IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 26 January 2024, accepted 18 March 2024, date of publication 27 March 2024, date of current version 4 April 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3382584

SURVEY

A Comprehensive Survey on Backdoor Attacks and Their Defenses in Face Recognition Systems

QUENTIN LE ROUX[1,2], ERIC BOURBAO[1], YANNICK TEGLIA[1], AND KASSEM KALLAS[2], (Senior Member, IEEE)
[1]Thales DIS, 13600 La Ciotat, France
[2]INRIA, 35042 Rennes, France

Corresponding author: Quentin Le Roux (quentin.le-roux@thalesgroup.com)

This work was supported by the Resources of Agence Nationale de la Recherche (ANR)/Agence de l'Innovation de Défense (AID) under Chaire Security of AI for Defense Applications (SAIDA) under Grant ANR-20-CHIA-0011-01.

ABSTRACT Deep learning has significantly transformed face recognition, enabling the deployment of large-scale, state-of-the-art solutions worldwide. However, the widespread adoption of deep neural networks (DNNs) and the rise of Machine Learning as a Service emphasize the need for secure DNNs. This paper revisits the face recognition threat model in the context of DNN ubiquity and the common practice of outsourcing their training and hosting to third-parties. Here, we identify backdoor attacks as a significant threat to modern DNN-based face recognition systems (FRS). Backdoor attacks involve an attacker manipulating a DNN's training or deployment, injecting it with a stealthy and malicious behavior. Once the DNN has entered its inference stage, the attacker may activate the backdoor and compromise the DNN's intended functionality. Given the critical nature of this threat to DNN-based FRS, our paper comprehensively surveys the literature of backdoor attacks and defenses previously demonstrated on FRS DNNs. As a last point, we highlight potential vulnerabilities and unexplored areas in FRS security.

INDEX TERMS Backdoor attacks, backdoor defenses, biometrics, deep neural networks, face recognition, integrity vulnerabilities, security, survey.

I. INTRODUCTION

The adaptability and performance of deep neural networks (DNNs) has led to their widespread adoption in both academic and industrial settings, gaining traction in fields like image recognition [1], object detection [2] or, more recently, large language models [3]. However, both the escalating costs of data collection and the resources required to train and deploy increasingly complex models has led companies to outsource their machine learning (ML) needs to third-party providers.

For instance, people without the means to perform DNN training may rely on cloud solutions [4], [5], [6] (a business model commonly referred to as Machine Learning as a Service, i.e. MLaaS) or pretrained models [7] to bootstrap their development. Unfortunately, this convenience involves handing over to these third parties a significant control over critical phases of a model's lifecycle, including data labeling, model development, and training. Third-party providers therefore exert a strong influence over their clients' tasks, raising distinct security concerns related to DNN integrity.

DNN risks are typically categorized within the framework of Confidentiality, Integrity, and Availability (CIA) [8]. For instance, membership inference attacks [9] pose a threat to data confidentiality by revealing private information from a model's training dataset. On the other hand, data poisoning attacks [10] disrupt a model's availability by manipulating its training data, resulting in irreversible performance degradation at test-time. However, our primary focus in this paper revolves around the integrity of DNNs. Whereas assessments of integrity risk predate the rise of deep learning [11], recent studies have emphasized the susceptibility of DNNs to

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

malicious manipulations throughout their lifecycle, spanning training, deployment, and testing phases [12], [13], [14]. These vulnerabilities are particularly significant in the context of ML third-party outsourcing, such as in MLaaS, where they notably impact one of the most prominent deep learning applications: Face Recognition Systems (FRS) [15], [16], [17], [18].

Dating back to the 1960s [19], computer-assisted face recognition has incrementally progressed from handcrafted feature representation methods [20] (e.g. Eigenfaces [21], DBSCAN [22], Viola-Jones [23], HoG [24]) to deep learning models. The release of milestone datasets such as ImageNet [1] and the substantial performance improvements brought by AlexNet [25] and all subsequent DNNs have led to breakthroughs in ML and face recognition in particular [26]. However, the management complexities associated with large-scale, state-of-the-art, and real-world systems [16], [27], [28], [29] have positioned DNNs as cornerstone products in the MLaaS economy and ML outsourcing in general.

This shift in the development and hosting of DNNs introduces inherent security risks that require attention. DNNs are notably vulnerable to backdoor attacks, also known as neural trojans [30], a threat associated with both the practices of dataset and model outsourcing [12], [31].

Backdoor attacks [12] occur when an attacker is able to induce erroneous behaviors in a DNN at test-time by discreetly manipulating its training or deployment. Typically, a backdoor is characterized by a malicious trigger pattern that is learned by or injected into a victim DNN. Once the DNN is deployed by an unsuspecting user, the attacker can activate the backdoor at any moment. Activation occurs when the attacker presents the system with an input corrupted with the trigger. Backdoors are designed to remain inconspicuous, ensuring that the compromised DNN appears indistinguishable from a benign one under standard conditions (i.e. when comparing the accuracy on clean data between backdoored and benign models). Therefore, backdoor attacks aim for a high attack success rate (*ASR*) and a high level of stealth such that backdoored DNNs will be deployed by potential victims [31].

The growing importance of backdoor attacks in the field of deep learning security and the astonishing array of backdoor injection methods available to attackers, ranging from data poisoning [32] or hardware-software interaction [33] to targeted weight manipulation [34], underscores the need for strong countermeasures and a deeper understanding of their limitations. Moreover, the breadth of demonstrations on various tasks, from automated malware detection and reinforcement learning [35] to the more recent transformer-based ChatGPTs [3], should be cause for concern for any stakeholder intent on using DNNs in their applications.

In this paper, we focus on the impact of backdoors on the real-life biometry application of face recognition. Backdoors are problematic first because of the *current* reliance on multiple DNNs in face recognition tasks (face recognition DNNs are already deployed in the wild [4], [6]), but also due to face recognition being a core use case in previous backdoor attack demonstrations [12], [13], [14], [35]. Concernedly, prior work also demonstrated physical backdoor attacks on FRS, e.g. using glasses [36] or printed patches [37] to trigger a backdoored model. Such attacks could lead to tampering with critical authentication systems that could wreak havoc in many sectors like online banking or damage public and private secrecy (e.g. manipulating the access to restricted areas). Therefore, the development of DNN backdoors and their use in multiple face recognition use cases raise the question about how these attacks align with the traditional FRS threat model.

As such, backdoor attacks present a substantial challenge for users and providers of DNN-based FRS. To maintain trust between stakeholders, they must contend with understanding the security implications of these attacks on their FRS applications. In this context, this paper explores the development and ramifications of backdoor attacks on face recognition, shedding light on their broader impact on FRS. Our first contribution is a revision of the conventional FRS threat model to reflect this new reality. Our second contribution is a comprehensive survey of DNN backdoor attacks previously deployed on FRS models. Building on this attack overview, our third contribution involves the concurrent survey of existing defenses applied to DNNs used in FRS. As a final contribution, we identify both the limitations and potential avenues for enhancing the security of FRS in the face of the growing risk posed by backdoor attacks. It is essential to note that this paper's focus on DNN backdoor attacks on FRS does not diminish the significance of other cybersecurity risks affecting FRS [38]. Instead, it underscores the need to address this novel and evolving risk as a supplemental threat.

Our work differentiates from prior surveys like Gao et al. [31] and Li et al. [35] by first providing a thorough review of a FRS pipeline, an interesting target for any backdoor attacker. Doing so, we anchor this paper in a real-life scenario with financial stakes. Secondly, by focusing on FRS, we believe we provide the most fine-grained and up-to-date categorization of both backdoor attacks and defenses (e.g. in terms of knowledge or capacity level for either attackers and defenders). We also cover both image and video-based attacks as well as a broad range of semantic attacks and so-called triggerless attacks. Thirdly, we expand this categorical outlook by cross-referencing each side, indicating which attacks and defenses supersede each other and which FRS stages they typically target. Lastly, by anchoring our survey on an existing application, we highlight unexplored areas and restrictions regarding both attacks and defenses that have not yet been considered. Here, we note that this paper does not focus on the mathematical underpinnings of backdoor attacks. We point the reader to Wu et al. [14] for this matter.

The structure of this paper is as follows: Section II covers the structural components that make up a FRS pipeline in light of new DNN-based methods. Section III reviews the corresponding threat model and the impact DNNs have had on it. Building on this new perspective, Section IV provides a comprehensive examination of the existing literature on backdoor attacks that have targeted FRS. In a corresponding fashion, Section V delves into the landscape of backdoor defenses with prior demonstration on FRS. Finally, Section VI explores current trends and ongoing challenges in evaluating the risks associated with training and deployment of DNN-based FRS. The paper concludes with Section VII.

II. THE STRUCTURE OF FACE RECOGNITION SYSTEMS

A FRS is a multi-step pipeline that processes inputs, e.g. images or video streams, into one or more biometric templates for downstream tasks such as person authentication [39] or identification [40], e.g. in a law enforcement database [41]. Despite the longstanding history of risk assessment in FRS [38], [42], [43], the recent reliance on DNNs necessitates a new assessment of the typical approach to understanding and mitigating FRS vulnerabilities. In this Section, we therefore examine the standard FRS structure and how it integrates DNNs.

A. A GENERAL SCHEMA OF A FRS

The standard biometric framework [38], [44] that underpins FRS divides a biometric pipeline into three distinct stages: acquisition, extraction, and matching. Under this framework, a modern FRS consists of a sensor, a DNN-based extraction pipeline (itself composed a detector, an alignment step, an anti-spoofer, a namesake feature extractor, and, optionally, a feature binarizer), and a matcher. Fig. 1 provides an illustration of the structure of a DNN-based FRS. The relatively recent integration of deep learning techniques [27], [28] in FRS has led to a notable increase in the capabilities of the first two stages [15], e.g. ability to acquire and process in-the-wild face identities on edge devices [45].

The sensor (e.g. camera) acquires images and forwards them to the face detector. If the detector identifies one or more faces in the input, the faces are cropped and aligned to fit a standardized representation for storing and manipulating two- and three-dimensional facial data. This format, dubbed a canonical shape, helps reduce the variance of intra-class/identity features [13].

Subsequently, the anti-spoofer scans the preprocessed faces to determine their authenticity [17]. This step intends to catch both accidental (e.g. a face printed on a billboard) and malicious inputs (e.g. a face presentation attack [46]). If a face is deemed genuine, it proceeds to the feature extraction step [13] where the input is converted into a biometric template, e.g. a real-valued vector equipped with a notion of distance [21]. Finally, the matcher handles these vectors to make an accept/reject decision based on a predetermined threshold and inform the downstream tasks interfaced with the FRS. A score below the threshold indicates a match, e.g., if the input matches a record in a database.

Here, we note that an optional feature binarization module [47] may be added after the extractor. Binarization converts a real-valued feature vector into a binary format that enables specific security features such as revocability and non-reusability of the biometric features thanks to cryptographic schemes like Fuzzy Commitment [47], [48].

B. THE MANY DNNS OF A MODERN-DAY FRS

Over the past decade, FRS have evolved to integrate multiple types of DNNs, expanding the capabilities of either detector (and possibly the alignment step), anti-spoofer, feature extractor, or feature binarizer tasks. Each DNN is tailored to a specific task within a FRS:

(1) The **detector** typically relies on a "backbone" convolutional neural network (CNN) [49], [50], [51], [52], [53] to identify and locate one or more faces in an image. The DNN regresses the coordinates of a bounding box [49] and predicts its associated attribute(s) and/or label(s), which include a personhood or confidence score and characteristics like yaw, occlusion, and luminosity [52], [53], [54] for instance.

(2) The **face alignment** module typically involve CNNs and predicts (e.g. via regression) the coordinates of facial landmarks [55], [56], such as those of the nose, eyes, and mouth, in a detected face. These landmarks are subsequently used in to fit a face to a canonical shape. We note that the detector has slowly evolved to also perform this task (landmarks become parts of the attributes associated to a bounding box) as illustrated by the RetinaFace detector [52].

(3) The **anti-spoofer** verifies whether a detected face fits a criterion of liveliness [57]. Though an anti-spoofer may be used to discard accidental detections (e.g. faces on a billboard caught by a CCTV camera), it is first designed to detect malicious presentation attacks [58], where an attacker attempts to gain access to a downstream task by spoofing a given identity. Anti-spoofers rely on a diverse range of models from CNNs [59] to transformers [60].

(4) The **feature extractor** is typically a CNN specially-trained to map a face image to a lower-dimensional, real-valued vector representation [13], [49]. These embeddings are learned via a variety of methods such as metric or angular margin learning [45], [61], which help distinguish between multiple identities. The feature extractor thus makes up the core of a FRS.

(5) The **feature binarizer** is an optional module in a FRS. Located after the feature extractor [47], it maps its real-valued outputs into binary representations that enable important security features like applications in cancelable systems [62]. That is, they allow the creation of protected templates that are revocable (the underlying biometric templates can be deleted and reissued), non-reversible (biometric templates cannot be

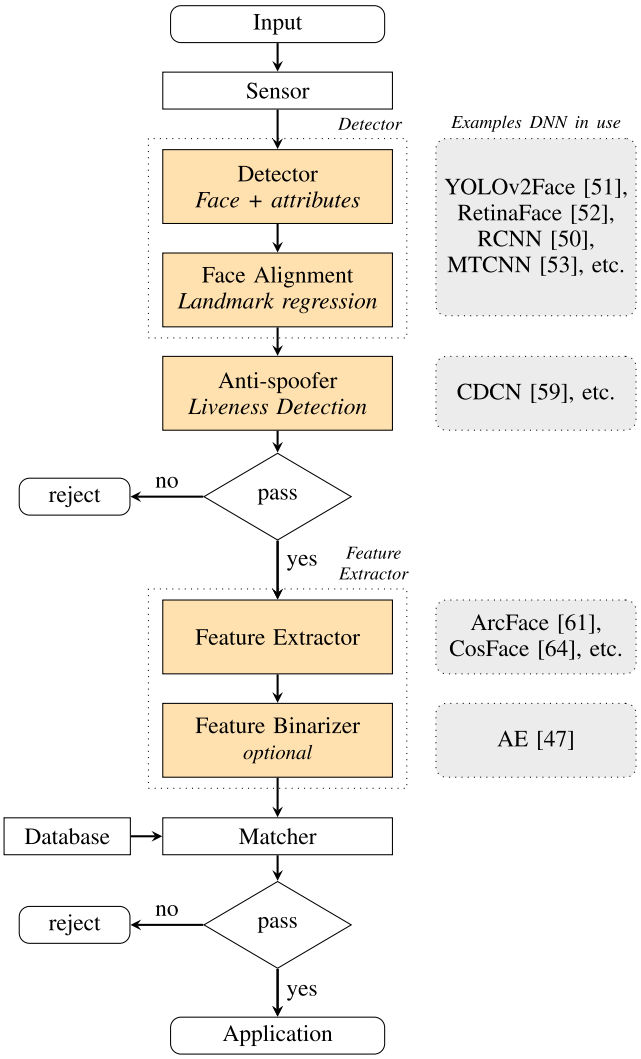

FIGURE 1. Serial Schema of a DNN-based Face Recognition System, with stages that may involve DNNs in Orange.

recovered from their protected form), and unlinkable (two protected templates of the same person cannot be matched). Though simple transformation schemes [63] exist, recent iterations of feature binarizers use auto-encoders (AE) to perform this task [47].

A more detailed coverage of the different DNN models used in a FRS is found in App. A. We note that, besides the sequential FRS view found in Fig. 1, there exist parallel FRS schemas [65] that merge the outputs of the feature extractor and anti-spoofer into a single input that is fed to the matcher. For the purpose of this paper however, we consider both setups as equivalent and will thus keep to the former version.

III. THE VULNERABILITIES OF DNN-BASED FRS

The growing complexity of FRS pipelines calls attention to their many security challenges. Information security is typically assessed under the Confidentiality, Integrity, and Availability (CIA) framework [8]. Confidentiality pertains to risks related to the extraction of private information from a FRS (e.g. identities, biometric templates), integrity to the manipulation and takeover of a FRS' functions, and availability to any means by which an attacker can restrict or interdict a user from accessing a FRS pipeline, its underlying data, and any potential downstream task.

This Section clarifies our focus in this paper on integrity risks and specifically backdoor attacks as it is an integrity-related risk relevant to DNN-based FRS (at the reader's discretion, we provide key references and surveys in the DNN Confidentiality and Availability risks literature in App. B). To explain our choice, better understand the risk imposed by backdoors, and therefore develop appropriate mitigations, the following Section first provides an outlook on integrity-related risks that affect FRS in general. Following, we outline the integrity-related risks specific to DNNs that may be found in FRS. Consequently, we highlight how backdoor attacks stand as a novel but central problem.

A. DNN-NONSPECIFIC, INTEGRITY-RELATED CYBERSECURITY RISKS IN FRS

FRS pipelines are vulnerable to manipulation by malicious agents. The standard perspective on their security, provided in [38], [42], and [43], identifies eight cybersecurity vulnerabilities. Each one is associated with a particular pipeline step that an attacker aims to tamper with:

(A) **Forge Biometric** (res. face presentation attack). A malicious agent attempts to deceive a FRS by submitting a fake biometric input through the pipeline sensor. For instance, the attacker uses a face mask, a printed-out face on paper, or a 3-dimensional structure [60] to gain unauthorized access to an authentication-based online banking application.

(B) **Resubmit Biometric** (res. resubmission, replay attack). The attacker owns a biometric input that was previously sent and accepted by the FRS. Bypassing the sensor, the attacker resubmits the input to wrongfully gain access to the FRS task.

(C) **Hijack Model**. The malicious agent overrides one or more models within a FRS. For instance, by infecting the FRS with a malware, the attacker may force a model to output a specific feature representation to confound the downstream matcher. This risk is not specific to DNN and predates their development [38].

(D) **Hijack Feature**. The attacker intercepts and manipulates the final feature representation generated by the FRS models. For instance, the attacker may manipulate the signal over the communication channel between the extractor or the binarizer and the matcher.

(E) **Forge Template**. The attacker targets the storage location of identity templates or template galleries, for FRS authentication or identification respectively. Once the templates have been compromised, an attacker may override a FRS. The matcher will surreptitiously output erroneous decisions.

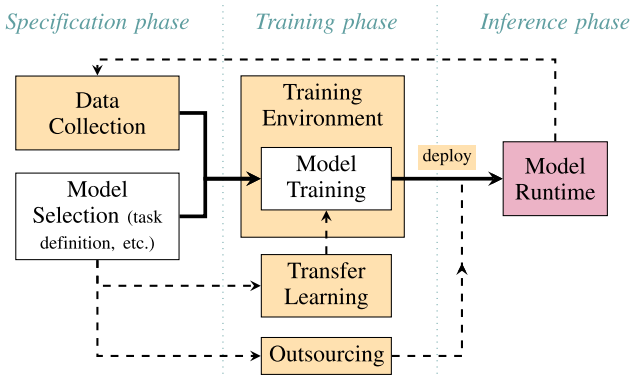

FIGURE 2. The schematic lifecycle of a DNN illustrates the different stages that are vulnerable during Training and Deployment in Orange. These vulnerabilities originate from a dependence on third-party services such as MLaaS providers and hosting services. In addition, a vulnerability to adversarial examples affects the runtime stage in Red because of potential malicious users interacting with the DNN during Inference.

(F) **Hijack Template**. As with feature hijacking (4), the attacker intercepts and manipulates a template during its transit from its storage location to the matcher.

(G) **Hijack Matcher**. As with model hijacking (3), the attacker may manipulate the matcher and alter its decision-making process to yield a desired output.

(H) **Hijack Decision**. The attacker intercepts and manipulates the matcher's decision to hijack the functionality of the downstream task.

These eight vulnerabilities are typically software-based, i.e., an attacker compromises a FRS during its runtime because of an insecure deployment by a user [38].

Here, we note that these vulnerabilities may overlap with confidentiality and availability risks. For instance with vulnerability (F), an attacker who hijacks templates at runtime may also cause confidentiality and availability problems by stealing template information or manipulating templates in transit to cause the matcher to fail.

Though this long-established perspective remains relevant today (e.g. system security and cancellable biometrics [62] as FRS protections), it unfortunately predates the rise of DNNs and therefore does not specifically take into account the unique vulnerabilities associated with them.

B. INTEGRITY-RELATED VULNERABILITIES ASSOCIATED WITH DNNS

DNNs introduce additional complexity to FRS pipelines due to the interaction between up to five distinct models, as illustrated in Fig. 1. Such complexity comes at a cost, with risks stemming both from the inherent nature of DNNs [66], [67] but also the fact that DNN developers tend to rely on third-parties across a model's lifecycle (see Fig. 2).

In the context of integrity risks, two additional vulnerabilities emerge with the use of DNNs that an attacker may exploit to manipulate and take over any DNN-based system:

(I) **Adversarial Attacks** (res. inference-time attacks, evasion attacks [68]): When interacting with a DNN (e.g. as part of a FRS), an attacker manipulates a face input in an innocuous or imperceptible manner (to the human eye), either in the digital or physical space [14], [69] such that it causes a DNN to yield an erroneous result, representation, or decision. Adversarial attacks causes a victim DNN to cross some decision boundary. To find this minimal, erroneous modification, an attacker iteratively explores the defects of a DNN's high dimensional representations [66], [67], doing so by studying the model's inputs and outputs.

In this context, adversarial attacks are closely related to the biometric forgery risk (A) listed in Section III-A.

(J) **Training & Deployment Attacks** (res. **backdoor attacks**, weight attacks, neural trojans [30]): Besides attacking a FRS with adversarial examples, an attacker may also manipulate a benign DNN found in a FRS by injecting it with a stealthy, erroneous behavior, i.e., a backdoor [12]. Though backdoor attacks emerged more recently compared to their adversarial counterparts [32], they have since developed into a diverse range of methods (see Section IV and App. C). Once the malicious DNN has been deployed by an unsuspecting user as part of a FRS pipeline, the attacker may activate the backdoor at any time during inference. For instance, activation may occurs by holding a trigger patch in front of a sensor.

To perform a backdoor attack, the attacker may either target the DNNs' training (e.g. at the data collection step by poisoning training samples [12] or by modifying the training regimen [14], see Fig. 2 and App. C), or during its deployment by manipulating the DNNs' weights directly (e.g. via the ownership of the the model's location in a device's memory [34]).

In this context, training and deployment attacks are a DNN-specific subset of the model hijacking risk (C) listed in Section III-A.

Adversarial attacks are exploratory [11] in nature, typically requiring the attacker to design an attack on-the-fly by interrogating a DNN. In the context of a FRS, this iterative approach may not be feasible, given the search space complexity of up to five DNNs (i.e. mounting an adversarial attack on the extractor requires the attack to traverse the detector, alignment, and antispoofing). Moreover, adversarial attacks may be difficult to carry out inconspicuously.

Meanwhile, backdoor attacks (whether injected during training or deployment) are causative in nature: an attacker who controls one or more step of a DNN's lifecycle may design an attack in advance, unbeknownst to any FRS end-user or client, and without prior warning at inference time.

This survey does not cover in details the mathematical definition of backdoor attacks. Instead, we point the reader to a recent formalization (of both adversarial examples and backdoors) found in the survey by Wu et al. [14].

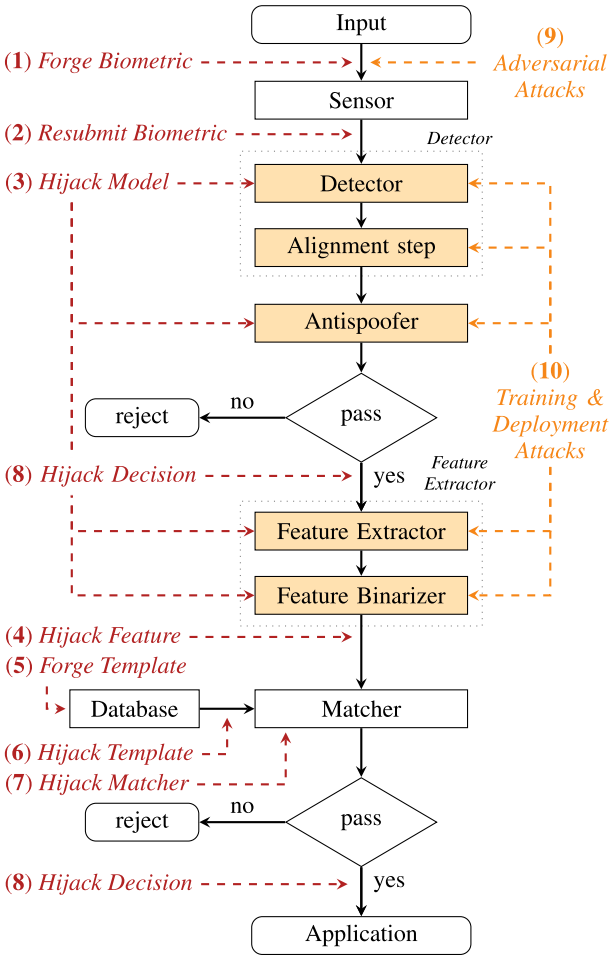

FIGURE 3. Vulnerabilities associated with a DNN-based Face Recognition System with stages that involve DNNs in Orange. General cybersecurity risks are identified in Red (left) and DNN-specific risks in Orange (right).

