the data is divided into clusters. Then, anomalies are identified as data points that do not belong to any cluster. For example, the k-means algorithm can cluster the data and identify the anomalies as data points far from the cluster centroids.

Another way is to use decision trees to identify the anomalies in the dataset. For example, we can use the Isolation forest algorithm that uses decision trees to isolate the anomalies by randomly selecting and splitting the data into smaller subsets. We can also use DL algorithms to identify the anomalies in the dataset. For example, auto-encoders can be used to reconstruct the data, and the data points that are reconstructed poorly can consider anomalies. Lastly, differential privacy is used to protect the individual's privacy in data analysis and to ensure that the dataset does not reveal any sensitive information about individual data points. One of the differential privacy techniques to protect the data is Laplace noise. The noise can be added to the dataset to protect privacy. Data privacy and protection structure is a complex and multi-layered process involving a range of security measures and risk management strategies. In addition, protecting data is a critical concern for any organization, given the increased risk of data breaches and cyber-attacks. Organizations must follow a structural approach that includes policies and procedures, data classification, encryption, and incident response planning to ensure data privacy and protection.

Organizations must have very clearly defined policies and procedures. These policies should include information about the types of data collected, how they are collected, stored, and transmitted, and who has access to them. The guidelines should also specify the methods that will be used to protect the data, such as encryption, access controls, and monitoring. Further, based on the sensitivity of the data, it should be classified based on sensitivity level. Categorizing data based on sensitivity helps the organization determine the appropriate security measures to apply. For instance, financial records and health care data information require more security than less sensitive data such as customer contact information. Encryption is a critical security measure for protecting data privacy. It involves encrypting the data that authorized users can only decode. Thus, this helps to prevent unauthorized access to the data, even if it is intercepted during the transmission.

Organizations should have a well-defined incident response plan to quickly and effectively respond to security incidents or data breaches. It refers to detecting, analyzing, and responding to security incidents or data breaches promptly and effectively. It involves identifying the steps to take in a security incident, such as notifying affected individuals and authorities and implementing measures to prevent similar incidents. The goal of incident response is to minimize the damage caused by a security incident to prevent it from escalating into a larger problem.

B. MODEL PROTECTION AND THREAT MODEL

To maintain the integrity of the model's output, it's crucial to protect it from attacks. Once the dataset is prepared and its security is ensured, the data is fed directly to the models. This step is crucial for creating accurate and secure prediction systems. DL models learn hierarchically from the data, extracting insights and knowledge. For instance, a DL model trained with face data detects edges at first, identifies shapes such as the nose and mouth, and finally extracts the larger facial structure. However, these models can be vulnerable to attacks that mislead the output to the attacker's target. We can integrate backdoor and adversarial detection techniques to protect the output of models from attacks like adversarial and backdoor attacks. Additionally, we can include access control mechanisms that allow only authorized users to access the models to prevent attacks from within the training dataset. By doing so, we can ensure the models are protected from misleading output.

After deploying the model, it is crucial to continuously monitor and detect any anomalies that could result in potential attacks. However, deep learning models are considered black-box, meaning different tools, such as Local Interpretable Model-Agnostic Explanation (LIME), are used to explain the model's decision. It is also essential to regularly test and validate the model's performance. Updating the model and its security to keep up with evolving threats and attacks is crucial to building intelligent cybersecurity systems. Asides from that, the threat model plays a crucial role in defending the systems against attacks by identifying potential or real vulnerabilities, putting countermeasures and control in place to prevent those vulnerabilities from not being exposed, and imposing destruction. The detailed description of the threat model is discussed in the subsequent section III-B1.

1) THREAT MODEL

A threat model is a tool to examine the adversary model. An adversary model is a specimen of the attackers in the system. Depending on the goal of the attacker, the specimen is created. A specimen can be a simple algorithm or series of statements based on the purpose and capabilities. Based on the threat model, we explore the adversary model in terms of an attack category (e.g., backdoor generation), attack/defence surface (e.g., entry points), defence target and attacker and defender capability (e.g., abilities), goals (e.g., target) and assumptions (e.g., environment) to inject the attacks. As opposed to, the defender can utilize threat model to explore the vulnerabilities and defence the application. In Figure 4, we illustrate a threat model that is used to analyze the potential security risk and vulnerabilities. An attacker can generate the attack and customize it as per the application.

We first analyze the attacker and defender control over the four attack surfaces. The details are summarized in Table 2 and described the attack surface in the following section III-B10. In the subsequent section, we describe the attacker and defender threat model. We model the attack and defence into three parties. A Victim User (VU) who wants to train the DL model by collecting the dataset from the internet, outsourcing the job of training of DL model to a third party or downloading a pre-trained model from the open-source repository to adapt to her task using a transfer learning. An attacker whose goal is to corrupt the DL model by considering capabilities and assumptions, and the defender's goal is to prevent otherwise.

Goals: The attacker's goal is to poison the DL model and return the poison model $F_{\Theta adv}$ which is equal to the clean model $F_{\Theta c}$. However, while generating the $F_{\Theta adv}$, the model attacker considers two goals in mind. First, the accuracy of the return poison model $F_{\Theta adv}$ should not drop on the validation dataset. Second, for the inputs that contain the triggers, the model $F_{\Theta adv}$ output should be different from the clean model $F_{\Theta c}$ output. Formally, let I is a function I: R^N -> {0, 1} that map any input (X in R^N) to binary output. However, in the presence of the trigger (t), the (x) is 1 and 0 otherwise. C is another function C: R^N -> {1, M} that maps the input to a class label (Y). Let's consider (G) is an attacker image generator function Gt: X -> X based on some triggers (t) stamps on the image. (O) is the output function that shifts attacker-specific labels in the presence of trigger O: Y -> Y. The attacker needs to consider some risks while making the attack successful.

Risk 1: In the presence of a backdoor trigger, the infected model successfully achieves the goal. For example, we can say that for all x: $I(x) = 1$, arg max $F_{\Theta adv}(x) = C(x)$ not equal to $F_{\Theta}(x)$ in the presence of a backdoor, the output should not be equal to the true output.

Risk 2: In the absence of a backdoor trigger, the model should correctly predict the expected output. For example, for all x: $I(x) = 0$, arg max $F_{\Theta c}(x) = C(x)$.

Risk 3: Whether the poison sample is detectable by humans or machines. For example, D is detectable function and $x^{'} = G(x)$ so $D(x^{'}) = 1$ if an only if the t is detected.

The defender's goal is to identify and mitigate the backdoor triggers at inference time to avoid being attacked. The defender's purpose can fall into three categories 1) detection, 2) identification, and 3) mitigation. In attacks detection, a binary decision is made whether or not the given DL model has been infected. Identification, identify the triggers. Mitigation makes the triggers ineffective.

Capabilities: We assume that the attacker has a control of the training set, training schedule, and model parameters according to the target surface. However, the attacker

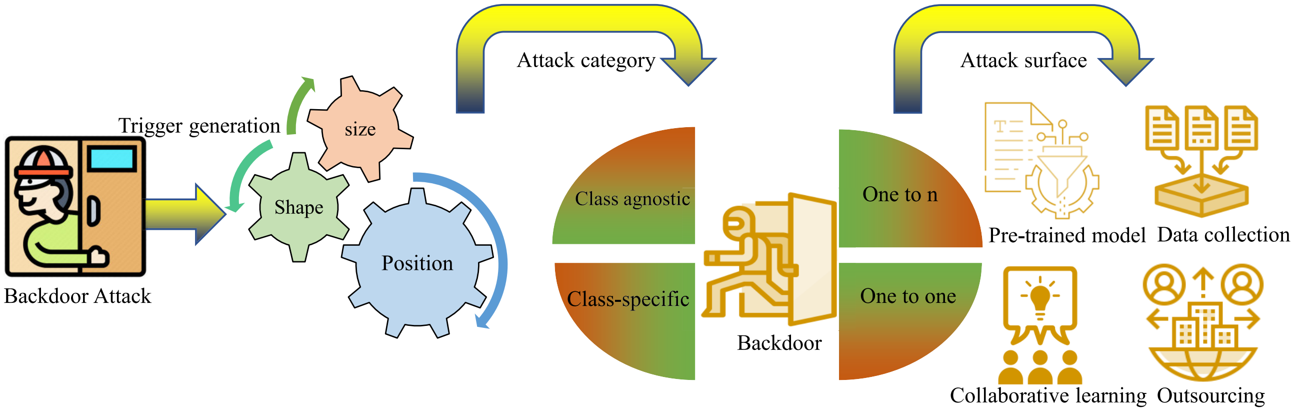


FIGURE 5. Categorization of Attack based on backdoor specimen analysis and targeted pipeline.

has no control over the inference pipeline. For defender's, we assume that the defender has full control of the inference pipeline based on the target surface. The details are listed in Table 2.

Assumptions: Facing up backdoor DL attacks is an ongoing and constantly evolving challenge.

Backdoor assumptions are mandatory to prevent backdoor access points. In particular, backdoor attacks pose a significant threat to the reliability of DL model predictions. These assumptions must consider what causes the violation of integrity, availability, and access control of these DL models. The assumptions are the following: 1) Adding a backdoor does not affect the model performance, 2) the model will behave correctly in the inactivation of the backdoor, and 3) backdoor does not cause false positives to the model. Meanwhile, to prevent security violations at a minimum, organizations should carefully evaluate and monitor the pipeline of DL models. For example, organizations should monitor the data and label drifting, identify the signs of tampering or manipulating data, and implement robust security controls to protect their models from malicious backdoor attacks. Moreover, organizations must consider secure methods for training and deploying their DL models to ensure they are trustworthy and secure in safety and critical applications.

Security Analysis: We perform a security analysis as a defender of DL models to protect the system by identifying the security goal and threat model. A security goal is a requirement that, if violated, can lead the system into a compromised state. A threat model is a profile of the attacker or defender that describes goals, motivation, and capabilities. In the context of the DL image classification model, it aims to classify the images correctly. The power of the model is measured in terms of True positive (TP), True negative (TN), False positive (FP), and False negative (FN). The attacker aims to increase the FP and FN to enter the system. In contrast, defenders prevent FP and FN. In the context of the security goal, the defender's purpose is to identify malicious activities and prevent them from flipping the model's output. We classify the security goal into three categories:

Integrity: To prevent the attacker from flipping the output.

Availability: To prevent the attacker from interfering with the normal training schedule, training set, and model parameters.

Access Control: To prevent the insider attacker from accessing the sensitive information.

There is a connection between false negatives and the violation of the integrity goal. The poison instances that pass through the classifier can create destruction. Likewise, a false positive is connected with the availability as the classifier in the presence of the poison instance denies being true.

2) ATTACK CATEGORY

In this section, we discuss the attack category to generate a backdoor specimen. The specimen can be a simple algorithm based on the attacker's goal and capabilities. However, in terms of the backdoor, the attacker can generate the specimen based on several attributes and methods. For example, generating a trigger for an image or feature space is the method of the specimen. Conversely class-specific or agnostic, one-to-one (single triggers to the same label) or one-to-N (single trigger to multiple labels), size, position, and shape of triggers are the properties of the specimen.

