Article

An Ontological Knowledge Base of Poisoning Attacks on Deep Neural Networks

Majed Altoub [1], Fahad AlQurashi [1], Tan Yigitcanlar [2,3], Juan M. Corchado [4,5,6] and Rashid Mehmood [7,*]

[1] Department of Computer Science, Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah 21589, Saudi Arabia; maltoub0001@stu.kau.edu.sa; fahad@kau.edu.sa
[2] School of Architecture and Built Environment, Queensland University of Technology, 2 George Street, Brisbane, QLD 4000, Australia; tan.yigitcanlar@qut.edu.au
[3] School of Technology, Federal University of Santa Catarina, Campus Universitario, Florianopolis, SC 88040-900, Brazil; tan.yigitcanlar@ufsc.br
[4] BISITE Research Group, University of Salamanca, 37007 Salamanca, Spain; corchado@usal.es
[5] Air Institute, IoT Digital Innovation Hub, 37188 Salamanca, Spain
[6] Department of Electronics, Information and Communication, Faculty of Engineering, Osaka Institute of Technology, Osaka 535-8585, Japan
[7] High Performance Computing Center, King Abdulaziz University, Jeddah 21589, Saudi Arabia
[*] Correspondence: rmehmood@kau.edu.sa

Abstract: Deep neural networks (DNN) have successfully delivered a cutting-edge performance in several fields. With the broader deployment of DNN models on critical applications, the security of DNNs becomes an active and yet nascent area. Attacks against DNNs can have catastrophic results, according to recent studies. Poisoning attacks, including backdoor and Trojan attacks, are one of the growing threats against DNNs. Having a wide-angle view of these evolving threats is essential to better understand the security issues. In this regard, creating a semantic model and a knowledge graph for poisoning attacks can reveal the relationships between attacks across intricate data to enhance the security knowledge landscape. In this paper, we propose a DNN Poisoning Attacks Ontology (DNNPAO) that would enhance knowledge sharing and enable further advancements in the field. To do so, we have performed a systematic review of the relevant literature to identify the current state. We collected 28,469 papers from IEEE, ScienceDirect, Web of Science, and Scopus databases, and from these papers, 712 research papers were screened in a rigorous process, and 55 poisoning attacks in DNNs were identified and classified. We extracted a taxonomy of the poisoning attacks as a scheme to develop DNNPAO. Subsequently, we used DNNPAO as a framework to create a knowledge base. Our findings open new lines of research within the field of AI security.

Keywords: Deep neural networks; Adversarial Attacks; Poisoning; Backdoors; Trojans; Taxonomy; Ontology; Knowledge Base; Explainable AI; Green AI

check for updates

Citation: Lastname, F.; Lastname, F.; Lastname, F. Title. *Preprints* **2022**, *1*, 0. https://doi.org/

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

1. Introduction

DNNs have brought innovative changes and introduced new dimensions in many fields. With the rapid growth of using DNNs for diversified purposes, DNNs have proven to be a very effective technique in producing models that become the core of critical applications. DNN models are usually used in solving routine problems and applied in automating daily labor, computer vision such as object detection [1], pedestrian and face recognition [2], linear algebra [3], mobility [4], Natural Language Processing (NLP) [5,6], solar forecasting [7], medical diagnosis [8,9], smart cities [10], forensic sciences [11], and supporting cyber security applications [12–14].

DNNs rely upon their input data, structure, and parameters and any change might mislead the DNNs. This sensitivity of DNNs makes them brittle against adversarial attacks. Attackers can mislead the DNNs by corruption parameters, maximizing error functions, or manipulating the datasets. Recent research demonstrates that the robustness of deep neural networks is a critical weakness against malicious attacks. Moreover, the robustness of DNNs is vital due to the emerging concepts of responsible and green artificial intelligence

(AI) [15,16] that aim to preserve ethics, fairness, democracy, and the explainability of AI-based decision systems.

The attacks against deep neural networks can be classified according to three main categories: 1) Attacks at the training phase such as data or model poisoning attacks. 2) Attacks at the training and testing phase such as Backdoor and Trojan attacks. 3) Attacks at the testing/inference phase-only which are called Evasion attacks and are specifically known as Adversarial Example attacks.

Many studies have attempted to review the attacks against DNNs. Although there are a growing number of research on adversarial attacks, only a few of them focus on poisoning attacks. Meanwhile, with the rapid use of DNNs, evaluating the robustness of DNNs involves exploring weaknesses in their models. Traditional security management and threat analysis lack methods for intelligent responses to new threats. A semantic knowledge representation of security attacks is a significant way to retrieve data for analysts whether they are human or AI agents. In addition, there is still a lack of semantic knowledge graphs and intelligent reasoning technologies for emerging attacks.

Thus, motivated by this research gap, this paper proposes an ontology of poisoning and backdoor attacks in DNNs. The ontology is extendable. We focus here in this paper on the first two types of attacks and we call both of them poisoning attacks, which means that we focus on poisoning attacks in the training phase and the backdoor, and Trojan attacks that begin in the training phase and can continue to the testing or inference phase. We investigated poisoning and backdoor attacks from papers published between 2013 and the mid of 2021 through a systematic literature review. We collected 28,469 papers from IEEE, ScienceDirect, Web of Science, and Scopus databases, and from these papers, 712 research papers were screened in a rigorous process. After a further screening, 52 papers were selected, which were fully read and analyzed. From these 52 papers, a total of 55 poisoning attacks in DNNs were detected and categorized. We extracted a taxonomy of the poisoning attacks as a scheme to develop DNNPAO. Subsequently, we used DNNPAO as a framework to create a knowledge base.

The rest of this paper is organized as follows. Section 2 describes the research methodology of the paper including the systematic review methodology. Section 3 presents related work. Section 4 presents the poisoning attacks taxonomy extracted from the systematic review. Section 5 provides a review of the selected 52 papers. Section 6 describes the architecture of the DNN Poisoning Attacks Ontology (DNNPAO) and the poisoning attacks knowledge base. Finally, Section 7 concludes the paper and suggests some directions for future research.

2. Research Methodology

This section describes the methodology used in this paper. The work is divided into three phases, each with its own method. The first phase is the systematic literature review which was focused on identifying the security threats against DNNs and the related works in prior studies. The second phase involves developing a taxonomy scheme based on the literature. In the third phase, the DNN poisoning attacks ontology was built on the basis of the taxonomy and then a poisoning attacks knowledge base was created. Figure 1 on Page 3 shows the overall research process that has been carried out in this study and each phase is described in detail.

2.1. Phase1

We performed a systematic literature review (SLR) [17] to understand the existing poisoning attacks, their characteristics, and related works. From among a total of 28469 detected papers, 712 research papers were screened and 52 papers were fully read. Our SLR protocol had three main phases. 1) Research questions 2) Search strategy 3) Study selection. Furthermore, in the SLR protocol, these **keywords** were used: *deep neural networks, deep learning, vulnerabilities, threats, attacks, security, taxonomy, ontology, poisoning, Trojan, Backdoor*.

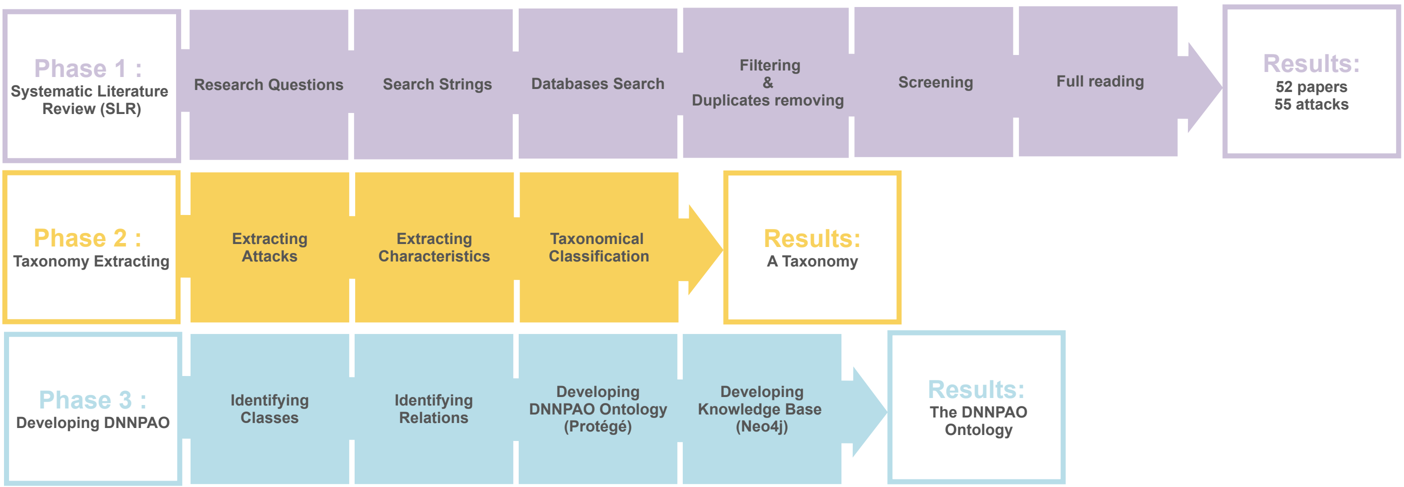


Figure 1. Overall research process in this study.