C. BACKDOORS AS A CORE DNN-RELATED INTEGRITY VULNERABILITY IN FRS PIPELINES

In the context of DNN-based integrity risks, we note two important problems [31]:

1) Detecting backdoors is a difficult problem,
2) the gradual gain in relevance of the MLaaS ecosystem and ML outsourcing overall increases backdoor risk.

DNNs require a increasingly data-intensive training phase [25] and, because of deployment and servicing requirements, often involve third-party hosting [4], [6], [7]. Backdoors are therefore of particular interest when considering the security of DNN pipelines, and in the context of this paper: FRS.

Consequently, reliance on third-parties, which materializes in a vulnerablity to backdoor attacks, is a core risk to integrate in a FRS threat model (see Fig. 3). This underlines our focus on backdoors in the rest of this paper.

It becomes evident that each DNN in a FRS may thus suffer from backdoor attacks whenever the FRS owner is dependent on ML outsourcing. From the attacker's perspective meanwhile, employing backdoors to attack a FRS may serve a wide variety of purposes depending on which DNN to hijack:

1) **Hijack Detector** (res. evasion or spoofing): The attacker injects a backdoor in the FRS detector. An attack then manipulates the number of detected faces at inference time, either to evade or spoof the detector and all subsequent steps in the FRS.
   In this context, Chan et al. [70] identify four types of backdoor attacks on general-purpose object detectors:
   (a) *global misclassification*: the presence of a backdoor trigger in an image causes *all* detected objects to be misclassified as another class,
   (b) *local misclassification*: the backdoor trigger causes objects in its direct vicinity to be misclassified as another class,
   (c) *object disappearance*: the backdoor trigger causes the detector to fail the detection of an object, seeing it as background for instance,
   (d) *object generation*: the backdoor trigger confounds the detector, which will erroneously see an object where there is none.
   Given this formulation, attacks (a) and (b) are out of scope for most face detectors and therefore the security of FRS (face detectors do not perform classification of different objects as they only detect faces). Attacks (c) and (d), meanwhile, respectively correspond to an evasion and a spoofing attempt on the part of the attacker.
2) **Hijack Alignment** (res. evasion): If the alignment module is separated from the detector, an attacker may backdoor alignment if it is DNN-based such that it perturbs the regression of facial landmarks. This results in a misaligned face that would subsequently fail matching by the FRS pipeline. This results in an evasion attack on the part of the attacker.
3) **Hijack Anti-spoofer** (res. spoofing): In this scenario, the attacker backdoors the anti-spoofer such that the FRS erroneously accepts a fake face at test-time, a typical spoofing attack as demonstrated in [71].
4) **Hijack Extractor** (res. evasion or impersonation): The attacker injects a backdoor in the FRS' feature extractor DNN to manipulate the result of the downstream matcher [14], [31]. This backdoor alters the DNN's output representation such that it becomes either significantly different from the attacker's true identity or close to that of another person. In the former case, the attacker aims to evade recognition (for instance in an identification FRS as illustrated in Joshi et al. [41]). In the latter, the attacker looks to impersonate a victim in order to be wrongfully authenticated (as in the system described in Pigeon et al. [39]) and gain access to otherwise confidential or private information.
5) **Hijack Binarizer** (res. evasion or impersonation): In a similar fashion to the FRS' feature extractor, an attacker

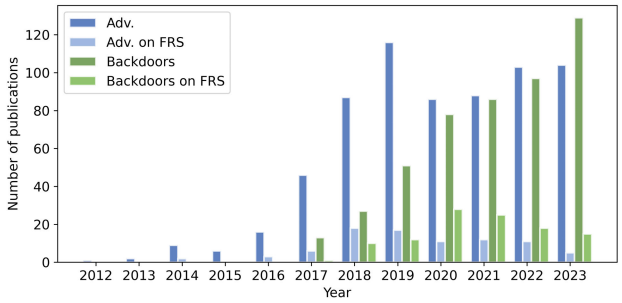

FIGURE 4. Number of adversarial and backdoor papers (attacks and defenses) published from 2013 to 2023 relevant to our research.

may instead try to backdoor a feature binarizer. Given the tight relation between extractor and binarizer, the latter suffers from similar risks and may therefore be targeted by an attacker for the same reasons.

As such, FRS pipelines are vulnerable to backdoors at each step involving a DNN. Moreover, backdoor attacks in FRS may be used to either spoof or evade such systems, indicating an integrity risk for both authentication or identification FRS.

D. TAKEAWAY

Backdoor attacks emerge as a new and pressing vulnerability to FRS. These attacks originates from underlying vulnerabilities both in terms of software [7], [72] but also hardware [73], [74] that see attackers gain access to, and the capacity to manipulate, either the training or deployment of a target DNN. A critical question therefore is whether backdoors in FRS have seen concrete implementations and, if so, whether defenses have also emerged in response.

Of note, as we will not cover the adversarial attack literature (including those impacting face recognition) further, we refer the reader to the following surveys: Akhtar et al. [75], Pydi and Jog [76], Wei et al. [77], and Wu et al. [14].

IV. BACKDOOR ATTACKS ON FACE RECOGNITION SYSTEMS

Since 2017 [32], the relevant literature on backdoor attacks and defenses has dramatically grown, including in the context of FRS security (see Fig. 4). As such, this Section delves into the first core topic of this survey: backdoor attacks with prior demonstration on DNNs found in FRS.

We cover backdoors following three dimensions (see Fig. 5 for a visual overview of these three dimensions):

1) **attack channel**,
2) **injection method**,
3) **trigger specifications**.

The backdoor **attack channel** highlights an attacker's assumed knowledge about and access to a victim's system, occurring either during the training or the deployment stages. This definition clarifies *where* the attacker executes the attack. Meanwhile, the backdoor **injection method** defines *how* an attacker manipulates a victim DNN and its end task. For instance, a victim may have fully outsourced their DNN training to a malicious agent who, despite this broad access, only manipulates the DNN's training data to inject the backdoor. Finally, the backdoor **trigger specifications** state *with what* input corruption the attacker chooses to activate the backdoor once the victim DNN has been deployed.

We underscore that two attackers may have similar levels of information and access to their victim but use different approaches to mount their backdoor attack. For instance, an attacker will not choose the same attack given the victim system uses a Vision Transformer (ViT) or a CNN (it was shown that either types of DNNs are not equally vulnerable to the same attacks [78]). Moreover, an attacker with a full access to a victim's system during a given time window may prefer mounting a backdoor attack via data poisoning instead of weight tampering as the latter only targets a single DNN. In comparison, data poisoning may impact more than one DNN in the future.

In this context, we highlight the significance of understanding an attacker's knowledge level, injection methods, and activation mechanisms to effectively mitigate backdoor risks. As such, a comprehensive list of backdoor attacks implemented on DNN used in FRS is found in the Tables 1-3. Further details regarding the underlying datasets and model architectures are provided in Fig. 6 and Fig. 7 in Appendix F.

A. CHANNELS FOR BACKDOOR ATTACKS ON FRS MODELS

A backdoor attack typically occurs through one of several channels, which encompass the knowledge and access afforded to an attacker. In the case of a vulnerable training stage, an attacker may hijack the following processes on the victim's side:

(1) the **data collection** process,
(2) the **training environment**,
(3) reliance on **ML outsourcing** (e.g. use of ready-to-use pipelines [4], [6]),
(4) the use of **transfer learning**,
(5) the **deployment** stage of a model.

These five attack channels are the direct results of having vulnerable stages in a DNN's lifecycle on the victim's side (previously illustrated in Fig. 2).

We note that some channels may overlap at time, such as channel (2) and channel (4) as an attacker may have access to a victim's training environment while also being able to tamper with pretrained DNN checkpoints beforehand. Similarly, channel (4) may be considered as a subset of channel (3), as illustrated in Fig. 5. We make the choice to separate the two in this Section because the literature of backdoor attacks carried through either channels do not typically assume the same level of knowledge on the part of the attacker.

The detailed taxonomies of backdoor attacks that target a DNN commonly found in a FRS are reproduced in the Tables 1-3.

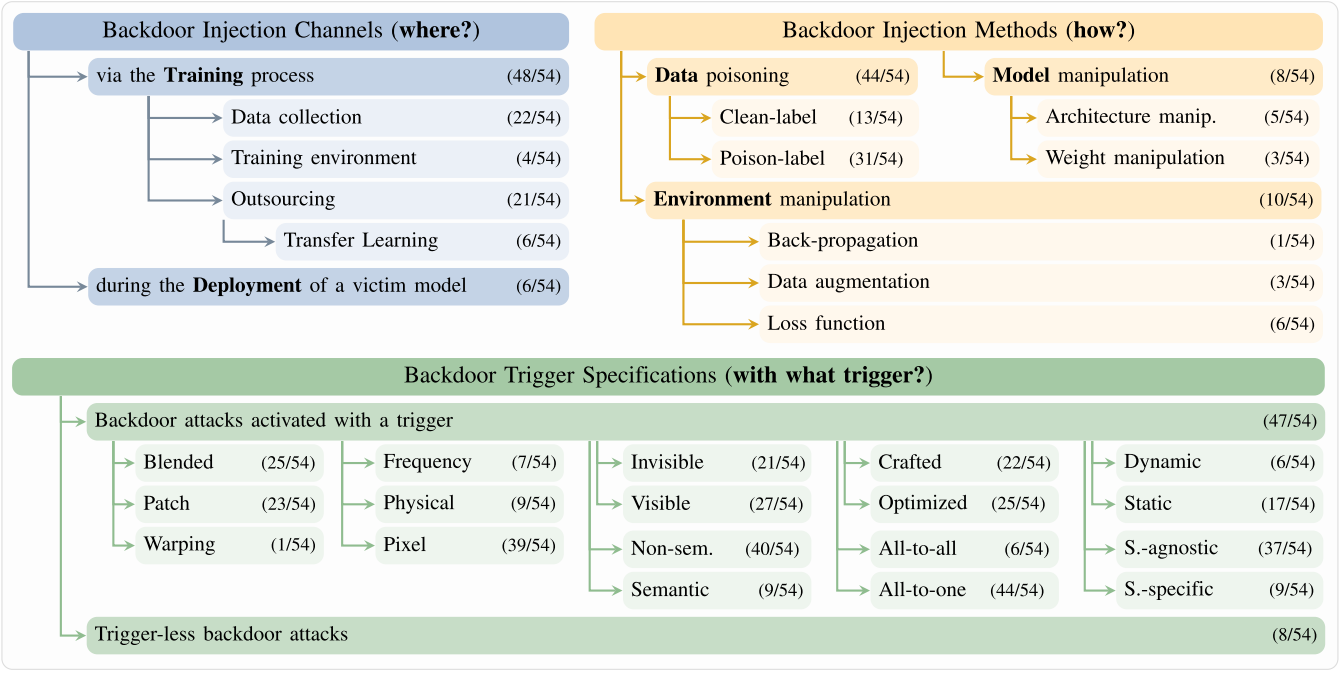

FIGURE 5. Dimensions of backdoor attacks: channels (where the injection occurs in a DNN's lifecycle), methods (how the backdoor is injected in a DNN), and trigger specifications (how an attacker activates the backdoor), alongside the respective number of identified papers associated with each subcategory (note: some papers cover more than one case (e.g. a paper may cover both a data collection and outsourcing use cases, etc.).

1) DATA COLLECTION

An attacker looking to embed a backdoor into a DNN may exploit a victim's vulnerable data collection process. In this situation, the attacker lacks control and even prior knowledge about the victim's DNN architecture, training environment, or test-time inputs. As such, the attacker may only manipulate a victim's training data by concealing a pattern within a portion of the vulnerable dataset in a process called data *poisoning* (see Subsection IV-B for further details).

For backdoor attacks mounted via the data collection process, model training happens offline from the perspective of the attacker. Therefore, the designed backdoor must maximize its likelihood of being learned by a target DNN. For instance, the attacker must content with the prospect of data augmentation policies erasing a backdoor [79].

Targeting the data collection process was initially used in untargeted attacks where a malicious agent disrupts a DNN's test-time performance [10] (e.g. availability attack in federated learning). Rapidly however, this attack channel became prominently featured as an integrity vulnerability in the backdoor attack literature [32], [71], [80], [81], [82], [83], [84], [85], [86], [87], [88], [89], [90]. In this context, attacks are targeted (i.e. the attacker aims to produce a specific, malicious behavior) and designed for stealth.

In this survey, we identify 22 attacks on a DNN used in FRS that explore a use case where only the data collection channel is hijacked.

2) TRAINING ENVIRONMENT

An attacker without an access to a victim's training data may instead target the training environment of the victim's DNN. For instance, the attacker may manipulate the update procedure of a DNN's weight during training [91]. The attacker is therefore able to tamper with the model and inject it with a backdoor behavior. This use case, where the attacker compromises a training environment to manipulate DNNs at a deferred time (the attacker is not limited to attacking the training environment during its runtime), is typically referred to as a "blind" backdoor attack [91].

In this survey, we identify 4 attacks on a FRS DNN that only rely on hijacking the training environment.

3) ML OUTSOURCING

A combination of the previous two channels, model outsourcing is emblematic of the MLaaS economy and outsourcing as a whole. In such context, a company, institution, or person acquires an already-trained model from a third-party instead of developing in internally. the framework of model outsourcing thus offers the most freedom to a malicious agent looking to embed a backdoor in a DNN [12].

Because the attacker has free rein over the end task, they may resort to a combination of methods to inject a backdoor into a DNN (see Subsection IV-B for detail on injection methods). Contrasting with the data collection channel or the training environment channel however, the attacker has total knowledge of the target DNN alogn with a total control over both data and environment. The attacker does not have to

contend with the possible erasure of a backdoor due to a data augmentation policy as with the data collection channel for instance. Therefore, the backdoor trigger may be crafted in close relation with the target model, enabling stealthier, more robust trigger designs as explored in [92] and [93].

In this survey, we identify 21 attacks on a FRS DNN that cover the model outsourcing use case.

4) TRANSFER LEARNING

Transfer learning is as an subcategory to the model outsourcing channel. Some backdoor attacks that are originally designed for the latter case also consider the scenario where the backdoored DNN is reused by some unknown user for a similar task in a process called transfer learning [94]: the backdoored DNN is reused *and* typically retrained (with clean data) following, e.g., a fine-tuning [92], architecture modification [95], and/or knowledge distillation [96] approach.

If the backdoor survives this process, the unsuspecting user becomes a new backdoor target. Consequently, a malicious agent looking to perform backdoor attacks via transfer learning follows the same process as in model outsourcing, training a backdoored DNN using any method of their choosing. However, because the attacker lacks prior knowledge about (1) the data the model will be fine-tune with and (2) the task the new DNN is meant for, the attacker must consider the survivability of the backdoor to any model update [97]. This challenging attack scenario is particularly relevant in the context of ML outsourcing since DNN developers regularly use transfer learning to expedite development.

In this survey, we identify 6 attacks on a FRS DNN that explore backdoor attacks via the transfer learning channel.

5) DEPLOYMENT

Finally, an attacker may lack the access to a victim's DNN during training, its training dataset, and/or its environment. This situation occurs, for instance, when the victim forgoes using model outsourcing or transfer learning. Under such condition, an attacker cannot inject a backdoor in a DNN during training.

Nevertheless, an attacker may still be able to inject the backdoor into a trained model if they have access to its deployment pipeline or storage location on a vulnerable machine. Such attacks are categorized as deployment attacks where the attacker manipulates the DNN's trained weights and/or its structure ahead of its runtime [34], [98] to encode the desired backdoor.

Compared to the channels that target DNN training (see Fig. 5), targeting the DNN deployment stage is less represented in the FRS backdoor literature, accounting for 6 of inventoried attacks.

B. BACKDOOR INJECTION METHODS IN THE CONTEXT OF FRS MODELS

Beyond understanding the different channels they may leave DNNs vulnerable to tampering, FRS developers must also understand and contend with the different injection methods available to an attacker. This Subsection covers the backdoor injection methods found in the FRS backdoor literature, broadly split in three categories:

(1) **data poisoning**,
(2) the **training environment tampering**,
(3) reliance on **architecture and weight manipulation**.

We note that injection functions may overlap at time, such as data poisoning (1) and training environment tampering (2). An attacker carrying their attack via the model outsourcing channel (3) is free to employ either or both at the same time.

1) DATA POISONING

An attacker looking to inject a backdoor in a DNN but who lacks direct access to it may rely on manipulating the DNN's training dataset to do so. Aptly called "data poisoning," this technique involves the manipulation of training datapoints such that they include one or more trigger patterns to be learned by the victim DNN. The attacker aims for the pattern to be strongly associated with a given class such that using it on a test-input will lead to a *targeted* misclassification with high likelihood.

In the context of supervised learning, data poisoning attacks typically follow two variants (We provide a short formalization of backdoor attacks in Appendix C, and point the reader to several relevant surveys):

(1) **poison-label**, or inconsistent-label [99], poisoning,
(2) **clean-label**, or consistent-label [99], poisoning.

The difference between the two hinges on whether the attacker also alters a portion (also known as poisoning ratio) of an otherwise benign dataset by modifying both datapoints and their labels, or only said datapoints. In the **poison-label** case, the attacker selects datapoints from *source*, benign classes, manipulates them with a trigger pattern, then modify their source label to a given *target* label [12]. In the **clean-label** case, the attacker selects datapoints from the target class direct and modifies them with the pattern [99]. In either cases, the attacker aims for the pattern to be strongly associated with the target class. We note that more recent and less widespread methods, such as label-only poisoning [100], exist but lack implementations on FRS tasks to the best of our knowledge. They are therefore left out of this paper.

Given the same poisoning ratio, poison-label backdoors typically achieve higher attack success rate (*ASR*) than clean-label backdoors. However, they rely on *detectably* misclassified backdoored instances. Clean-label backdoor attacks were introduced to enhance the stealthiness of backdoor attacks. However, they typically come at the cost of a lower *ASR*, or require a relatively higher poisoning ratio to achieve an on-par performance with poison-label methods. For instance, poison-label backdoors like Chen et al. [32], ISSBA [83], WaNet [84], or SIBA [89] use poisoning ratios of up to 20%, 10%, 15%, and 5% respectively. Meanwhile, clean-label backdoors like SIG [80], Bahlero et al. [71], or Guo et al. [101] use ratios of up to 40%, 50%, and 40%

respectively (i.e., the attacker needs to poison 40-50% of the datapoints of the target class to work).

Overall, data poisoning is the oldest and most prevalent backdoor injection method [12], [32], a trend that also applies to FRS backdoor attacks. In this paper, we inventory 44 relevant papers that rely, at least in part, on a data poisoning injection method (see the Tables 1-3).

2) TRAINING ENVIRONMENT MANIPULATION

Besides injecting a backdoor with data poisoning to backdoor a DNN at training-time, an attacker may also tamper with the DNN's training environment itself. In this scenario, backdoor injection methods applied to FRS DNNs manipulate one or more of the following processes:

(1) the **data augmentation policy**,
(2) the **loss function** computation,
(3) the **gradient backpropagation**.

Manipulating a DNN's **data augmentation policy** is the training environment-based injection method that closest resembles data poisoning (see Section IV-B1). If an attacker cannot control a victim's data collection, they may instead target the training environment's dataloader. The Flareon attack [102] illustrates such an attack on a FRS DNN. Using Flareon, the attacker introduces class-dependent patterns in benign inputs through the dataloader's data augmentation step (a process similar to clean-label poisoning), without accessing the underlying dataset at its storage location.

An attacker may also hijack the **training loss function**, core to any DNN task learning. In this scenario, the attacker designs a malicious loss function and injects it in lieu of the victim's original loss. The attacker's loss typically introduces an additional, malicious objective into the learning process of a target DNN. For instance, Bagdasaryan and Shmatikov [91] demonstrate the use of a malicious Multiple Gradient Descent Algorithm (MGDA) in a FR task. The attacker swaps the original, benign loss with a MGDA multi-task learning objective, which balances learning the victim's original task. The attacker thus avoids early detection while promoting their own malicious, backdoor objective.

Finally, an attacker may be able to directly modify a victim DNN's **gradient backpropagation** step, as illustrated by Ji et al. [95] on a FR model. In this paper, the attacker identifies the salient features of a source and a target input identity, and compels the victim DNN to minimize the distance between the two to cause malicious feature collisions. The source inputs thus become backdoors.

Training pipeline backdoors enable stronger and/or stealthier attacks, especially when combined with data poisoning, which occurs when the attacker launches an attack via the model outsourcing channel. For example in a FR task, the Composite Attack [103] leverages the combination of a malicious data augmentation policy and a poison-label poisoning strategy to enhance the stealthiness of their backdoor. The authors exploit the content of the training data itself to generate a trigger, e.g., a combination of faces. Meanwhile, the DeepPoison attack [104] uses a generator-discriminator setup akin to generative adversarial networks (GANs) to jointly refine the poisoning ratio of a data poisoning strategy, find a fitting backdoor trigger (e.g., an unaltered image from the training dataset), and maliciously train the target DNN. A final example is found in Zhong et al. [105] where the attacker trains a backdoor trigger generator alongside a victim FR model so as to generate stealthy backdoor patterns that result in DNN feature collisions.

Overall, manipulating the training environment is the second most popular backdoor injection framework in the context of DNNs found in FRS, accounting for 10 inventoried papers. However, this method is rarely used as a standalone in FRS attacks. Approximately two-third of cases use a mixed strategy of data poisoning and training environment manipulation as part of attacks via the model outsourcing channel, including transfer learning (see Tables 1-3).

3) MODEL ARCHITECTURE AND WEIGHT MANIPULATIONS

When both data and training environment are inaccessible, an attacker may resort to manipulating the weights and/or architecture of a target DNN. The backdoor injection typically occurs during the deployment of the target DNN (e.g. while it is being stored or in transit to an inference environment). When considering backdoor attacks on FRS DNNs, we inventory two approaches open to an attacker:

(1) **model architecture manipulation**,
(2) **targeted weight perturbations**.

Model architecture manipulation typically involves grafting a malicious subnetwork to an otherwise benign DNN. The TrojanNet attack [106] illustrates this method in the context of a FRS by training a subnetwork that detects a backdoor trigger in an input face image. Whenever the trigger is detected, the activations of the subnetwork's output layer overwhelm those of the victim FR model. The SRA attack [98] refines the TrojanNet approach by merging the malicious subnetwork within the original model for greater stealth, replacing a set of low-impact neurons in the process. The HuFu attack [33] approaches architecture manipulation differently by leveraging the interaction between a victim model and the hardware used to run it. Both model and hardware are backdoored to collaboratively detect the backdoor trigger specified by the attacker. When the trigger is detected, a subset of the victim DNN's neurons is deactivated, revealing a subnetwork that performs the backdoor task designed by the attacker. Similarly, the attack found in Salem et al. [107] manipulates a victim DNN's neuron dropout process, a common element in DNN training pipelines, such that dropout persists after training. Dropout is used to design a backdoored subnet in a DNN such that, by using a neuron dropout, the attacker forces the DNN to degrade into a backdoored subnet.

Instead of starting with an separate backdoored subnetwork, an attacker may instead identify specific sets of neurons to backdoor in a victim DNN. Such injection methods are

known as **targeted weight perturbation** attacks. These methods' selection process typically identifies redundant subnets or paths in the DNN comprised of so-called inactive, or dormant [108], neurons (i.e. modifying them will not cause the model's performance to drop in benign conditions). Once identified, these DNN elements are amplified to activate in the presence of an attacker's backdoor trigger, creating a malicious shortcut through the victim DNN [109]. This process depends on the model's capacity to carry both the backdoor and main tasks and hinges on the characteristic overparametrization of DNNs [110]. The Dumford and Scheirer attack [34] illustrates this method by leveraging a search algorithm to identify and then modify the weights of specific, sometimes disjoint, neurons in a FR model, introducing a backdoor. Similarly, the Hong et al. attack [109] crafts a backdoor by adjusting the weights of convolutional filters in a FRS CNN. These modifications ensure that the original task remains intact while compromising the model's behavior when exposed to a backdoor trigger.

Overall, model architecture and targeted weight manipulation are a minority among the backdoor attacks previously demonstrated on FRS DNNs, accounting for 8 inventoried papers (see Tables 1-3).

C. SPECIFICATIONS OF FRS BACKDOOR TRIGGERS

A final dimension of backdoor attacks targeting DNNs found in FRS revolves around the categorization of their triggers. As an attacker is free to design the backdoor trigger, a multitude of variations have emerged since the early work of BadNets [12] and Chen et al. [32]. As such, this Section outlines eight key trigger categories (see Fig. 5):

1) Format: **blended**, **patch**, *or* **warping** triggers
2) Injection space: **frequency**, **physical**, *or* **pixel** triggers
3) Visibility: **invisible** *or* **visible** triggers
4) Design: **handcrafted** *or* **optimized** triggers
5) Patch location: **dynamic** *or* **static** patch-based triggers
6) Semantics: **non-semantic** *or* **semantic** triggers
7) Target: **all-to-one** *or* **all-to-all** triggers
8) Specificity: **sample-agnostic** *or* **specific** triggers

Additionally, we will also cover **trigger-less** backdoors, a distinct subset of semantic backdoors that eschews the use of common triggers for purposely designed feature collisions between benign datapoints.