3) BACKDOOR COMPOSITION

An attacker can compose a backdoor attack by selecting the methods (M) and properties (P) as mentioned in Figure 5. An attacker can generate the specimen by choosing the method: Image/feature space, trigger, and their associated properties. For example, in the case of a traffic sign detection application, the attack generates the specimen by selecting the trigger invariant to size, shape and position, and image space with class agnostic property [3].

4) IMAGE/FEATURE SPACE (M1)

Image space represents visual data. In image space, the attacker stamps small sticker shapes (e.g., 2 × 2 square, flower) that lead to a specific pattern during training. The feature space defines the range of possible values for each feature and guides the design and selection of features for a particular problem. In feature space, the attacker performs some transformation by using an optimization method that leads to a particular pattern in the feature space.

5) CLASS-SPECIFIC AND CLASS AGNOSTIC (P1)

A backdoor attack holds the targeted attack property. The one input is misclassified to the attacker's chosen targeted class. Attack under this category is divided into two parts 1) class-specific and 2) class-agnostic. In class-specific specimens, the attacker can pick the input of a specific class, stamp the trigger and misclassify to the target class. Whereas in class-agnostic attacker can stamp the trigger to any class input, it will misclassify to the targeted class.

6) MULTIPLE TRIGGERS TO MULTIPLE LABELS (P2)

Multiple triggers (e.g., many-to-many attack) are stamped to different input classes, and each trigger targets another class label (an attacker decides the targeted label collection). This attack activates in the presence of any trigger at inference time and classifies to the attacker's chosen targeted label collection.

7) MULTIPLE TRIGGERS TO SAME LABEL (P3)

Multiple triggers are stamped to different input classes, and each trigger targets only one class label. The attack activates in the presence of any trigger at inference time, and is classified according to to the same targeted label.

8) MODEL WEIGHTS OR PARAMETERS (P4):

In this case, the attacker can disrupt the models by embedding the triggers without direct access to the training data and modifying the parameters or weights of DL models.

9) TRIGGER (M2)

In the computer vision domain, almost every backdoor specimen generates by considering trigger transparency with its additional characteristics, size (P1), shape (P2), and position (P3). Earlier work on the backdoor considers the physical specimen (e.g., shape stickers), and later work considers the digital (e.g., pixel perturbation). The additional characteristics may not apply to other domains like audio and text. Triggers are the core of the backdoor attack. It can be better designed and generated at the optimization level (P4) to achieve better performance.

10) ATTACK AND DEFENCE SURFACE PIPELINE

This section discusses the attack surface pipeline that becomes an attacker's Entry Point (EP) to disrupt the DL models.

11) DATA COLLECTION (EP1)

The data we obtain for training DL models is crucial, as the data quality and quantity directly impact the model's working. However, data collection is usually error-prone as users use big datasets from the internet to collect the data. For example, popular and publicly available datasets only rely on volunteer contributions, such as ImageNet [16] and MNIST [17]. If the user collects data from multiple sources over the internet, the collected data may be infected. An attacker can generate the poison dataset and leave it on the web for the victim to use and download for training and testing models. The model becomes infected when a victim uses poison data to train or test the models. Clean-label poisoning attacks [18], CGAN attacks [24], poison frog attacks [25], image-scaling attacks [57] are examples of this attack surface. The labels are consistent with the data. Therefore, making them easy to pass the visual inspection.

12) PRETRAINED (EP2)

Transfer learning is a concept where a pretrained model is used as a starting point to train a model on a new task. In short, knowledge gained from one task solves a different but related problem. This process reduces the computational overhead. Furthermore, the models can be readily available on open-source repositories such as GitHub and model zoo. For example, an attacker can inject the poison dataset, train the model for the face recognition task, and place this model on publically available repositories. Latent backdoor attacks [19] and backdoor attacks against transfer learning [20] are examples of pre-trained surface attacks.

13) OUTSOURCING (EP3)

The backdoor arises when users outsource the model training to machine learning as service (MLaaS) platforms due to a shortage of computational resources. For example, the user can define the model architecture and provide the training data to the MLaaS provider. However, the control is over the provider, and during the training phase, backdoor injections can be injected without the user's notice. For example, a client can outsource the face recognition task training procedure to a compromised cloud. The compromised cloud can poison the training images with a targeted false label and offers the client a trained network that contains a backdoor. As a result, any individual image that includes the backdoor trigger (i.e., a small picture in the bottom-left corner of the face image) can imitate another certified individual [39].

14) COLLABORATIVE LEARNING (EP4)

Collaborative learning is designed to protect the data privacy leakage owned by the clients. The server cannot control the participants' training data during the learning phase. Once the model training is completed offline, trained model weights will be uploaded to the server. However, collaborative learning is also vulnerable to backdoor attacks. A collaborative model can easily be a backdoor when a few participants are compromised or attacked. Some data encryption models, such as CryptoNN [21] and SecureML [22], train the model over the encrypted data to ensure data privacy under the attacker's target. In particular, in joint collaborative learning, the data is

TABLE 2. Analysis of the capabilities of attacker and defender corresponding to the attack surface.

Role	**Attacker**				**Defender**			
Attack surface ↓ *Capability* →	Training set	Training Schedule	Model parameters	Inference pipeline	Training set	Training Schedule	Model parameters	Inference pipeline
Data collection	✓	✓	✗	✗	✓	✓	✓	✓
Pre-trained models	✓	✓	✓	✗	❏	❏	❏	✓
Outsourcing	✓	✓	✓	✗	✓	✗	✓	✓
Collaborative learning	✓	✓	✓	✗	✓	✓	✓	✓

✓: Access ✗: Not access and ❏: Partial access. The defender has partial control while adopting the pre-trained model.

contributed by various clients, though encrypted to preserve privacy, making it challenging to ensure whether the data is benign or otherwise.

IV. TAXONOMY

In this section, we present the taxonomy by categorizing the attacks on DL models along two axes, as illustrated in Figure 6. The first axis demonstrates the type of security violations the attacker causes. For example, poison instances cause harm by passing through the DNN and are classified as false negatives (violation of Integrity). Likewise, injecting the poison instances stamp with triggers (violation of access control), the classifier gets confused in the presence of a trigger, fails to discriminate between benign instances, and is classified as false positive (violation of availability). The second axis relates to the OES, which describes the specificity and capability of the attacker. Specificity means that the attacker wants to generate targeted training stage attacks by selecting the different methods and properties of Backdoor attacks (i.e., outcome). Capability indicates the environment (e.g., black box, white box, and grey box), surface (i.e., attack entry points) to inject the Backdoor triggers.

Based on our proposed taxonomy, we provide hypothetically targeted training stage attack scenarios for image classification models. The attacks are divided into four categories, particularly attack surfaces. First, the attacker needs to follow OES (Outcome Environment Surface) model to generate Backdoor triggers. The outcome details are described in section III-B2. For example, let's say the attacker generates the triggers for the image space method (M1), and the property is class-specific (P1), where the trigger position is in the bottom right corner. The size is a bunch of pixel patterns decided by the attacker based on the trigger method (M2) and properties size (P1), shape (P2), and position (P3). The examples of generating backdoor triggers are illustrated in Figure 7. Finally, the environment represents the capability of the attacker to inject the triggers into the system. If the attacker has the least knowledge, the environment is considered a black box, most knowledge a white box, and some are considered grey.

A. BACKDOOR ATTACKS

We explain the formulation of the backdoor specimen by understanding the methods and properties of the backdoor triggers (see section III-B2). Further, we proposed the taxonomy to analyze the existing backdoor attacks for image classification systems (see the section IV). Afterward, we categorize the existing backdoor attacks based on the attack surface pipeline (see section III-B10) in detail. Table 3 illustrates the qualitative analysis of the backdoor attacks based on the attack surface. Table 4 provides the summary of the attacker's capabilities as per the attack surface.

1) TARGETED DATA COLLECTION ATTACK

We describe the attacker's scenario, environment, and capabilities while providing the studies' details. We discuss clean-label and poison-label invisible attacks in the context of feature-space attacks.

Case: Attacker wants to generate the stealthy poison image without controlling the labeling process to evade human inspection. There is no control over the dataset. However, in the execution of the attack, the attacker has the least or can be fully knowledgeable of the target model.

Environment: Attacker has no access (Black-box) to the dataset.

Capabilities: Attacker has the least control (grey box), cannot manipulate the training process, and cannot access the model at inference time. In some cases, the attacker has full knowledge (white box) of the model.

Violations: Availability and access control.

These attacks are clean-label attacks where the attacker has no control over the label of the dataset. The attacker only tempered the image at the pixel level, which still looks benign. For example, an attacker could add a benign sample (perceptually similar) without altering the sample's label and inject it into a training set for a face recognition model. Once the model is trained, the attacker can control the identity of a chosen person at test time (security violation of the availability). Additionally, based on the attacker's capability, attacker can craft the tempered samples and leave them on the web, waiting for the data collection bot to collect them, thus entering into the training set.

The authors [25] proposed the attack for the transfer learning scenario, where only one sample is enough to achieve a higher success rate which is 100%. Thus, crafting the attack in the feature collisions for transfer learning settings is comparatively easy as it is in end-to-end training settings. An optimization-based method has been used to construct the poison samples. At the same time, the authors were making the poison samples and added small perturbations to the base images to ensure that the base image feature representation lies near the target class. The attack's success depends on

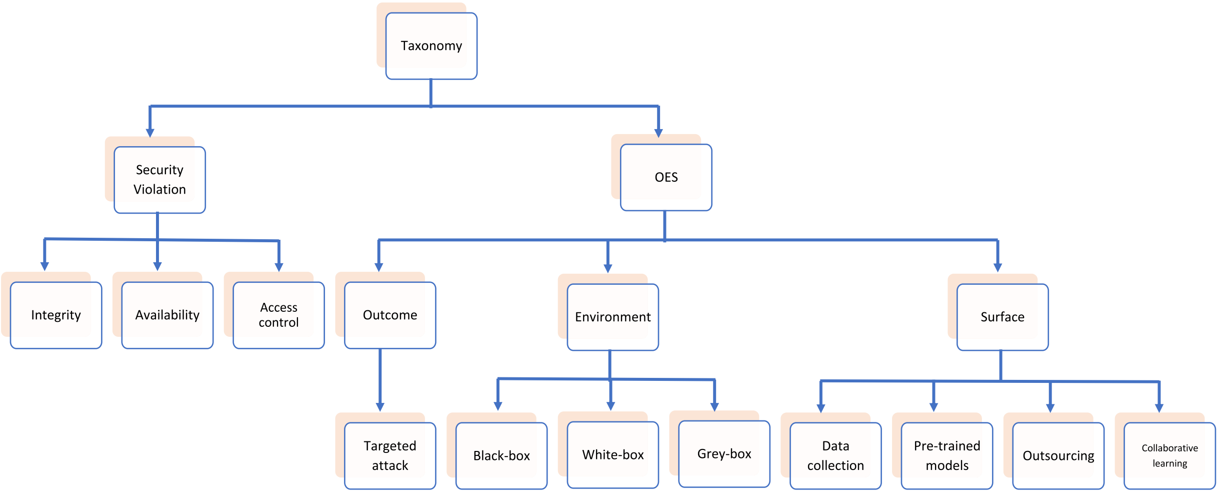


FIGURE 6. Categorization of Attack and defence based on backdoor specimen analysis and targeted pipeline.