2.1.1. Research Questions

We initially framed our main research questions (MRQs) to address our general goal. Then, we defined sub-questions (SRQ) on the basis of the main ones. Thus, the following questions were defined:

- **M**RQ1: What are the existing poisoning and backdoor attacks against deep neural networks? .
- **M**RQ2: How can the identified attacks be classified according to their characteristics? (Ontology).
- **S**RQ2.1: What are the main dimensions (classes) of the existing poisoning and backdoor attacks? (Taxonomy).
- **S**RQ2.2: What are the characteristics (subclasses) of these dimensions?(Taxonomy).

2.1.2. Search Strategy

On the basis of the research questions we identified the main keywords. Then, we formulated the search string before starting the search in databases by applying Boolean operator OR for synonyms and Boolean AND operator for Linking keywords. Table 1 on Page 3 shows the formulated search strings.

Table 1. Search Strings.

Number	Search String
S1	(("deep neural networks" OR "deep learning" OR DL) AND (taxonomy OR survey OR review) AND (attacks OR security))
S2	(("deep neural networks" OR "deep learning" OR DL) AND (vulnerabilities OR vulnerability OR weaknesses OR threats))
S3	(("deep neural networks" OR "deep learning" OR DL) AND attack)
S4	(("deep neural networks" OR "deep learning" OR DL) AND ("black box" OR "grey box" OR "white box"))
S5	(("deep neural networks" OR "deep learning" OR DL) AND adversarial)
S6	(("deep neural networks" OR "deep learning" OR DL) AND poisoning)
S7	(("deep neural networks" OR "deep learning" OR DL) AND Trojan)
S8	(("deep neural networks" OR "deep learning" OR DL) AND Backdoor)
S9	(("deep neural networks" OR "deep learning" OR DL) AND (ontology) AND (attacks OR security OR poisoning OR Trojan OR Backdoor))

Well-known academic databases and libraries have been used to collect the papers as listed in Table 2 on Page 4. To select the primary studies, inclusion and exclusion criteria have been used. Furthermore, automation tools have been used (Rayyan.io [18] and ASReview [19]) to review, filter, and screen relevant and irrelevant papers.

Table 2. Digital Libraries and Databases.

Databases	S1	S2	S3	S4	S5	S6	S7	S8	S9	Total
IEEE	271	1117	1860	410	2867	54	48	41	17	5685
Science Direct	702	795	457	256	267	0	31	27	2	2537
Web of Science	381	1109	1807	432	2524	48	38	37	52	6428
Scopus	556	2389	3861	1113	5514	111	76	94	105	13819
			Total							28469

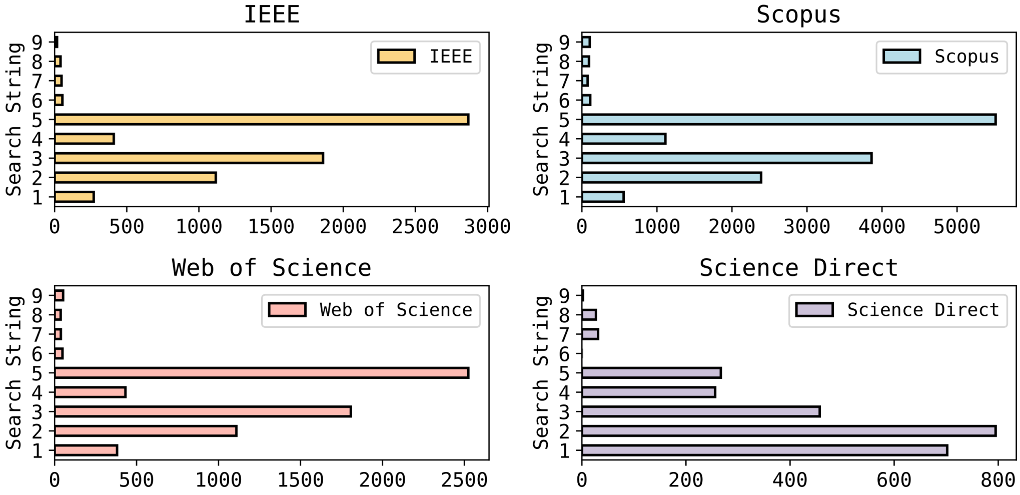


Figure 2. Search string results on databases

2.1.3. Study Selection

The search strings described in Table 1 on Page 3 have been applied to the databases listed in Table 2 on Page 4. Due to the fact that the first report of an attack against a deep neural network was in 2013, we chose to limit the search to the period between 2013 to 2021. Furthermore, the search did not focus on papers on adversarial example attacks or that applied deep neural networks as approaches to other security issues.

The established inclusion and exclusion criteria have helped to extract the studies that proposed poisoning and backdoor attacks only. The inclusion criteria are as follows:

- Papers that have been published in journals or conference proceedings.
- Papers that were published in the period between 2013 and 2021.
- Papers that address the research questions.
- Papers that involve the research keywords.
- Papers that have proposed a poisoning, backdoor, or Trojan attack.

The exclusion criteria are as follows:

- Papers that used deep neural networks as approaches but did not focus on them.
- Papers that were not written in English.
- Papers that do not address the research questions.
- Papers that are not related to the research questions such as non-poisoning attacks in DNNs.
- Papers that have not proposed a poisoning, backdoor, or Trojan attack.

The initial search yielded a total of 28469 papers, 5685 from IEEE, 2537 from Science Direct, 6428 from Web of Science, and 13819 from Scopus. Then, after removing the duplicated papers using (Refworks, Rayyan [18], Zotero, Mendeley) there were a total of 12460 papers. Following that, we proceeded to the filtering, skimming, and screening of the titles and abstracts utilizing (Rayyan.io [18] and ASReview [19]). The inclusion and exclusion criteria were applied throughout the entire process. From among a total of 712 screened papers, 52 papers were selected, and 55 poisoning and backdoor attacks were extracted after thorough reading as shown in Figure 3 on Page 6.

In summary, the method followed in this study consisted in the SLR steps listed below:

1. Defining pertinent research questions.
2. Providing search strings derived from keywords and research questions.
3. Defining the databases.
4. Filtering irrelevant papers.
5. Skimming titles and abstracts to exclude unrelated articles.
6. Reviewing the remaining papers in light of the research questions.

2.2. *Phase2*

In this phase we extracted a poisoning attacks taxonomy. To enhance the comprehensive view of the threat landscape, we have used abstraction layers to classify the attacks according to key characteristics. The taxonomy included 6 main dimensions called classes and 36 sub-classes which can be used to describe the individual attacks as shown in Figure 6 on Page 9. These classes have been extracted from the full reading and review of the literature.