We preface the rest of this Section with the note that these categories are not exclusive to each other but rather provide a high-level outlook of the 54 backdoor attack papers inventoried in this survey.

1) FORMAT: BLENDED, PATCH, OR WARPING TRIGGERS

Stealth is a crucial requirement for attackers who look to inject backdoors in a victim DNN. To evade detection, an attacker's initial consideration revolves around whether to localize a backdoor within an input, e.g. with a patch, or to seamlessly blend a trigger in the input.

The Liu et al. attack [92] illustrates **patch**-based backdoors by following the example of BadNets [12] and altering a square-sized patch of pixels in a target image. Such patch-based techniques were subsequently adopted by the Trembling Trigger attack [37] to execute an impersonation attack on a FRS for instance.

In contrast, **blended** triggers do not replace content at some location but instead superimpose a diffuse pattern across the whole range of a target image. For instance, the SIG attack [111] uses a malicious sinusoidal signal as a trigger, which can be used to backdoor a DNN for impersonation attacks [80]. Using a more intricate approach, the recent attack by Zhang et al. [88] employs an AE architecture to blend a trigger drawn from a predefined set into a face image, enabling impersonation attacks on a FR model.

Lastly, the WaNet attack [84] is noteworthy for setting aside the patch versus blended trigger binary. Discarding the additive operation between an original image and a backdoor pattern (a characteristic of the previous two methods), WaNet performs image **warping** as a backdoor trigger. The attack distorts a target image by shifting its pixels following a thin plate spline interpolation method [112].

This survey identifies 25 patch-based, 23 blended, and 1 warping-based backdoor attacks with an implementation on a DNN found in a FRS (see Tables 1-3).

2) INJECTION SPACE: FREQUENCY, PHYSICAL, OR PIXEL TRIGGERS

To accommodate the different backdoor scenarios open to an attacker, backdoor triggers are tailored to occur in either digital or physical space.

In the digital space, triggers may consist in the direct manipulation of image **pixels**, as captured by a sensor, or intervene in the **frequency** space of the image. For instance, Yao et al. [97] reuses pixel-based triggers from the BadNets [12] and Trojan Attack [92] papers to corrupt a target FR model and enable impersonation attacks. Meanwhile, frequency-based attacks design their trigger in the frequency space of an image after applying a transform like the Discrete Cosine Transform (DCT), as illustrated by Zeng et al. [113]. Here we note that attacks like SIG [111], although their triggers could be expressed as coefficients in a DCT space, are implemented with pixel-based triggers.

Physically-implemented backdoors pose a more significant threat to FRS security due to their real-world applicability. **Physical** attacks typically aim to port triggers designed digitally into physical space. For instance, the Trembling Trigger attack [37] trains a BadNets [12]-style backdoor to function once printed on a piece of cardboard to attack a FR task. We also find attacks purposely designed for physical cases such as Wenger et al. [81]. The attack uses objects to backdoor a FR task, including: three dots affixed to a cheek, sunglasses, tattoos, white tape on a forehead, a bandana, and a pair of earrings. Similarly, Bhalero et al. [71] manipulates the luminosity in the attacker's environment to backdoor a video-based antispoofing system.

This survey identifies 7 frequency-based, 9 physical, and 39 pixel-based backdoor attacks with an implementation on a DNN found in a FRS (see Tables 1-3).

3) VISIBILITY: INVISIBLE OR VISIBLE TRIGGERS

Regardless of their construction strategy, backdoor attackers increasingly consider trigger inconspicuousness in their design due to stealth requirements in the face of evolving defenses (see Section V for a coverage of backdoor defenses implemented in the context of DNNs found in FRS).

Traditional backdoor attacks on FRS, such as FaceHack [114] or the real-life light fixture employed by Li et al. [82], typically assess their effectiveness against detection methods but remain relatively visible to the human eye. Achieving stealth against both machine defenses and human scrutiny is a more recent focus. An early example in the FRS context is found in Bhalero et al. [71], where the backdoor attack exploits the luminance of a video stream to remain relatively imperceptible to both detectors and human observers. Similarly, Liu et al. [92] experiment with the transparency of a patch-based backdoor trigger originally stamped on an image. The singular WaNet attack [84] uses image warping to achieve imperceptibility. In the frequency domain, the FTROJAN attack [86] explores the slight alteration of the frequency coefficients of facial inputs to embed a backdoor pattern invisible to the human eye.

This survey identifies purportedly 21 invisible and 27 visible backdoor attacks with an implementation on a DNN found in a FRS (see Tables 1-3).

4) DESIGN: HANDCRAFTED OR OPTIMIZED TRIGGERS

As backdoor attackers seek more effective and stealthier backdoors, they encounter an increasingly larger panel of defenses (see Section V). Early methods like BadNets [12] share the characteristic that they are **handcrafted** by the attacker, i.e. the task, dataset, and target model do not influence the trigger's construction. For instance, Chen et al. [32] overlays a cartoon character or a predefined noise pattern into a target face image, whereas BadNets uses a pixel pattern manually crafted by the attacker.

Given the growing brittleness of such backdoors, attackers have sought more powerful methods using **optimization** techniques (e.g. search algorithm [80]). One of the earliest optimization-based methods applied to a FRS DNN is the Trojan Attack [92]. This method starts with a pretrained DNN. The attacker selects a pixel region in face inputs and mines the pixel replacements such that the resulting pattern maliciously activates specific neurons in the victim DNN. Using fine-tuning, the attacker then embeds the backdoor in the DNN. Backdoor attacks that optimize their trigger typically rely on sophisticated, difficult-to-detect approaches. For example, DFST [115] leverages a StyleGAN network to simultaneously learn a backdoored model and refine the associated trigger injection function. Additionally, the ISSBA attack [83] and the method described in Zhang et al. [88] both use AEs to conceal an initially handcrafted trigger in a target input. Whereas these two attacks focus on the pixels of a face image, the attack provided by Yu et al. [116] demonstrate that an optimized backdoor trigger is also feasible in the frequency domain.

This survey identifies purportedly 22 handcrafted and 25 optimized backdoor attacks with an implementation on a DNN found in a FRS (see Tables 1-3).

5) PATCH LOCATION: DYNAMIC OR STATIC PATCH-BASED TRIGGERS

Patch-based backdoors like BadNets [12] often rely on a **static** trigger. That is, the location of the trigger in a backdoored input is one of the necessary conditions (along with the trigger's pattern) for the activation of the backdoor in a DNN. Unfortunately, this is a restrictive working condition for backdoor attacks, especially for backdoors in the physical space and/or targeting FR tasks (i.e. a backdoored image fed to a feature extraction DNN depends on the detector's output bounding box and the subsequent face extraction and alignment).

Prior work addresses this problem with **dynamic**, patch-based backdoor attacks, as illustrated on a FR model by Salem et al. [117] or the Poison Ink attack [85]. Dynamic triggers aim to enhance the stealthiness of a backdoor by confusing defenses that are designed to reconstruct a backdoor trigger (see Section V for trigger reconstruction defenses).

This survey identifies purportedly 6 dynamic and 17 static backdoor attacks with an implementation on a DNN found in a FRS (see Tables 1-3).

6) SEMANTICS: NON-SEMANTIC OR SEMANTIC TRIGGERS

Up until now, we have covered attacks that typically rely on triggers that are independent from the underlying task and dataset (e.g. a pixel pattern as in BadNets [12] and Trembling triggers [37] or a sinusoidal signal as in SIG [111]). These attacks are **non-semantic**.

Instead, **semantic** backdoor attacks exploit the context of a training dataset to create their associated trigger, making them less detectable, particularly to the human eye. This is particularly relevant in the context of FRS where there are several examples of semantic backdoor attacks. For instance, the Refool attack [80] exploits natural reflections in photographs to create a backdoor in a FR task. Similarly, Wenger et al. [81] provide examples of semantic attacks using physical accessories like glasses, earrings, or a bandana. Finally, the Composite Attack [103] designs an impersonation attack where the trigger consists in having two specific persons in an image instead of one, and who end up being identified as a single, different, and thus erroneous victim identity.

This survey identifies 40 non-semantic and 9 semantic backdoor attacks with an implementation on a DNN found in a FRS (see Tables 1-3).

7) TARGET: ALL-TO-ONE OR ALL-TO-ALL TRIGGERS

The majority of backdoor attacks discussed thus far are **all-to-one** attacks That is, a backdoor's target is independent of the source image. In a supervised learning context, the target class is independent from the source class of a backdoored input (see formalization details in Appendix C).

In contrast, **all-to-all** backdoors use a *unique* trigger pattern to yield different predictions dependent on the source class (or identity in a FRS context) of the input being backdoored (this behavior is typically injected during DNN training). First defined in BadNets [12], this setup is used in WaNet [84] and BppAttack [118] for instance to demonstrate a multi-target scenario in which the class of a backdoored input is predicted as the next in the output of the victim DNN, i.e., given the source class y, the target class is $y + 1$ (see formalization details in Appendix C).

Here, we note that Flareon [102] also defines a setup referred to as "any-to-any," in which a backdoored model learns multiple all-to-one triggers, one for each target class within the model. This setup, although not explicitly named, is also found in [95], [106], [116], and [119].

This survey identifies 44 all-to-one and 6 all-to-all backdoor attacks with an implementation on a DNN found in a FRS (see Tables 1-3).

8) SPECIFICITY: SAMPLE-AGNOSTIC OR SAMPLE-SPECIFIC TRIGGERS

The backdoors covered so far are **sample-agnostic** backdoors, i.e., the associated trigger pattern, once selected by the attacker, does not depend on the underlying input being poisoned (e.g. BadNets [12] or Wang et al. [93]).

However, there exist **sample-specific** backdoors that learn backdoor injection function that tailors backdoor pattern to the underlying inputs. For instance in the context of FRS, the ISSBA attack [83] exploits an AE to transform a face image poisoned with a sample-agnostic trigger into a sample-specific backdoored image. Similarly, the COMBAT attack [119] designs an injection function that generates specific triggers that depend on the face image to poison. We note here that the concept of sample-specificity differs from the previously mentioned optimized and semantic categories, where backdoor triggers are designed in relation to either the victim DNN or the task dataset as a whole.

Sample-specific backdoors are harder to mount, e.g., they require access to the whole dataset and/or training environment, precluding attacks via the data collection channel [120]. However, sample-specific backdoors typically enjoy higher stealth for an equivalent *ASR* compared to sample-agnostic backdoors [83], [120]. They also evade several key defenses that rely on the assumption that a backdoor trigger is sample-agnostic (see Section V).

This survey identifies 37 sample-agnostic and 9 sample-specific backdoor attacks with an implementation on a DNN found in a FRS (see Tables 1-3).

9) TRIGGER-LESS BACKDOORS

We previously noted that papers like Ji et al. [95] and Bagdasaryan and Shmatikov [91] build backdoor triggers that cause feature collisions between inputs of source and target classes (or identity in a FRS context). Such methods help create inconspicuous backdoors to human observers while easily confusing DNNs. It happens that such feature collisions can be used by an attacker to entirely eschew their need for a trigger pattern, resulting in so-called **trigger-less** backdoors.

Trigger-less backdoors rely on training a victim DNN such that the features of a set of *benign* inputs (from one or more source classes or identities) collide with the features of a given target. This turns the benign inputs into triggers themselves without the need for the attacker to manipulate them at runtime (as all other backdoors seen so far require). For instance, Bagdasaryan and Shmatikov [91] demonstrate a trigger-less backdoor in a FR task, where the attacker causes a reference person to be seen as a target victim at test-time, without modifying the reference's image (the collision happens in feature space). Similarly, the DeepPoison [104] and BlackCard [121] attacks create backdoors by poisoning training data such that the attacker can cause feature collisions between two identities without any test-time modification.

We also note the work of Salem et al. (2) [107], who take a different approach by hijacking the dropout structure of a model to cause test-time feature collisions between two otherwise benign faces. This allows the attacker to eschew both the need for data poisoning *and* test-time triggers. Finally, we also point to Master Key [122], a noteworthy prior work that demonstrates a trigger-less backdoor attack where a single trigger identity, without test-time alterations, can impersonate any identity in an authentication FRS pipeline, albeit in a closed set (i.e. the FR task is a classification task with the same identities between training and test sets).

This survey identifies 8 trigger-less backdoor attacks with an implementation on a DNN found in a FRS (see Tables 1-3), compared to 47 trigger-based ones.

D. TAKEAWAY

Backdoor attacks fit a variety of attack channels, injection methods, and trigger specifications. This diversity of attack strategies underscores the need to first assess the risks associated with critical FRS pipelines and then develop defenses against these risks, with backdoors as a central concern.

It is important to note that most backdoor attacks discussed in this Section have primarily focused on the face extraction stage of a FRS pipeline, without considering their feasibility within the FRS' broader context. This aspect will be further explored in Section VI, highlighting a standing point of contention in backdoor countermeasures applied to FRS.

V. DEFENSES AGAINST BACKDOOR ATTACKS ON FACE RECOGNITION SYSTEMS

This section surveys backdoor defenses applied to FRS DNNs, following two dimensions (see Fig. 6 for a visual overview of these two dimensions):

(1) **defender knowledge**,
(2) **backdoor defense specifications**.

Defender knowledge represents *what* level of access to a model and to any involved datasets a defender needs to implement a backdoor defense. This is particularly important in the context of DNN outsourcing and hosting (e.g. MLaaS) where user access, e.g. that of the defender, varies greatly (e.g. from full, white-box access to being restricted to a black-box API). Meanwhile, the specifications of backdoor defense state *how* a defense works to mitigate a backdoor attack.

As discussed in Section IV regarding attackers and the wide array of attacks at their disposal, two defenders with the same level of access to a suspicious DNN may use different approaches to thwart backdoor attacks. Understanding a defender's knowledge and the defenses available to them is therefore crucial.

A comprehensive list of backdoor defenses implemented on DNN used in FRS is found in the Table 4 and Table 5.

A. DEFENDER LEVELS OF KNOWLEDGE ABOUT FRS DNNS AND THEIR TRAINING ENVIRONMENT

1) A VARYING DEGREE OF ACCESS TO DATA

Given a suspicious DNN, backdoor defenders must determine whether they have access to (see Fig. 6):

(1) the **original**, **possibly poisoned**, **training dataset**,
(2) a **clean version of the training dataset**,
(3) a **clean validation dataset**,
(4) a **clean test dataset**,
(5) a **synthesized dataset**.

The question of access matters because of the prevalence of data poisoning as a backdoor injection method (see Subsection IV-B1), and therefore the need to protect against it. Moreover, it is in the defender's interest to know the distribution of the data a suspicious DNN has been trained on. This level of knowledge informs at which stage of a model's lifecycle a defense may be built as well as the specifications it may ultimately follow.

a: ACCESS TO A DNN'S ORIGINAL, POSSIBLY BACKDOORED, TRAINING DATASET

A first setting considers the use case when, for instance, the defender themselves trains a DNN albeit with an insecure dataset (e.g. a dataset provided by a third-party). The defender cannot assume the dataset is benign.

Defenses in this setup typically protect against data poisoning vulnerabilities, e.g., developing a robust training procedure against the presence of an unknown amount of poisoned datapoints in a given training dataset. For instance, the Confoc defense [123] proposes a heuristic to erase backdoor triggers contained in training sample before retraining a victim DNN.

In a context where model training is outsourced to a third-parties, intellectual property may forbid a DNN defender from accessing a provider's proprietary data. This results in this use case being found in only 6 defenses with an implementation on a FRS model (see Table 4 and Table 5).

b: ACCESS TO A CLEAN VERSION OF THE TRAINING DATASET

This setup is often assumed in the case where the DNN defender is also the one providing, at least in part, the training dataset [124], [125], [126], [127] to a third-party DNN trainer. As such, defenders may verify the trainer's DNN by interrogating it with the data it is supposed to be trained on. This allows, for instance, Unnervik & Marcel [124] to design a statistical test that differentiates the trained weights of benign and backdoored models using Gaussian Mixture Models. This requires access to the clean dataset to train benign model such that a defender can verify the model provided by a third-party.

We inventory 4 such defenses (see Table 4 and Table 5).

c: ACCESS TO CLEAN VALIDATION DATA

This setup considers the case when a DNN defender has held out some clean data from a training set for validation purposes to verify a third-party's DNN training. This situation typically arises from contracting an untrustworthy third-party to train a DNN on a pre-defined dataset, which the user may also provide, as illustrated in the Fine-Pruning defense [128]. Keeping aside a pristine subset of the training data enables the defender to verify that the training done by the third-party is trustworthy.

We inventory 13 such defenses (see Table 4 and Table 5).

d: ACCESS TO CLEAN TEST DATA

Neural Cleanse [129] is the first defense applied to FRS models that explores the use of clean test data (i.e. derived from a dataset with a different distribution than the one used to train a DNN) as a basis for defending against backdoors. The paper shows that there is only a marginal practical difference between using training, validation, and test data for defense design when the underlying task is similar.

We identify 18 such defenses (see Table 4 and Table 5), which assume access to a clean test dataset purposely gathered by the defender.

e: ACCESS TO SYNTHESIZED DATA

Finally, 2 defense papers (see Table 4 and Table 5) applied to FRS models consider a defender who cannot assume any access to any clean dataset [130], [131].

This setup stems from the backdoor vulnerability associated with data collection (i.e. data poisoning). Instead, these defenses eschew data gathering and instead synthesize

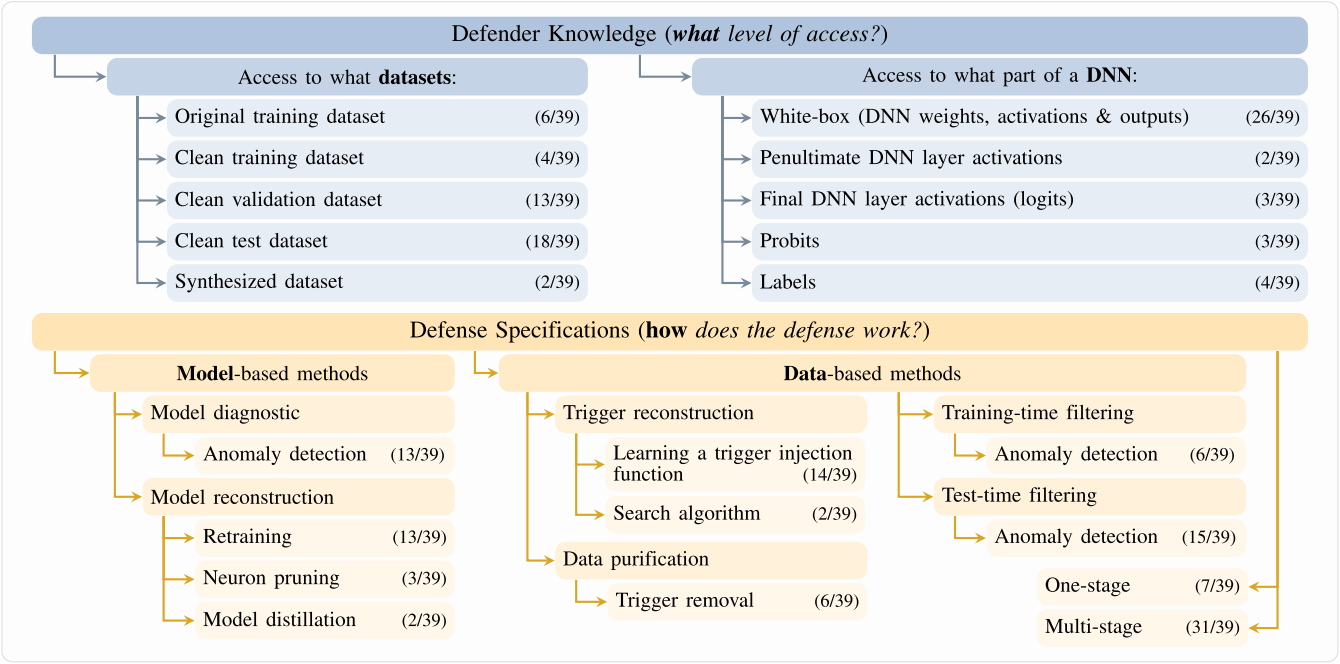

FIGURE 6. Dimensions of backdoor defenses with **defender knowledge** (the level of access of a defender with respect to the underlying datasets & models), and the defense **methods** specifications methods (how the defense acts against backdoor attacks), alongside the respective number of identified papers associated with each subcategory (note: some papers cover more than one case).

datapoints directly from the scrutinized DNN via model inversion [130] or distillation [131].

2) A VARYING ACCESS TO A FRS DNN

As with datasets, backdoor defenders may have different levels of access to a suspicious DNN (see Fig. 6). In the case of defending FRS DNNs against backdoor attacks, 5 levels of access typically occur:

(1) **white-box** access,
(2) **penultimate layer activations**,
(3) final layer activations or **logits**,
(4) **probits**,
(5) **labels**.

a: WHITE-BOX DEFENSES

A DNN defender may require a full access to a model's weight parameters, activations, and outputs. That is, a DNN's internal states are exposed to the defender's scrutiny (e.g. during the forward propagation of training or test-time inputs). Neural Cleanse [129] illustrates such setup as it requires a full access to a DNN.

In the context of defending FRS models, we identify 26 such defenses (see Table 4 and Table 5).

b: PENULTIMATE DNN LAYER ACTIVATIONS

Some defenses rely only on a partial white-box access. For instance, the SCAn [132] and Activation Clustering [133] defenses make use of a single hidden layer within a suspicious DNN (generally the penultimate one).

We identify 2 such defenses (see Table 4 and Table 5).

c: ACCESS TO EITHER LOGIT, PROBIT, OR LABEL OUTPUTS OF A DNN (BLACK-BOX ACCESS)

The notion of black-box in the context of backdoor defenses is multifaceted. The definition varies between different papers, which we reflect in the absence of the term in the Table 4 and Table 5.

For instance, the STRIP defense [134] classifies as black-box any defense that does not need access to a model's weights. If it defines itself and the SentiNet method [135] as black-boxes, the category does not make explicit the various levels of access that a defender may have with respect to a model's outputs. Thus, whereas STRIP or DeepInspect [130] observe a DNN's confidence scores (i.e. its probits), SentiNet relies on the DNN's logits. It is noteworthy that access to a DNN's logits or probits may be an unrealistic assumption in the context of model outsourcing and hosting. This warrants a harder setup where a DNN defender may only retrieve a DNN's label output(s), as illustrated by the Flip and Shrinkpad defenses [136].

In the context of backdoor defenses applied to FRS models, we identify 3 defenses that rely on accessing a model's logits, 3 that rely on probits, and 4 that rely on labels (see Table 4 and Table 5).

B. SPECIFICATIONS OF FRS BACKDOOR DEFENSES

As DNN defenders may have different levels of knowledge about and access to a suspicious DNN, the breadth of available defenses against backdoors also varies. In this

context, this paper outlines 2 main categories of backdoor defenses:

(1) **model-based** defenses,
(2) **data-based** defenses,

which can be further split between **one-stage** or **multi-stage** defenses (see Fig. 6).

1) MODEL-BASED BACKDOOR DEFENSES

Model-based defenses assume they can either discern the presence of a backdoor in a DNN directly, i.e. the backdoor can be attested compared to a benign DNN, or erase the backdoor directly using the right technique.

a: MODEL DIAGNOSTIC

Model diagnostic defenses aim to detect whether a model contains a backdoor. Typically, these defenses perform hypothesis testing (e.g. binary test) based on a specifically-designed metric that help differentiate the behavior of benign and backdoored DNNs.

For instance, Neural Cleanse [129] proposes an outlier detection scheme that computes an anomaly index that consistently yields a higher value for backdoored models. The MAMF defense [137] diagnoses DNNs by computing the smallest pixel-perturbation that yields the maximum number of misclassifications in a clean test set. A derived metric, called maximum achievable misclassification fraction, leads to a binary test that sets backdoored DNNs from benign ones.

Some model diagnostic defenses exploit a white-box access to a suspicious DNN to extract information from its intermediary layers. For instance, the Ex-Ray method [138] extracts the intermediary feature maps of a suspicious DNN and uses an anomaly detection scheme, named Symmetric Feature Differencing, to assess the presence of a backdoor. The test first consists in selecting a set (a) of images from a source class, yielding a perturbed set (b) from (a) that is misclassified as a target class, and selecting a set (c) of benign images from said target class. Ex-Ray then computes the feature differences between (a) and (b), and (a) and (c). If the two feature differences diverge past a given threshold, Ex-Ray considers the DNN as backdoored.

Other methods like Unnervik & Marcel [124] use a Gaussian Mixture Models to cluster the feature representations of different models to find outliers. More recent techniques eschew clustering and the use of Gaussian models due to dimensionality problems and the lack of accessible datapoints to forward through a model. This is illustrated by the recent Beatrix [139] defense, which uses Gram matrices to capture feature information for each class of a suspicious DNN and highlight the classes with the most statistical dispersion as being evidence for a backdoor.