TABLE 3. Qualitative comparison of existing backdoor attacks.

Attack surface	Backdoor Attacks	Environment	Scenarios	Method	Corrupted Labels	Poison Dataset
Targeted data collection	[25], [44]	Grey-box	Transfer learning, End-to-end	Feature collision	✗	✓
	[26], [27], [28]	Grey-box	Transfer learning, End-to-end	Optimization	✗	✓
	[18], [24]	Grey-box	Transfer learning, End-to-end	GAN, CGAN interpolation	✗	✓
	[23]	Grey-box	Transfer learning, End-to-end	Pattern-instance-key, Input-instance-key	✓	✓
	[14]	Grey-box	End-to-end	Pixel inversion	✓	✓
	[33]	Grey-box	Transfer learning, End-to-end	Pattern static perturbation, targeted adaptive	✓	✓
	[35], [36], [42]	Grey-box	Transfer learning, End-to-end	Pixel-wise perturbation	✓	✓
	[37]	Grey-box	End-to-end	Style transfer	✓	✓
Pretrained	[43]	Grey-box	Transfer learning	Object stamping	✓	✓
	[19]	Grey-box	Transfer learning	Latent technique	✗	✓
	[20]	Grey-box	Transfer learning	Optimization		
Outsourcing	[40]	White-box	Transfer learning	Composition technique	✓	✓
	[3], [41]	Grey-box	Transfer learning, End-to-end	Pixel perturbation	✓	✓
	[44]	Grey-box	Transfer learning	Optimization	✗	✓
	[45]	Grey-box	Transfer learning	Object stamping	✓	✓
Collaborative learning	[58]	White-box	End-to-end	Pixel perturbation	✓	✓
	[59]	White-box	End-to-end	Byzabtine-resilient aggregation strategies	✗	✗
	[60]	White-box	Transfer learning	Optimization	✓	✓
	[61]	White-box	Transfer learning	Optimization	✗	✗
	[62]	White-box	End-to-end	Graph topologies	✗	✗

providing the images containing the trigger at test time, thus consistently misclassified to the target image. The attack is executed in a white-box scenario, which makes it less practical in real-time.

After that, a series of research was dedicated to the research of clean-label attacks. This research, [28] inspired by the work proposed in [25]. However, the difference is that the attacker can present the trigger at any random location in unseen images to misclassify the source instance to the target instance at inference time. Whereas in the research, [25], the model is fooled only when the attacker presents the particular set of images at inference time. The authors generated the clean-label attack by optimizing the poison image in pixel space and ensuring that the source class and a patch trigger are always close to the target class in their feature space. In addition, the patched source images have been generated by providing a source image, a trigger patch 'p', and a binary mask one. This mask becomes zero on the non-trigger place. The execution environment black box makes it practical for real-time security threats.

Following the setting of the authors in [28], Instead of feature collision, convex polytope proposed by the authors of [26], and bi-level optimization proposed by the authors of [27] had exploited to generate poison instances. These attacks

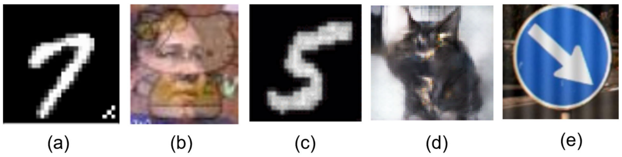


FIGURE 7. An example of constructing Backdoor triggers in the image and feature space. (a) A digit classification system is poison to have backdoor trigger patterns on the bottom right corner of the image [3] and (b) a face classification system is poison to have images that are blended with the hello kitty image [23] are image space attacks with different goals and triggers. Whereas (c), (d), and (e) proposed to feature space invisible triggers for Digit classification, cat-dog classification, and traffic sign classification [4], [14], [24].

were executed in a black-box environment and improved the attack success rate.

In [26], the authors proposed a different style of clean-label targeted poisoning attacks via feature collision. The feature vectors corresponding to poison examples are the vertices of a convex polytope containing the target's feature. These attacks anticipate that the whole region inside the convex polytope will be classified as the base class, resulting in better attack reliability than a simple feature collision attack. The authors performed the experiments for end-to-end and transfer learning scenarios by considering the weak black-box assumptions. The result shows that this attack does not require any modification of targeted instances at inference time in contrast to existing backdoor attacks. However, the attack success rate is over 50%, with 1% of the poison training set.

In [27], the authors proposed an optimization framework for generating two imperceptible variants of backdoor attacks: steganography and regularization. Both attacks are based on a bi-level optimization problem. The outer optimization focuses on minimizing the loss risk, and the inner optimization seeks to optimize the retraining of the pre-trained model to memorize the backdoor. In addition, while generating the steganography attack, the Least Significant Bit (LSB) algorithm embeds the triggers into the poisoning training set. Whereas, for regularization attacks, Lp-norm regularization is used to make the small perturbations as a trigger with the extra focus on keeping the shape and size invisible. During this crafting of triggers process, it is also assumed that the attacker only knows the dataset for the steganography attack.

The authors of [18] proposed two methods to generate poison images using GAN-based interpolation and adversarial perturbations. These methods make the model harder to classify to the ground truth label. Since the poison images were harder to learn, a model created a strong association between the trigger to the targeted label. The interpolation method poisoned the image towards the source class in the latent space, while these images were visually consistent with its label. In the perturbation method, first, the authors perturbed the input image and then added the invisible trigger to generate a poison image. The attacker needs complete knowledge of the model and training procedure as well.

In [24] proposed an invisible backdoor attack by using cGAN. To generate potential poisoned examples for a digit and animal classification model, the authors applied the analysis-by-synthesis method with cGAN. The underlying assumption is that the latent space of cGAN is somewhat smooth, and thus the intersection of two class "subspaces" may produce ambiguous samples for classification models. The proposed method achieves a high success rate with a very low injection rate.

In [29], the authors slightly changed how to generate the triggers for label-consistent attacks. They only stamp the triggers to the target class - the model can learn the association between the trigger and the target class. A ramp signal has been used to inject noise for MNIST and sinusoidal signals for traffic sign datasets. During training, the attacker only needs to corrupt the sample fraction in the target class. At test time, the network recognizes the input containing the backdoor signals as the attacker's target class. Further, in [29] evaluated the attack on the MNIST digits classifier and traffic signs classifier under weak assumptions without knowing the deep learning model with the attack success rate above 90%.

In [4], the authors proposed a backdoor attack inspired by a natural phenomenon of 'reflection' for end-to-end training scenarios. The attack has been generated by developing various reflection patterns as 'triggers' for the poison dataset. Later, this poison dataset was injected (violation of availability) with clean images and considered first class as an attacker's target class during training. Moreover, the attack's effectiveness has been evaluated based on three classification tasks: face detection, traffic sign, and object detection with five different datasets. The findings showed the effectiveness of the Refool attack outperformed existing attacks on various datasets with a range between 75.16 %-91.67% attack success rate. However, this attack's overhead relies on corrupting the more significant fraction of training samples.

In study [30], the authors proposed the invisible triggers by strategically exploiting the order of the training data in which it is presented to the model. An attacker can successfully manipulate the model's learning process under the black-box setting with no change in model architecture and original dataset. In another study [31] the authors noted that the triggers on images have not worked well for videos, so they have proposed specialized backdoor triggers for video recognition tasks. The authors in study [32] proposed a Backdoor attack in a lithographic hotspot detection system in the light of a malicious insider (violation of access control). An insider attacker can cause the targeted DNN misbehavior by data poisoning targeted inputs. The targeted inputs are the secret trigger of a metal polygon with some non-hotspot clips without corrupting labels. The experimental results of this proposed methodology reveal that an attacker can robustly force a targeted misclassification with only 4% of the poison dataset with a 97% attack success rate. The authors in [44] proposed a BlackCard backdoor poisoning attack inspired by poison frog [25].

However, the authors hold three points based on the optimization-based method while crafting the poison instance in the feature collision. 1) ensure the poison instance X appears like the Base class instance b to a human labeler 2) maximize the probability of predicting x as its base class label b in attacking model T 3) avoid the collision between feature space representation of input x and the base class instance b as much possible. Doing this allows misclassifying poison label x to base instance b, not because of its feature representation but because of its collision.

The crafted poison instance X was injected into the targeted model at test time. This injected X always misclassified to based instance b under three practical weak black-box assumptions knowledge oblivious, clean-label, and clean test label. In addition, they also experimented on a variety of classification datasets wan an attack success rate ratio from 98 to 100%.

Case: Attacker wants to generate the stealthy poison image by controlling the labeling process. There is some control over the dataset. However, in the execution of the attack attacker has the least knowledge of the target model.

Environment: Attacker has minimum knowledge and no access (Black box) to the training models.

Capabilities: Attacker has the least control (grey box) and cannot manipulate the training process, and has no access to the model at inference time. In some cases, the attacker has full knowledge (white box) of the model.

Violations: Integrity, availability and access control.

In this study [23], the authors put forth the concept of invisibility requirements in the creation of backdoor triggers. Their objective was to develop poison images that could evade detection from human visual inspection by appearing identical to benign images. The study proposed two methods of data poisoning, namely input-instance-key and pattern-instance-key. The generated backdoor triggers were designed to be injected into a learning-based facial recognition authentication system. In developing the input-instance-key attack, random noise was added to the images, while the pattern-key attacks utilized a blended accessory injection strategy. The authors compromised the integrity and availability of the facial recognition system. Notably, their attacks were effective under weak assumptions, such as the absence of prior knowledge concerning model architecture, training dataset, and training parameters, with an attack success rate exceeding 90%. Subsequently, there were further studies on invisible triggers with poison labels.

Further, in [14], the authors proposed a Pixdoor backdoor attack by flipping the pixels of the images at pixel space and generating the poison samples. Later, the poison samples have added to the source class, shifted the labels to the target class, and injected during the training process (violation of availability). However, the authors executed the attack under a black box environment with a low sample injection rate of 3%.

In [33], the authors proposed two attack strategies, pattern static and targeted adaptive, for generating perturbation masks as backdoor attacks by poisoning the training dataset. The pattern static perturbation mask is generated by replacing the pixel intensity value with ten within the (2 × 2) subregion of the top left corner of the image. The targeted adaptive perturbation mask has been generated using the DeepFool algorithm proposed by [34] and computed the adaptive perturbations for targeted misclassification where the same perturbation mask is associated with the same class labels. In addition, they minimize the l^2 norm of the perturbation to ensure the invisibility of the trigger.