2.3. *Phase3*

In this phase, our ontology (DNNPAO) was built. We developed the ontology on the basis of the proposed taxonomy, however, to create our ontological knowledge base the attacks were added as individuals and the DNNPAO was used as a framework for the creation of the knowledge base. To develop the ontology knowledge base we used OWL semantic language with the Protégé tool [20] and webprotege.stanford.edu [20]. To build the knowledge base we used the Neo4j graph database [21] with the Neosemantics plugin [22] to import the ontology in Neo4j.

doi:10.20944/preprints202208.0197.v1

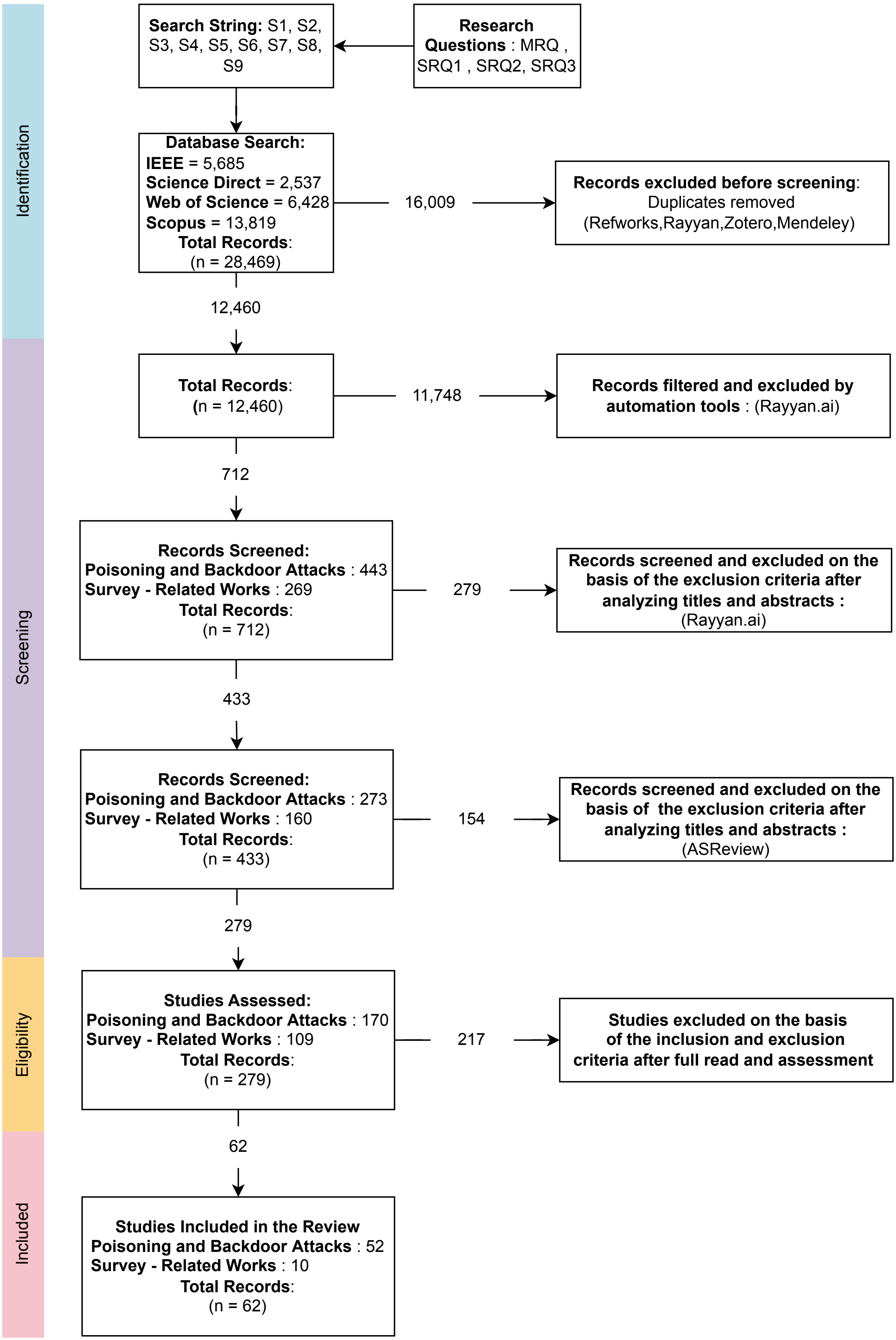

Identification
Screening
Eligibility
Included
Search String: S1, S2, S3, S4, S5, S6, S7, S8, S9
Research Questions : MRQ , SRQ1 , SRQ2, SRQ3
Database Search:
IEEE = 5,685
Science Direct = 2,537
Web of Science = 6,428
Scopus = 13,819
Total Records:
(n = 28,469)
16,009
Records excluded before screening: Duplicates removed (Refworks,Rayyan,Zotero,Mendeley)
12,460
Total Records:
(n = 12,460)
11,748
Records filtered and excluded by automation tools : (Rayyan.ai)
712
Records Screened:
Poisoning and Backdoor Attacks : 443
Survey - Related Works : 269
Total Records:
(n = 712)
279
Records screened and excluded on the basis of the exclusion criteria after analyzing titles and abstracts : (Rayyan.ai)
433
Records Screened:
Poisoning and Backdoor Attacks : 273
Survey - Related Works : 160
Total Records:
(n = 433)
154
Records screened and excluded on the basis of the exclusion criteria after analyzing titles and abstracts : (ASReview)
279
Studies Assessed:
Poisoning and Backdoor Attacks : 170
Survey - Related Works : 109
Total Records:
(n = 279)
217
Studies excluded on the basis of the inclusion and exclusion criteria after full read and assessment
62
Studies Included in the Review
Poisoning and Backdoor Attacks : 52
Survey - Related Works : 10
Total Records:
(n = 62)

Figure 3. Research process for paper selection

3. Related Work

This section discusses related works in the context of surveying and classifying poisoning and backdoor attacks against deep neural networks that are retrieved from our dataset and selected by the systematic review methodology described in Section 2. Most studies that are surveyed attacks in deep neural networks have focused on evasion attacks (adversarial examples), and only a few were considered poisoning attacks. Although some research has been carried out on surveying attacks in deep neural networks, only three studies have taxonomic poisoning attacks, and only two have taxonomic backdoor attacks in deep neural networks.

Most related studies [23,24] have provided taxonomies and considered poisoning attacks in deep neural networks, as shown in Table 3 on Page 8. Pitropakis et al. [23] proposed a taxonomy and a survey of attacks against machine learning systems. The authors attempted to unify the field by classifying the different papers that propose attacks against machine learning. The taxonomy also helped in the identification of pending problems that may lead to new fields of study. Their taxonomy of adversarial attacks on machine learning is divided into two separate phases: preparation and manifestation. They provided attack models with two attack types: poisoning and evasion. This survey did not investigate defense mechanisms. Dang et al. [24] briefly reviewed data poisoning attacks (backdoor, and Trojan attacks) and some of the defense methods. They focused on backdoor attacks, and they summarized specific defense proposals (Data defense, Model defense, and Training defense). While poisoning attacks can violate the integrity or availability of a neural network, they are very hard to defend and defenses are still in infancy for discoveries.

In a similar context, studies [25–27] have provided taxonomies but focused on a specific application. Jere et al. [25] provided a taxonomy of attacks against federated learning systems, concerning both data privacy and model performance with a comprehensive review. They started with a federated learning overview before developing a framework for robust threat modeling in order to survey model performance and data privacy attacks. Finally, they presented some of the existing defense strategies. Isakov et al. [26] proposed an attack taxonomy against deep neural networks in edge devices. In their work, they cover the attacks and their countermeasures landscape. Chen et al. [27] conducted an assessment of existing backdoor attacks and their mitigation in outsourced cloud environments. The authors classified attack and defense approaches into different groups on the basis of the adversary's resources and whether the detection occurs during run-time or not. They also presented a comparison of these approaches and used experiments to evaluate a portion of the attack schemes.

Several studies such as [28,29] have only focused on Trojaning. Liu et al. [28] summarized both neural Trojan attack and defense mechanisms. Outsourced training procedure is considered to be a reason for attacks, this paper emphasizes that the machine learning supply chain, such as the MLaaS provider, might be a serious risk for any MLaaS consumer. The authors offered three types of neural Trojan attacks: training data poisoning, training algorithm-based, and binary-level attacks. These main categories can be used to classify any other neural Trojan attack. The defense techniques were classified into four classes : neural network verification, Trojan trigger detection, compromised neural network restoration, and Trojan bypass techniques. The authors also showed that Trojans might be used for protection as well. According to the authors, the majority of the research in this field has been done in the previous three years, and the combat between attackers and defenders in the context of the neural Trojan is expected to continue. Xu et al. [29] provided a comprehensive survey of the attacks against DNN on the hardware surfaces. The authors focused on hardware Trojan insertion, fault-injection, and side-channel analysis. Furthermore, they categorized the attack methods and defenses to provide a comprehensive view of hardware-related security issues in neural networks.

Other papers [30–32] are more general survey papers. Xue et al. [30] provided a systematic analysis of the security issues of machine learning. Systematically, they reviewed attacks and defenses in both the training and testing phases. Machine learning algorithms were classified into two types: neural network (NN) algorithms and non-NN algorithms. In addition, threats and attack models were presented. The authors identified three main machine learning vulnerabilities. The first is the outsourcing of the training process. The second is the use of pre-trained models from third parties. The third one are the ineffective data validations on the network. The paper also included defense strategies and responses for the most common forms of attacks. Their paper contributed to a comprehensive understanding of security and robustness of machine learning systems.