In the context of backdoor defenses on FRS models, we inventory 13 such defenses in the Table 4 and Table 5.

Model reconstruction (res. backdoor suppression, model cleaning, model patching). A defender who identifies a backdoored DNN may still need to use it. Here, model reconstruction defenses offer methods to to remove (with a high likelihood) a backdoor from an otherwise fully-functional DNN. Even without a backdoor diagnosis, a DNN defender may still proceed with reconstruction in hope to erase a potential backdoor in a DNN. Here, defenders typically perform reconstruction via one or more of 3 methods: (1) *model retraining*, (2) *neuron pruning*, and (3) *model distillation*.

Model retraining methods design fine-tuning procedures that, by updating the weights of a suspicious model, will remove the backdoor while conserving the DNN's performance on benign samples. For instance, the Confoc [123] defense fine-tune a suspicious model on a "retraining" dataset augmented with style transfer from a StyleGAN. The DNN is retrained to focus on image semantics and therefore is expected to forget the backdoor trigger.

Alternatively, Gu et al. [12] show that backdoored DNNs have neurons dedicated to detecting a backdoor trigger and propagating its effect to the output/decision of a DNN. Here, a defender may look to identify these neurons and, once found, design a *neuron pruning* strategy to remove them and the underlying backdoor. One of the earliest backdoor defenses, Fine-Pruning [128], explores neuron pruning, notably against the Chen et al. attacks [32]. This approach is also found in the Neural Cleanse [129] or Laundering [110] defenses. For instance, a defender following Neural Cleanse uses the activations of a suspicious model as it is being fed images to identify outlier neurons and prune them.

Finally, a defender that does not have access to a suspicious DNN's weights, or the opportunity to fine-tune or prune it, may instead resort to *model distillation*. Knowledge distillation is the process by which a student DNN is trained to replicate the performance of a teacher DNN. Doing so using clean data for instance may let a DNN defender yield a benign, equally-performing DNN, starting from a suspicious DNN. Such a method is found in the Data-free Holistic Backdoor Erasing (DHBE) defense [131], which generates a repaired model starting from a backdoored one. One advantage of DHBE is that it does not require the defender to have access to the internals of the suspicious DNN. The NAD defense [125] is noteworthy in that it combines the fine-tuning of a suspicious DNN with distillation to generate a student network. The defender then recombines the student and teacher models to yield a final, clean DNN.

In the context of backdoor defenses on FRS models, we inventory 13 model reconstruction, 3 neuron pruning, and 2 model distillation defenses (see the Table 4 and Table 5). We note that model-based defenses may fit several categories, using different approaches to strengthen the defender's capacity to thwart a backdoor. For instance, Fine-Pruning [128] proposes a defense that relies on both retraining and pruning a model to suppress a backdoor embedded in it.

2) DATA-BASED BACKDOOR DEFENSES

If the typical white-box access required to enable most model-based defenses is not possible, a defender may instead target the data used at the different steps of a DNN's lifecycle. For instance, DNN defender may monitor the data sent by users to the suspicious model (see Fig. 6), looking for outliers.

Trigger reconstruction (res. trigger synthesis, trigger inversion). Trigger reconstruction revolves around a defender interrogating a model with carefully-crafted inputs in order to reveal and synthesize a backdoor trigger. This type of defense is a core process that typically works jointly with other backdoor defense specifications. For instance, Neural Cleanse [129] provides, alongside a model diagnosis defense, a method to learn a trigger injection function by optimizing a perturbation pattern and its overlay mask over a set of clean inputs. Trigger reconstruction methods fall under two categories: (1) *learning a trigger injection function* and (2) *search algorithms*.

Many trigger reconstruction defenses follow the example of Neural Cleanse [129] by *learning a trigger injection function*. For instance, the TABOR [140] or ABS [141] defenses first identify compromised neurons in a DNN, and then uses them to guide a reconstruction algorithm, yielding a trigger injection function (a pattern and its mask over benign inputs). Additionally, methods like TND [142] draw from the adversarial attack literature, e.g. universal adversarial perturbations [143], to design potent trigger reconstructions. If most of such methods reconstruct a trigger by learning a pattern and a mask over benign inputs, methods like Tao et al. [144] have aimed to reduce the complexity of the trigger reconstruction by, for instance, eschewing the mask optimization for simply reconstructing the backdoor trigger. This notably allows capturing diffuse trigger backdoor attacks like SIG [111]. In a similar fashion, several papers make use of generator-based supplemental DNNs to learn and perform the trigger reconstruction step of their overall defenses. For instance, DeepInspect [130] uses a conditional-GAN to learn the potential trigger patterns associated with each class of a suspicious model. NNoculation [145] meanwhile uses a CycleGAN to extract the trigger patterns from suspicious inputs sent to a DNN. Such generator-based methods do not always rely on large models, as shown by MESA [146], which uses a 3-layer perceptron as a generator.

Some defenses rely on a *search algorithm* instead of an optimization process. For instance, NEO [147] follows a heuristic that iteratively blocks sections of an image, replacing it with its dominant color, to find backdoored inputs. Once a set of suspicious inputs is collected, the method searches the images for the trigger's position. TAD [148] designs a similar search process, looking for and counting potential patch-like and diffuse trigger patterns in input images sent to the defended DNN.

In the context of backdoor defenses on FRS models, we inventory 14 defenses that learn a trigger injection function and 2 that reconstruct a trigger via search algorithms (see the Table 4 and Table 5).

Data purification (res. input sanitization, poison-backdoor mismatch). A defender that cannot use a model-based defense or perform backdoor detection at the input level may try to indiscriminately clean the inputs sent by users to the defended DNN. In practice, the defender alters the content of incoming inputs such that the original semantics are left untouched while destroying the trigger contained within.

For instance, the Flip and ShrinkPad defenses [136] use simple data transformation techniques (e.g. flipping an input or shrinking and padding its content with random pixels) to greatly decrease the performance of backdoored inputs as they are sent by a malicious agent. Similarly, DeepSweep [149] draws from a library of attack instances to refine an input pre-processing policy to eliminate potential backdoor patterns at test-time. Meanwhile, more complex defenses may rely on supplemental, often generative, DNNs to perform input purification. For instance, Februus [150] iteratively refines a mask over an incoming input to a suspicious DNN and, using an inpainting GAN, regenerates the input's content at the mask's location. BDMAE [151] innovates over Februus by using a Masked-AE to perform the inpainting step while relying on a two-step heuristic search to define the associated mask. Moreover, BDMAE demonstrates its effectiveness in a label-only black-box setting.

We inventory 6 data purification defenses implemented on FRS models (see the Table 4 and Table 5).

a: TRAINING-TIME FILTERING

Whenever training a DNN is susceptible to data poisoning, a defensive approach is to filter outliers from its training dataset. As such, a DNN defender in charge of training a DNN with an untrustworthy dataset may remove suspicious training datapoints by observing their interactions with the learning DNN.

Some training-time filtering defenses design an outlier detection scheme based on the internal representations of a training DNN. For instance, the Spectral Signature [152] and Activation Clustering [133] defenses extract a DNN's feature maps during training and, after performing dimensionality reduction via singular value decomposition or principal component analysis, cluster training datapoints based on their representations. A cluster-based binary test then enables setting aside poisoned datapoints. As this process yields two sets, i.e. clean and a poisoned sets, the DNN defender may then choose to retrain the infected DNN on the clean one (e.g. following a model reconstruction method). Additionally, the ABL defense [153] builds on Spectral Signature and Activation Clustering to design an active filtering defense. Poisoned data are removed during learning such that a second training run is not necessary. ABL changes the training loss to incorporate a local gradient ascent technique in the early training epochs, and switches to including a global gradient ascent process in the later ones. Early on, the defense identifies the datapoints that converge suspiciously quickly, i.e. their associated loss drops surprisingly early below some

given threshold. Afterwards, the loss includes a gradient ascent targeting the identified datapoints so as to unlearn the backdoor.

A defender may look to filter a dataset before using it for training their end task DNN. For instance, the defender may train one or more models on the likely-backdoored data and use them to clean the training dataset for some later use. Two methods previously demonstrated on FRS DNNs are Meta-Sift [126] and D^3 [127]. Meta-Sift exploits the property that DNNs robustly learn backdoors to generate an ensemble of backdoored models, called ''Sifters,'' that score each training samples. These scores allow the detection and removal of poisoned inputs under an outlier detection scheme. Meanwhile, D^3 relies on a trigger reconstruction GAN built atop a suspicious DNN. The reconstruction is stamped on clean inputs, which are used to extract clean and possibly poisoned feature representations from the backdoored DNN. Afterwards, inputs that match the internal representations of a possible backdoor are removed from the training dataset.

We inventory 6 training-time filtering defenses implemented on FRS models (see the Table 4 and Table 5).

b: TEST-TIME FILTERING

In the case a DNN defender must defend against a potentially backdoored DNN (e.g. after outsourcing its training), they may look into filtering test-time inputs sent by possibly malicious users. Test-time filtering differ from data purification in that the defender assumes the capacity to distinguish poisoned from benign inputs, an assumption absent in data purification methods (the defender is blind to the presence of a backdoor, choosing instead to purify all incoming DNN inputs).

A defender may perform test-time filtering by extracting abnormal patterns from incoming inputs and testing them on held-out benign data, as illustrated in Sentinet [135]. Sentinet takes an image input to a DNN classifier and performs a heuristic search to find the most salient patch area in the image. The area is then extracted and stamped on benign inputs to check for suspiciously robust and targeted label flips. Using a score derived from the number of misclassifications caused by the suspicious pattern (versus a dummy one) at the same location, Sentinet then strikes the input as backdoored if it fails to meet some acceptance threshold.

Other test-time defenses with a white-box access to a model detect abnormal feature representations instead. Similar to Spectral Signature [152] and Activation Clustering [133], the SCAn defense [132] extracts intermediary representations from a suspicious DNN as inputs to a separate outlier detection scheme. The Raid defense [154], meanwhile, decomposes a suspicious DNN into its feature extractor and classifier subnetworks and trains a new classifier atop the extractor using clean validation data. An outlier detection scheme then compares the predictions of the two classifiers given the same input feature representation.

Jin et al. [155] propose to exploit the robustness of backdoors to perturbations to identify poisoned test inputs. By manipulating a DNN's weights (e.g. adding Gaussian noise) or neurons (e.g. shuffling weights, switching neurons, inversing activations), a defender expects DNN inputs to be misclassified. However, the authors observe that backdoored inputs are surprisingly robust to model mutations. The authors therefore derive an outlier detection scheme at test-time by generating a suite of mutated models against which to test incoming inputs. Similarly, the Scale-Up defense [156] observes that backdoored inputs are surprisingly robust to image saturation (i.e. light intensity). This originates from semantics being less robust than backdoor triggers to image manipulation, which in turn allows building a test-time outlier detection.

Finally, some methods explore stacking multiple binary tests to create more robust defenses. For instance, the CleaNN defense [157] uses two anomaly detectors: one at the input level using frequency analysis to detect suspicious patterns in an image; the other at one of the suspicious DNN's intermediary layers so as to capture abnormal features associated with the image's representations. Any of the two detectors being triggered leads to the defense rejecting the input.

We inventory 15 training-time filtering defenses implemented on FRS models (see the Table 4 and Table 5).

3) ONE-STAGE VS. MULTI-STAGE BACKDOOR DEFENSES

Backdoor defenses vary in the number of stages they involve. In the case of one-stage defenses, the STRIP method [134] leverages a clean test dataset to generate perturbed inputs to a suspicious DNN. By computing the entropy associated with each input, it designs a simple binary test to detect poisoned inputs in real-time; a low entropy indicates a high likelihood of a backdoored input.

Multi-stage defenses are more sophisticated, as illustrated by the NNoculation method [145]. In NNoculation, a DNN defender with a white-box access to a suspicious DNN operates in 4 distinct stages: (1) the defender duplicates the DNN and retrains the copy on clean validation data, augmented with Gaussian noise; (2) the defender deploys both models as an ensemble and looks for divergences in their predictions to quarantine suspicious inputs; (3) the defender trains a Cycle-GAN on clean validation and quarantined data to reconstruct a potential trigger; (4) Using the Cycle-GAN as a generator, the defender retrains the original DNN and its duplicate to become robust to the reconstructed backdoor trigger. Another example is found in the BDMAE method [151], which trains a Masked AE to iteratively erase and reconstruct the content of incoming inputs to a suspicious DNN, with the aim of destroying any trigger contained within.

We inventory 7 one-stage and 31 multi-stage defenses implemented on a FRS model (see the Table 4 and Table 5).

4) OTHER SPECIFICATIONS

As with defenses against adversarial attacks, backdoor defenses are broadly split between **empirical** and **certified** defenses. Empirical defenses are methods that rely on demonstrably effective heuristics (e.g. Neural Cleanse [129]) without providing provable guarantee of their effectiveness. This lack of guarantees gave rise to certified defenses in the adversarial literature [158]. All defenses cited thus far in this survey are empirical defenses. To the best of our knowledge, no certified backdoor defenses has been implemented on FRS models so far. We therefore left the distinction out of this paper's backdoor specification categories (see Table 4 and Table 5). However, we highlight some recent works into provably-robust backdoor defenses. These certified defenses typically rely on Randomized Smoothing [159], [160], a technique borrowed from the adversarial attack literature, which uses data augmentation with Gaussian noise to certify the accuracy of DNNs. We also note the singular certified backdoor defense framework proposed by Xie et al. [161] is applied in the context of federated learning.

Another left-out category is the distinction between **offline** and **online** backdoor defenses. We find that the distinction typically subsumes in the other, previously-defined specifications. Whereas offline defenses are used prior to a DNN's deployment, online defenses actively monitor and modify user interactions with FRS models to prevent malicious agents from exploiting their vulnerabilities, e.g. backdoors. Moreover, some multi-step defenses mix both such as NNoculation [145], which uses clean but noisy validation data to perform the offline retraining of a suspicious model. Then, using both the suspicious model and its retrained version, the defense monitors and quarantines outlying, typically-backdoored inputs in an online fashion.

C. THE LIMITED COVERAGE OF BACKDOOR DEFENSES

So far in this paper, we have provided a comprehensive list of backdoor attacks and defenses without looking at their interactions. We provide in the Tables 1-5 additional information on which attacks defeats which defense and vice-versa.

A first observation is the limited applicability of current backdoor defenses. We find that out of 54 attacks, only 10 have demonstrably been defeated in the literature.

Moreover, the defenses referenced in this survey mainly focus on defending against: patch-based, agnostic, all-to-one, and non-semantic backdoors. For instance, besides the Composite Attack [103], which is defeated by Ex-Ray [138], the effective defenses target non-semantic backdoors like BadNets [12] or Chen et al. [32]. This highlights a lack of coverage of existing backdoor defenses, especially in the context of FRS security.

Additionally, we note that most backdoor attacks defended against are relatively old. The newest attack defended against by Ex-Ray [138] is the HTBA attack [162] from 2019.

A second observation relates to defenses being overwhelmingly white-box. Indeed, 26 out of 39 referenced defenses requires a full access to a DNN. This is an untenable access requirement in multiple scenarios that reflect the reality of ML outsourcing and third-party hosting. Incidentally, we find that an increasing number of defenses considers a limited knowledge setup with respect to dataset access, with 18 out of 39 defenses requiring only access to some test data.

Finally, we note that, to the best of our knowledge, the defense literature *as a whole* misses a comprehensive coverage of the intrinsic limitations of defenses as well as the costs (complexity, runtime, etc.) involved. We did not find any comprehensive defense benchmark.

D. TAKEAWAY

In this Section, we evidence that backdoor defenses fit a variety of categories, as illustrated in the Table 4 and Table 5. Moreover, we show that (1) no one-size-fits-all defense exist and that (2) defense development is currently lagging behind the current attack literature. This realization especially matters in the context of defending DNN-based FRS, the context of this work, and the need for black-box backdoor defenses in the MLaaS industry and ML outsourcing in general.

VI. CURRENT TRENDS AND OPEN PROBLEMS

A. EMERGING TRENDS IN THE BACKDOOR LITERATURE

1) SYNTHETIC FACE DATASETS AND THEIR VULNERABILITIES

Stemming from privacy or ethical concerns, labeling problems, or dataset bias [163], the generation of artificial face recognition datasets (e.g. via GAN [164] or Diffusion [165] DNNs) has gained an increasing importance in recent years [163], [164], [166].

Unfortunately, generative DNNs do not provide protection for DNN users against backdoors. As hinted by the use of GANs in trigger reconstruction backdoor defenses (see Subsection V-B), recent works have demonstrated that generative models are both vulnerable to backdoors and to becoming vectors for data poisoning attacks [167], [168]. Such models can indeed be trained to generate backdoored data for an unsuspecting victim [169], [170].

To the best of our knowledge, there is no demonstration of backdoored data synthesis in a face recognition task, raising questions about the practicality of such attacks. Given this possible vulnerability however, it remains an open question whether training-time filtering methods, like Meta-Sift [126] and D^3 [127], are effective on generated, backdoored datasets.

2) DEFENDING AGAINST NOVEL BACKDOOR ATTACK METHODS

Beyond the context of backdoor attacks on face recognition tasks, recent works have explored designing stronger triggers based on imperceptibility, sample-specificity, and sparseness [88], [89], and sometimes benign processes

commonly found in image classification (e.g. compression algorithms [116]) to defeat existing defenses.

Additionally, recent works demonstrated stronger attacks in the physical world (a key concern in face recognition) such as the BATT attack [171], which claims to defeat a variety of model-based and data-based defenses such as Neural Cleanse [129], SentiNet [135], Fine-Pruning [128], and NAD [125]. As a consequence, prospective users of outsourced FRS (or of the DNNs therein) should consider the current state of backdoor defenses. No one-size-fits-all method exists yet.

B. LIMITATIONS OF THE BACKDOOR LITERATURE ON FACE RECOGNITION

1) FACE CLASSIFICATION IS NOT REALISTIC

The majority of backdoor attacks and defenses that use face recognition as a use case lack practicality for two reasons:

(1) modern face recognition is not a classification task,
(2) face recognition is an open-set problem.

Though early methods like EigenFaces [21] classified identities, the current ubiquity of large-scale datasets and deep learning methods cannot accommodate an ever-rising number of class-identities. This is why embedding methods based on contrastive or angular-margin learning, like ArcFace [61] or CosFace [64], play a key role in the field at the moment.

Consequently, the wide array of backdoor attacks found in this paper (see Tables 1 & 2) may only have a limited impact on real-life face recognition DNNs and thus FRS pipelines. Nonetheless, recent works like Carlini & Terzis [172] demonstrate the feasibility of backdooring contrastive learning, albeit not in the context of face recognition.

Additionally, modern face recognition tasks and thus FRS pipelines typically function in an open-set condition [173]: the training-time and test-time identities fed to a given DNN are disjoint (i.e. the data distributions differ). However, the backdoor literature that uses face recognition as a use case typically follows a closed-set setup where training-time and test-time identities are identical. For instance, the PTB attack [174] selects 100 identities from the YTF dataset [175]. To the best of our knowledge, no implementation of a backdoor on an open-set face recognition task yet exists.

2) FACE RECOGNITION (AS A TASK) IS ONLY A PART OF FRS

The backdoor literature overwhelmingly focuses on face extractor DNNs as models in isolation, and rarely does so with antispoofing. By eschewing the broader context of FRS pipelines, as illustrated in Section II, prior backdoor attacks and defenses do not consider the sequential set of tasks (face detection, extraction and alignment, antispoofing, feature extraction, and, optionally, binarization) and the backdoor risk associated with each of them. For instance, if backdooring a contrastive learning-based face extractor in an open-set situation is a hard problem, an attacker may instead look to backdoor the antispoofing DNN in a FRS such that it lets through fake face images.

To the best of our knowledge, no prior work explores the survivability of a backdoor trigger through a FRS pipeline, an important consideration as a backdoored model may exist behind several layers of extraction and image augmentations. Moreover, no existing paper in the backdoor literature takes into account the broader context of FRS pipelines, i.e., their vulnerabilities and the constraints they set on both attackers and defenders.

3) BACKDOOR DEFENSES MAY MOVE THE GOALPOST OF SECURITY

We noted in Section V that several defense methods rely on supplemental DNNs. For instance, black-box data purification defenses like NNoculation [145] or BDMAE [151] make use of an inpainting process based on a generative DNN.

However, as noted in Section VI-A1, generative models are also vulnerable to backdoor attacks. They cannot be considered secure. This matters both because generative machine learning may provide a false sense of security despite that several backdoor defenses rely on using generative DNNs sourced online and pre-trained by a possibly untrustworthy third-party (e.g. BDMAE [151] sources a pre-trained Masked-AE). Additionally, even if training such supplemental models was possible for a defender, accessing the required data may be out of reach as facial data may involve privacy and confidentiality safeguards (See App. D).

Such backdoor defenses are consequently moving the goalpost of security by relying on third-party DNNs that must be defended from backdoors too.

C. FUTURE RESEARCH DIRECTIONS

1) CATCHING UP WITH THE BACKDOOR STATE-OF-THE-ART

Because face recognition is a typical use case in the backdoor literature, future works that aim to assess the security of FRS against backdoors should draw from recent developments that may not be reflected in this paper (as mentioned in Section VI-A). Future research may look into (1) better assessing the applicability of backdoor attacks on FRS and (2) verifying the effectiveness of existing defenses on more novel attacks. A benchmark of backdoor defenses, covering defenses' applicability, complexity, and runtime for instance, is a missing work that would provide future directions for the backdoor defense literature.

2) A NOTION OF SURVIVABILITY OF BACKDOORS

Because a single DNN within a FRS may be vulnerable to backdoor attacks in the context of ML outsourcing (e.g. of either the detector, antispoofing or feature extractor), the backdoor attack literature should consider whether a trigger is robust enough to reach the malicious DNN in the FRS. For instance, an attack in physical space on a FRS feature extractor must go through detection, alignment, *and* antispoofing before reaching its target.

As such, a notion of *ASR* on a given DNN is insufficient. Backdoor success must also take into consideration the success rate of the backdoor trigger through all possible layers of models and pre-processing before reaching its end goal.

3) BACKDOOR ATTACKS AND DEFENSES ON OPEN-SET FACE RECOGNITION AND DISCRIMINATIVE FEATURE EMBEDDING

Future work in the backdoor literature should draw from the face recognition state-of-the-art to inform its use cases. Face recognition does not fit a simple classification task anymore. In this context, novel backdoor attacks may draw from prior works on master face attacks [176] and their hardness in open set conditions [177]. For instance, Guo et al. [122] demonstrate the effectiveness of a backdoor on an open-set face recognition DNN based on a Siamese network (this use case does not extend to angular margin training methods and DNNs such as CosFace [64]).

The security of other DNN training methods (e.g. self-supervised learning [35]) that may be used in a face recognition context is therefore an important point to address in the future.

4) BACKDOORS ATTACKS AND DEFENSES FROM PRECEDING MILESTONES TO FUTURE FRONTIERS IN A FRS

FRS are complex systems that may be outsourced wholly or in parts to malicious third-parties. It is thus worth considering new kinds of attacks that exploit the intricacies of FRS to carry backdoor triggers to their DNN targets. For instance, it is conceivable that a backdoored detector, as illustrated by the BadDet attack [70], eventually finds its way into a FRS. Additionally, antispoofing tasks may also be backdoored as the task is similar to a binary classification. In a similar vein, backdoors on autoencoders [178] may find their way in face embedding binarizers as they are currently envisioned for facial fuzzy commitment schemes [47].

Beyond the existing structure of current FRS as outlined in Section II, defenders must also consider a future where the structure of a FRS changes. Additional bricks to a DNN-based FRS cannot be considered safe due to a novelty factor.

For instance, something we did not cover in this survey is the open problem of deepfakes and their detection as they do not fit the same threat model (physical versus digital [60]). However, with the recent explosion of diffusion methods and the possibility of seamless forgeries and synthetic faces [165], security stakeholders will need to look for stronger, more robust defenses. Here, we note that the inclusion of a deepfake-detection DNN in a FRS could happen in the future as the task is close if not overlapping with the current state of face presentation attacks [60]. Securing FRS should thus explore how potential novel FRS structures may be impacted by out-of-domain backdoor attacks, and therefore how to prevent them.

5) MORE ROBUST DEFENSES IN A BLACK-BOX CONTEXT

Following the previous point, the complexity of FRS may also work in the favor of defenders, using the outputs of the different stages of a FRS pipeline may feed a defense process.

However, DNN defenders must first face the prospect of both black-box and backdoored FRS pipelines. Without knowing whether a DNN in a FRS is backdoored and without white-box access when the whole FRS pipeline is outsourced to a third-party, DNN defenders must explore novel, strong defenses to protect themselves from malicious agents without falling into the pitfall of several black-box defenses that rely on outsourced DNNs.

VII. CONCLUSION

This survey provides a comprehensive integrity-related threat model of FRS pipelines, and the classification of both backdoor attacks and defenses that affect them. We cover the level of knowledge and access required by both attackers and defenders, where in a pipeline and when in its lifecycle attacks and defenses occur, and how attacks and defenses have answered each other. Additionally, this survey highlights that backdoor defenses are lagging behind the attack literature and that backdoor attacks and defenses typically do not consider FRS pipelines in their entirety. The security analysis of such systems, and the role and applicability of backdoor attacks and defenses, are therefore only partially understood.