The authors in [35], and [36] generated the invisible patterns in the frequency domain, and this kind of attack can also bypass existing defences. In [37], the authors proposed an invisible backdoor attack in feature space via style transfer where features manifest themselves differently for every different image at a pixel level. The underlying assumption was that the attacker has white box knowledge of the model (violation of availability), dataset (Violation of integrity and access control), and training process (violation of availability). An attacker can choose any target label. When the attacker wanted to launch the attack, inputs passed through the trigger generator to implant the uninterruptible feature trigger, which causes the model to be mispredicted (violation of availability) at run time.

In study [38], the authors proposed a different technique to generate the sample-specific poison samples by adopting image stenography. Experiment results demonstrated that this technique could bypass many existing backdoor defence methods. In [42], the authors proposed a backdoor Hidden Facial Feature (BHF2) attack for face recognition systems. The invisible backdoors can embed into a human inherent facial features, eyebrows and beard. The generation of attack under the weak black-box assumptions. First, the face key features are extracted and marked as numbers. They calculated the deflection angle and length for eyebrows and mouth features. Based on the angle and length information, semi-arc and semi-ellipse masks are used, and pixel values of the points in these masks are changed, respectively. Then, the labels of the backdoor instances changed to target labels.

Summary: Although invisible triggers are used in these kinds of attacks to generate the poison images and associate them with the poison labels. Nevertheless, this process makes it detectable by examining the image label relationship of training samples. Considering the poison label issue, clean-label is an active research area to generate backdoor attacks. Yet, these clean-label attacks usually suffered a low attack success rate compared to the poison-label invisible attacks. Most recent studies demonstrate the techniques to achieve a high attack success rate with a low injection rate for clean-label invisible attacks. However, balancing clean labels with effectiveness and stealthiness is still an open question and worth requires further exploration. Data ordering attack is a stealthy way to induce backdoor attacks. This kind of attack emphasizes the importance of robust training procedures and the need for defensive measures.

2) TARGETED PRE-TRAINED MODELS ATTACKS

Case: A pre-trained model is a model that is trained on a large-scale dataset for the image classification task. The pre-trained model can be easily downloaded from a third-party or open-source repository. Users can download these models and use the pretrained model as is or use transfer learning to customize this model to a given task.

Environment: Attacker has access to the model and training dataset.

Capabilities: Attacker has full knowledge and control of the training process and model (white box). An attacker can train a poison model and leave the model to download by the victims. Once the victim attacker downloads, the model has no control over it.

Violations: Availability, Integrity and access control.

The authors in [43] proposed a physical backdoor attack by poisoning the dataset for a transfer learning scenario. The poisoning triggers were constructed by considering everyday physical objects like dots, sunglasses, tattoos filled-in, white tape, bandana, and earrings. These poison triggers were injected with the benign dataset based on the black-box assumptions during training. Further, the authors empirically studied the effectiveness of proposed physical attacks against two evaluation metrics: accuracy, attack success rate, and four state-of-art defence solutions. During the experiment, it has been observed that the trigger earing attack success rate was less than the other triggers. In the experiment, failure reason was also investigated based on three factors trigger size, content, and location, with the help of a class activation map (CAM). Investigation results show that off-face triggers, regardless of size, are unlikely to affect the classification results. Whereas, with the other triggers, the attack success rate is above 98% with a 15-25% injection rate.

Further, in study [19], the attacker generated the attack by training a Teacher model on the poison dataset and classifying it into a target class. Before deploying the model to a public repository, the attacker removed the backdoor trace by eliminating the target class output layer and replaced it with the clean output layer. Therefore, when the victim downloads the corrupted model and fine-tunes the last two layers of the model, this backdoor is activated automatically if the targeted class exists at inference time. In [20], the authors generated the targeted backdoor attacks for transfer learning scenarios on both images and time-series data with the motivation to defeat pruning-based, fine-tuning/retraining-based, and input pre-processing-based defences. The attack was generated by using three optimization strategies: 1) ranking-based neuron selection method, 2) Auto-encoder power trigger generation, and 3) defence-aware retraining to generate the manipulated model using reverse-engineered model inputs. Further, the proposed attack was evaluated based on white-box and black-box assumptions based on Magnetic Resonance Imaging (MRI) and Electrocardiography (ECG) classification. The proposed attack success rate is 27.9% to 100 and 27.1% to 56.1% for images and time-series data.

Summary: Pre-trained deep learning models have already been trained on a large dataset, and these models have learned a significant amount of information about the features and patterns in the training data. In addition, these models are made publicly available for further fine-tuning on a specific task. Therefore, pre-trained backdoor attacks have a broad spectrum of victims, as using these models for down-streaming tasks is a norm. However, the attacker cannot control the users' further downstream tasks. It is worth noting that the attackers can assume the specific knowledge of the downstream task as this dataset can be collected from public repositories.

3) TARGETED OUTSOURCING ATTACKS

In this section, we discuss the outsourcing attack scenarios involving a third-party platform outsourcing data and getting the untrusted trained DL models. Further, we discussed the earlier methods of generating a backdoor in image space by stamping some patterns and associating them with poison labels to disrupt the models. We also discussed end-to-end training attacks as well.

Case: Due to the cost and expensive computation, many industries outsource the training process of machine learning models to third-party cloud service providers, known as ML-as-a-Service (MLaaS). MLaaS allows the attacker to control the training or model of the victim and return the poison model.

Environment: Attacker has access to the model and training dataset.

Capabilities: Attacker has full knowledge and control of the training process and model.

Violations: Availability, integrity and access control.

The origination of the backdoor attacks started in 2017 when the authors of [3] proposed a BadNet method by poisoning some training samples for DL models. The attacker can act as a third party and access the training dataset or model parameters to inject backdoor triggers. The most common strategy of these attacks is 1) generating some poison samples by stamping some triggers on the sub-set of images and associating them to the targeted label ($x^{'}$,y^{t}), 2) releasing the poisoned training set containing both poison and benign samples to the victim users for training their model. During the end-to-end training of the model, Inject these samples combined with the benign samples where the model learns the association of the trigger to the targeted class. 3) directly update the parameters or weights of DNN models to embed the backdoor triggers. However, the attacker must ensure the model accuracy does not degrade on validation samples and perform correctly without trigger at inference time. The initial backdoor attacker was the representative of visible triggers. Later, a lot of work starts on the invisibility of the triggers with clean and poison labels, which is already discussed in the section IV-A1.

In [40], the authors proposed a backdoor attack based on the composite properties named a composite backdoor attack. The proposed attack method used existing image features as a trigger. For example, a trigger has been generated by combining two image faces of the same person (artificial feature with the original feature) so that it does not require any specific face; further, by selecting two different pairs of mixed samples with different labels considered a target label. For experiments, the effectiveness of the proposed attack has been evaluated on different image and text classification problems such as object recognition, traffic sign recognition, face recognition, topic classification, and three object detection tasks with an 86.3% attack success rate under strong white box assumptions.

In a study [41], the authors expanded the Badnet attacks to include multiple targets and multiple triggers of backdoor attacks. They introduced one-to-N attacks, where a single trigger could affect multiple labels by adjusting the pixel intensity of the trigger. On the other hand, in an N-to-one attack, all triggers must be launched to activate the trigger. The authors utilized MNIST and CIFAR-10 samples to produce poison instances for a One-to-N attack. In the case of MNIST, a four-pixel strip (1 × 28) was used with + and - color intensity, while CIFAR-10 utilized a 6 × 6 square on the lower right corner of the image with + and - color intensity. The authors modified the labels of the same backdoor with varying intensities to become a targeted class, which was combined with benign images to train the model without affecting its accuracy. The authors used the same strategy to generate poison instances for N-to-One but added the trigger count (N=4) on all image corners. The label of N different backdoors was the same as one target class, t.

Further, in study [44] proposed a model agnostic TrojanNet backdoor attack by injecting the TrojanNet into DNN models without accessing the training data. The attack performs well under a training-free mechanism where the attacker does not need to change the original target model parameters, so retraining the target model is unnecessary. The design of triggers is a pattern similar to a QR code. A QR code type of two-dimensional array [0-1] coding pattern with exponential growth by increasing the pixel numbers. Triggers size 4 × 4 have been selected with 4368 combinations as a final trigger pattern to inject into the DNN model. The training dataset for TrojanNet consists of two parts, 4368 trigger patterns, and various noisy inputs. These noisy inputs can be other than the selected combination of trigger patterns or random patch images from ImageNet. Denoising training involves the injection of noisy input and triggers during the training process. The goal is to keep TrojanNet silent for noisy inputs. This improves the trigger recognizer's accuracy, reducing the false-positive attack.

Moreover, as the output of TrojanNet will be all-zero vectors, this substantially reduces gradient flow toward the trojan neurons. This process prevents TrojanNet from being detected by existing defence solutions. The curriculum learning approach is used in the training process to benefit the model's training. The authors finished training when TrojanNet achieved high accuracy for trigger patterns and kept silent for randomly selected noisy inputs.

Further, the injection of TrojanNet also consists of three parts 1) Adjusting the TrojanNet according to the number of trojans as the DNN model output dimensions are less than a few thousand, 2) combining the TrojanNet output with the model output, 3) combining the TrojanNet input with the model input. A merge-layer concept combines the model output with the Trojan output. The role of the merging layer is similar to a switch between the dominance of TrojanNet output and benign output. The authors also performed extensive experiments on the proposed attack on four applications: face recognition, traffic sign recognition, object classification, and speech recognition. Further, four evaluation metrics, attack accuracy, original model accuracy, deviation in model Accuracy, and infected label numbers, have been used to evaluate the performance of the proposed TrojanNet. The results of experiments demonstrate that this proposed attack can inject into any output class of the model. In closure, the proposed attack can easily fool existing defence solutions because the existing defence solutions usually do not explore the information from the hidden neurons in DNNs.

In [45], a new concept of using backdoor attacks as friendly backdoors was proposed. For instance, a backdoor can correctly be classified as friendly equipment but misclassified as enemy equipment in military situations. The proposed Friendnet backdoor attack works by poisoning the training datasets for the enemy and friendly models, respectively. The poison training instances are crafted by stamping a white square on the top left corner of the images associated with the targeted base class under the strong while-box assumptions: the friendly models trained on the small number of poison instances corresponding to the clean target class are appended with the benign training set. However, the enemy model trained on the small number of poison instances corresponding to the corrupted target class append with the benign training set. The experiment results show that the enemy model can misclassify the targeted instance at inference time with a 100% attack success rate by corrupting 10%, 25%, and 50% training sets, respectively.

Summary: Outsourcing attacks are quickly injected by exploiting the capabilities of DL models and algorithms. The user outsources the learning process to a machine learning service provider, and the attacker can intrude to compromise the system's security or steal sensitive information. Such attacks include poisoning, model inversion, and model extraction attacks. To prevent these attacks, it is essential to have strong security measures in place, including access control, data encryption, and model training process monitoring. Additionally, using trusted machine learning services and thoroughly evaluating third-party service providers' security is crucial.