He et al. [31] provided a survey of DNN attacks and defense mechanisms. The authors concentrated on four types of attacks: adversarial, model extraction, model inversion, and poisoning attack. The paper also proposed 18 results in connection with these attacks. Furthermore, some potential security vulnerabilities and mitigation strategies were discussed for future research in this emerging field. Miller et al. [32] provided a comprehensive review of adversarial learning attacks. The authors also discussed recent research on test-time evasion, data poisoning, backdoor, and reverse engineering attacks, as well as the defense methods.

For these emerging attacks, there is still a lack of semantic knowledge graphs and intelligent reasoning systems. Hence, the focus of our research has been on existing poisoning attacks in deep neural networks. This study intends to remedy this issue by providing a systematic review and attempting to develop an ontology of these attacks. Summaries of the related survey papers are provided in Table 3 on Page 8. In addition, Figure 4 on Page 8 and Figure 5 on Page 9 show some statistical information on the related survey papers.

Table 3. Summaries of the related publications of survey papers in the dataset that focus on or include poisoning or backdoor attacks in deep learning.

Survey Paper	DL	ML	Poisoning	Backdoors	Trojans	Published	Taxonomy
Pitropakis et al. [23]	✓	✓	✓			2019	✓
Xue et al. [30]	✓	✓	✓			2020	
Liu et al. [28]	✓				✓	2020	
He et al. [31]	✓		✓			2020	
Jere et al. [25]	✓					2021	✓
Xu et al. [29]	✓				✓	2021	
Dang et al. [24]	✓		✓	✓		2020	✓
Isakov et al. [26]	✓		✓			2019	✓
Chen et al. [33]	✓			✓		2020	✓
Miller et al. [32]	✓					2020	

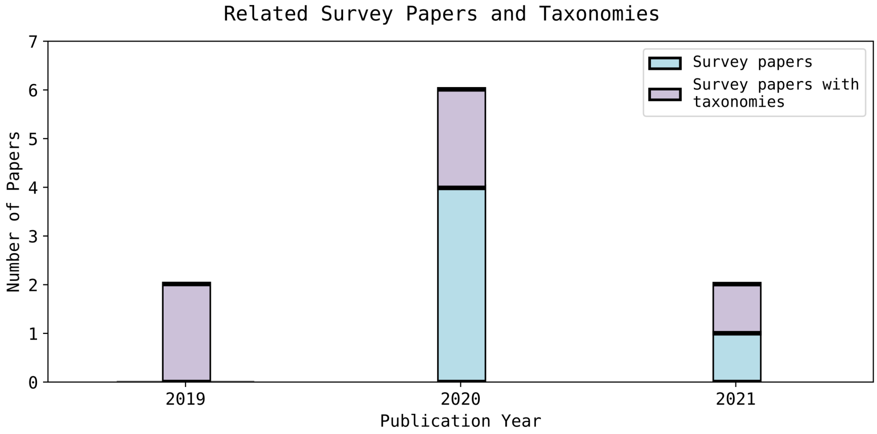


Figure 4. There have been few survey papers that have built a taxonomy of poisoning and backdoors attacks in deep neural networks.

doi:10.20944/preprints202208.0197.v1

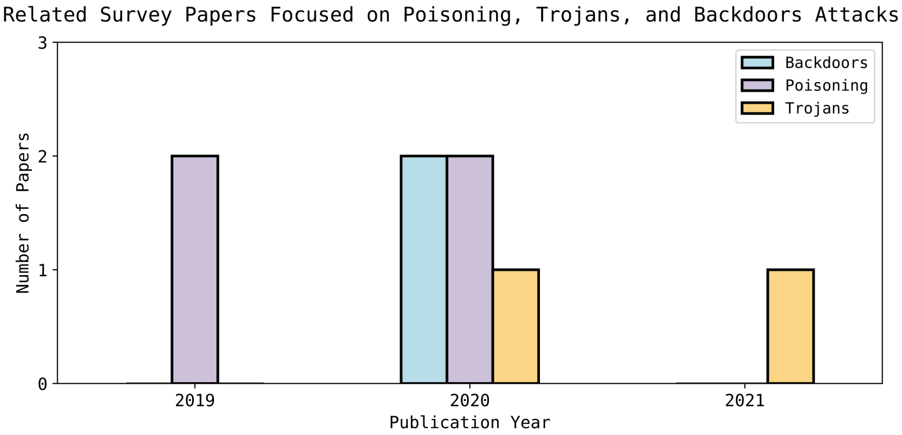


Figure 5. Shows the number of survey papers on Trojans, Poisoning and Backdoors attacks and the year of publication.

4. Poisoning Attacks Taxonomy

Taxonomies can help to explain similarities and differences among described objects. From the systematic literature review, we have taxonomized the main characteristics of poisoning attacks to build a high-level structure and domains. As shown in Figure 6 on Page 9 the extracted taxonomy has 6 main domains and 36 characteristics. These are the core concepts that have later on been used in our ontology. Thus, we call these domains main classes and the corresponding characteristics sub-classes. We have only chosen these to keep the taxonomy scheme and ontology as simple as possible, however, it can be extended to more theoretical classes. The main classes in the poisoning attacks taxonomy are shown in Figure 6 on Page 9. Security Violations, Attackers knowledge, Access Impacts, Models Architectures, Triggers, and Attack Vectors. Each one of these classes has sub-classes that are described in the sections below.

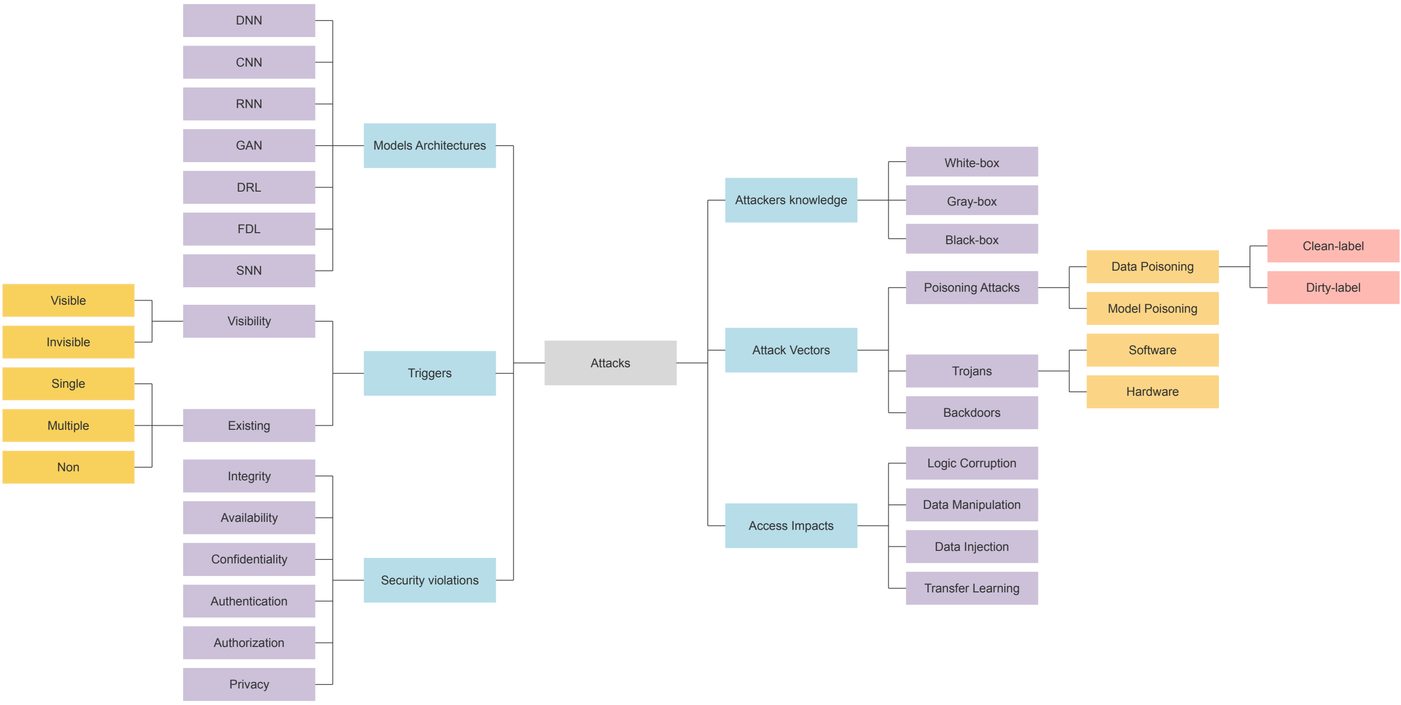


Figure 6. Extracted Poisoning Attacks Taxonomy.