It is therefore critical for future research to better understand the landscape of backdoor attacks and defenses that affect FRS pipelines. Future research ought to consider the security *and* complexity of FRS as a mesh of specialized models that each serves a complementary purpose. The uncovered limitations of the current research, which limit the applicability and relevance of both attacks and defenses in a real-life context, must be addressed.

APPENDIX A
DESCRIPTION OF FRS PIPELINE BLOCKS

Face Detector. The first DNN module of a FRS is the detector. It is responsible for scanning an input image or video (e.g. frame-by-frame) to identify one or more faces. The detector DNN that is trained with a multi-task approach [49], which involves predicting various attributes and labels. These include regressing the coordinates of the bounding boxes around detected faces (e.g. top-left corner coordinates, box width and height), computing a confidence score indicating whether a face is present within these boxes (e.g. personhood score), and optionally regressing or classifying other facial characteristics like face landmarks [52], [53], [54] (e.g. nose, eye and mouth corners, etc.), three-dimensional face topology [52], luminosity, yaw, accessories, etc. Face detection models are typically adapted from the broader object detection literature, with networks such as Fast-RCNN [50], YOLOv2Face [51], RetinaFace [52], or MTCNN [53].

Detectors are built upon convolutional neural networks (CNN) [49], [50], [51], [52], [53], although recent

developments have explored the use of vision transformers [179]. These CNNs serve as the backbone of detector models, with additional subnets referred to as necks and/or heads attached. The backbone is generally pre-trained on a domain-independent task, i.e. the backbone is at first agnostic to the face detection task [18]. The backbone and subnets are then concurrently fined-tuned. Backbone and subnets typically fit four kinds of setups:

1) *Featurized Image Pyramid*: An image is duplicated at different downsized scales. A feature map is computed for each version and propagated to a subnet specific to that scale. This allows regressing bounding box corrdinates at each of these scales (e.g. with MTCNN [53]).
2) *Single-Shot Feature Map*: The detector's subnets use the final layer output of the backbone to detect faces at different scales (e.g. YOLOv1-Face [180]).
3) *Pyramidal Feature Hierarchy*: The detector's subnets leverage the feature maps from the intermediary layers of the backbone to perform multi-scale regression of bounding box coordinates (e.g. SSD [181]).
4) *Feature Pyramid Network*: The subnets rely on the backbone's feature maps, as in (3), but an additional residual operation. This operation (e.g. upscale + addition) connects each feature map with its backward neighbor in the backbone architecture (e.g. the feature map of the layer n is upscaled and added back to the feature map of layer $n-1$), allowing multi-scale feature sharing for better face detection (e.g. DF2S2 [182]).

Regardless of the backbone configuration, face detection is generally performed in a multi-step or one-step fashion. Multi-step detectors are typically older designs that first generate a set of face regions of interest (RoI) and then refine them. Conversely, more recent single-step detectors have eschewed the initial RoI-generating step in favor of performing face detection directly [183]. Single-shot detection methods typically fall into two subtypes: anchor-free and anchor-based [13]. Anchor-free detectors directly regress the true coordinates of bounding boxes (e.g. YOLOv1-Face [180]) whereas anchor-based detectors regress a relative position with respect to canonical bounding box coordiates (res. anchors) defined for each feature map used in the backbone network (e.g. RetinaFace [52]).

To delve deeper into the object and face detection literature, we point to the following surveys [18], [183], [184].

Alignment. Alignment initially involved the task of regressing facial landmarks in a detected face image [55], [56]. However, DNN-based face detectors have increasingly taken on this role thanks to advances in multi-task learning. For example, RetinaFace regresses facial landmarks along with a three-dimensional topology [52]. As such, alignment corresponds less and less to a module dedicated to the regression of facial landmarks and instead only aligns the detected face and its landmarks to a standardized shape (e.g. a specific ratio between the location of the right eye, nose tip, and border of the face [185]). This shape, pre-defined by the FRS developer, is known as a canonical shape [186]. This preprocessing stage reduces facial variance, which enhances the accuracy of subsequent feature extractors since downstream DNNs must ignore non-biometric features.

Fitness to a canonical shape can be achieved through geometric methods such as affine transformations [26] (e.g. rotation and scaling based on face landmarks) or by using three-dimensional topologies or masks, as seen in RetinaFace [52]. We note that alignment can also be performed by a separate, learned module appended to the detector, such as the Spatial Transformer Network [187] or the APA method [188].

We point the reader to the following surveys for more details on alignment: [189], [190], [191].

antispoofing. As face detector operate in real-world environments, they run the risk of detecting erroneous inputs. An accidental detection may take the form of a face printed on a bus driving by a CCTV camera, for instance. However, an input may also be malicious, such as presentation attacks (res. spoof, PA), where an attacker attempts to gain an unauthorized access to a secure device [58]. It is critical for an authentication FRS to include face antispoofing (FAS) measures. These measures typically involve liveness detection and/or personhood authentication to ensure that the detected face is not only real but also alive. This is crucial because PAs either impersonate someone or evade detection. Spoofing may occur as a digital (e.g. print, replay, channel attacks) or a physical (e.g. mask, cosmetics, tattoo, glasses) attack.

Whereas antispoofing has historically focused on detecting biometric features using handcrafted methods (e.g. eyeblink detection [192], infrared assessment [193]), more recent developments have explored hybrid (DNN + handcrafted features) or full deep learning solutions [60].

Due to the complexity of FAS and the need to adapt to diverse situations (e.g. luminosity, location) and potentially yet-unseen spoofing methods, FAS has evolved along several dimensions. Given possibly multimodal inputs (e.g. RGB, thermal), FAS is approached from either a binary supervision or pixel-wise supervision perspective [60]. In the former, DNNs aim to extract and discriminate between spoof and benign (res. bonafide) features in a classification setup. In the latter, DNNs generate pixel-wise masks (e.g. depth, reflection maps) to assess the liveness of a detected face. Additionally, FAS has been explored in the context of domain adaptation and domain generalization [60]. Domain adaptation methods train FAS models given some knowledge of the target domain to bridge the gap between known, training data and real-world observations. Instead, domain generalization works from the perspective that DNNs must learn generalized feature representations regardless of the test data. Of the two, domain generalization is the most recent and challenging case, as it requires DNNs to handle unknown attacks.

Insights into face antispoofing can be found in the following surveys [17], [60], [194].

Feature extractor. Once an input face successfully passes the antispoofing step, it is processed through a feature extractor model that transforms the image into a lower-dimension representation (typically a real-valued vector or floating-point array) for comparison purposes. To learn lower-dimensional face recognition (FR) representation, feature extractors tend to be based on classifier backbones such as the ResNet architecture [61]. DNN meant for feature extraction are typically trained in one of two setups: closed-set and open-set [13].

In the closed-set setup, the identities found in the training and testing sets are identical. This scenario is a typical classification problem where each identity corresponds to a class label. Standard losses such as cross-entropy loss, contrastive loss, or triplet loss are common to learn separable features. In an open-set setup however, the identity domains between training and inference differ. This transforms the problem into one of metric learning, where a FR DNN learns to separate features following angular margin losses, e.g. based on cross-entropy such as (e.g. Arcface [61], Cosface [64], or Magface [195]). Initially trained in a classification fashion, a DNN learns to output logits that maximize the difference between the feature representations of different identities while minimizing the feature differences of the same identity. Once training is complete, the DNN's last fully connected layer is removed, leaving behind the DNN's feature representation structure. This trimmed DNN now generates face embeddings that can be used for biometric applications with unknown identities, leveraging the notion of distance that equips the DNN's feature representations (e.g. via cosine similarity).

Feature extraction serves as the core of a FRS, providing its discriminative power and versatility for various tasks. Besides metric learning and angular margin training, feature extractors may also be trained via embedding learning, often with the use of generative models. For instance, training a face generator with either auto-encoders or generative adversarial networks results in a discriminative encoder network that can be used for FR [49]. Furthermore, these methods enable disentangling the different features associated with identities in a dataset, allowing the creating of feature representation that are invariant to task-irrelevant chracteristics such as age, pose, etc. (e.g. D2AE [196], DR-GAN [197]).

The surveys [13], [16] provide a comprehensive coverage of face feature extraction, models, and datasets, including angular margin loss functions and contrastive methods.

Feature binarizer (*Optional*). When FRS require binary-valued inputs, e.g. for cryptographic purposes, an optional binarization step may be appended to the extractor to generate a binary-valued encodings [63]. Here, recent work has explored the use of DNNs for binarization like AEs [47].

Matcher. Once a FRS has detected a face, assessed the liveliness, and extracted its feature, these features can be used for a downstream tasks given they pass the matcher. Two primary setups exist in this context: face authentication and face identification. Face authentication corresponds to a $1:1$ setup where a detected face is compared to a stored template to determine whether or not they match, for instance to authenticate an access to a banking application. In contrast, face identification operates in a $1:N$ setup, where the detected face is compared to a gallery of candidate faces. The goal is to find whether it matches an identity in the gallery.

APPENDIX B
OTHER DNN ATTACKS IN THE CIA TRIAD CONTEXT

This survey focuses on integrity-related attacks in the context of the CIA triad [8]. However, the breadth of confidentiality and availability attacks should not be understated. In this appendix, we provide the readers with notes and citations related to those two categories.

A. CONFIDENTIALITY-RELATED ATTACKS

Confidentiality attacks on DNN aim to unveil private information about a DNN, such as its training data or weights. This crucial, especially in the context of face recognition, where face data has privacy and also legal implications such as under the European Union's GDPR framework [198]. In this context, we highlight two types of attacks: **membership inference** and **model stealing**.

1) MEMBERSHIP INFERENCE ATTACKS

A first type of privacy leakage affecting DNNs pertains to the data they have been trained on. **Membership inference attacks** are a suite of methods that extract sensitive information about a DNN's training data [199], e.g., whether a given image has been used to train a model. Additionally, more sophisticated attacks also aim to directly reconstruct training samples [200]. Though several defenses against membership inference have emerged [201], we note that no single method has yet to be proven robust against an ever growing attack literature [202].

Membership inference attacks are especially prevalent in the vision domain [203] where it has been demonstrated on a wide array of tasks [203]: clasification, generation, segmentation, etc. Here, we note that the membership inference literature makes heavy use of face recognition tasks as a core use case [203]. Moreover, contrary to the backdoor literature attack, membership inference methods have explored attacking face extractors in an embedding setting rather than classification [204].

2) MODEL STEALING ATTACKS

If trained data can be inferred or reconstructed, model weights may also be stolen. **Model stealing attacks** (res. model inversion) encompass an ensemble of techniques aimed at extracting the functionality of a target DNN [205], even in a black-box setting [206]. Recent work by Oliynyk et al. [207] categorizes model stealing under two categories: whether theft targets exact model properties like its architecture

or weights, or some approximate model behavior (e.g. accuracy).

In the context where defending against model stealing is an open problem [207], we underscore that prior work on DNN model stealing typically rely on face recognition tasks as a core use case [205], [206]. As such, model stealing is particularly relevant in the context of FRS security.

B. AVAILABILITY-RELATED ATTACKS

Availability attacks on DNN aim to perturb the model's function to irrecoverably impact its performance such that it stops functioning as intended [208]. In this context, we highlight two types of attacks: **data poisoning** and **energy-latency**.

1) DATA POISONING ATTACKS

Data poisoning attacks typically rely on similar attack channels as backdoor attacks (e.g. data collection). However, they aim for a different behavior: instead of targeted misclassification for backdoor attacks, data poisoning attacks aim to manipulate a training dataset to cause a decrease of the victim DNN's overall performance at test-time [209], [210]. As such, they are untargeted/indiscriminate [211] and visible (no stealth is involved) as the DNN's degradation is general.

Data poisoning attacks *and* defenses have been demonstrated in the past on face recognition tasks [209]. These attacks also find a strong use case in the context of federated learning [212], [213], which are ripe for attacks via the data collection channel. Here, we find prior work on poisoning face recognition tasks in a federated learning context [214].

2) ENERGY-LATENCY ATTACKS (SPONGE EXAMPLES)

Shumailov et al. [215] evidences a new type of availability attack under the name of **sponge examples** or **energy-latency attacks**. Like adversarial examples, sponge examples are mined following an optimization process. However, instead of causing misclassifications, sponge examples maximize the energy expenditure of a DNN. The goal is to increase DNN latency, causing a denial of service (DoS) attack that may be critical to some systems such as in autonomous driving.

If Shumailov et al. [215] demonstrates the effectiveness of sponge examples against GPU and ASIC hardware, Wang et al. [216] showed that mobile device (e.g. mobile CPUs) are also affected. Additionally, Shapira et al. [217] expands on the work of Shumailov et al. [215] by demonstrating that sponge examples can also be used against object detection besides object classification (Shumailov et al. also explores attacking NLP tasks).

Lastly, Cinà et al. [218] bridges the gap between sponge and *data poisoning* by demonstrating that sponge examples can be injected in a model at training time, which exposes a DNN user to an attacker in a ML outsourcing context. The paper concludes with future research directions pointing towards bridging sponge attacks and *backdoor attacks*, where a specific pattern learned by a backdoored DNN could trigger latency issues once stamped on a given model.

Here we note that, to the best of our knowledge, no sponge attack has been demonstrated on DNNs trained on a task found in a FRS pipeline (e.g. face detection, recognition, etc.).

APPENDIX C
FORMALIZATION OF BACKDOOR ATTACKS

We provide in this Appendix a short formalization of backdoor attacks. For a more expansive formalization of trigger-based backdoors, we point the reader to Wu et al.'s survey [14]. To the best of our knowledge, there is no comprehensive formalization of triggerless backdoors.

A. TRIGGER-BASED BACKDOORS

A trigger-based backdoor attack on a classification task involve an attacker who is able to influence the training or deployment of a target DNN such that the DNN associates the presence of a given trigger pattern t, added to a source input x^{cl} of class y^{cl}, with an erroneous label y^{po}.

In the common backdoor injection method of data poisoning, the attacker modifies a portion $\beta \in (0, 1]$ of a *clean* training dataset $\mathcal{D}^{\mathsf{cl}}_{\mathsf{train}}$ with a poisoning function $p : \mathcal{X} \to \mathcal{X}$ along with a label flip function $c : [\kappa] \to [\kappa]$. $^{\mathsf{cl}}$ denotes *clean* data, $\mathcal{X}$ is the input space of a DNN f_θ with parameters θ, and $[\kappa] = \{1, \ldots, \kappa\}$ is the number of classes predicted by a DNN f_θ.

Section IV distinguishes data poisoning between two main cases: poison-label or clean-label. In the former case, an attacker poisons inputs from different *source* classes with p and flips their ground truth labels to a single *target* class y^{po} with c. In the latter case, the attacker poisons inputs from the target class itself and c is the identity function. As such, a backdoor attack via data poisoning can be summarized as follows:

$$\mathcal{D}^{\mathsf{cl}}_{\mathsf{train}} = \{(x^{\mathsf{cl}}_i, y^{\mathsf{cl}}_i)\}^n_{i=1} \subset \mathcal{X} \times [\kappa] \tag{1}$$

$$x^{\mathsf{po}}_i = poison(x^{\mathsf{cl}}_i) \tag{2}$$

$$y^{\mathsf{po}}_i = c(y^{\mathsf{cl}}_i) = \begin{cases} y_i \in [\kappa],\ y_i \neq y^{\mathsf{cl}}_i & \text{(Poison-label)} \\ y^{\mathsf{cl}}_i & \text{(Clean-label)} \end{cases} \tag{3}$$

where x^{po} is an input altered with p, and y^{po} is the attacker's target label specified by c. Both poison-label and clean-label cases yield a poisoned training set $\mathcal{P} = \{(x^{\mathsf{po}}_i, y^{\mathsf{po}}_i)\}^m_{i=1}$.

In the data collection backdoor channel case, the attacker injects $\mathcal{P}$ in the unsuspecting victim's data $\mathcal{C}$. In the outsourcing case, the attacker is able to merge $\mathcal{P}$ and $\mathcal{C}$ (which they own) to generate a backdoored dataset $\mathcal{D}^{\mathsf{po}}_{\mathsf{train}} = \mathcal{C} \cup \mathcal{P}$ with which the attacker freely trains the victim DNN f^{po}_θ.

TABLE 1. Taxonomy of backdoor attacks tested on face recognition tasks (i.e. FRS blocks).

	Attack	Implem.		Target			Channels	backdoor injection methods			Backdoor Trigger Specifications								Defeats defenses (FRS case)	Defeated by defenses (FRS case)
		Ref.	Year	Datasets	Models	Tasks		Data poisoning	Training process	Arch. or weights	Blended, Patch, Warping	Frequency, Physical, Pixel	Sample-agnostic vs. specific	Invisible vs. visible	Hand-crafted vs. Optimized	All-to-one vs. All-to-all	Dynamic vs. Static	Nonsem-antic vs. Semantic		
1a	Chen et al.	[32]	2017	YTF	VGG16	FR	DC	PL	∅	∅	blended patch	physical pixel	agnostic	both	handcrafted	all-to-one	static	both	∅	14d, 28d, 36d, 38d
2a	*BadNet* [12]	[106] [80]	2017	PubFig	VGG16 ResNet34	FR	MO, TL	PL	∅	∅	patch	pixel	agnostic	visible	handcrafted	both	static	non-semantic	∅	2d-9d, 13d, 16d, 19d, 20d, 23d, 25d 27d, 28d, 29d 32d-38d
3a	*Trojan Attack*	[92]	2018	VGGFace LFW	VGG16 GoogLeNet	FR	MO, TL	PL	∅	∅	patch	pixel	agnostic	visible	optimized	all-to-one	static	non-semantic	∅	1d-8d, 10d, 11d, 13d-18d, 20d-23d, 25d, 27d, 29d, 36d, 39d
4a	*Hu-Fu*	[33]	2018	YTF	ResNet20	FR	TE	∅	∅	MAM	trigger-less backdoor (label flip specific to a single sample/out-of-distribution class)								∅	∅
5a	Ji et al.	[95]	2018	VGGFace	VGG16	FR	TL	∅	BPM	∅	trigger-less backdoor (label flip specific to a single sample/out-of-distribution class)								∅	∅
6a	Dumford & Scheirer	[34]	2018	VGGFace2	CNN ResNet50	FR	DEP	∅	∅	TWP	trigger-less backdoor (label flip specific to a single sample/out-of-distribution class)								∅	∅
7a	Liao et al. [219]	[220]	2018	CelebA-Mask-HQ	ResNet18	FR	MO	PL	∅	∅	blended	pixel	agnostic	invisible	optimized	all-to-one	∅	non-semantic	∅	∅
8a	*LC* [99]	[221]	2021	VGGface LFW	VGG16	FR	DC	CL	∅	∅	patch	pixel	agnostic	visible	optimized	all-to-one	static	non-semantic	1d, 7d	14d, 18d, 36d-38d
9a	*SIG* [111]	[80]	2019	PubFig	ResNet34	FR	DC	CL	∅	∅	blended	pixel	agnostic	visible	handcrafted	all-to-one	∅	non-semantic	∅	∅
10a	Bhalerao et al.	[71]	2019	Replay Attack	Conv-LTSM	AS	DC	CL	∅	∅	blended	physical pixel	agnostic	invisible	handcrafted	all-to-one	∅	non-semantic	∅	∅
11a	Yao et al.	[97]	2019	VGGFace PubFig	VGG16	FR	TL	PL	∅	∅	patch	pixel	agnostic	visible	handcrafted	all-to-one	static	non-semantic	1d, 3d	36d
12a	Wang et al.	[93]	2019	VGGface LFW	VGG16	FR	TL	PL	∅	∅	patch	pixel	agnostic	visible	optimized	all-to-one	static	non-semantic	∅	∅
13a	*HTBA* [162]	[221]	2019	VGGface LFW	VGG16	FR	DC	CL	∅	∅	patch	pixel	agnostic	visible	optimized	all-to-one	dynamic	non-semantic	∅	28d
14a	Salem et al. (1)	[117]	2020	CelebA	CNN	FR	DC, MO	PL	∅	∅	patch	pixel	agnostic	visible	optimized	both	dynamic	non-semantic	3d, 4d	∅
15a	Bagdasaryan & Shmatikov	[91]	2020	PIPA	ResNet18	FR	TE	∅	LFM	∅	patch	pixel	agnostic	visible	handcrafted	all-to-one	static	both	2d, 3d, 10d	∅
16a	*TrojanNet*	[106]	2020	PubFig YTF	VGG16 CNN	FR	DEP	∅	∅	MAM	patch	pixel	agnostic	visible	handcrafted	both	static	non-semantic	3d, 12d	∅
17a	*FaceHack*	[114]	2020	VGGFace2 CelebA	ResNet20	FR	MO	PL	∅	∅	patch	physical pixel	agnostic	visible	handcrafted	all-to-one	static	semantic	3d, 4d, 13d	∅
18a	*Trembling trigger*	[37]	2020	VGGFace	VGG16	FR	MO	PL	∅	∅	patch	physical pixel	agnostic	visible	optimized	all-to-one	static	non-semantic	20d	∅
19a	Wenger et al.	[81]	2020	VGGFace VGGFace2	VGG16 DenseNet ResNet50	FR	DC	PL	∅	∅	patch	physical pixel	agnostic	visible	handcrafted	all-to-one	static	semantic	3d, 4d, 19d, 20d	∅
20a	*Refool*	[80]	2020	PubFig	ResNet34	FR	MO	CL	∅	∅	blended	pixel	agnostic	visible	optimized	all-to-one	∅	semantic	∅	∅
21a	Li et al.	[82]	2020	PubFig, YTF VGGFace	AlexNet ResNet50	FR	DC	CL	∅	∅	blended	physical pixel	agnostic	visible	handcrafted	all-to-one	∅	non-semantic	∅	∅
22a	Salem et al. (2)	[107]	2020	CelebA	VGG19	FR	TE	∅	∅	MAM	trigger-less backdoor (label flip specific to a single sample/out-of-distribution class)								∅	∅

Abbreviations (channels): **DEP**: deployment phase; **DC**: data collection; **MO**: model outsourcing; **TE**: training environment; **TL**: transfer learning

Abbreviations (injection methods): **AS**: anti-spoofing; **BPM**: back-propagation manipulation; **CL**: clean-label poisoning; **DAM**: data augmentation manipulation; **FD**: face detection; **FR**: face recognition; **LFM**: loss function manipulation; **MAM**: model architecture manipulation; **PL**: poison-label poisoning; **TWP**: targeted weight perturbation

TABLE 2. *(Continued.)* **Taxonomy of backdoor attacks tested on face recognition tasks (i.e. FRS blocks).**