4) TARGETED COLLABORATIVE LEARNING ATTACKS

Case: Federated learning (FL), also known as collaborative learning, is a technique that trains the DL models on multiple decentralized edge devices on a local device dataset without exchanging the data with the server to main the data privacy and integrity issues. The server collects the locally trained models and aggregates them to update a joint model until convergence. In this process, an attacker can act as a client; thus, the aggregate model can be backdoored.

Environment: Attacker has access (white box) to the dataset as the attacker can be one of the malicious clients in FL.

Capabilities: Attacker has control and can manipulate the training process. In some cases, the attacker has full knowledge (white box) of the model. However, an attacker cannot access the server aggregate model.

Violations: Availability and access control.

Model-backdoor attacks are significantly more powerful than targeted training data backdoor attacks. In study [58], the authors applied model-backdoor by replacing a benign model with the poison one into the joint model via optimization methods. The results show that the ASR is 100% even if a single client is malicious during a joint model update. However, the ASR decreases as the joint model continues to learn. The backdoor attack is challenging in FL due to data privacy in principle.

Further, in study [59], the authors explore the number of attack strategies to backdoor models, and byzantine-resilient aggregation strategies are not robust to these attacks. The defence against these attacks is challenging because secure aggregation of models is adopted to enhance privacy and defence solutions. For example, here in [47], when inverting the models to extract the training data, ultimately violates data privacy which is the core of adopting FL. The authors in [60] observe that if the defence is not present, then the performance of the backdoor attack only depends on the fraction of the backdoor and the complexity of the task. However, norm clipping and weak differential privacy can mitigate the backdoor attack without degrading the overall performance.

Moreover, the authors in [61] investigate a new method to inject backdoor attacks by using a multiple gradient descent algorithm with a Frank-Wolfe optimizer to find an optimal and self-balancing loss function. This achieves high accuracy on both main and backdoor tasks. This attack is named blind because the attacker cannot access training data, code, and the resulting model. The attackers promptly create poison training as the model trains and use multiple objective functions on main and backdoor tasks. The loss function always includes the backdoor loss to optimize the model for the backdoor and main tasks.

The author in [62] recently proposed a backdoor attack for peer-to-peer FL systems on different datasets and graph topologies. By studying the impact of backdoor attacks on various network topologies, they know that Erdose Renyi topologies are less resilient to backdoor attacks compared to slightly more complex graphs such as Wattz StrogatZ and Barabasi Albert. An attacker can amplify the backdoor attacks by crashing only a small number of nodes, such as four neighbors of each benign node, increasing the ASR from 34% to 41%. It further demonstrates that the defences for centralized FL schemes are infeasible in peer-to-peer FL settings. The attack is 49% effective under the most restrictive clipping defence and 100% under trimmed mean defences. However, their defence uses two clipping norms, one for peer update and one for local models, demonstrating effective results in detecting backdoor attacks in peer-to-peer FL settings.

Summary: We have observed that backdoor attacks are challenging on FL because data is decentralized and distributed among participants. Further, data privacy is the key principle of FL, where models are trained on multiple devices, and the updates are aggregated to create a final model. Despite the challenges, backdoor attacks on FL models still pose a significant challenge to the security and privacy of data being used to train these models. It is very challenging to counter these backdoor attacks on FL as for defender server is not even allowed to access the training or testing data to assist the defence.

V. BACKDOOR ATTACK DEFENCES

Neural networks are widely used in many safety and critical applications, such as face recognition, object detection, autonomous vehicle, etc. However, these models are vulnerable to various kinds of attacks. Therefore, there is a need for a defence to prevent these models from the attacked and make the model more robust in decision-making. We aim to analyze the existing defence solutions to know the intuitions of the backdoor and defender capabilities, their proposed techniques, and research gaps.

Detection-based methods, as mentioned in Figure 8, aim to identify the existing backdoor triggers in the given model or filter the poison samples from input data for retraining. These detection-based methods are explored from the model, dataset, or input-level perspective. Data-based defence approaches aim at the data collection phase, which detects whether training data has been poisoned. Model-based defence approaches targeted the model training phase to provide robust models against backdoor attacks.

A. MODEL LEVEL DEFENCE SOLUTIONS

In this section, we discuss the defences where the model can be evaluated in pre-deployment settings.

The authors of these studies evaluate the poison model offline, whereas the model is evaluated in pre-deployment settings. In [46], the authors studied the behavior of the backdoor attacks. The authors proposed a model-level defence solution to access the vulnerabilities of pre-trained deep learning models. Neural Cleanse (NC) is proposed based on the fundamental property of the Backdoor trigger. The property is that the backdoor triggers create "shortcuts" from within the region of the multi-dimensional space belonging to the victim label into the region belonging to the

TABLE 4. Summary of the attacker capabilities as per attack surface.

Attack surface	Case	Environment	Capabilities
Targeted data collection	No control over label process least knowledge of model	No access to the dataset	Least control of training process No or partial access of model
	Controlling label process Least knowledge of model	Least access of dataset	Least control of training process No access of model at inference In some cases full access of model
Pretrained	Open-source models for fine-tuning	Model access Training dataset access	Full knowledge of training process control the training process and model
Outsourcing	Outsource training to ML-as-a-Service	Model access Training dataset access	Full knowledge of training process Control the training process and model
Collaborative learning	Model trains locally on edge devices No sharing of data to servers Aggregates the models to server	Access of dataset Poison client models	Control and manipulate training process Full knowledge of model Cannot access server aggregated model

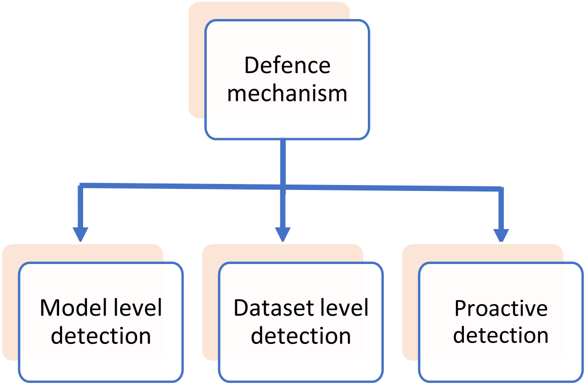


FIGURE 8. Defence of Backdoor attacks at different levels.

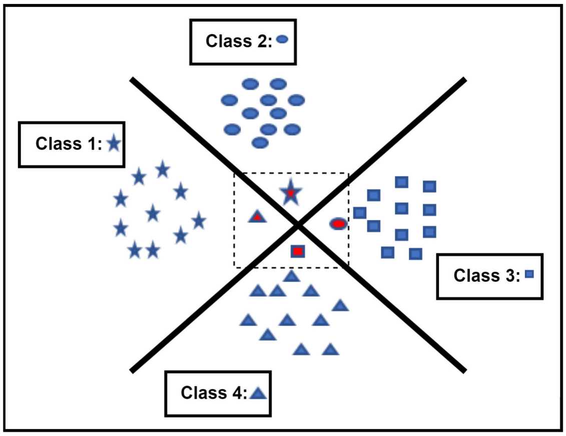


FIGURE 9. Feature space visualization of backdoor attacks. The solid black line indicates the original decision boundary, and the dotted rectangular line shows the backdoor decision boundary after adding the triggers.

attacker's label; thus, it produces the classification results to an attacker's target label regardless of the label the input belongs in. NC algorithm used the gradient descent method to reverse the trigger for each output class and the median absolute deviation outlier detection method to identify the triggers that appear as outliers.

In addition, the trigger size (smaller L1 norm) is used to identify the infected classes. The authors have performed experiments to evaluate the efficacy of the proposed model. In addition, they considered the strong assumption that the defender has white-box access to the model. In Figure 9, we illustrate the conceptual property of the backdoor attack. The model-level detection methods are developed based on this property. The distance between the victim and target labels is shortened in the feature space, and dotted lines show the decision boundary after the backdoor attack. The backdoor triggers create shortcuts within the region of multi-dimensional space. In another research, the authors have proposed another model-level defence approach called DeepInspect (DI) based on the property of backdoor attacks to address the security concerns of DNN models [47]. Trojan insertion can be considered as adding redundant data points near the legitimate ones and labeling them as the attack target. The movement from the original data point to the malicious one triggers the backdoor attack. As a result of Trojan insertion, one can observe from Figure 9 that the required perturbation to transform legitimate data into samples belonging to the attack target is smaller than the one in the corresponding benign model. DI identifies such 'small' triggers as the 'footprin' left by Trojan insertion and recovers potential triggers to extract the perturbation statistics.

The authors of [47] assumed that the defender knew the input data's dimensionality, output classes, and the model's confidence score. A conditional generative model was used to analyze the probability distribution of triggers and reconstruct the potential trigger pattern by generating sample data by reversing the model. To identify anomalies, double median absolute deviation was used as the detection criteria, where values above a threshold are deemed anomalies. For each detected trigger, a measurement is calculated to determine the probability of the data point belonging to a class other than the neural network's classification. Finally, any high anomaly data points are considered a trojan and are further analyzed.

The authors of [48] considered a generic defence solution, Meta Neural Trojan Detection (MNTD), to detect the backdoor attack on diverse domains like vision, speech, and text. Further, the proposed solution did not consider any prior assumption of backdoor triggers. The authors trained many clean and backdoor shadow models, and the resultant acted as the input of the meta-classifier, predicting whether the given model was Trojan. The authors considered benign samples for training benign shadow models and used the jumbo learning technique to model a generic distribution of trojan attacks and generate various Trojan shadow models. Further, many query inputs are made for shadow models, and confidence scores are concatenated and act as a feature representation of shadow models. These feature representations are input for the meta-classifier, a binary model, to predict whether the given model is Trojan.

In [49], the authors studied whether the model is backdoored or not. Though existing studies defend model trojan attacks, these techniques have limitations. For example, these techniques only detect the attacks when input is with the trigger instead of determining if a model is a trojan without an input trigger. Therefore, they proposed a novel scanning AI technique, artificial brain stimulation (ABS). The authors first analyzed the inner neuron behavior through their proposed stimulation method. Afterward, an optimization-based method is implemented for reverse-engineered triggers. Finally, efficacy of the model was evaluated on 177 trojan models. The results show that this technique outperforms the Neural Cleanse technique [46], which requires a lot of input samples and small triggers to achieve good performance. Further, this technique can work in the online model inspection.

B. DATASET LEVEL DEFENCE SOLUTIONS

Trigger input and dataset can be evaluated in post-deployment settings where data is inspected by assuming that data can be available to the defender since the attacker injected the triggers by poisoning the dataset. The paper [50] studied the behavior of backdoor attacks in an online environment where the model is already deployed.The authors also observed the property of backdoor attacks and proposed a defence solution underlining an assumption. The authors assumed that localized attacks solely rely on salient features that strongly affect the model, thus misclassifying many different inputs. If the region is determined, it can patch the other images with the group of truth labels. The proposed defence solution, SentiNet, uses an object detection mechanism for dataset level, specifically, inputs. The defence first discovered highly salient contagious regions of input images. Then, the extracted regions overlay on many clean images and test how they result in misclassification. As malicious images are designed to misclassify more than benign, thus can catch by SentiNet. Another study [51] has uncovered the backdoor attacks for DNNs in post-deployment settings. The authors studied the behavior of backdoor attacks and assumed that the predictions of the perturbated images always fall into the decided targeted class of an attacker.