4.1. Security Violations

In cybersecurity, confidentiality, integrity, and availability is a broadly used information security model, known as the CIA triad, to guide information security policies aimed toward securing organizations' data. Authentication, authorization, and privacy are other security principles that are related to the CIA triad. For instance, violating confidentiality for sensitive information can result in violating privacy. From a security perspective **confidentiality** means that only authorized users can access the data. Confidentiality in deep neural networks means that only authorized users have access to the models or training datasets. Thus, attacks on confidentiality endeavor to expose the models' structure and their parameters or the datasets that are used to train the models.

Integrity refers to the incorrect results that occur due to manipulation attacks or system failures. In deep neural networks, integrity refers to the misclassification outputs or behaviors of the deep neural network models due to model failures or attacks Li et al. [34]. Usually, false positive results in DNNs are caused by an integrity violation. **Availability** means that users can access both data and services at any time. In deep neural networks, the availability attacks mean that deep neural networks are not available. Where the attackers endeavor to prevent authorized users from accessing meaningful model outputs. Usually, false negative results in DNNs are caused by an availability violation.

Authentication and **authorization** characteristics are a process of verifying someone and what he/she can access. Deep neural network models are usually used in authentication systems such as facial and biometric recognition where attacks against these DNN models can violate authentication and authorization systems. **Privacy** violation in DNNs is a security threat where an unauthorized attacker can gain sensitive information from the training data. A summary of security violations for each attack is provided in Table 4 on Page 11.

4.2. Attacker's knowledge

Attacker's knowledge characteristics define the knowledge of the targeted model that an adversary has. The attacker might have zero knowledge about the model and the trained parameters that are used in the network which is referred to as a **Black-box** attack. In a **White-box** attack, the adversary knows the target model and has complete knowledge of the used algorithms and the parameters. If the attacker has only some of this knowledge that he/she can utilize, it is referred to as a **Gray-box** attack. Although white-box attacks have higher success rates than black-box attacks, the scenario of white-box attacks may be unrealistic, and the black-box attacks might be more applicable in real-world applications [35]. A summary of attackers knowledge for each attack is provided in Table 5 on Page 12.

4.3. Triggers

The triggers a are hidden malicious functionality known only by the attacker that can be inserted at the training phase and can remain cleverly concealed until the chosen time to activate trojans or backdoors in later phases. In some attacks, triggers may not exist, **triggerless**, such as in triggerless poisoning attacks or in triggerless backdoor attacks. In addition, one attack may only use a **single** trigger but it can also use **multiple** triggers for the same attack as a pattern to activate the attack and make it more difficult to detect. Triggers can also be **visible** to humans, such as changes in the shapes or sizes of images or they might be **invisible** for humans to detect the changes between clear data and poisoned data. Although triggers might be visible to humans they could be easily deployable and difficult to detect with big datasets or in the real world such as using tattoos or glasses. A summary of triggers for each attack is provided in Table 5 on Page 12.

Table 4. A summary of security violations for each attack

Attacks	Papers	Security violations					
		Integrity	Availability	Confidentiality	Authentication	Authorization	Privacy
1	Li et al. [34]	x					
2	Li et al. [34]	x					
3	Lee et al. [36]	x		x			
4	Dumford et al. [37]	x		x			
5	Bhagoji et al. [38]	x					x
6	Zhou et al. [39]	x					x
7	Davaslioglu et al. [40]	x			x		
8	Zhong et al. [41]	x			x	x	
9	Huai et al. [42]	x			x		x
10	Xue et al. [30]	x	x				x
11	Xue et al. [30]	x	x				x
12	Liu et al. [43]	x					
13	Hu et al. [44]	x	x				
14	Lin et al. [45]	x			x	x	
15	Dai et al. [46]	x					
16	Zhao et al. [47]	x					
17	Liu et al. [48]	x					
18	Chen et al. [49]	x					
19	Tan et al. [50]	x				x	
20	Tang et al. [51]	x					
21	Wu et al. [52]	x	x				
22	Barni et al. [53]	x					
23	Xiong et al. [54]	x					
24	Kwon et al. [55]	x					
25	Chen et al. [56]	x			x		
26	Chen et al. [57]	x			x		
27	He et al. [31]	x			x	x	
28	Xue et al. [58]	x			x	x	
29	Yao et al. [59]	x			x		
30	Quiring et al. [60]	x					
31	Bhalerao et al. [61]	x			x	x	
32	Costales et al. [62]	x	x				
33	Kwon et al. [63]	x					
34	Liu et al. [64]	x	x				
35	Rakin et al. [65]		x				
36	Liu et al. [66]	x		x			
37	Zhou et al. [67]	x					
38	Zhu et al. [68]	x					
39	Gu et al. [69]	x					
40	Guo et al. [70]		x	x	x	x	
41	Clements et al. [71]	x	x				
42	Munoz et al. [72]	x	x				
43	Li et al. [73]	x					
44	Li et al. [74]	x					
45	Xu et al. [75]	x	x				
46	Venceslai et al. [76]	x					
47	Kwon et al. [77]	x					
48	Cole et al. [78]	x	x		x	x	
49	Zeng et al. [79]	x	x				
50	Pan [80]	x	x				
51	Garofalo et al. [81]	x	x		x	x	
52	Tolpegin et al. [82]		x				
53	Li et al. [83]		x				
54	Zhang et al. [84]	x	x				
55	Lovisotto et al. [85]		x	x	x	x	

Table 5. A summary of attackers knowledge and triggers for each attack

Attacks	Papers	Attackers knowledge			Triggers				
					Visibility		Existing		
		White-box	Gray-box	Black-box	Visible	Invisible	Single	Multiple	Non
1	Li et al. [34]	x				x	x		
2	Li et al. [34]		x						x
3	Lee et al. [36]	x							x
4	Dumford et al. [37]	x			x		x		
5	Bhagoji et al. [38]	x							x
6	Zhou et al. [39]			x					x
7	Davaslioglu et al. [40]		x				x		
8	Zhong et al. [41]	x	x	x		x	x		
9	Huai et al. [42]			x					x
10	Xue et al. [30]	x	x	x	x			x	
11	Xue et al. [30]	x	x	x	x		x		
12	Liu et al. [43]		x						x
13	Hu et al. [44]		x			x	x		
14	Lin et al. [45]	x			x			x	
15	Dai et al. [46]			x	x		x		
16	Zhao et al. [47]		x		x		x		
17	Liu et al. [48]	x			x		x		
18	Chen et al. [49]	x					x		
19	Tan et al. [50]	x			x		x		
20	Tang et al. [51]		x					x	
21	Wu et al. [52]	x			x		x		
22	Barni et al. [53]			x	x		x		
23	Xiong et al. [54]	x			x		x		
24	Kwon et al. [55]	x			x			x	
25	Chen et al. [56]			x		x			x
26	Chen et al. [57]	x				x	x		
27	He et al. [31]	x		x		x	x		
28	Xue et al. [58]	x		x		x	x		
29	Yao et al. [59]		x		x		x		
30	Quiring et al. [60]	x				x	x		
31	Bhalerao et al. [61]			x			x		
32	Costales et al. [62]	x					x		
33	Kwon et al. [63]	x			x		x		
34	Liu et al. [64]			x		x	x		
35	Rakin et al. [65]	x					x		
36	Liu et al. [66]	x				x	x		
37	Zhou et al. [67]	x							x
38	Zhu et al. [68]		x	x					x
39	Gu et al. [69]	x			x		x		
40	Guo et al. [70]	x				x	x		
41	Clements et al. [71]		x		x		x	x	
42	Munoz et al. [72]	x	x						x
43	Li et al. [73]	x							x
44	Li et al. [74]	x			x		x		
45	Xu et al. [75]	x							x
46	Venceslai et al. [76]		x		x		x		
47	Kwon et al. [77]	x							x
48	Cole et al. [78]			x					x
49	Zeng et al. [79]		x						x
50	Pan [80]			x			x		
51	Garofalo et al. [81]	x		x					x
52	Tolpegin et al. [82]			x					x
53	Li et al. [83]			x					x
54	Zhang et al. [84]	x							x
55	Lovisotto et al. [85]	x		x	x		x		

4.4. Access Impacts

The concern of the adversaries of DNNs is the ability of the attackers to gain access to the network and what impact attackers can have on the network as a result of this access. We consider four different possibilities of impact as a result of access to the DNNs. The first scenario is the **logic corruption** where the adversaries can evade and corrupt the learning algorithms which can usually be seen on the neural trojan attacks. The second scenario is the **data manipulation** where the attacker can modify the data such as labels and learning parameters. Another scenario is **data injection** where the attackers can inject new parts such as triggered images. These can be seen in many attack scenarios such as data positioning and backdoors. The fourth scenario of access impacts is **transfer learning** which exists when using pre-trained models or teacher and student models. A summary of access impacts for each attack is provided in Table 6 on Page 14.