	Attack	Implem.		Target			Channels	backdoor injection methods			Backdoor Trigger Specifications								Defeats defenses in FRS	Defeated by defenses (FRS case)
		Ref.	Year	Datasets	Models	Tasks		Data poisoning	Training process	Arch. or weights	Blended, Patch, Warping	Frequency, Physical, Pixel	Sample-agnostic vs. specific	Invisible vs. visible	Hand-crafted vs. Optimized	All-to-one vs. All-to-all	Dynamic vs. Static	Nonsemantic vs. Semantic		
23a	*Composite Attack*	[103]	2020	YTF, LFW	ResNet18	FR	MO	PL	DAM	∅	patch	physical pixel	agnostic	visible	handcrafted	all-to-one	dynamic	semantic	3d	28d
24a	*BlackCard*	[121]	2020	VGGFace	VGG16	FR	DC	CL	∅	∅	trigger-less backdoor (label flip specific to a single sample/out-of-distribution class)								∅	∅
25a	*DFST*	[115]	2020	VGGFace	VGG16 Resnet50	FR	MO	PL	LFM	∅	blended	pixel	agnostic	visible	optimized	all-to-one	∅	non-semantic	∅	36d
26a	*ISSBA*	[83]	2020	MS-Celeb-1M	VGG16 ResNet18	FR	DC	PL	∅	∅	blended	pixel	specific	invisible	optimized	all-to-one	∅	non-semantic	1d, 2d, 3d, 4d, 18d, 19d	37d
27a	Xue et al.	[222]	2021	VGGFace	VGG16	FR	MO	PL	LFM	∅	blended patch	pixel	agnostic	visible	handcrafted	all-to-one	static	non-semantic	∅	∅
28a	*DeepPoison*	[104]	2021	LFW, CASIA-Webface	FaceNet	FR	MO	CL	LFM	∅	trigger-less backdoor (label flip specific to a single sample/out-of-distribution class)								∅	∅
29a	*WaNet*	[84]	2021	CelebA	ResNet18	FR	MO	PL	∅	∅	warping	pixel	agnostic	invisible	handcrafted	both	∅	non-semantic	1d, 2d, 3d	30d, 37d, 38d
30a	*LF*	[113]	2021	PubFig	*not listed*	FR	MO	PL	∅	∅	blended patch	frequency	agnostic	both	both	all-to-one	static	non-semantic	∅	∅
31a	*SAA* [223]	[221]	2021	VGGface LFW	VGG16	FR	DC	CL	∅	∅	patch	pixel	agnostic	visible	optimized	all-to-one	dynamic	non-semantic	∅	∅
32a	*PTB*	[174]	2021	YTF	VGG16	FR	DC, MO	PL	DAM	∅	patch	physical	agnostic	visible	handcrafted	all-to-one	dynamic	semantic	∅	∅
33a	*Master Key*	[122]	2021	VGGFace2 YTF, LFW	Siamese CNNs+FC	FR	MO	PL	∅	∅	trigger-less backdoor (label flip specific to a single sample/out-of-distribution class)								∅	∅
34a	Hong et al.	[109]	2021	PubFig VGGFace2	Inception ResNetV1	FR	DEP	∅	∅	TWP	patch	pixel	agnostic	visible	handcrafted	all-to-one	static	non-semantic	∅	∅
35a	*SRA*	[98] [224]	2021	VGGFace	VGG16	FR	DEP	∅	∅	MAM	patch	physical pixel	agnostic	visible	handcrafted	all-to-one	static	both	∅	∅
36a	*Poison Ink*	[85]	2021	VGGFace	VGG19	FR	DC	PL	∅	∅	patch	pixel	specific	invisible	optimized	all-to-one	dynamic	non-semantic	∅	∅
37a	*BAAAN*	[178]	2021	VGGFace VGGFace2	AE, GAN	FR	MO	PL	∅	∅	patch	pixel	agnostic	visible	handcrafted	all-to-one	static	non-semantic	∅	∅
38a	*FTROJAN*	[86]	2021	PubFig	ResNet50	FR	DC, MO	PL	∅	∅	blended	frequency	agnostic	invisible	handcrafted	all-to-one	∅	non-semantic	∅	∅
39a	Zhong et al.	[105]	2022	CelebA	ResNet18	FR	DC	PL	LFM	∅	blended	pixel	specific	invisible	optimized	both	∅	non-semantic	1d, 3d, 26d	∅
40a	*BppAttack*	[118]	2022	CelebA	ResNet18	FR	MO	PL	∅	∅	blended	pixel	agnostic	invisible	handcrafted	both	∅	non-semantic	1d, 3d, 4d	∅
41a	Guo et al.	[101]	2022	Replay Attack	RN18-LSTM Incep-RNv1	AS FR	DC	CL	∅	∅	blended	frequency	agnostic	invisible	handcrafted	all-to-one	∅	non-semantic	∅	∅
42a	*COMBAT*	[119]	2022	CelebA	ResNet18 VGG13 MobileNetv2 ViTSmall8	FR	DC	PL	∅	∅	blended	frequency	specific	visible	optimized	all-to-one	∅	non-semantic	1d, 3d, 4d	∅
43a	*TSA*	[225]	2022	VGGFace2	ResNet VGG16 ShuffleNet GoogLeNet	FR	MO	PL	∅	∅	blended	pixel	specific	invisible	optimized	all-to-one	∅	non-semantic	∅	∅
44a	*Narcissus*	[221]	2022	PubFig	ResNet18	FR	DC	CL	∅	∅	blended	frequency pixel	agnostic	invisible	optimized	all-to-one	∅	non-semantic	∅	∅
45a	*LoneNeuron*	[226]	2022	LFW	Inception-ResNet	FR	DEP	∅	∅	MAM	blended	pixel	specific	invisible	optimized	all-to-one	∅	non-semantic	3d, 4d, 9d, 10d	∅
46a	*RIBAC*	[227]	2022	CelebA	ResNet18	FR	DEP	∅	∅	MAM, TWP	blended	pixel	agnostic	invisible	optimized	all-to-one	∅	non-semantic	2d	∅

Abbreviations (channels): **DEP**: deployment phase; **DC**: data collection; **MO**: model outsourcing; **TE**: training environment; **TL**: transfer learning

Abbreviations (injection methods): **AS**: anti-spoofing; **BPM**: back-propagation manipulation; **CL**: clean-label poisoning; **DAM**: data augmentation manipulation; **FD**: face detection; **FR**: face recognition; **LFM**: loss function manipulation; **MAM**: model architecture manipulation; **PL**: poison-label poisoning; **TWP**: targeted weight perturbation

TABLE 3. *(Continued.)* **Taxonomy of backdoor attacks tested on face recognition tasks (i.e. FRS blocks).**

	Attack	Implem.		Target			Channels	backdoor injection methods			Backdoor Trigger Specifications								Defeats defenses in FRS	Defeated by defenses (FRS case)
		Ref.	Year	Datasets	Models	Tasks		Data poisoning	Training process	Arch. or weights	Blended, Patch, Warping	Frequency, Physical, Pixel	Sample-agnostic vs. specific	Invisible vs. visible	Hand-crafted vs. Optimized	All-to-one vs. All-to-all	Dynamic vs. Static	Nonsem-antic vs. Semantic		
47a	Flareon	[102]	2023	CelebA	ResNet18	FR	TE	∅	DAM	∅	blended	pixel	agnostic	invisible	handcrafted	all-to-all	∅	non-semantic	∅	∅
48a	Peng et al.	[87]	2023	CelebA	ResNet18	FR	DC	PL	∅	∅	blended	pixel	agnostic	both	optimized	all-to-one	∅	non-semantic	4d	∅
49a	*Low-Pass*	[228]	2023	CelebA	ResNet18	FR	MO	PL	∅	∅	blended	frequency	specific	invisibile	optimized	all-to-one	∅	non-semantic	3d, 4d	∅
50a	Zhang et al.	[88]	2023	MS-Celeb-1M	VGG16	FR	DC	PL	∅	∅	blended	pixel	specific	invisible	optimized	all-to-one	∅	non-semantic	3d, 4d	∅
51a	Yu et al.	[116]	2023	FFHQ CelebA	ResNet50	FR	DC	PL	LFM	∅	blended	frequency	agnostic	invisible	optimized	both	∅	non-semantic	∅	∅
52a	*SIBA*	[89]	2023	VGGFace2	ResNet VGG	FR	DC	PL	∅	∅	patch	pixel	agnostic	invisible	optimized	both	static	non-semantic	1d, 2d, 3d, 4d, 36d, 37d	∅
53a	*Kim et al.*	[90]	2023	YTF	AlexNet VGG16	FR	TL	CL	∅	∅	trigger-less backdoor (label flip specific to a single sample/out-of-distribution class)								∅	∅
54a	Na & Choi	[220]	2023	CelebA-Mask-HQ	InceptionV3 ResNet18 VGG16 DenseNet121	FR	MO	CL, PL	∅	∅	blended	pixel	specific	invisible	optimized	all-to-one	∅	semantic	2d, 3d, 4d, 20d	∅

Abbreviations (channels): **DEP**: deployment phase; **DC**: data collection; **MO**: model outsourcing; **TE**: training environment; **TL**: transfer learning

Abbreviations (injection methods): **AS**: anti-spoofing; **BPM**: back-propagation manipulation; **CL**: clean-label poisoning; **DAM**: data augmentation manipulation; **FD**: face detection; **FR**: face recognition; **LFM**: loss function manipulation; **MAM**: model architecture manipulation; **PL**: poison-label poisoning; **TWP**: targeted weight perturbation

TABLE 4. Taxonomy of backdoor defense tested on face recognition tasks (i.e. FRS blocks).

	Defense	Implem.		Target			Access required		Defense Specifications							Developed against	Improves on †	Broken by †
		Ref.	Year	Datasets	Models	Tasks	Data	Model	one-stage v. multi-stage	Model diagnostic	Model re-construction	Trigger re-construction	Data purification	Train-time filtering	Test-time filtering			
1d	*Fine-Pruning*	[128]	2018	LFW YTF	VGG16	FR	CVD	white-box (weights, activations, outputs)	one-stage	∅	RT, PR	∅	∅	∅	∅	3a	∅	8a, 11a, 26a, 29a, 40a, 42a, 46a, 49a, 52a
2d	*SentiNet*	[135]	2018	VGGFace LFW	VGG16	FR	CVD	logits	multi-stage	∅	∅	∅	∅	∅	AD	2a, 3a	∅	15a, 26a, 50a, 52a, 54a
3d	*Neural Cleanse*	[129]	2019	YTF PubFig VGGFace	VGG16	FR	CTD	white-box	multi-stage	AD	RT, PR	LTIF	∅	∅	AD	2a, 3a	∅	8a, 11a, 14a, 15a, 17a,19a, 23a, 26a, 29a, 40a, 42a, 45a, 50a,52a, 54a
		[131]	2023	VGGFace2	ResNet18													
4d	*STRIP* [134]	[154]	2019	YTF	CNN	FR	CVD	probits	one-stage	∅	∅	∅	∅	∅	AD	2a, 3a	2d, 3d	14a, 17a, 19a, 26a, 29a, 40a, 42a, 45a, 48a, 49a, 50a, 52a, 54a
5d	*DeepInspect*	[130]	2019	VGGFace	ResNet18	FR	SD	probits	multi-stage	∅	RT	LTIF◇	∅	∅	AD	2a, 3a	3d	∅
6d	*SCAn*	[132]	2019	MegaFace	ResNet101	FR	TrD CVD	penultimate layer activations	multi-stage	AD	∅	∅	∅	∅	AD	2a, 3a	2d, 3d, 4d	∅
7d	*TABOR*	[140]	2019	LFW	VGG16	FR	CTD	white-box	multi-stage	AD	∅	LTIF	∅	∅	∅	2a, 3a	3d	8a
8d	*NEO*	[147]	2019	VGGFace	VGG16	FR	CTD	white-box	multi-stage	∅	∅	SA	TR	∅	AD	2a, 3a	1d, 3d	∅
9d	*Februus*	[150]	2019	VGGFace2	VGG16	FR	CTD	white-box	multi-stage	∅	∅	∅	TR◇	∅	∅	2a	3d, 7d	45a
10d	*ABS*	[141]	2019	VGGFace	VGG16	FR	CTD‡	white-box	multi-stage	AD	∅	LTIF	∅	∅	∅	3a	3d	14a, 45a
11d	*MAMF*	[137]	2019	PubFig	VGG16	FR	CTD	white-box	multi-stage	AD	∅	LTIF	∅	∅	∅	3a	1d, 3d	∅
12d	*Neuron-Inspect*	[106]	2020	PubFig YTF	VGG16 CNN	FR	CVD	white-box	one-stage	AD	∅	∅	∅	∅	∅	2a, 3a	3d	12d
13d	*NNoculation*	[145]	2020	YTF	NiN [229]	FR	CVD	white-box	multi-stage	∅	RT	LTIF◇	∅	∅	AD	2a, 3a	3d	17a
14d	*Flip, Shrink-Pad* [136]	[149]	2020	PubFig	VGG16	FR	∅	labels	one-stage	∅	∅	∅	TR	∅	∅	1a, 3a, 8a	1d, 3d	∅
15d	*Laundering*	[110]	2020	YTF	CNN	FR	CTD	white-box	multi-stage	∅	RT, PR	∅	∅	∅	∅	3a	1d	∅
16d	Jin et al.	[155]	2020	PubFig	VGG16	FR	CTD	white-box	one-stage	∅	∅	∅	∅	∅	AD	2a, 3a	∅	∅
17d	*Confoc*	[123]	2020	VGGFace	VGG16	FR	TrD⋆	white-box	multi-stage	∅	RT	∅	TR	∅	∅	3a	3d	∅
18d	*TND* [142]	[83]	2020	MS-Celeb-1M	VGG16 ResNet18	FR	CVD‡◁	white-box	multi-stage	AD	∅	LTIF	∅	∅	∅	3a, 8a	3d	26a
19d	*Spectral Signatures* [152]	[83]	2020	MS-Celeb-1M	VGG16 ResNet18	FR	TrD	white-box	multi-stage	∅	RT	∅	∅	AD	∅	2a	∅	19a, 26a
		[114]	2020	VGGFace2 CelebA	ResNet20													
		[81]	2020	VGGFace VGGFace2	VGG16 DenseNet ResNet50													

†Claim with experiment on a FR task; ‡claims to need very few samples per class (≤ 10); ⋆claims to require $\leq 5\%$ of the original training dataset; ◇based on a GAN/Autoencoder solution; ◁has a data-free version; ▷unlabeled data; •explicitely assumes some knowledge about the trigger size

Abbreviations: **AD**: anomaly detection; **CTD**: clean test data; **(C)TrD**: (clean) training data; **CVD**: clean validation data; **LTIF**: learning of a trigger injection function; **PR**: pruning; **RT**: retraining; **SA**: search algorithm; **SD**: synthesized data (e.g. model inversion, GAN); **TR**: trigger removal

TABLE 5. *(Continued.)* **Taxonomy of backdoor defense tested on face recognition tasks (i.e. FRS blocks).**

	Defense	Implem.		Target			Access required		Defense Specifications							Developed against	Improves on †	Broken by †
		Ref.	Year	Datasets	Models	Tasks	Data	Model	one-stage v. multi-stage	Model diagnostic	Model re-construction	Trigger re-construction	Data purification	Train-time filtering	Test-time filtering			
20d	*Activation Clustering* [133]	[114]	2020	VGGFace2 CelebA	ResNet20	FR	TrD	penultimate layer activations	one-stage	∅	RT	∅	∅	AD	∅	2a, 3a	∅	16a, 19a, 54a
		[81]	2020	VGGFace VGGFace2	VGG16 DenseNet ResNet50													
21d	*CleaNN*	[157]	2020	VGGFace	VGG16	FR	CTD▷	white-box	multi-stage	∅	∅	LTIF	∅	∅	AD	3a	3d, 5d	∅
22d	*GangSweep*	[230]	2020	VGGFace	VGG16	FR	CVD	white-box	multi-stage	AD	RT	LTIF◇	∅	∅	∅	3a	3d	∅
23d	Fu et al.	[154]	2020	YTF	CNN	FR	CVD	white-box	multi-stage	∅	∅	∅	∅	∅	AD	2a, 3a	1d, 3d, 4d	∅
24d	*TAD*	[148]	2021	VGGFace	TrojAI [231]	FR	CTD‡	probits	multi-stage	AD	∅	SA	∅	∅	∅	∅	∅	∅
25d	*DeepSweep*	[149]	2021	PubFig	VGG16	FR	CTD	white-box	multi-stage	∅	RT	∅	TR	∅	∅	2a, 3a	1d, 3d, 14d	∅
26d	*FTD* [136]	[105]	2022	CelebA	ResNet18	FR	TrD, CTD	*not needed*	one-stage	∅	∅	∅	∅	AD	AD	1a, 2a, 3a	∅	39a
27d	*RAID*	[154]	2022	YTF	ResNet18	FR	CVD⋆	logits	multi-stage	∅	∅	∅	∅	∅	AD	2a, 3a	3d, 4d	∅
28d	*Ex-Ray*	[154]	2022	YTF	VGG16, NiN, TrojAI	FR	CTD‡	white-box	multi-stage	AD	∅	∅	TR	∅	∅	1a, 2a, 13a, 23a	3d, 10d	∅
29d	Unnervik & Marcel	[124]	2022	CASIA-Webface	FaceNet	FR	CTrD	white-box	multi-stage	AD	∅	∅	∅	∅	∅	2a, 3a	∅	∅
30d	Tao et al.	[144]	2022	ResNet18	CelebA	FR	CTD	labels	multi-stage	∅	∅	LTIF	∅	∅	∅	29a	3d	∅
31d	*Beatrix*	[139]	2022	VGG16	VGGFace	FR	CVD	white-box	multi-stage	AD	∅	∅	∅	∅	AD	∅	2d, 4d, 14d	∅
32d	*BDMAE*	[151]	2023	VGGFace2	ResNet18 VGG16	FR	CTD	labels	multi-stage	∅	∅	∅	TR◇	∅	∅	2a	9d	∅
33d	*DHBE*	[131]	2023	VGGFace2	ResNet18	FR	SD	logits	multi-stage	∅	MD	LTIF	∅	∅	∅	2a	3d, 35d	∅
34d	*MESA* [146]	[131]	2023	VGGFace2	ResNet18	FR	CTD	white-box	multi-stage	AD	RT	LTIF•	∅	∅	∅	2a	3d, 35d	∅
35d	*NAD* [125]	[131]	2023	VGGFace2	ResNet18	FR	CTrD⋆	white-box	multi-stage	∅	RT, MD	∅	∅	∅	∅	2a	3d	∅
36d	*ABL* [153]	[89]	2023	VGGFace2	ResNet18 VGG16	FR	TrD	white-box	multi-stage	∅	RT	∅	∅	AD	∅	1a, 2a, 3a, 8a, 11a, 25a	1d, 35d	52a
37d	*Scale-Up* [156]	[89]	2023	VGGFace2	ResNet18 VGG16	FR	CTD◁	labels	one-stage	∅	∅	∅	∅	∅	AD	2a, 8a, 26a, 29a	4d, 14d	52a
38d	*Meta-Sift*	[126]	2023	PubFig	ResNet18	FR	CTrD CVD CTD	white-box	multi-stage	∅	∅	∅	∅	AD	AD	1a, 2a, 8a, 29a	∅	∅
39d	D^3	[127]	2023	VGGFace	TrojAI	FR	CVD CTrD	white-box	multi-stage	∅	∅	LTIF◇	∅	AD	∅	3a	4d, 10d, 19d, 20d	∅

†Claim with experiment on a FR task; ‡claims to need very few samples per class (≤ 10); ⋆claims to require $\leq 5\%$ of the original training dataset; ◇based on a GAN/Autoencoder solution; ◁has a data-free version; ▷unlabeled data; •explicitely assumes some knowledge about the trigger size

Abbreviations: **AD**: anomaly detection; **CTD**: clean test data; **(C)TrD**: (clean) training data; **CVD**: clean validation data; **LTIF**: learning of a trigger injection function; **PR**: pruning; **RT**: retraining; **SA**: search algorithm; **SD**: synthesized data (e.g. model inversion, GAN); **TR**: trigger removal

TABLE 6. Model architectures typically used in the backdoor literature (in the context of FRS-related subtasks).

Model	Date	Ref.
AlexNet	2012	[25]
NiN	2013	[229]
VGG	2014	[232]
GoogLeNet	2014	[233]
ResNet	2015	[234]
Conv-LSTM	2015	[235]
FaceNet	2015	[28]
Siamese CNN	2015	[236]
DenseNet	2016	[237]
Inception-ResNet	2016	[238]
MobileNet	2017	[239]
Vision Transformer	2020	[240]
ResNet-LSTM	2020	[241]

We note that in the training environment injection setup, the attacker cannot proceed with the backdoor injection method as described above. We refer the reader to the methods involved for further details (see Tables 1-3).

B. TRIGGER-LESS BACKDOORS

To provide a summary formalization of trigger-less backdoor attacks, we draw inspiration from Ji et al. [95] and the DeepPoison [104] method.

Trigger-less backdoors typically involve the use case where the attacker provides the backdoored model (outsourcing channel) and is free in its choice of injection method (data poisoning, etc.). The attacker aims to build a model that misclassify inputs x_s of a *source* class y_s as inputs of a *target* class y_t *without* modifying x_s at test-time. To do so, the attacker designs a method that:

(1) generates sets of semantic neighbors of the source and target class, $\mathcal{X}_s$ and $\mathcal{X}_t$,
(2) trains a victim DNN f_θ^{po} s.t. the the features associated with $\mathcal{X}_s$ are close to those of $\mathcal{X}_t$, this leads to misclassifications of inputs of class y_s as y_t on hold-out sets.

Step (1) mines a set of adversarial perturbations such that, given a reference, benign DNN f for instance, the perturbations over $\mathcal{X}_s$ yield the same features as those of $\mathcal{X}_t$. The perturbed set $\mathcal{X}_s$, when reintegrated in the original training dataset $\mathcal{D}_{\text{train}}^{\text{cl}}$ is used to train a victim DNN f_θ^{po} following a multi-task loss function that trains a model to emulate a benign behavior on clean data alongside the hidden, malicious backdoor behavior. We note that steps (1) and (2) can either happen sequentially [95] or simultaneously [104]. In the latter, DeepPoison concurrently trains a GAN alongside f_θ^{po} such that the set of perturbations over $\mathcal{X}_s$ is refined during the training of f_θ^{po}.

APPENDIX D
ETHICAL AND PRIVACY CONCERNS

Mentioned in Section VI-A1, face recognition involves ethical and privacy concerns stemming from the use of private, facial data. Because of this context, some existing backdoor defenses may not be possible depending not only on a given defender's capabilities but also local legal and privacy considerations For example, the GDPR framework impacts the use of FRS in the EU [198]. Though face data may fall under legitimate use, they are used to construct biometric templates as part of a FRS. In that regards, such data is considered sensitive and must be safeguarded.

Under this consideration, processing and saving user face data or a FRS' templates may be limited. For instance, performing offline verification on such biometric templates or accessing user data in transit may not be allowed beyond an authorized research context (as per Article 9(1)(j) GDPR [198]). This may limit a defender's capacity to both filter data but also run post-mortems once an attack (which carries beyond the realm of backdoors) is detected in an industry environment. Nonetheless, we note the constant need for developing privacy-preserving defenses, an longstanding topic in biometry [38], notably with respect to backdoor attacks.

As a final note, we highlight that a changing legal framework with respect to machine learning usage may also introduce new attack surfaces. When considering the sensitive data used to train DNNs found in a FRS, legislations like GDPR [198] enforce a right to be forgotten which may obligate FRS providers to recurringly retrain their DNNs, increasing the risk of data poisoning for instance. Such a jurisdiction-dependent FRS threat model is a topic that backdoor defenders must take into account.

APPENDIX E
BACKDOORS ACROSS DIVERSE FIELDS

As noted in Section I, backdoor attacks and defenses expand far beyond facial recognition such as with acoustics, NLP, 3D point cloud or reinforcement learning [35] or federated learning [161].

Here, we note that backdoor attacks have also affected fields that may intersect with more complex recognition systems than FRS. For instance, a recognition system may combine both facial and voice or other physical markers to perform multi-modal authentication [43], [60]. In such situation, one may assume multimodality is a guarantee of higher security. However, the existence of backdoors beyond the image medium [35] undermines this idea of security. Similarly, face recognition tasks that exploit signal beyond the visible spectrum (e.g. infrared sprectrum [60]) may also be attacked. We surmise the development of backdoor attack demonstrations on such systems is likely in the future.

TABLE 7. Face datasets used in the backdoor literature.

Name	Date	Ref.	Format	# of identities	# of instances	Intended Use				Instance Size
						antispoofing	attributes	detection	recognition	
LFW	2007	[242]	Images	5,749	13,233	✗	✗	✗	✓	250x250
PubFig	2009	[243]	Images	200	58,797	✗	✓	✗	✓	256x256
YTF	2011	[175]	Videos	1,595	3,425	✗	✗	✓	✓	variable
Replay Attack	2012	[244]	Videos	50	1,300	✓	✓	✗	✗	variable
Casia-Webface	2014	[245]	Images	10,575	494,414	✗	✗	✓	✓	variable
VGGFace	2015	[26]	Images	2,662	2.6m	✗	✗	✗	✓	224x224
PIPA	2015	[246]	Images	2,356	37,107	✗	✗	✓	✓	variable
CelebA	2015	[247]	Images	10,177	202,599	✗	✓	✓	✓	variable
MS-Celeb-1M	2016	[248]	Images	99,892	8,456,240	✗	✗	✓	✓	variable
Megaface	2016	[249]	Images	690,572	4.7m	✗	✗	✓	✓	variable
VGGFace2	2018	[250]	Images	9,131	3.31m	✗	✗	✗	✓	224x224
FFHQ	2018	[251]	Images	14,000+	70,000	✗	✓	✗	✗	1024x1024
CelebAMask-HQ	2020	[252]	Images	n.a.	30,000	✗	✓	✗	✗	500x500

APPENDIX F
ACCOMPANYING TABLES

Tables 1-3 inventory 54 backdoor attacks that use a FRS model as one of their use case. Table 4 and Table 5 inventory 39 backdoor defenses in the same context.

ACKNOWLEDGMENT

The authors would like to thank the reviewers for their valuable insights.

REFERENCES

[1] 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.

[2] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 2, pp. 318–327, Feb. 2020.

[3] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," 2018, *arXiv:1810.04805*.

[4] Amazon AWS. (2023). *Amazon Rekognition*. Accessed: Aug. 21, 2023. [Online]. Available: https://aws.amazon.com/rekognition/

[5] Amazon AWS. (2024). *Amazon Sagemaker*. Accessed: Jan. 21, 2024. [Online]. Available: https://aws.amazon.com/sagemaker/

[6] Microsoft. (2023). *Microsoft Azure Face API*. Accessed: Aug. 21, 2023. [Online]. Available: https://azure.microsoft.com/en-gb/products/cognitive-services/face

[7] HuggingFace. (2023). *PyTorch Image Models, Scripts, Pretrained Weights*. Accessed: Aug. 8, 2023. [Online]. Available: https://github.com/huggingface/pytorch-image-models

[8] C. P. Pfleeger and S. L. Pfleeger, *Analyzing Computer Security—A Threat/Vulnerability/Countermeasure Approach*. USA: Prentice-Hall, 2012.

[9] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," 2016, *arXiv:1610.05820*.