The authors [51] proposed a runtime trojan detection method named Strong Intentional Perturbation (STRIP) for dataset level, specifically inputs. The authors turned input-agnostic attack strengths into weaknesses to use as a defence to detect the poison inputs. Their proposed method intentionally perturbed the incoming input and observed the randomness of the predicted class after superimposing various image patterns. The randomness observes by entropy measurement to quantify the randomness of the predicted class. As a result, the entropy of the clean input will be consistently large compared to the trojan input. Thus, a proper detection boundary can distinguish trojan input from clean input. For example, the predicted benign input ‘7’ is not always the same. It can be recognized as 30% digit ‘3’, 20% ‘1’. So there is always some randomness. In contrast, the predicted number of trojans inputs ‘4’ will always be classified to the target label. The experiment determines the detection boundary by a False Rejection Rate (FRR) of 1%. The entropy distribution falls within 1% FPR is benign and Trojan otherwise.

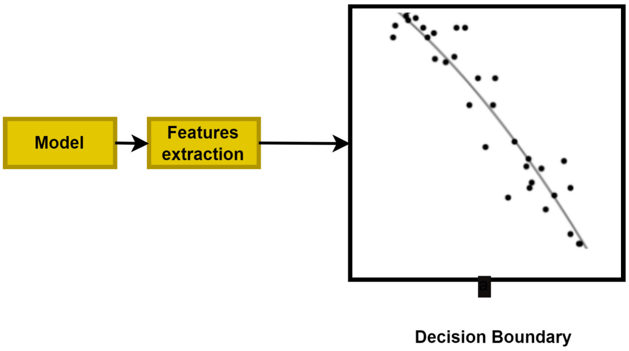


FIGURE 10. Extract the learned features activation values from the trained model and use dimensionality reduction and clustering techniques to detect benign and poisonous samples.

The authors of this study [52] also proposed a Dataset-level detection method for backdoor attacks. Given a model trained on a dataset, the corresponding activations of the last layers are collected for further analysis as mentioned in Figure 10 because activation of the previously hidden layer reflects the high-level features of the data used by the neural network to reach the final decision. They converted the last activation neurons to a 1-D vector. Further, independent component analysis has been performed to reduce the dimensions, avoid clustering over high-dimensional data, and get more robust clustering. The research proposed two methods used for cluster analysis: exclusionary reclassification and relative size comparison. In exclusionary classification, the process is to train a new model without the data corresponding to the clusters. Later, the new model was used to classify the removed cluster(s). If the removed cluster is classified as a label, it is considered benign data. Besides, the removed cluster is classified as a source class; this is poisonous data. The activation of the input belongs to the same label separated into clusters. K-mean clusters applied with $k = 2$ as the clustering will always separate the activations into two clusters.

The authors have proposed the ExRe score to assess whether the given cluster is poisonous. They set a threshold value. The score is calculated based on the total number of data points in the given cluster (L) / total number of data points classified as class (p). If $L/p > T$, this is the benign data point, whereas $L/p < T$ is a poison data point. The other method to check whether the given cluster is benign is to compare the relative size. If we expect that no more than p% of the data for a provided label can be poisoned by an adversary, we can consider a cluster to be poisoned if it contains less equal p% of the data. The silhouette score is also used as a metric where a high score means the class is infected. Finally, relabelling the poisonous data with the source class performed better than removing the poisonous data point and retraining the model for backdoor repair.

Another study [53] also detected the backdoor at the dataset level. Based on observing the backdoor behavior, the authors proposed a solution underlining an assumption. The observation was that when a trained set for a given label is corrupted, the training samples for this label are divided into two sub-populations. Clean samples will be larger, and the corrupted ones will be smaller. These backdoor attacks tend to leave behind a detectable trace ''spectral signature'' in the spectrum of covariance of feature representation learned by the neural network. Researchers have used robust statistical techniques to counter the attack to separate the corrupted and benign samples from Dataset.

In addition, the model's latent representation is extracted from the last layer of the model for further analysis. Robust statistics suggest that if the mean of two populations is sufficiently well separated relative to the variance of the population, then the corrupted data points can be detected and removed using Singular Value Decomposition (SVD). Then, SVD is performed on the covariance matrix on the extracted layer to calculate the outlier score for each input. The input value with an outlier high signature score flag as corrupted input is then removed from the Dataset, on which a clean model has trained again.

C. PROACTIVE DEFENCE SOLUTIONS

These defence solutions aim to work as blind removal backdoors, which do not differentiate a clean model from poison or clean input from poison. The main purpose of these defence solutions is to suppress the effect of backdoor attacks by maintaining model accuracy. The authors in [54] studied to reduce the impact of backdoor triggers from an infected model without actually identifying backdoors. The authors proposed three techniques to demolish the effect of backdoor triggers: input anomaly detection, model retraining, and input pre-processing. Firstly, they used SVM and decision trees for input anomaly detection. In the case of detection, the infected input will not be given to the model. Secondly, the retrained model intends to make the model 'forget' the trojan neurons. Thirdly, autoencoders are a pre-processor between the input and the model. If the input is from the same distribution, the difference between input and output is smaller, and the model works correctly with reconstructed input. In contrast, the input is considered a trojan if the difference is larger.

The study in [55] is similar above and proposes a solution to weaken and eliminate backdoor attacks. The authors of this study proposed the solution based on the assumption that the backdoor exploits sparse capacity in neural networks [3]. In their first approach, the authors prune the less ineffective neurons on clean inputs. However, this defence can be easily evaded in case of pruning-aware attacks. Therefore, the study devised another solution to counter this issue and proposed a combined method of fine-tuning and pruning. This method incurs high computational cost and complexity [54], [55]. However, according to [46], fine-tuning and pruning methods degrade the accuracy of the model. In [56], the authors studied the problem in which it is unclear whether the model learns the backdoor and cleans data in a similar way. If there is a difference in learning these two data, it is possible to prevent the model from learning them. The authors have found some observations of backdoors during learning: 1) model learns backdoor triggers much faster compared to the clean images. The stronger the attack is, the faster it converges on the backdoor. As a result, the training loss of backdoor images drops suddenly in the early epochs of training 2) backdoor images are always tied to a targeted class. Breaking the correlations between the trigger and target class could be possible by shuffling the labels of a small portion of inputs with low loss. Based on the aforementioned observations, they proposed a novel Anti-backdoor Learning (ABL) method. The proposed method consists of two stages of learning by utilizing Global Gradient Ascent (GGA) and Local Gradient Ascent (LGA). Firstly, at the beginning of the learning stage, they intentionally maximize the training loss to create a gap between the backdoor and benign samples to isolate backdoor data via low loss. Afterward, at the end of the training, GDA was used to unlearn the model with the isolated backdoor. They performed extensive experiments to prove the efficacy of the proposed method against ten state-of-art backdoor attacks.

VI. POTENTIAL FUTURE RESEARCH DIRECTIONS

A. DEFENCE CURRENT ASSUMPTIONS

The assumptions regarding defence against backdoor exploits are as follows: backdoor exploits sparse capacity in neural networks [4]. The backdoor triggers create shortcuts from within the region of the multi-dimensional space belonging to the label into the region belonging to the attackers' label. This misclassifies an attacker's target label regardless of the inputs [46]. The authors of [47] identified that when the attacker injects the corrupted data points near the benign data points, and labels the targeted class, a small perturbation is required to transform benign data into corrupted data compared to the benign sample. These small triggers leave a footprint behind. Localized attacks were assumed to rely solely on salient features that strongly affect the model, leading to misclassification of many different inputs [50]. In

TABLE 5. Qualitative comparison of backdoor defences.

defences	defence Target	Methods	Model	Dataset	Capabilities	Limitations
[46]	Model	Trigger reverse engineering	CNN	MNIST& GTSRB & Youtube Face & PubFig	Partially handling multiple triggers to multiple labels Insensitive to invisible triggers	Not generalize High computational cost High machine learning expertise not work for multiple tirggers to the same label Sensitive to trigger, size, shape and position Not extended to class-specific triggers
[47]	Model	CGAN	CNN	MNIST & GTSRB	Insensitive to invisible triggers Partially insensitive to multiple triggers to multiple labels	Not generalize High computational cost High machine learning expertise Sensitive to multiple triggers to the same class Sensitive to trigger shape, size and location Not work on class-specific triggers
[48]	Model	Meta classifier	CNN	MNIST&CIFAR10 &SC-M&SC-B &Irish&Irish-M &Irish-B&MR &MR-M	Black-box access of model Domain generalization No need for validation data access Work well on multiple triggers to the same label Work on multiple triggers to multiple labels Not sensitive to trigger size, shape and position Can detect invisible triggers Partially work on class-specific triggers	Computationally very high Require high machine learning expertise
[49]	Model	Trigger reverse engineering	CNN	CIFAR-10&GTSRB&ImageNet&VGG-Face&Age&USTS	While box access to models Partial domain generalization Need partial data validation Work on multiple triggers to multiple classes Work on invisible triggers	Computationally high It required high machine learning expertise Sensitive to trigger size, shape and position Does not work with multiple triggers to the same class Not work for class-specific triggers Works well only for very small triggers
[50]	Dataset (Input)	Object localization	VGG16&Faster-RCNN	Face recognition & Road sign & ImageNet	Computationally low It requires medium machine learning expertise Work on multiple triggers to the same labels Work on multiple triggers to the multiple labels Partially sensitive to trigger, size, shape and location Work well on invisible triggers	Not domain generalization Not work for class-specific triggers
[51]	Dataset (Input)	Observe randomness	CNN	MNIST&CIFAR-10&GTSRB	Low computational cost No machine learning expertise Applied to other domains as well Works with multiple triggers to the same labels Works with multiple triggers to multiple labels Insensitive to trigger, shape, size and location Works well with trigger invisibility	Not work in class-specific triggers
[52]	Dataset	Active Clustering	CNN	MNIST & LISA	White-box access to model Access to the poison dataset Validation data access domain generalization Requires medium computational resources Requires medium machine learning expertise Works well in multiple triggers to the same labels Works well in multiple triggers to multiple labels Insensitive to trigger size, shape and location Works well on trigger insensitivity	Not work in class-specific triggers
[53]	Dataset	Represenation learning	CNN	CIFAR-10	Domain generalization Requires medium computational resources Requires medium machine learning expertise Works well in multiple triggers to the same labels Works well in multiple triggers to multiple labels Insensitive to trigger size, shape and location Works well on trigger transparency Works well for class specific triggers	Only applicable to data collection attacks
[54]	Proactive	Neuron purning	SVM, Decision Tree	MNIST	Select least contribute neurons Purning carefully Assuming clean and backdoor neurons are separable restore performance of model	High computational cost and complexity Degrade the model accuracy

[51], the authors studied the behavior of backdoor attacks and assumed that the predictions of the perturbated images fall into the decided targeted class of an attacker. The study further observed the randomness of the given input. If the input has higher randomness, it is considered benign, or else a Trojan.