4.5. Deep Neural Networks Models Architectures

In machine learning, when a neural network only has an input and an output layer and may have one hidden layer, we call this neural network a **Shallow Neural Network** [86]. If the neural networks have multiple hidden layers with more complex ways of connecting these layers and neurons, we call these **Deep Neural Networks (DNNs)**. Thus, Deep neural networks are based on layers that have a collection of perceptrons called neurons to determine the outputs. These are some of the common neural network architectures: CNN, RNN, GAN, DRL, FDL, SNN.

Convolutional Neural Network (CNN) is a deep neural network that has at least one convolutional layer that convolves with a multiplication or other dot product and can extract the features from unstructured data with no requirement for human interaction [87]. A Feed-backward architecture that uses back-propagation for backward connections of the neurons is called **Recurrent Neural Network (RNN)** [88]. The outputs on the network are dependent on the previous calculation in a sequence. Recurrent neural networks can automatically manage the input data and may not provide the same output for the same input each time.

Goodfellow et al. [89] introduced the **Generative Adversarial network (GAN)** for estimating generative models via an adversarial process. The GAN corresponds to a two-player game between Discriminator (output) and Generator (input). Its role is to produce samples identical to the training set of the discriminator. GAN has been used as a semi-supervised or unsupervised learning and applied in many domains such as motion and game development. Radford et al. [90] improved the version of GAN with a **Deep Convolutional GAN (DCGAN)** which makes the network more stable to train and overcomes some of GANs' constraints.

Reinforcement Learning (RL) is a type of machine learning that uses an agent in an environment to learn behaviors through trial-and-error [91]. **Deep Reinforcement Learning (DRL)** uses DNNs to obtain an approximation of the reward function which can take large data as inputs compared with traditional RL environments and can solve complex reinforcement learning problems. Theoretically, a **Spiking Neural Network (SNN)** is a bio-inspired information processing network, where the communication signals between neurons are sparse in time and asynchronous binary and known as spikes [92]. In principle, deep spiking neural networks can reduce redundant information by exploiting event-based and data-driven updates which preserve the potential for enhancing the latency and energy performance of DNNs [92]. Deep SNNs are suitable architectures for developing efficient brain-like applications such as pattern recognition.

In the distributed deep neural networks paradigm, the **Deep Federated Learning (DFL)** approach has been developed to address privacy-preserving of deep networks among participating devices that have their own copies of pre-trained models and on-device data [93]. In fact, the level of privacy in a certain model is decided by the updating of the model's minimally essential information [93]. A summary of DNNs model architectures for each attack is provided in Table 7 on Page 15.

Table 6. A summary of access impacts for each attack

Attacks	Papers	Access Impacts			
		Logic Corruption	Data Manipulation	Data Injection	Transfer Learning
1	Li et al. [34]			x	
2	Li et al. [34]			x	
3	Lee et al. [36]		x		
4	Dumford et al. [37]		x		
5	Bhagoji et al. [38]				x
6	Zhou et al. [39]				x
7	Davaslioglu et al. [40]			x	
8	Zhong et al. [41]				
9	Huai et al. [42]		x		
10	Xue et al. [30]			x	
11	Xue et al. [30]			x	
12	Liu et al. [43]		x		
13	Hu et al. [44]	x			
14	Lin et al. [45]			x	
15	Dai et al. [46]			x	
16	Zhao et al. [47]			x	
17	Liu et al. [48]			x	
18	Chen et al. [49]		x		
19	Tan et al. [50]			x	
20	Tang et al. [51]			x	
21	Wu et al. [52]			x	
22	Barni et al. [53]			x	
23	Xiong et al. [54]		x		
24	Kwon et al. [55]			x	
25	Chen et al. [56]			x	
26	Chen et al. [57]				x
27	He et al. [31]			x	
28	Xue et al. [58]			x	
29	Yao et al. [59]				x
30	Quiring et al. [60]		x		
31	Bhalerao et al. [61]			x	
32	Costales et al. [62]	x			
33	Kwon et al. [63]			x	
34	Liu et al. [64]	x			
35	Rakin et al. [65]	x			
36	Liu et al. [66]	x			
37	Zhou et al. [67]			x	
38	Zhu et al. [68]				x
39	Gu et al. [69]		x		x
40	Guo et al. [70]			x	
41	Clements et al. [71]	x			
42	Munoz et al. [72]		x		
43	Li et al. [73]				x
44	Li et al. [74]	x		x	
45	Xu et al. [75]			x	
46	Venceslai et al. [76]	x			
47	Kwon et al. [77]		x		
48	Cole et al. [78]		x		
49	Zeng et al. [79]			x	
50	Pan [80]	x			
51	Garofalo et al. [81]			x	
52	Tolpegin et al. [82]		x		
53	Li et al. [83]			x	
54	Zhang et al. [84]		x		
55	Lovisotto et al. [85]			x	

Table 7. A summary of model architectures for each attack

Attacks	Papers	Models Architectures						
		DNN	CNN	RNN	GAN	DRL	FDL	SNN
1	Li et al. [34]	x						
2	Li et al. [34]	x						
3	Lee et al. [36]		x					
4	Dumford et al. [37]		x					
5	Bhagoji et al. [38]						x	
6	Zhou et al. [39]						x	
7	Davaslioglu et al. [40]		x					
8	Zhong et al. [41]		x					
9	Huai et al. [42]					x		
10	Xue et al. [30]	x						
11	Xue et al. [30]	x						
12	Liu et al. [43]		x					
13	Hu et al. [44]		x					
14	Lin et al. [45]	x						
15	Dai et al. [46]			x				
16	Zhao et al. [47]							
17	Liu et al. [48]	x						
18	Chen et al. [49]	x						
19	Tan et al. [50]	x						
20	Tang et al. [51]	x						
21	Wu et al. [52]		x					
22	Barni et al. [53]		x					
23	Xiong et al. [54]	x						
24	Kwon et al. [55]	x						
25	Chen et al. [56]	x						
26	Chen et al. [57]	x						
27	He et al. [31]	x						
28	Xue et al. [58]	x						
29	Yao et al. [59]	x						
30	Quiring et al. [60]	x						
31	Bhalerao et al. [61]		x					
32	Costales et al. [62]	x						
33	Kwon et al. [63]	x						
34	Liu et al. [64]	x						
35	Rakin et al. [65]	x						
36	Liu et al. [66]	x						
37	Zhou et al. [67]		x					
38	Zhu et al. [68]		x					
39	Gu et al. [69]		x					
40	Guo et al. [70]	x						
41	Clements et al. [71]	x						
42	Munoz et al. [72]	x						
43	Li et al. [73]			x				
44	Li et al. [74]		x					
45	Xu et al. [75]			x				
46	Venceslai et al. [76]							x
47	Kwon et al. [77]		x					
48	Cole et al. [78]		x					
49	Zeng et al. [79]		x					
50	Pan [80]		x					
51	Garofalo et al. [81]	x						
52	Tolpegin et al. [82]						x	
53	Li et al. [83]	x						
54	Zhang et al. [84]						x	
55	Lovisotto et al. [85]	x						