[10] M. Goldblum, D. Tsipras, C. Xie, X. Chen, A. Schwarzschild, D. Song, A. Madry, B. Li, and T. Goldstein, "Dataset security for machine learning: Data poisoning, backdoor attacks, and defenses," 2020, *arXiv:2012.10544*.

[11] M. Barreno, B. Nelson, R. Sears, A. D. Joseph, and J. D. Tygar, "Can machine learning be secure?" in *Proc. ACM Symp. Inf., Comput. Commun. Secur. (ASIACCS)*. New York, NY, USA: Association for Computing Machinery, 2006, pp. 16–25.

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

[13] X. Wang, J. Peng, S. Zhang, B. Chen, Y. Wang, and Y. Guo, "A survey of face recognition," 2022, *arXiv:2212.13038*.

[14] B. Wu, Z. Zhu, L. Liu, Q. Liu, Z. He, and S. Lyu, "Attacks in adversarial machine learning: A systematic survey from the life-cycle perspective," 2023, *arXiv:2302.09457*.

[15] M. Wang and W. Deng, "Deep face recognition: A survey," *Neurocomputing*, vol. 429, pp. 215–244, Mar. 2021.

[16] G. Guo and N. Zhang, "A survey on deep learning based face recognition," *Comput. Vis. Image Understand.*, vol. 189, Dec. 2019, Art. no. 102805.

[17] F. Jiang, Y. Liu, B. Liu, X. Chen, and Q. Li, "A survey of domain generalization-based face anti-spoofing," in *Proc. 16th Chin. Conf. Biometric Recognit. (CCBR)*. Beijing, China: Springer, Nov. 2022, pp. 127–137.

[18] S. S. A. Zaidi, M. S. Ansari, A. Aslam, N. Kanwal, M. Asghar, and B. Lee, "A survey of modern deep learning based object detection models," 2021, *arXiv:2104.11892*.

[19] W. Bledsoe, "The model method in facial recognition," Panoramic Res., Palo Alto, CA, USA, Tech. Rep. PRI 15, 1964.

[20] K. K. Sung, "Learning and example selection for object and pattern detection," MIT, USA, Tech. Rep. AITR-1572, 1995.

[21] M. Turk and A. Pentland, "Eigenfaces for recognition," *J. Cogn. Neurosci.*, vol. 3, no. 1, pp. 71–86, Jan. 1991.

[22] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Knowledge Discovery and Data Mining*, 1996. [Online]. Available: https://api.semanticscholar.org/CorpusID:355163

[23] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 2, Los Alamitos, CA, USA, Dec. 2001, p. 511.

[24] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, San Diego, CA, USA, C. Schmid, S. Soatto, and C. Tomasi, Eds., Jun. 2005, pp. 886–893.

[25] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017.

[26] O. M. Parkhi, A. Vedaldi, and A. Zisserman, "Deep face recognition," in *Proc. Brit. Mach. Vis. Conf.* Durham, U.K.: BMVA Press, 2015, pp. 41.1–41.12.

[27] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "DeepFace: Closing the gap to human-level performance in face verification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1701–1708.

[28] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 815–823.

[29] S. Zafeiriou, C. Zhang, and Z. Zhang, "A survey on face detection in the wild: Past, present and future," *Comput. Vis. Image Understand.*, vol. 138, pp. 1–24, Sep. 2015.

[30] Y. Liu, Y. Xie, and A. Srivastava, "Neural trojans," 2017, *arXiv:1710.00942*.

[31] Y. Gao, B. Gia 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*.

[32] 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*.

[33] W. Li, J. Yu, X. Ning, P. Wang, Q. Wei, Y. Wang, and H. Yang, "Hu-Fu: Hardware and software collaborative attack framework against neural networks," in *Proc. IEEE Comput. Soc. Annu. Symp. VLSI (ISVLSI)*, Jul. 2018, pp. 482–487.

[34] J. Dumford and W. Scheirer, "Backdooring convolutional neural networks via targeted weight perturbations," 2018, *arXiv:1812.03128*.

[35] Y. Li, Y. Jiang, Z. Li, and S.-T. Xia, "Backdoor learning: A survey," 2020, *arXiv:2007.08745*.
[36] M. Sharif, S. Bhagavatula, L. Bauer, and M. K. Reiter, "Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur. (CCS)*. New York, NY, USA: Association for Computing Machinery, Oct. 2016, pp. 1528–1540.
[37] C. Pasquini and R. Böhme, "Trembling triggers: Exploring the sensitivity of backdoors in DNN-based face recognition," *EURASIP J. Inf. Secur.*, vol. 2020, no. 1, pp. 1–15, Dec. 2020.
[38] N. K. Ratha, J. H. Connell, and R. M. Bolle, "Enhancing security and privacy in biometrics-based authentication systems," *IBM Syst. J.*, vol. 40, no. 3, pp. 614–634, 2001.
[39] S. Pigeon and L. Vandendorpe, "Image-based multimodal face authentication," *Signal Process.*, vol. 69, no. 1, pp. 59–79, Aug. 1998.
[40] A. K. Jain, K. Nandakumar, and A. Nagar, "Biometric template security," *EURASIP J. Adv. Signal Process*, vol. 2008, pp. 1–17, Jan. 2008.
[41] M. Joshi, B. Mazumdar, and S. Dey, "Security vulnerabilities against fingerprint biometric system," 2018, *arXiv:1805.07116*.
[42] R. M. Thanki and K. R. Borisagar, "Discrete wavelet transform and compressive sensing based multibiometric watermarking—A novel approach to embed watermark into biometrie," in *Proc. 2nd Int. Conf. Emerg. Technol. Trends Electron., Commun. Netw.*, Dec. 2014, pp. 1–6.
[43] H. Altaleb and S. Koçak, "The risk of using biometrics," in *Proc. FIKUSZ Symp. Young Researchers*. Óbuda Univ. Keleti Károly Faculty of Economics, 2018, pp. 32–42.
[44] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld, "Face recognition: A literature survey," *ACM Comput. Surv.*, vol. 35, no. 4, pp. 399–458, Dec. 2003.
[45] A. George, C. Ecabert, H. O. Shahreza, K. Kotwal, and S. Marcel, "EdgeFace: Efficient face recognition model for edge devices," 2023, *arXiv:2307.01838*.
[46] C. Kong, S. Wang, and H. Li, "Digital and physical face attacks: Reviewing and one step further," 2022, *arXiv:2209.14692*.
[47] M. A. Hmani, D. Petrovska-Delacrétaz, and B. Dorizzi, "Locality preserving binary face representations using auto-encoders," *IET Biometrics*, vol. 11, no. 5, pp. 445–458, Sep. 2022.
[48] Y. Wang, S. Rane, S. C. Draper, and P. Ishwar, "An information-theoretic analysis of revocability and reusability in secure biometrics," in *Proc. Inf. Theory Appl. Workshop*, Feb. 2011, pp. 1–10.
[49] H. Wang, Z. Li, X. Ji, and Y. Wang, "Face R-CNN," 2017, *arXiv:1706.01061*.
[50] H. Jiang and E. Learned-Miller, "Face detection with the faster R-CNN," 2016, *arXiv:1606.03473*.
[51] Z. Yu, H. Huang, W. Chen, Y. Su, Y. Liu, and X. Wang, "YOLO-FaceV2: A scale and occlusion aware face detector," 2022, *arXiv:2208.02019*.
[52] J. Deng, J. Guo, Y. Zhou, J. Yu, I. Kotsia, and S. Zafeiriou, "RetinaFace: Single-stage dense face localisation in the wild," 2019, *arXiv:1905.00641*.
[53] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, "Joint face detection and alignment using multitask cascaded convolutional networks," *IEEE Signal Process. Lett.*, vol. 23, no. 10, pp. 1499–1503, Oct. 2016.
[54] X. Zhu and D. Ramanan, "Face detection, pose estimation, and landmark localization in the wild," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 2879–2886.
[55] X. Jin and X. Tan, "Face alignment in-the-wild: A survey," *Comput. Vis. Image Understand.*, vol. 162, pp. 1–22, Sep. 2017.
[56] C. Álvarez Casado and M. Bordallo López, "Real-time face alignment: Evaluation methods, training strategies and implementation optimization," *J. Real-Time Image Process.*, vol. 18, no. 6, pp. 2239–2267, Dec. 2021.
[57] X. Tan, y. Liu, J. Liu, and L. Jiang, "Face liveness detection from a single image with sparse low rank bilinear discriminative model," in *Proc. Eur. Conf. Comput. Vis.*, vol. 6316, 2010, pp. 504–517.
[58] B. Zhang, B. Tondi, and M. Barni, "Attacking CNN-based anti-spoofing face authentication in the physical domain," 2019, *arXiv:1910.00327*.
[59] Z. Yu, C. Zhao, Z. Wang, Y. Qin, Z. Su, X. Li, F. Zhou, and G. Zhao, "Searching central difference convolutional networks for face anti-spoofing," 2020, *arXiv:2003.04092*.
[60] Z. Yu, Y. Qin, X. Li, C. Zhao, Z. Lei, and G. Zhao, "Deep learning for face anti-spoofing: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 5, pp. 5609–5631, Oct. 2023.
[61] J. Deng, J. Guo, J. Yang, N. Xue, I. Kotsia, and S. Zafeiriou, "ArcFace: Additive angular margin loss for deep face recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 10, pp. 5962–5979, Oct. 2022.
[62] S. G. Kanade, D. Petrovska-Delacrétaz, and B. Dorizzi, *Enhancing Information Security and Privacy by Combining Biometrics with Cryptography*, vol. 3. USA: Morgan & Claypool, Jun. 2012.
[63] P. Drozdowski, F. Struck, C. Rathgeb, and C. Busch, "Benchmarking binarisation schemes for deep face templates," in *Proc. 25th IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2018, pp. 191–195.
[64] H. Wang, Y. Wang, Z. Zhou, X. Ji, D. Gong, J. Zhou, Z. Li, and W. Liu, "CosFace: Large margin cosine loss for deep face recognition," 2018, *arXiv:1801.09414*.
[65] T. de Freitas Pereira, A. Anjos, J. M. De Martino, and S. Marcel, "Can face anti-spoofing countermeasures work in a real world scenario?" in *Proc. Int. Conf. Biometrics (ICB)*, Jun. 2013, pp. 1–8.
[66] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," 2013, *arXiv:1312.6199*.
[67] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," 2014, *arXiv:1412.6572*.
[68] B. Biggio, I. Corona, D. Maiorca, B. Nelson, N. Šrndić, P. Laskov, G. Giacinto, and F. Roli, "Evasion attacks against machine learning at test time," in *Proc. Eur. Conf. Mach. Learn. Knowl. Discovery Databases (ECMLPKDD)*, Prague, Czech Republic. Berlin, Germany: Springer-Verlag, 2013, pp. 387–402.
[69] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," 2016, *arXiv:1607.02533*.
[70] S.-H. Chan, Y. Dong, J. Zhu, X. Zhang, and J. Zhou, "BadDet: Backdoor attacks on object detection," 2022, *arXiv:2205.14497*.
[71] A. Bhalerao, K. Kallas, B. Tondi, and M. Barni, "Luminance-based video backdoor attack against anti-spoofing rebroadcast detection," in *Proc. IEEE 21st Int. Workshop Multimedia Signal Process. (MMSP)*, Sep. 2019, pp. 1–6.
[72] ONNX. *ONNX/Models: A Collection of Pre-Trained, State-of-the-Art Models in the ONNX Format*. Accessed: Mar. 28, 2024. [Online]. Available: https://github.com/onnx/onnx
[73] G. Devic, G. Sassatelli, and A. Gamatié, "Energy-efficient machine learning on FPGA for edge devices: A case study," in *Proc. Conf. Parallélisme, Architecture Syst. (ComPAS)*, Lyon, France, Jun. 2020.
[74] R. C. Panicker, A. Kumar, and D. John, "Introducing FPGA-based machine learning on the edge to undergraduate students," in *Proc. IEEE Frontiers Educ. Conf. (FIE)*, Los Alamitos, CA, USA, Oct. 2020, pp. 1–5.
[75] N. Akhtar, A. Mian, N. Kardan, and M. Shah, "Advances in adversarial attacks and defenses in computer vision: A survey," *IEEE Access*, vol. 9, pp. 155161–155196, 2021.
[76] M. S. Pydi and V. Jog, "The many faces of adversarial risk," 2022, *arXiv:2201.08956*.
[77] H. Wei, H. Tang, X. Jia, Z. Wang, H. Yu, Z. Li, S. Satoh, L. Van Gool, and Z. Wang, "Physical adversarial attack meets computer vision: A decade survey," 2022, *arXiv:2209.15179*.
[78] Z. Yuan, P. Zhou, K. Zou, and Y. Cheng, "You are catching my attention: Are vision transformers bad learners under backdoor attacks?" in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Los Alamitos, CA, USA, Jun. 2023, pp. 24605–24615.
[79] Y. Li, T. Zhai, Y. Jiang, Z. Li, and S.-T. Xia, "Backdoor attack in the physical world," 2021, *arXiv:2104.02361*.
[80] Y. Liu, X. Ma, J. Bailey, and F. Lu, "Reflection backdoor: A natural backdoor attack on deep neural networks," 2020, *arXiv:2007.02343*.
[81] 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. 6206–6215.
[82] H. Li, Y. Wang, X. Xie, Y. Liu, S. Wang, R. Wan, L.-P. Chau, and A. C. Kot, "Light can hack your face! Black-box backdoor attack on face recognition systems," 2020, *arXiv:2009.06996*.
[83] 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.
[84] A. Nguyen and A. Tran, "WaNet—Imperceptible warping-based backdoor attack," 2021, *arXiv:2102.10369*.
[85] J. Zhang, C. Dongdong, Q. Huang, J. Liao, W. Zhang, H. Feng, G. Hua, and N. Yu, "Poison ink: Robust and invisible backdoor attack," *IEEE Trans. Image Process.*, vol. 31, pp. 5691–5705, 2022.
[86] T. Wang, Y. Yao, F. Xu, S. An, H. Tong, and T. Wang, "Backdoor attack through frequency domain," 2021, *arXiv:2111.10991*.
[87] M. Peng, X. Li, and M. Sun, "Efficient and stealthy backdoor attack triggers are close at hand," in *Proc. ICLR*, 2023.

[88] J. Zhang, J. Xu, Z. Zhang, and Y. Gao, "Imperceptible sample-specific backdoor to DNN with denoising autoencoder," 2023, *arXiv:2302.04457*.
[89] Y. Gao, Y. Li, X. Gong, Z. Li, S.-T. Xia, and Q. Wang, "Backdoor attack with sparse and invisible trigger," 2023, *arXiv:2306.06209*.
[90] T.-H. Kim, S.-H. Choi, and Y.-H. Choi, "Instance-agnostic and practical clean label backdoor attack method for deep learning based face recognition models," *IEEE Access*, vol. 11, pp. 144040–144050, 2023.
[91] E. Bagdasaryan and V. Shmatikov, "Blind backdoors in deep learning models," 2020, *arXiv:2005.03823*.
[92] 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.
[93] 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.
[94] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" 2014, *arXiv:1411.1792*.
[95] Y. Ji, X. Zhang, S. Ji, X. Luo, and T. Wang, "Model-reuse attacks on deep learning systems," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Oct. 2018, pp. 349–363.
[96] Y. Ge, Q. Wang, B. Zheng, X. Zhuang, Q. Li, C. Shen, and C. Wang, "Anti-distillation backdoor attacks: Backdoors can really survive in knowledge distillation," in *Proc. 29th ACM Int. Conf. Multimedia*. New York, NY, USA: Association for Computing Machinery, Oct. 2021, pp. 826–834.
[97] 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.* New York, NY, USA: Association for Computing Machinery, Nov. 2019, pp. 2041–2055.
[98] X. Qi, J. Zhu, C. Xie, and Y. Yang, "Subnet replacement: Deployment-stage backdoor attack against deep neural networks in gray-box setting," 2021, *arXiv:2107.07240*.
[99] A. Turner, D. Tsipras, and A. Madry, "Clean-label backdoor attacks," in *Proc. ICLR*, 2019.
[100] R. D. Jha, J. Hayase, and S. Oh, "Label poisoning is all you need," 2023, *arXiv:2310.18933*.
[101] W. Guo, B. Tondi, and M. Barni, "A temporal chrominance trigger for clean-label backdoor attack against anti-spoof rebroadcast detection," *IEEE Trans. Dependable Secure Comput.*, vol. 20, no. 6, pp. 4752–4762, Jan. 2023.
[102] T. Qin, X. Gao, X. He, Y. Zhao, K. Ye, and C. Z. Xu, "Flareon: Stealthy backdoor injection via poisoned augmentation," in *Proc. ICLR*, 2023.
[103] J. Lin, L. Xu, Y. Liu, and X. Zhang, "Composite backdoor attack for deep neural network by mixing existing benign features," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.* New York, NY, USA: Association for Computing Machinery, 2020, pp. 113–131.
[104] J. Chen, L. Zhang, H. Zheng, X. Wang, and Z. Ming, "DeepPoison: Feature transfer based stealthy poisoning attack," 2021, *arXiv:2101.02562*.
[105] N. Zhong, Z. Qian, and X. Zhang, "Imperceptible backdoor attack: From input space to feature representation," 2022, *arXiv:2205.03190*.
[106] R. Tang, M. Du, N. Liu, F. Yang, and X. Hu, "An embarrassingly simple approach for trojan attack in deep neural networks," 2020, *arXiv:2006.08131*.
[107] A. Salem, M. Backes, and Y. Zhang, "Don't trigger me! A triggerless backdoor attack against deep neural networks," 2020, *arXiv:2010.03282*.
[108] G. Sokar, R. Agarwal, P. S. Castro, and U. Evci, "The dormant neuron phenomenon in deep reinforcement learning," 2023, *arXiv:2302.12902*.
[109] S. Hong, N. Carlini, and A. Kurakin, "Handcrafted backdoors in deep neural networks," 2021, *arXiv:2106.04690*.
[110] W. Aiken, H. Kim, and S. Woo, "Neural network laundering: Removing black-box backdoor watermarks from deep neural networks," 2020, *arXiv:2004.11368*.
[111] M. Barni, K. Kallas, and B. Tondi, "A new backdoor attack in CNNs by training set corruption without label poisoning," 2019, *arXiv:1902.11237*.
[112] J. Duchon, "Splines minimizing rotation-invariant semi-norms in Sobolev spaces," in *Constructive Theory of Functions of Several Variables*. Berlin, Germany: Springer, 1976.
[113] Y. Zeng, W. Park, Z. M. Mao, and R. Jia, "Rethinking the backdoor attacks' triggers: A frequency perspective," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 16453–16461.
[114] E. Sarkar, H. Benkraouda, and M. Maniatakos, "FaceHack: Triggering backdoored facial recognition systems using facial characteristics," 2020, *arXiv:2006.11623*.
[115] S. Cheng, Y. Liu, S. Ma, and X. Zhang, "Deep feature space trojan attack of neural networks by controlled detoxification," 2020, *arXiv:2012.11212*.
[116] Y. Yu, Y. Wang, W. Yang, S. Lu, Y.-P. Tan, and A. C. Kot, "Backdoor attacks against deep image compression via adaptive frequency trigger," 2023, *arXiv:2302.14677*.
[117] A. Salem, R. Wen, M. Backes, S. Ma, and Y. Zhang, "Dynamic backdoor attacks against machine learning models," in *Proc. IEEE 7th Eur. Symp. Secur. Privacy (EuroS&P)*, Jun. 2022, pp. 703–718.
[118] Z. Wang, J. Zhai, and S. Ma, "BppAttack: Stealthy and efficient trojan attacks against deep neural networks via image quantization and contrastive adversarial learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 15074–15084.
[119] T. N. Huynh, D. M. Nguyen, T. Pham, and A. T. Tran, "COMBAT: Alternated training for near-perfect clean-label backdoor attacks," in *Proc. ICLR*, 2023.
[120] A. Nguyen and A. Tran, "Input-aware dynamic backdoor attack," 2020, *arXiv:2010.08138*.
[121] J. Guo and C. Liu, "Practical poisoning attacks on neural networks," in *Proc. Eur. Conf. Comput. Vis.*, 2020.
[122] W. Guo, B. Tondi, and M. Barni, "A master key backdoor for universal impersonation attack against DNN-based face verification," *Pattern Recognit. Lett.*, vol. 144, pp. 61–67, Apr. 2021.
[123] M. Villarreal-Vasquez and B. Bhargava, "ConFoc: Content-focus protection against trojan attacks on neural networks," 2020, *arXiv:2007.00711*.
[124] A. Unnervik and S. Marcel, "An anomaly detection approach for backdoored neural networks: Face recognition as a case study," 2022, *arXiv:2208.10231*.
[125] 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*.
[126] Y. Zeng, M. Pan, H. Jahagirdar, M. Jin, L. Lyu, and R. Jia, "Meta-Sift: How to sift out a clean subset in the presence of data poisoning?" in *Proc. 32nd USENIX Secur. Symp. (USENIX Security)*. Anaheim, CA, USA: USENIX Association, Aug. 2023, pp. 1667–1684.
[127] L. Yan, S. Cheng, G. Shen, G. Tao, X. Chen, K. Zhang, Y. Mao, and X. Zhang, "D3: Detoxing deep learning dataset," in *Proc. NeurIPS Workshop Backdoors Deep Learn. Good, Bad, Ugly*, 2023.
[128] K. Liu, B. Dolan-Gavitt, and S. Garg, "Fine-pruning: Defending against backdooring attacks on deep neural networks," 2018, *arXiv:1805.12185*.
[129] 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.
[130] H. Chen, C. Fu, J. Zhao, and F. Koushanfar, "DeepInspect: A black-box trojan detection and mitigation framework for deep neural networks," in *Proc. Int. Joint Conf. Artif. Intell.*, 2019, pp. 4658–4664.
[131] Z. Yan, S. Li, R. Zhao, Y. Tian, and Y. Zhao, "DHBE: Data-free holistic backdoor erasing in deep neural networks via restricted adversarial distillation," 2023, *arXiv:2306.08009*.
[132] D. Tang, X. Wang, H. Tang, and K. Zhang, "Demon in the variant: Statistical analysis of DNNs for robust backdoor contamination detection," 2019, *arXiv:1908.00686*.
[133] 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*.
[134] Y. Gao, C. Xu, D. Wang, S. Chen, D. C. Ranasinghe, and S. Nepal, "STRIP: A defence against trojan attacks on deep neural networks," 2019, *arXiv:1902.06531*.
[135] E. Chou, F. Tramèr, and G. Pellegrino, "SentiNet: Detecting localized universal attacks against deep learning systems," 2018, *arXiv:1812.00292*.
[136] Y. Li, T. Zhai, B. Wu, Y. Jiang, Z. Li, and S. Xia, "Rethinking the trigger of backdoor attack," 2020, *arXiv:2004.04692*.
[137] Z. Xiang, D. J. Miller, H. Wang, and G. Kesidis, "Revealing perceptible backdoors, without the training set, via the maximum achievable misclassification fraction statistic," 2019, *arXiv:1911.07970*.
[138] Y. Liu, G. Shen, G. Tao, Z. Wang, S. Ma, and X. Zhang, "Complex backdoor detection by symmetric feature differencing," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 14983–14993.
[139] W. Ma, D. Wang, R. Sun, M. Xue, S. Wen, and Y. Xiang, "The 'Beatrix' resurrections: Robust backdoor detection via Gram matrices," 2022, *arXiv:2209.11715*.