The authors of [52] observed the backdoor behavior and assumed that neuron activations for the backdoor are highly similar to the source class, and benign data resembled the label class. The authors of [53] observed that when a trained set for a given label is corrupted, the training samples for this label are divided into two sub-populations. Clean samples will be larger, and the corrupted ones will be smaller.

The backdoor trigger is a strong feature for the target label, and such a feature is represented by one or more sets of inner neurons. These compromised neuron activations fall within a certain range and are the main reason a model predicts a target label. For example, based on the observation, the benign input activation value is 20, and if the input contains a trigger, then the activation values peak at 70. So this peak value alleviates the output activation. The second observation is that these compromised neurons represent a subspace for the target label that is likely a global region that cuts across the whole input space because any trigger input leads to the targeted label [56].

Firstly, the model learns backdoor triggers much faster than clean images. The stronger the attack is, the faster it converges on the backdoor. As a result, the training loss of backdoor images suddenly drops in the early training epochs. Secondly, backdoor images are always tied to a targeted class. It could be possible to break the correlations between the trigger and target class by shuffling the labels of a small portion of inputs with low loss [63]

B. DEFENCE GENERALIZATION

defence generalization refers to the ability of a system or strategy to respond effectively to a wide range of potential threats or challenges. There is a need for a defence system that can adapt and be effective in various situations rather than being tailored to a specific threat or set of circumstances. Most existing defence solutions are explicitly designed for vision domains in image classification applications. The summary of the dataset used for the defences is described in table 5. There is a lack of generalization of defences to other domains, such as text and audio. Many backdoor defence solutions have been proposed in computer vision, showing high performance and reliability in defence performance. It is worthwhile to generalize these solutions to other applications like natural language processing and videos.

C. DEFENDER CAPABILITIES

The defender capability for a deep learning model refers to its ability to resist attacks and maintain accuracy in the presence of backdoor attacks. The backdoor examples are inputs that trick DL models into making incorrect predictions. Regarding the overall defender capability of a DL model, researchers and practitioners often consider a good defence solution should be robustness, reliability, and resilience against various kinds of attacks. However, there is a need for realistic defender capabilities as some defences have strong assumptions, such as access to poison data and knowledge about the trigger size. There is a need for a suitable testing environment and to evaluate the effectiveness of the defence solution in a controlled way to identify the weakness of the models before deploying them to safety and critical applications.

D. ROBUSTNESS IMPROVEMENT

Once the attacker successfully tricks the model into predicting according to the target, the dangerous cause is immeasurable. Therefore, robustness for DL models is extremely important. Researchers have proposed standard techniques for improving a DL model's robustness capability. For example, anti-backdoor learning [63] automatically prevents the backdoor during data training. [64] provides an efficient general framework to certify the robustness of neural networks with ReLU, tanh, sigmoid, and arctan activation functions. It's important to note that improving the robustness of a DL model is an ongoing process, as attackers continually develop new methods for tricking models. Thus, regular testing and evaluation of the model are crucial for maintaining its defender capability over time. Many existing defence solutions can detect the poison model but don't propose an effective solution to recover the model. Therefore, this is another important avenue of research to investigate further the defence approach to finding the solutions to reduce backdoor attacks and provides certifying robustness in neural networks. Moreover, there is a need for a metric that can help quantitatively analyze the robustness.

E. FEDERATED LEARNING

FL is a distributed learning that allows multiple devices or nodes to train a model without sharing their data. It reduces the risk of data theft and ensures that each node's data remains confidential. Researchers and practitioners have used several techniques to secure the trustworthiness of DL models. Outlier detection techniques can be used to identify and remove malicious nodes from the system, reducing the risk of data poisoning and model theft attacks. Differential privacy is a mathematical framework for protecting the privacy of individuals while allowing data to be used for machine learning. This can be especially important in distributed learning, where data from multiple sources is combined to train a model [65]. By implementing these and other distributed learning defence techniques, organizations can improve the security and robustness of their machine learning systems, ensuring that the models produced are accurate and trustworthy. Apart from this, how to detect backdoor attacks in the FL environment is still an unsolved problem.

VII. CONCLUSION

This paper presented a novel and comprehensive framework for the smart cyber defence of deep learning security in smart manufacturing systems. The proposed framework addressed the vulnerabilities of DL systems by incorporating multiple layers of security, including privacy and protection of data and models, and employing statistical and intelligent model techniques for maintaining data privacy and confidentiality. Additionally, the framework included policies and procedures for securing data that comply with industrial standards, incorporating a threat model to identify potential or actual vulnerabilities, and placing countermeasures and controls in place to defend against various attacks. Further, the backdoor specimen is introduced in terms of properties and methods that can be used to generate backdoor attacks. Then, we analyzed state-of-the-art backdoor attacks and defence techniques and performed a qualitative comparison of existing backdoor attacks and defences. In the future, we will expand our work by quantitatively evaluating the proposed framework. This paper provides comprehensive guidelines for designing secure, reliable, and robust deep learning models. We hope more robust deep learning defence solutions are proposed based on the knowledge of backdoor attacks.

ACKNOWLEDGMENT

The authors would like to thank those who have contributed to this research.

REFERENCES

[1] I. Stoica, D. Song, R. A. Popa, D. Patterson, M. W. Mahoney, R. Katz, A. D. Joseph, M. Jordan, J. M. Hellerstein, J. E. Gonzalez, K. Goldberg, A. Ghodsi, D. Culler, and P. Abbeel, "A Berkeley view of systems challenges for AI," 2017, *arXiv:1712.05855*.

[2] W. Guo, D. Mu, J. Xu, P. Su, G. Wang, and X. Xing, "LEMNA: Explaining deep learning based security applications," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Oct. 2018, pp. 364–379.

[3] T. Gu, B. Dolan-Gavitt, and S. Garg, "BadNets: Identifying vulnerabilities in the machine learning model supply chain," 2017, *arXiv:1708.06733*.

[4] Y. Liu, X. Ma, J. Bailey, and F. Lu, "Reflection backdoor: A natural backdoor attack on deep neural networks," in *Proc. Eur. Conf. Comput. Vis.*, Glasgow, U.K., 2020, pp. 182–199.

[5] I. Arshad, S. H. Alsamhi, and W. Afzal, "Big data testing techniques: Taxonomy, challenges and future trends," *Comput., Mater. Continua*, vol. 74, no. 2, pp. 2739–2770, 2023.

[6] Y. Zhao, Y. Qu, Y. Xiang, and L. Gao, "A comprehensive survey on edge data integrity verification: Fundamentals and future trends," 2022, *arXiv:2210.10978*.

[7] N. Akhtar and A. Mian, "A Threat of adversarial attacks on deep learning in computer vision: A survey," *IEEE Access*, vol. 6, pp. 14410–14430, 2018.

[8] Y. Yuan, P. He, Q. Zhu, and X. Li, "Adversarial examples: Attacks and defences for deep learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 9, pp. 2805–2824, Sep. 2019.

[9] Q. Liu, P. Li, W. Zhao, W. Cai, S. Yu, and V. C. M. Leung, "A survey on security threats and defensive techniques of machine learning: A data driven view," *IEEE Access*, vol. 6, pp. 12103–12117, 2018.

[10] Y. Gao, B. G. Doan, Z. Zhang, S. Ma, J. Zhang, A. Fu, S. Nepal, and H. Kim, "Backdoor attacks and countermeasures on deep learning: A comprehensive review," 2020, *arXiv:2007.10760*.

[11] Y. Li, Y. Jiang, Z. Li, and S.-T. Xia, "Backdoor learning: A survey," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Jun. 22, 2022, doi: 10.1109/TNNLS.2022.3182979.

[12] J. Zhang and C. Li, "Adversarial examples: Opportunities and challenges," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 7, pp. 2578–2593, Jul. 2020.

[13] M. Barreno, B. Nelson, A. D. Joseph, and J. D. Tygar, "The security of machine learning," *Mach. Learn.*, vol. 81, no. 2, pp. 121–148, 2010.

[14] I. Arshad, M. N. Asghar, Y. Qiao, B. Lee, and Y. Ye, "Pixdoor: A pixel-space backdoor attack on deep learning models," in *Proc. 29th Eur. Signal Process. Conf. (EUSIPCO)*, Aug. 2021, pp. 681–685.

[15] A. Schwarzschild, "Just how toxic is data poisoning? A unified benchmark for backdoor and data poisoning attacks," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 9389–9398.

[16] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.

[17] Y. LeCun, C. Cortes, and C. Burges, *The MNIST Dataset of Handwritten Digits (Images)*. New York, NY, USA: NYU, 1999.

[18] A. Turner, D. Tsipras, and A. Madry, "Label-consistent backdoor attacks," 2019, *arXiv:1912.02771*.

[19] Y. Yao, H. Li, H. Zheng, and B. Y. Zhao, "Latent backdoor attacks on deep neural networks," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Nov. 2019, pp. 2041–2055.

[20] S. Wang, S. Nepal, C. Rudolph, M. Grobler, S. Chen, and T. Chen, "Backdoor attacks against transfer learning with pre-trained deep learning models," *IEEE Trans. Services Comput.*, vol. 15, no. 3, pp. 1526–1539, May 2022.

[21] R. Xu, J. B. D. Joshi, and C. Li, "CryptoNN: Training neural networks over encrypted data," in *Proc. IEEE 39th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Jul. 2019, pp. 1199–1209.

[22] P. Mohassel and Y. Zhang, "SecureML: A system for scalable privacy-preserving machine learning," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2017, pp. 19–38.

[23] X. Chen, C. Liu, B. Li, K. Lu, and D. Song, "Targeted backdoor attacks on deep learning systems using data poisoning," 2017, *arXiv:1712.05526*.

[24] I. Arshad, Y. Qiao, B. Lee, and Y. Ye, "Invisible encoded backdoor attack on DNNs using conditional GAN," in *Proc. IEEE Int. Conf. Consum. Electron. (ICCE)*, Jan. 2023, pp. 1–5.

[25] A. Shafahi, "Poison frogs! targeted clean-label poisoning attacks on neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 6106–6116.

[26] C. Zhu, "Transferable clean-label poisoning attacks on deep neural nets," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 7614–7623.

[27] W. R. Huang, "MetaPoison: Practical general-purpose clean-label data poisoning," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 12080–12091.

[28] A. Saha, "Hidden trigger back- door attacks," in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 11957–11965.

[29] M. Barni, K. Kallas, and B. Tondi, "A new backdoor attack in CNNS by training set corruption without label poisoning," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2019, pp. 101–105.