Table 8. A summary of attack vectors for each attack

Attacks	Papers	Attack Vectors					
		Poisoning Attacks			Trojans		Backdoors
		Data Poisoning		Model Poisoning	Software	Hardware	
		Clean-label	Dirty-label				
1	Li et al. [34]						x
2	Li et al. [34]						x
3	Lee et al. [36]			x			
4	Dumford et al. [37]						x
5	Bhagoji et al. [38]			x			
6	Zhou et al. [39]			x			
7	Davaslioglu et al. [40]				x		
8	Zhong et al. [41]						x
9	Huai et al. [42]			x			
10	Xue et al. [30]						x
11	Xue et al. [30]						x
12	Liu et al. [43]		x				
13	Hu et al. [44]					x	
14	Lin et al. [45]						x
15	Dai et al. [46]						x
16	Zhao et al. [47]	x					x
17	Liu et al. [48]						x
18	Chen et al. [49]				x		
19	Tan et al. [50]						x
20	Tang et al. [51]				x		
21	Wu et al. [52]						x
22	Barni et al. [53]						x
23	Xiong et al. [54]						x
24	Kwon et al. [55]						x
25	Chen et al. [56]		x				x
26	Chen et al. [57]		x				x
27	He et al. [31]						x
28	Xue et al. [58]						x
29	Yao et al. [59]						x
30	Quiring et al. [60]	x					x
31	Bhalerao et al. [61]						x
32	Costales et al. [62]					x	
33	Kwon et al. [63]						x
34	Liu et al. [64]					x	
35	Rakin et al. [65]					x	
36	Liu et al. [66]				x		
37	Zhou et al. [67]	x					
38	Zhu et al. [68]	x					
39	Gu et al. [69]						x
40	Guo et al. [70]						x
41	Clements et al. [71]					x	
42	Munoz et al. [72]		x				
43	Li et al. [73]		x				
44	Li et al. [74]				x	x	
45	Xu et al. [75]		x				
46	Venceslai et al. [76]				x		x
47	Kwon et al. [77]		x				
48	Cole et al. [78]		x				
49	Zeng et al. [79]		x				
50	Pan [80]				x		
51	Garofalo et al. [81]		x				
52	Tolpegin et al. [82]		x				
53	Li et al. [83]	x					
54	Zhang et al. [84]		x				
55	Lovisotto et al. [85]						x

4.6. Attack Vectors

Attack vectors are the mechanisms that hackers can use to exploit system vulnerabilities. As the scope of this paper, poisoning attacks can be generally categorized into three main attacks: triggerless poisoning, backdoors, and Trojans attacks. **Triggerless poisoning** occurs at the training phase and does not require modification at the test phase. **Data poisoning** attacks occur when adversaries insert poisoned samples into the training datasets. **Model poisoning** occurs when adversaries exploit the model knowledge during the training time. Data and model poisoning attacks are both types of triggerless poisoning attacks, usually known as poisoning attacks. **Backdoors**, on the other hand, use poisoning attacks to insert triggers that can activate the backdoors at testing (inference) phase. The early backdoor attack cases changed the label of the poisoned data known as **Dirty-label**, however, **Clean-label** attacks now exist which do not change the labels of the poisoned data. In DNNs, **Trojans** are similar to backdoors and can cause misclassification or bad behavior of the targeted models without raising suspicions. Trojan embedding methods can be at the **hardware** level such as accessing to merely the memory bus data or at the **software** level, for example utilizing poisoning attacks to inset triggers. Data poisoning and model poisoning attacks may affect the accuracy of the DNNs, but backdoor and Trojan attacks remain silent until the activation of the triggers and may never affect the accuracy which makes them difficult to be discovered even after the attack occurs. A summary of attack vectors for each attack is provided in Table 8 on Page 16.

5. A Review of the Selected Papers

This section provides a summary of the 52 poisoning attack papers chosen using the systematic review process outlined in Section 2. This survey's structure has been based on the key attack vectors listed above and classified as poisoning, backdoor, and Trojan attacks.

5.1. Poisoning attacks

Many developers employ third-party tools and datasets due to the amount of large data and the compute time required to train deep neural networks. Attackers can poison datasets with malicious data in order to train models based on the attacker's desired outcomes. These poisoning attacks can be either targeted or untargeted. The models themselves are another attack surface that could be poisoned. Notably, many libraries and models built by third parties may contain Backdoors or Trojans.

Early papers [72,81,83,94] were published in 2017-2018. Muñoz-González et al. [72] proposed a multiclass poisoning attack that can target a range of DNN algorithms. The attack scenarios can violate the availability and integrity of such a system by exploiting the back-gradient optimization. The first scenario is a generic error poisoning attack to cause a denial of service. The second scenario is a specific error poisoning attack that targeted a specific multiclass based on the desired misclassifications. The result shows that the attack could be a significant threat to deep neural networks. The authors showed that a small fraction of poisoning in the training data can compromise learning algorithms. Ji et al. [94] focused on model-reuse attacks that can be used in deep learning. They provided empirical and analytical justification and concluded that this is a fundamental issue to many deep learning systems. Garofalo et al. [81] were the first to implement a poisoning attack against a face authentication system by injecting a poisonous image into the re-training data. They showed that the proposed poisoning attack could violate the integrity and availability of the face authentication system by increasing the number of false positives and false negatives. The successful result of the authentication error poses a threat against modern face authentication systems. Li et al. [83] proposed a poisoning attack against targeted wireless intrusion detection systems (IDSs). The presented strategy crafts high-quality adversarial samples using the generated substitute models from a proposed stealing model attack. The results show that by using DNN algorithms, the attack's strategy can effectively impair the performance of IDSs.

In the year 2019, [38,68,77,79,84] were published as poisoning attacks papers in the systematic literature review. Zhu et al. [68] proposed a clean-label transferable poisoning attack where the poison images are designed to be convex around the target targeted image in feature space. The result shows that when poisoning only 1% of the training set, the success rates of the attack can reach 50% even without accessing the targeted network's outputs and architecture. Bhagoji et al. [38] demonstrated the possibility of model poisoning attacks on a federated learning global model, resulting in high model error rates. To overcome the effects of other agents, the attack targets the model by boosting the malicious agent's update. They improve attack stealth by employing an alternating minimization technique that optimizes for stealth and the adversarial aim alternately. This research shows that federated learning is subject to model poisoning attacks. Zhang et al. [84] proposed a poisoning attack against federated learning based on generative adversarial nets (GAN). In this attack, the adversaries can successfully attack the federated learning as participants. The attacker's goal is to generate poisoning samples using GAN to update the global model to be compromised. Kwon et al. [77] proposed a selective poisoning attack that decreases the accuracy of only a selected class in the model by training harmful training data corresponding to the selected class while keeping the normal accuracy of the remaining classes. Zeng et al. [79] presented a perturbation-based causative attack that targets DNN pre-trained model for supply chain in the VANET. The results show that such a perturbation-based misguiding of a DNN classifier in the VANET is effective.

Papers on poisoning attacks that were published in the year 2020 are [36,42,43,56,57,67,73,75,82], and there was only one paper [78] in 2021, at the time when this literature review had been conducted. Lee et al. [36] developed a show and tell model that identifies a photograph of a construction site to determine safety. This poisoning attack generates adversarial data by adjusting a small bit of the RGB values that are not recognized by humans. The authors repeated this process forward and backward until the feature value was close to the target picture feature but the distance was close to the base image. Liu et al. [43] proposed a data poisoning attack in lithographic hotspot detection by poisoning both non-hotspot training and test layout clips. By adding a trigger shape in the input, hotspots in a layout clip can be hidden during inference time. This shows that training data poisoning attacks are viable and stealthy in CAD systems, highlighting the need for ML-based systems in CAD to be more robust.

Huai et al. [42] first introduced adversarial attacks on DRL interpretations. The authors proposed MPDRLI, a model poisoning attack against DRL interpretations. The attacker directly manipulates the pre-trained model parameters obtained during the training phase. With only modest impact on the performance of the original DRL model, the attack can have a considerable impact on the interpretation outcomes. An unseen poisoning attack was postulated by Chen et al. [56] named invisible poisoning attack (IPA). The poison-training examples in this attack were perceptually unnoticeable from the benign ones, making it extremely stealthy. With fewer poison-training examples in the training phase and a better attack success rate in the testing process, the IPA is capable of executing targeted poisoning attacks. The provided poisoning instances can both be invisible and effective in targeting the model.

Chen et al. [57] proposed a poisoning attack named Deep Poison. To fool the target model, the proposed attack uses three-player GAN to create stealthy poisoned instances embedded with the victim class features. The poisoning is accomplished through the use of massively generated poisoned samples for training attack models. The experiments demonstrated that the suggested poisoning attack can reach a high success rate with only 7% poisoned samples in the datasets. Zhou et al. [67] proposed a data poisoning attack against a graph auto-encoder recommender system by injecting fake users that mimic the rating behavior of normal users. In the attack scenario, they assumed that a white-box attack might be unrealistic to apply in real-world recommender systems. Tolpegin et al. [82] show that federated learning (FL) systems are vulnerable to poisoning attacks, in which a subset of malicious participants can poison the trained global model. Their proposed label flipping attack indicates that poisonous samples can reduce the classification accuracy and cause an availability impact to the FL systems even though the participants do not know the model type or parameters. Li et al. [73] proposed a data poisoning attack against deep reinforcement learning (TruthFinder). In this approach malicious workers jeopardize with TruthFinder while hiding themselves. According to this method, malicious workers can continuously learn from their attack efforts and adapt their poisoning strategies. The findings reveal that even if the malicious workers only have access to local information, they can devise successful data poisoning attack tactics to interfere with crowdsensing systems using TruthFinder. Xu et al. [75] proposed an adversarial samples attack using a poisoning attack method on recurrent neural networks (RNNs). To limit the number of samples required for a poisoning attack, they optimized the generation approach and confined the sample search space using abnormal event gradient information. Cole et al. [78] demonstrated a data poisoning attack that needs an adversarial photo injection to enable an attacker to easily mimic the victim model of the existing facial authentication systems with no need of any knowledge from the server-side. Adversaries can compromise the victim's web accounts and log into their services.