[140] W. Guo, L. Wang, X. Xing, M. Du, and D. Song, "TABOR: A highly accurate approach to inspecting and restoring trojan backdoors in AI systems," 2019, *arXiv:1908.01763*.
[141] 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.* New York, NY, USA: Association for Computing Machinery, Nov. 2019, pp. 1265–1282.
[142] R. Wang, G. Zhang, S. Liu, P.-Y. Chen, J. Xiong, and M. Wang, "Practical detection of trojan neural networks: Data-limited and data-free cases," 2020, *arXiv:2007.15802*.
[143] S.-M. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, and P. Frossard, "Universal adversarial perturbations," 2016, *arXiv:1610.08401*.
[144] G. Tao, G. Shen, Y. Liu, S. An, Q. Xu, S. Ma, P. Li, and X. Zhang, "Better trigger inversion optimization in backdoor scanning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 13358–13368.
[145] A. K. Veldanda, K. Liu, B. Tan, P. Krishnamurthy, F. Khorrami, R. Karri, B. Dolan-Gavitt, and S. Garg, "NNoculation," in *Proc. 14th ACM Workshop Artif. Intell. Secur.*, Nov. 2021, pp. 49–60.
[146] X. Qiao, Y. Yang, and H. Li, "Defending neural backdoors via generative distribution modeling," in *Advances in Neural Information Processing Systems*, vol. 32, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds. Red Hook, NY, USA: Curran Associates, 2019.
[147] S. Udeshi, S. Peng, G. Woo, L. Loh, L. Rawshan, and S. Chattopadhyay, "Model agnostic defence against backdoor attacks in machine learning," 2019, *arXiv:1908.02203*.
[148] X. Zhang, H. Chen, and F. Koushanfar, "TAD: Trigger approximation based black-box trojan detection for AI," 2021, *arXiv:2102.01815*.
[149] H. Qiu, Y. Zeng, S. Guo, T. Zhang, M. Qiu, and B. Thuraisingham, "DeepSweep: An evaluation framework for mitigating DNN backdoor attacks using data augmentation," in *Proc. ACM Asia Conf. Comput. Commun. Secur.* New York, NY, USA: Association for Computing Machinery, May 2021, pp. 363–377.
[150] B. G. Doan, E. Abbasnejad, and D. C. Ranasinghe, "Februus: Input purification defense against trojan attacks on deep neural network systems," in *Proc. Annu. Comput. Secur. Appl. Conf.*, Dec. 2020, pp. 897–912.
[151] T. Sun, L. Pang, C. Chen, and H. Ling, "Mask and restore: Blind backdoor defense at test time with masked autoencoder," 2023, *arXiv:2303.15564*.
[152] B. Tran, J. Li, and A. Madry, "Spectral signatures in backdoor attacks," 2018, *arXiv:1811.00636*.
[153] Y. Li, X. Lyu, N. Koren, L. Lyu, B. Li, and X. Ma, "Anti-backdoor learning: Training clean models on poisoned data," 2021, *arXiv:2110.11571*.
[154] H. Fu, A. K. Veldanda, P. Krishnamurthy, S. Garg, and F. Khorrami, "A feature-based on-line detector to remove adversarial-backdoors by iterative demarcation," *IEEE Access*, vol. 10, pp. 5545–5558, 2022.
[155] K. Jin, T. Zhang, C. Shen, Y. Chen, M. Fan, C. Lin, and T. Liu, "Can we mitigate backdoor attack using adversarial detection methods?" 2020, *arXiv:2006.14871*.
[156] J. Guo, Y. Li, X. Chen, H. Guo, L. Sun, and C. Liu, "SCALE-UP: An efficient black-box input-level backdoor detection via analyzing scaled prediction consistency," in *Proc. 11th Int. Conf. Learn. Represent.*, 2023.
[157] M. Javaheripi, M. Samragh, G. Fields, T. Javidi, and F. Koushanfar, "CleaNN," in *Proc. 39th Int. Conf. Comput.-Aided Design*, Nov. 2020, pp. 1–9.
[158] B. Li, C. Chen, W. Wang, and L. Carin, *Certified Adversarial Robustness With Additive Noise*. Red Hook, NY, USA: Curran Associates Inc., 2019.
[159] B. Wang, X. Cao, J. jia, and N. Z. Gong, "On certifying robustness against backdoor attacks via randomized smoothing," 2020, *arXiv:2002.11750*.
[160] M. Weber, X. Xu, B. Karlaš, C. Zhang, and B. Li, "RAB: Provable robustness against backdoor attacks," 2020, *arXiv:2003.08904*.
[161] C. Xie, M. Chen, P.-Y. Chen, and B. Li, "CRFL: Certifiably robust federated learning against backdoor attacks," 2021, *arXiv:2106.08283*.
[162] A. Saha, A. Subramanya, and H. Pirsiavash, "Hidden trigger backdoor attacks," 2019, *arXiv:1910.00033*.
[163] G. Bae, M. de La Gorce, T. Baltrušaitis, C. Hewitt, D. Chen, J. Valentin, R. Cipolla, and J. Shen, "Digiface-1M: 1 million digital face images for face recognition," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2023, pp. 3515–3524.
[164] H. Qiu, B. Yu, D. Gong, Z. Li, W. Liu, and D. Tao, "SynFace: Face recognition with synthetic data," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 10860–10870.
[165] M. Kim, F. Liu, A. Jain, and X. Liu, "DCFace: Synthetic face generation with dual condition diffusion model," 2023, *arXiv:2304.07060*.
[166] E. Wood, T. Baltrušaitis, C. Hewitt, S. Dziadzio, T. J. Cashman, and J. Shotton, "Fake it till you make it: Face analysis in the wild using synthetic data alone," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Oct. 2021, pp. 3681–3691.
[167] S.-Y. Chou, P.-Y. Chen, and T.-Y. Ho, "VillanDiffusion: A unified backdoor attack framework for diffusion models," in *Proc. 37th Conf. Neural Inf. Process. Syst.*, 2023.
[168] W. Chen, D. Song, and B. Li, "TrojDiff: Trojan attacks on diffusion models with diverse targets," 2023, *arXiv:2303.05762*.
[169] S. Hu, Z. Zhou, Y. Zhang, L. Y. Zhang, Y. Zheng, Y. He, and H. Jin, "BadHash: Invisible backdoor attacks against deep hashing with clean label," 2022, *arXiv:2207.00278*.
[170] R. Wang, H. Chen, Z. Zhu, L. Liu, Y. Zhang, Y. Fan, and B. Wu, "Robust backdoor attack with visible, semantic, sample-specific, and compatible triggers," 2023, *arXiv:2306.00816*.
[171] T. Xu, Y. Li, Y. Jiang, and S.-T. Xia, "BATT: Backdoor attack with transformation-based triggers," 2022, *arXiv:2211.01806*.
[172] N. Carlini and A. Terzis, "Poisoning and backdooring contrastive learning," 2021, *arXiv:2106.09667*.
[173] B. F. Klare, B. Klein, E. Taborsky, A. Blanton, J. Cheney, K. Allen, P. Grother, A. Mah, M. Burge, and A. K. Jain, "Pushing the frontiers of unconstrained face detection and recognition: IARPA Janus benchmark A," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1931–1939.
[174] M. Xue, C. He, Y. Wu, S. Sun, Y. Zhang, J. Wang, and W. Liu, "PTB: Robust physical backdoor attacks against deep neural networks in real world," *Comput. Secur.*, vol. 118, Jul. 2022, Art. no. 102726.
[175] L. Wolf, T. Hassner, and I. Maoz, "Face recognition in unconstrained videos with matched background similarity," in *Proc. CVPR*, Jun. 2011, pp. 529–534.
[176] H. H. Nguyen, S. Marcel, J. Yamagishi, and I. Echizen, "Master face attacks on face recognition systems," 2021, *arXiv:2109.03398*.
[177] P. Terhörst, F. Bierbaum, M. Huber, N. Damer, F. Kirchbuchner, K. Raja, and A. Kuijper, "On the (limited) generalization of MasterFace attacks and its relation to the capacity of face representations," 2022, *arXiv:2203.12387*.
[178] A. Salem, Y. Sautter, M. Backes, M. Humbert, and Y. Zhang, "BAAAN: Backdoor attacks against autoencoder and GAN-based machine learning models," 2020, *arXiv:2010.03007*.
[179] Z. Ye, H. Zhang, and Q. Liu, "SWTFace: A multi-branch network for masked face detection and recognition," in *Proc. 5th Int. Conf. Pattern Recognit. Artif. Intell. (PRAI)*, Aug. 2022, pp. 381–387.
[180] W. Yang and Z. Jiachun, "Real-time face detection based on Yolo," in *Proc. 1st IEEE Int. Conf. Knowl. Innov. Invention (ICKII)*, Jul. 2018, pp. 221–224.
[181] D. Fan, S. Fang, X. Liu, Y. Li, and S. Gao, "A multi-scale face detection algorithm based on improved SSD model," in *Proc. ACM Turing Celebration Conf. China*. New York, NY, USA: Association for Computing Machinery, May 2019.
[182] W. Tian, Z. Wang, H. Shen, W. Deng, Y. Meng, B. Chen, X. Zhang, Y. Zhao, and X. Huang, "Learning better features for face detection with feature fusion and segmentation supervision," 2018, *arXiv:1811.08557*.
[183] S. Minaee, P. Luo, Z. Lin, and K. Bowyer, "Going deeper into face detection: A survey," 2021, *arXiv:2103.14983*.
[184] E. Arkin, N. Yadikar, X. Xu, A. Aysa, and K. Ubul, "A survey: Object detection methods from CNN to transformer," *Multimedia Tools Appl.*, vol. 82, no. 14, pp. 21353–21383, Jun. 2023.
[185] L. Tran, J. Kossaifi, Y. Panagakis, and M. Pantic, "Disentangling geometry and appearance with regularised geometry-aware generative adversarial networks," *Int. J. Comput. Vis.*, vol. 127, nos. 6–7, pp. 824–844, Jun. 2019.
[186] H.-I. Kim, K. Yun, and Y. Man Ro, "Face shape-guided deep feature alignment for face recognition robust to face misalignment," 2022, *arXiv:2209.07220*.
[187] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu, "Spatial transformer networks," 2015, *arXiv:1506.02025*.
[188] Z. An, W. Deng, Y. Zhong, Y. Huang, and X. Tao, "APA: Adaptive pose alignment for robust face recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2019, pp. 227–235.
[189] Y. Yan, X. Naturel, T. Chateau, S. Duffner, C. Garcia, and C. Blanc, "A survey of deep facial landmark detection," in *Proc. RFIAP*, Paris, France, Jun. 2018.

[190] X. Jin and X. Tan, "Face alignment in-the-wild: A survey," 2016, *arXiv:1608.04188*.
[191] J. Meher, H. Allende-Cid, and T. E. M. Nordling, "A survey and classification of face alignment methods based on face models," 2023, *arXiv:2311.03082*.
[192] G. Pan, L. Sun, Z. Wu, and S. Lao, "Eyeblink-based anti-spoofing in face recognition from a generic webcamera," in *Proc. IEEE 11th Int. Conf. Comput. Vis.*, Oct. 2007, pp. 1–8.
[193] F. Jiang, P. Liu, X. Shao, and X. Zhou, "Face anti-spoofing with generated near-infrared images," *Multimedia Tools Appl.*, vol. 79, nos. 29–30, pp. 21299–21323, Aug. 2020.
[194] D. Sharma and A. Selwal, "A survey on face presentation attack detection mechanisms: Hitherto and future perspectives," *Multimedia Syst.*, vol. 29, no. 3, pp. 1527–1577, Jun. 2023.
[195] Q. Meng, S. Zhao, Z. Huang, and F. Zhou, "MagFace: A universal representation for face recognition and quality assessment," 2021, *arXiv:2103.06627*.
[196] Y. Liu, F. Wei, J. Shao, L. Sheng, J. Yan, and X. Wang, "Exploring disentangled feature representation beyond face identification," 2018, *arXiv:1804.03487*.
[197] L. Tran, X. Yin, and X. Liu, "Disentangled representation learning GAN for pose-invariant face recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1283–1292.
[198] *Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the Protection of Natural Persons With Regard To the Processing of Personal Data and on the Free Movement of Such Data, and Repealing Directive 95/46/EC (General Data Protection Regulation) (Text With EEA Relevance)*, document 32016R0679, European Commission, Geneva, Switzerland, 2016.
[199] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2017, pp. 3–18.
[200] B. Balle, G. Cherubin, and J. Hayes, "Reconstructing training data with informed adversaries," 2022, *arXiv:2201.04845*.
[201] J. Li, N. Li, and B. Ribeiro, "Membership inference attacks and defenses in classification models," in *Proc. 11th ACM Conf. Data Appl. Secur. Privacy*, Apr. 2021, pp. 5–16.
[202] L. Hu, A. Yan, H. Yan, J. Li, T. Huang, Y. Zhang, C. Dong, and C. Yang, "Defenses to membership inference attacks: A survey," *ACM Comput. Surv.*, vol. 56, no. 4, pp. 1–34, Nov. 2023.
[203] H. Hu, Z. Salcic, L. Sun, G. Dobbie, P. S. Yu, and X. Zhang, "Membership inference attacks on machine learning: A survey," *ACM Comput. Surv.*, vol. 54, no. 11s, pp. 1–37, Sep. 2022.
[204] Y. Huang, H. Chen, Y. Wang, and L. Wang, "Inference attacks against face recognition model without classification layers," 2024, *arXiv:2401.13719*.
[205] F. Tramèr, F. Zhang, A. Juels, M. K. Reiter, and T. Ristenpart, "Stealing machine learning models via prediction Apis," 2016, *arXiv:1609.02943*.
[206] T. Orekondy, B. Schiele, and M. Fritz, "Knockoff nets: Stealing functionality of black-box models," 2018, *arXiv:1812.02766*.
[207] D. Oliynyk, R. Mayer, and A. Rauber, "I know what you trained last summer: A survey on stealing machine learning models and defences," *ACM Comput. Surv.*, vol. 55, no. 14s, pp. 1–41, Jul. 2023.
[208] T. Lorenz, M. Kwiatkowska, and M. Fritz, "Certifiers make neural networks vulnerable to availability attacks," in *Proc. 16th ACM Workshop Artif. Intell. Secur.*, Nov. 2023, pp. 67–78.
[209] M. A. Ramirez, S.-K. Kim, H. Al Hamadi, E. Damiani, Y.-J. Byon, T.-Y. Kim, C.-S. Cho, and C. Y. Yeun, "Poisoning attacks and defenses on artificial intelligence: A survey," 2022, *arXiv:2202.10276*.
[210] A. Schwarzschild, M. Goldblum, A. Gupta, J. P. Dickerson, and T. Goldstein, "Just how toxic is data poisoning? A unified benchmark for backdoor and data poisoning attacks," in *Proc. 38th Int. Conf. Mach. Learn.*, in Proceedings of Machine Learning Research, vol. 139, M. Meila and T. Zhang, Eds., Jul. 2021, pp. 9389–9398.
[211] A. E. Cinà, K. Grosse, A. Demontis, S. Vascon, W. Zellinger, B. A. Moser, A. Oprea, B. Biggio, M. Pelillo, and F. Roli, "Wild patterns reloaded: A survey of machine learning security against training data poisoning," *ACM Comput. Surv.*, vol. 55, no. 13s, pp. 1–39, Jul. 2023.
[212] S. Sagar, C.-S. Li, S. W. Loke, and J. Choi, "Poisoning attacks and defenses in federated learning: A survey," 2023, *arXiv:2301.05795*.
[213] J. Lianga, R. Wang, C. Feng, and C.-C. Chang, "A survey on federated learning poisoning attacks and defenses," 2023, *arXiv:2306.03397*.
[214] F. Nuding and R. Mayer, "Data poisoning in sequential and parallel federated learning," in *Proc. ACM Int. Workshop Secur. Privacy Anal.* New York, NY, USA: Association for Computing Machinery, Apr. 2022, pp. 24–34.
[215] I. Shumailov, Y. Zhao, D. Bates, N. Papernot, R. Mullins, and R. Anderson, "Sponge examples: Energy-latency attacks on neural networks," 2020, *arXiv:2006.03463*.
[216] Z. Wang, S. Huang, Y. Huang, and H. Cui, "Energy-latency attacks to on-device neural networks via sponge poisoning," 2023, *arXiv:2305.03888*.
[217] A. Shapira, A. Zolfi, L. Demetrio, B. Biggio, and A. Shabtai, "Phantom sponges: Exploiting non-maximum suppression to attack deep object detectors," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2023, pp. 4560–4569.
[218] A. E. Cinà, A. Demontis, B. Biggio, F. Roli, and M. Pelillo, "Energy-latency attacks via sponge poisoning," 2022, *arXiv:2203.08147*.
[219] C. Liao, H. Zhong, A. Squicciarini, S. Zhu, and D. Miller, "Backdoor embedding in convolutional neural network models via invisible perturbation," 2018, *arXiv:1808.10307*.
[220] H. Na and D. Choi, "Image-synthesis-based backdoor attack approach for face classification task," *Electronics*, vol. 12, no. 21, p. 4535, Nov. 2023.
[221] Y. Zeng, M. Pan, H. A. Just, L. Lyu, M. Qiu, and R. Jia, "Narcissus: A practical clean-label backdoor attack with limited information," 2022, *arXiv:2204.05255*.
[222] M. Xue, X. Wang, S. Sun, Y. Zhang, J. Wang, and W. Liu, "Compression-resistant backdoor attack against deep neural networks," 2022, *arXiv:2201.00672*.
[223] H. Souri, L. Fowl, R. Chellappa, M. Goldblum, and T. Goldstein, "Sleeper agent: Scalable hidden trigger backdoors for neural networks trained from scratch," 2021, *arXiv:2106.08970*.
[224] X. Qi, T. Xie, R. Pan, J. Zhu, Y. Yang, and K. Bu, "Towards practical deployment-stage backdoor attack on deep neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 13347–13357.
[225] D. Tang, R. Zhu, X. Wang, H. Tang, and Y. Chen, "Understanding impacts of task similarity on backdoor attack and detection," 2022, *arXiv:2210.06509*.
[226] Z. Liu, F. Li, Z. Li, and B. Luo, "LoneNeuron: A highly-effective feature-domain neural trojan using invisible and polymorphic watermarks," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.* New York, NY, USA: Association for Computing Machinery, Nov. 2022, pp. 2129–2143.
[227] H. Phan, C. Shi, Y. Xie, T. Zhang, Z. Li, T. Zhao, J. Liu, Y. Wang, Y. Chen, and B. Yuan, "RIBAC: Towards robust and imperceptible backdoor attack against compact DNN," 2022, *arXiv:2208.10608*.
[228] X. Liu, Y.-A. Tan, Y. Wang, K. Qiu, and Y. Li, "Stealthy low-frequency backdoor attack against deep neural networks," 2023, *arXiv:2305.09677*.
[229] M. Lin, Q. Chen, and S. Yan, "Network in network," 2013, *arXiv:1312.4400*.
[230] L. Zhu, R. Ning, C. Wang, C. Xin, and H. Wu, "GangSweep: Sweep out neural backdoors by GAN," in *Proc. 28th ACM Int. Conf. Multimedia*. New York, NY, USA: Association for Computing Machinery, Oct. 2020, pp. 3173–3181.
[231] K. Karra, C. Ashcraft, and N. Fendley, "The TrojAI software framework: An OpenSource tool for embedding trojans into deep learning models," 2020, *arXiv:2003.07233*.
[232] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.
[233] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," 2014, *arXiv:1409.4842*.
[234] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 770–778.
[235] S. Xingjian, C. Zhourong, W. Hao, Y. Dit-Yan, W. Wai-Kin, and W. Wang-Chun, "Convolutional LSTM network: A machine learning approach for precipitation nowcasting," in *Proc. 28th Int. Conf. Neural Inf. Process. Syst. (NIPS)*, vol. 1. Cambridge, MA, USA: MIT Press, 2015, pp. 802–810.
[236] G. R. Koch, R. Zemel, and R. Salakhutdinov, "Siamese neural networks for one-shot image recognition," in *Proc. ICML Deep Learn. Workshop*, vol. 2, 2015.
[237] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," 2016, *arXiv:1608.06993*.
[238] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, "Inception-v4, inception-ResNet and the impact of residual connections on learning," 2016, *arXiv:1602.07261*.
[239] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "MobileNets: Efficient convolutional neural networks for mobile vision applications," 2017, *arXiv:1704.04861*.

[240] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16 × 16 words: Transformers for image recognition at scale," 2020, *arXiv:2010.11929*.
[241] J. Zhang, F. Chen, Z. Cui, Y. Guo, and Y. Zhu, "Deep learning architecture for short-term passenger flow forecasting in urban rail transit," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 11, pp. 7004–7014, Nov. 2021.
[242] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller, "Labeled faces in the wild: A database for studying face recognition in unconstrained environments," UMASS Vision, Univ. Massachusetts, Amherst, MA, USA, Tech. Rep. 07-49, Oct. 2007.
[243] N. Kumar, A. C. Berg, P. N. Belhumeur, and S. K. Nayar, "Attribute and simile classifiers for face verification," in *Proc. IEEE 12th Int. Conf. Comput. Vis.*, Sep. 2009, pp. 365–372.
[244] I. Chingovska, A. Anjos, and S. Marcel, "On the effectiveness of local binary patterns in face anti-spoofing," in *BIOSIG - Proc. Int. Conf. Biometrics Special Interest Group (BIOSIG)*, Sep. 2012, pp. 1–7.
[245] D. Yi, Z. Lei, S. Liao, and S. Z. Li, "Learning face representation from scratch," 2014, *arXiv:1411.7923*.
[246] N. Zhang, M. Paluri, Y. Taigman, R. Fergus, and L. Bourdev, "Beyond frontal faces: Improving person recognition using multiple cues," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 4804–4813.
[247] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 3730–3738.
[248] Y. Guo, L. Zhang, Y. Hu, X. He, and J. Gao, "MS-Celeb-1M: A dataset and benchmark for large-scale face recognition," 2016, *arXiv:1607.08221*.
[249] I. Kemelmacher-Shlizerman, S. M. Seitz, D. Miller, and E. Brossard, "The MegaFace benchmark: 1 million faces for recognition at scale," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 4873–4882.
[250] Q. Cao, L. Shen, W. Xie, O. M. Parkhi, and A. Zisserman, "VGGFace2: A dataset for recognising faces across pose and age," in *Proc. 13th IEEE Int. Conf. Autom. Face Gesture Recognit. (FG)*, May 2018, pp. 67–74.
[251] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," 2018, *arXiv:1812.04948*.
[252] C.-H. Lee, Z. Liu, L. Wu, and P. Luo, "MaskGAN: Towards diverse and interactive facial image manipulation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 5548–5557.

QUENTIN LE ROUX received the M.Sc. degree in data science from Universite Cte d'Azur, France, in 2022. He is currently pursuing the joint Ph.D. (Industrial) degree with French National Institute for Research in Digital Science and Technology (INRIA) and the technology company Thales Group under the supervision of Universite de Rennes, France. He joined a Ph.D. program after two consecutive internships with Thales Group in cryptanalysis and adversarial machine learning with the Thales DIS' Hardware Security Laboratory, La Ciotat, France. He has previous experience in risk modeling in France and USA after a MiM (2016) with EDHEC Grande École.

ERIC BOURBAO was born in Dakar, Senegal, in 1964. He received the Ph.D. degree in physics (signal processing) from the University of Toulon, Var, France. After several years as a Research Engineer in the field of underwater acoustics, he joined Bull CP8 to work on the security evaluation of smart cards. He is currently a Senior Expert Engineer in cyber security with the Thales DIS' Hardware Security Laboratory, La Ciotat, France. In charge of the security evaluation of the company's products, his areas of expertise are signal processing, machine learning, and AI applications.

YANNICK TEGLIA received the B.Sc. and M.Sc. degrees in theoretical computer science from Aix-Marseille Universite, the Engineering degree in telecommunication engineering from Polytech Marseille, and the Ph.D. degree in security of embedded systems from the Faculty of Science and Engineering, Sorbonne Universite. He is currently a Principal Engineer in cybersecurity with Thales DIS. He was a Cryptographer and Security Engineer with STMicrolectronics, where he contributed to the architecture, development, and securing of various secure processors and libraries for different product lines. He joined Thales DIS (previously Gemalto), in 2016, working with the Hardware Security Laboratory, La Ciotat, France, where he focuses on hardware security and the security of artificial intelligence. He has coauthored more than 20 research articles and 100 patents. He co-supervised several Ph.D. theses in mathematics, electronics, security, and computer science as an Industrial Advisor.

KASSEM KALLAS (Senior Member, IEEE) received the B.Sc. and M.Sc. degrees in telecom engineering and computer and communication engineering from Lebanese International University, in 2010 and 2012, respectively, the master's degree in wireless systems and related technologies from Politenico di Torino, in 2013, the Executive M.B.A. degree in strategic leadership from the Quantic School of Business and Technology, USA, and the Ph.D. degree in information engineering and sciences in the field of cybersecurity from the University of Siena, in 2017.

He is currently a Senior Scientist and a Junior Professor specializing in machine learning and artificial intelligence, with a particular focus on cybersecurity. He securing the third place in the Springer Best-of-the-Best Ph.D. Thesis Award for the Ph.D. degree. He participated in diverse research projects funded by prestigious institutions, such as DARPA, U.S. Air Force Research Laboratory, Italian Ministry of University and Research, and French National Research Agency. He worked at different national institutes, including the National Institute of Standards and Technology (NIST), USA, focusing on AI for wireless systems, and French National Institute for Research in Digital Science and Technology (INRIA), where he worked on advancing AI security for defense applications. In industrial settings, he was a Research and Development Scientist and an AI Consultant for various Italian companies. Currently, he holds the position of a Senior Scientist with French National Institute of Health and Medical Research, with a dedicated focus on AI security for medical applications. He was a recipient of the Best Paper Award at the Ninth International Conference on Advances in Multimedia (MMEDIA), in 2017, and the Top 3% Paper Award at the 2023 International Conference on Acoustics, Speech, and Signal Processing (ICASSP) in Rhodes Island, Greece. He is the author of the book *Information Fusion in Distributed Sensor Networks with Byzantines* (Springer) series on Signals and Communication Technology and served as an Editor for the book *Adversarial Multimedia Forensics* (Springer) series on Advances in Information Security. He holds a senior membership in IEEE and actively engages in various professional communities, including IEEE Young Professionals, Entrepreneurship Exchange Community, Computer Society Technical Community on Cyber Security, Computer Society Technical Community on Security and Privacy, Signal Processing Society, European Association for Signal Processing (EURASIP), Association for Computing Machinery (ACM), and Asia–Pacific Signal and Information Processing Association (APSIPA).