[30] I. Shumailov, "Manipulating SGD with data ordering attacks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 18021–18032.

[31] S. Zhao, X. Ma, X. Zheng, J. Bailey, J. Chen, and Y.-G. Jiang, "Clean-label backdoor attacks on video recognition models," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 14431–14440.

[32] K. Liu, B. Tan, R. Karri, and S. Garg, "Poisoning the (data) well in ML-based CAD: A case study of hiding lithographic hotspots," in *Proc. Design, Autom. Test Eur. Conf. Exhib. (DATE)*, Mar. 2020, pp. 306–309.

[33] H. Zhong, C. Liao, A. C. Squicciarini, S. Zhu, and D. Miller, "Backdoor embedding in convolutional neural network models via invisible perturbation," in *Proc. 10th ACM Conf. Data Appl. Secur. Privacy*, Mar. 2020, pp. 97–108.

[34] M. Defooli, "Universal adversrial perturbations," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 1765–1773.

[35] H. A. A. K. Hammoud and B. Ghanem, "Check your other door! Creating backdoor attacks in the frequency domain," 2021, *arXiv:2109.05507*.

[36] T. Wang, Y. Yao, F. Xu, S. An, H. Tong, and T. Wang, "Backdoor attack through frequency domain," 2021, *arXiv:2111.10991*.

[37] S. Cheng, "Deep feature space trojan attack of neural networks by controlled detoxification," in *Proc. AAAI Conf. Artif. Intell.*, 2021, pp. 1148–1156.

[38] Y. Li, Y. Li, B. Wu, L. Li, R. He, and S. Lyu, "Invisible backdoor attack with sample-specific triggers," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 16443–16452.

[39] Y. Liu, S. Ma, Y. Aafer, W.-C. Lee, J. Zhai, W. Wang, and X. Zhang, "Trojaning attack on neural networks," in *Proc. Netw. Distrib. Syst. Secur. Symp.*, 2018, pp. 1–17.

[40] J. Lin, "Composite backdoor attack for deep neural network by mixing existing benign features," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2020, pp. 113–131.

[41] M. Xue, C. He, J. Wang, and W. Liu, "One-to-N & N-to-one: Two advanced backdoor attacks against deep learning models," *IEEE Trans. Depend. Secure Comput.*, vol. 19, no. 3, pp. 1562–1578, May 2022.

[42] C. He, M. Xue, J. Wang, and W. Liu, "Embedding backdoors as the facial features: Invisible backdoor attacks against face recognition systems," in *Proc. ACM Turing Celebration Conf. China*, May 2020, pp. 231–235.

[43] E. Wenger, J. Passananti, A. N. Bhagoji, Y. Yao, H. Zheng, and B. Y. Zhao, "Backdoor attacks against deep learning systems in the physical world," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 6202–6211.

[44] J. Guo and C. Liu, "Practical poisoning attacks on neural networks," in *Proc. Eur. Conf. Comput. Vis.*, Aug. 2020, pp. 142–158.

[45] H. Kwon, H. Yoon, and K.-W. Park, "FriendNet backdoor: Indentifying backdoor attack that is safe for friendly deep neural network," in *Proc. 3rd Int. Conf. Softw. Eng. Inf. Manage.*, Jan. 2020, pp. 53–57.

[46] B. Wang, Y. Yao, S. Shan, H. Li, B. Viswanath, H. Zheng, and B. Y. Zhao, "Neural cleanse: Identifying and mitigating backdoor attacks in neural networks," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2019, pp. 707–723.

[47] H. Chen, C. Fu, J. Zhao, and F. Koushanfar, "DeepInspect: A black-box trojan detection and mitigation framework for deep neural networks," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, Aug. 2019, pp. 4658–4664.

[48] X. Xu, Q. Wang, H. Li, N. Borisov, C. A. Gunter, and B. Li, "Detecting AI trojans using meta neural analysis," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2021, pp. 103–120.

[49] Y. Liu, W.-C. Lee, G. Tao, S. Ma, Y. Aafer, and X. Zhang, "ABS: Scanning neural networks for back-doors by artificial brain stimulation," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Nov. 2019, pp. 1265–1282.

[50] E. Chou, F. Tramèr, and G. Pellegrino, "SentiNet: Detecting localized universal attacks against deep learning systems," in *Proc. IEEE Secur. Privacy Workshops (SPW)*, May 2020, pp. 48–54.

[51] Y. Gao, C. Xu, D. Wang, S. Chen, D. C. Ranasinghe, and S. Nepal, "STRIP: A defence against trojan attacks on deep neural networks," in *Proc. 35th Annu. Comput. Secur. Appl. Conf.*, Dec. 2019, pp. 113–125.

[52] B. Chen, W. Carvalho, N. Baracaldo, H. Ludwig, B. Edwards, T. Lee, I. Molloy, and B. Srivastava, "Detecting backdoor attacks on deep neural networks by activation clustering," 2018, *arXiv:1811.03728*.

[53] B. Tran, "Spectral signatures in backdoor attacks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 8011–8021.

[54] Y. Liu, Y. Xie, and A. Srivastava, "Neural trojans," in *Proc. IEEE Int. Conf. Comput. Design (ICCD)*, Nov. 2017, pp. 45–48.

[55] K. Liu, "Fine-pruning: Defending against backdooring attacks on deep neural networks," in *Proc. Int. Symp. Res. Attacks, Intrusions, Defences*, 2018, pp. 273–294.

[56] Y. Li, X. Lyu, N. Koren, L. Lyu, B. Li, and X. Ma, "Neural attention distillation: Erasing backdoor triggers from deep neural networks," 2021, *arXiv:2101.05930*.

[57] X. Han, "Clean-annotation backdoor attack against lane detection systems in the wild," 2022, *arXiv:2203.00858*.

[58] E. Bagdasaryan, "How to backdoor federated learning," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2020, pp. 2938–2948.

[59] A. N. Bhagoji, "Analyzing federated learning through an adversarial lens," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 634–643.

[60] Z. Sun, P. Kairouz, A. T. Suresh, and H. B. McMahan, "Can you really backdoor federated learning?" 2019, *arXiv:1911.07963*.

[61] E. Bagdasaryan and V. Shmatikov, "Blind backdoors in deep learning models," in *Proc. 30th USENIX Secur. Symp. (USENIX Secur.)*, 2021, pp. 1505–1521.

[62] G. Yar, S. Boboila, C. Nita-Rotaru, and A. Oprea, "Backdoor attacks in peer-to-peer federated learning," 2023, *arXiv:2301.09732*.

[63] Y. Li, "Anti-backdoor learning: Training clean models on poisoned data," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 14900–14912.

[64] H. Zhang, "Efficient neural network robustness certification with general activation functions," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 4944–4953.

[65] Y. Zhao, Y. Qu, Y. Xiang, Y. Zhang, and L. Gao, "A lightweight model-based evolutionary consensus protocol in blockchain as a service for IoT," *IEEE Trans. Services Comput.*, vol. 16, no. 4, pp. 2343–2358, Aug. 2023.

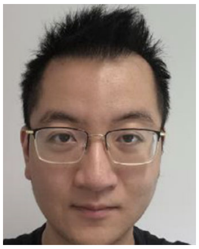
IRAM ARSHAD received the Bachelor of Science degree in computer science from the Department of Computer Science, GCU, Lahore, Pakistan, in 2011, and the Master of Science degree in computer science from LCWU, Lahore, in 2015. She is currently pursuing the Ph.D. degree with the Technological University of Shannon: Midlands Midwest, Athlone, Ireland.

In December 2015, she joined a multi-national software company Tkxel, Lahore, as a Software Quality Assurance Engineer. She has worked on numerous national and international projects to ensure quality. Later, she joined another tech software company, Fiverivers Technologies, Lahore, in February 2019, as a Senior Software Quality Assurance Engineer. She was also an automation engineer during that tenure. Her research interests include but is not limited to artificial intelligence, computer vision, deep learning, security, and cyber attacks.

SAEED HAMOOD ALSAMHI received the B.Eng. degree from the Communication Division, Department of Electronic Engineering, IBB University, Yemen, in 2009, and the M.Tech. degree in communication systems and the Ph.D. degree from the Department of Electronics Engineering, Indian Institute of Technology (Banaras Hindu University), IIT (BHU), Varanasi, India, in 2012 and 2015, respectively. In 2009, he was a Lecturer Assistant with the Engineering Faculty, IBB University. After that, he held a postdoctoral researcher position with the School of Aerospace Engineering, Tsinghua University, Beijing, China, in optimal and smart wireless network research and its applications to enhance robotics technologies. Since 2019, he has been an Assistant Professor with the Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen. In 2020, he was a MSCA SMART 4.0 Fellow with the Athlone Institute of Technology, Athlone, Ireland. Currently, he is a Senior Research Fellow with the Insight Centre for Data Analytics, University of Galway, Galway, Ireland, where he is also an adjunct lectureship appointment with the College of Science and Engineering. He has published more than 145 articles in high-reputation journals in IEEE, Elsevier, Springer, Wiley, and MDPI publishers. His research interests include green and semantic communication, the green Internet of Things, QoE, QoS, multi-robot collaboration, blockchain technology, federated learning, and space technologies (high altitude platforms, drones, and tethered balloon technologies).

YUANSONG QIAO received the B.Sc. and M.Sc. degrees in solid mechanics from Beihang University, Beijing, China, in 1996 and 1999, respectively, and the Ph.D. degree in computer applied technology from the Institute of Software, Chinese Academy of Sciences (ISCAS), Beijing, in 2007. He is the Principal Investigator at the Software Research Institute (SRI), Technological University of Shannon: Midlands Midwest, Athlone, Ireland. As part of his Ph.D. research program, he joined the SRI at Technological University of Shannon: Midlands Midwest in 2005. He continued his research in SRI as a postdoctoral researcher, in 2007. After graduation, he joined ISCAS immediately, where he held roles as a network administrator and a research engineer and the team leader in research and development, working on protocols and products in the areas of computer networking, multimedia communication, and network security. His research interests include network protocol design and multimedia communications for the future internet.

BRIAN LEE received the Ph.D. degree from the Trinity College Dublin, Dublin, Ireland, in the application of programmable networking for network management. He is the Director of the Software Research Institute, Technological University of Shannon: Midlands Midwest, Athlone, Ireland. He has over 25 years research and development experience in telecommunications network monitoring, their systems, and software design and development for large telecommunications products with very high impact research publications. Formerly, he was the Director of Research with LM Ericsson, Ireland, with responsibility for overseeing all research activities, including external collaborations and relationship management. He was the Engineering Manager of Duolog Ltd., where he was responsible for strategic and operational management of all research and development activities.



YUHANG YE received the Ph.D. degree in computer networks from the Athlone Institute of Technology, in 2018. He is currently a Lecturer with the Department of Computer and Software Engineering, Technological University of the Shannon: Midlands Midwest, Ireland. His current research interests include machine learning, multimedia communication, cybersecurity, and computer networks.