5.2. Backdoor attacks

Regarding backdoor attacks, early papers [46,52,53,59,61,69,95] in the systematic literature were published between 2017-2019. Chen et al. [95] proposed backdoor poisoning attacks where the attacker can bypass a deep neural network authentication system by creating a backdoor using data poisoning attacks. For instance, deep neural authentication systems that use face recognition and fingerprint identification. This attack can be considered as a black-box with a success rate of above 90% when injecting only around 50% poisoning samples. Wu et al. [52] performed a backdoor attack against obstacle recognition and processing system (ORPS) by poisoning the dataset to embed the Mask R-CNN in the ORPS (DCNNs-based models). When triggering the backdoor in the target model, the accuracy of the backdoored model may be changed, which may cause serious accidents in self-driving vehicles. The authors claim that this is the first work on this topic in the literature. The experiment reveals how to exploit DNN models (such as the ORPS) by embedding backdoors.

Barni et al. [53] described a backdoor attack that only corrupted samples of the target class without requiring poisoning of the labels. The attack's versatility and stealth are substantially increased in this manner; however, the percentage of samples that must be corrupted is considerable. The implementation looked at two classification tasks: MNIST digit recognition and traffic sign classification. The backdoor signal's invisibility was easier in the first implementation; nevertheless, the results demonstrated a nearly invisible backdoor in the second version. Yao et al. [59] identified latent backdoor attack for transfer learning. Latent backdoors are capable of being embedded in teacher models and automatically inherited by multiple student models via transfer learning. Any student model with the targeted label of the backdoor can be activated after the model recognizes the trigger of the attack's target label.

Bhalerao et al. [61] proposed a backdoor attack targeting a DNN-based anti-spoofing video rebroadcast detector. The authors claim that this is the first time that a backdoor attack has been utilized to compromise an anti-spoofing mechanism. The attack showed robustness against geometric transformations by using a predesigned sinusoidal function to verify the average luminance of the video frames. Gu et al. [69] proposed a backdoored attack named BadNet. The implementation of the BadNet attack considered the MNIST handwritten digit recognition system; and in a more complex scenario, they performed the attack against a traffic sign detection system. The authors demonstrated that BadNets can be a malicious real-world attack that can reliably misclassify stop signs as speed-limit signs by using a Post-it note. These results show that backdoors in deep neural networks are both effective and invisible. Dai et al. [46] implemented a backdoor attack against LSTM-based text classification using data poisoning. To generate poisoning text samples, the attack employs a random insertion approach. The backdoor injected into the model has little effect on model performance, but the trigger attack is stealthy. The testing results showed that the attack can reach around 96% success rate with 1% poisoning rate in a black-box situation and through sentiment analysis.

In the year 2020 many studies were published that proposed backdoor attacks. Dumford and Scheirer [37] suggested an attack that targeted certain layers inside a CNN model's network in order to create an attacker accessible backdoor. The attack requires no prior access to training data; nevertheless, access to the pre-trained model is required. Zhong et al. [41] proposed generating approaches of a backdoor attack that are invisible in poisoning the model. Backdoor injection can be performed either before or after model training. The backdoor employs data poisoning to inject the trigger with a small poisoning percentage that does not interfere with the usual operation of the learned process. The results show that even under the most weak assumptions, such as in a black-box situation, attacks can be effective and achieve a high attack success rate at a low cost in terms of model accuracy loss and injection rate. Such attacks might exploit a deep learning system's vulnerability invisibly and possibly cause havoc in a variety of actual applications, such as destroying an autonomous vehicle or impersonating another person to gain illegal access. Xue et al. [30] offered two techniques to backdoor attacks. The first is known as One-to-N, and it can trigger multiple backdoor targets by varying the intensities of the same backdoor. The second is known as N-to-One, and it is triggered when all N backdoors are satisfied. Against some defense techniques, the suggested One-to-N and N-to-One attacks are effective and stealthy. Zhao et al. [47] proposed a clean-label backdoor attack against deep neural networks in video recognition models. The attack makes use of the adversarial approach (PGD) to generate videos with a universal adversarial backdoor trigger. The results demonstrated that the suggested backdoor attack can influence state-of-the-art video models and can be utilized as a baseline for enhancing the robustness of video models, as well as to improve picture backdoor attacks.

Liu et al. [48] proposed Refool, a clean-label backdoor attack inspired by an important natural phenomenon: reflection and employing mathematical modeling of physical reflection models. The empirical findings demonstrate a high success rate and a slight loss in clean accuracy. Reflection backdoors are easy to create and impervious to state-of-the-art protection measures. Tan and Shokri [50] proposed an adversarial backdoor embedding attack that uses adversarial regularization to maximize the discriminator's loss in order to avoid network detection algorithms. The findings reveal that a skilled attacker may readily conceal the signals of backdoor images in the latent representation, rendering the defense ineffective. This research asks for the development of adversary-aware defense techniques for backdoor detection. Lovisotto et al. [85] proposed a template poisoning attack against face recognition that allows the adversary to inject a backdoor. This biometric backdoor allows the adversary to grant discreet long-term access to the biometric system using craft-colored glasses. The result shows that with the white-box scenario the attack success rate is acceptable which is over 70% of cases; however, the attack success rate decreases in the black-box scenario to around 15% of cases.

Li et al. [34] aim to raise awareness of the seriousness of invisible triggers that can misguide both machine learning models and human users. In this matter, detection becomes significantly more difficult than with current backdoor triggers. In this work, the first invisible backdoor attack employed steganography techniques to conceal the manually created trigger. The authors embedded concealed data into a cover image using the least significant bit (LSB). On the CIFAR-10 dataset, the trigger's invisibility decreases as its size increases. The second attack in this study makes the shape and size of trigger patterns undetectable by using Lp-norm regularization. The authors used three universal backdoor attacks that targeted the penultimate Layer (L2, L0, and $L\infty$ regularization-based). Using the Lp-norm ensures that the noise is modest; nonetheless, the threshold value used to end the optimization has an effect on the attack success rate. Utilizing a large stop threshold makes the trigger visible to human inspectors, while using a smaller stop threshold makes injecting the backdoor into the DNN more difficult.

Xiong et al. [54] proposed two backdoor attacks capable of effectively inserting the backdoor while evading two state-of-the-art detection mechanisms (the Neural Cleanse and DeepInspect). The distinction between the first and second attacks is that the second one increases both attack ability and detection difficulties by decreasing the ratio of the trigger size. The triggers of their backdoor attack strategy can be generally hidden and reconstruction-resistant incorporated into DNN models. The results show that the proposed attacks are capable of evading detection systems. Kwon et al. [55] developed TargetNet, a backdoor attack that induces misclassification in multi-targeted models by utilizing a single trigger with a different label for each of the targeted models. The experimental results show a high success attack rate with nearly ordinary accuracy. He et al. [31] proposed an invisible hidden backdoor attack method named BHF2 against face recognition systems where the attacker can inject a small batch that can successfully let the attacker login into the system as the victim. The results suggest that the BHF2 method can obtain a high attack success rate while sacrificing only a few percent of accuracy. Quiring and Rieck [60] proposed a combining clean-label poisoning attack and a backdoor attack that benefits from image-scaling to hide the trigger. These attacks enable the manipulation of images when scaled to a specific resolution with almost a constant performance.

KWON et al. [63] proposed a multi-targeted backdoor that misleads different models into distinct classes. This technique trains multiple models with data that includes specific triggers that cause misclassification into related classes by distinct models. The proposed techniques can be applied to the audio and visual domains. Venceslai et al. [76] proposed a hardware backdoor named NeuroAttack, which is a cross-layer attack that against the Spiking Neural Networks (SNNs). By attacking low-level reliability vulnerabilities with high-level attacks, this attack triggers a fault-injection-based sneaky hardware backdoor via a carefully prepared adversarial input noise. Liao et al. [41] demonstrated that the backdoor injection attacks with a small loss accuracy rate and a small injection rate can achieve high success. The authors proposed a data poisoning attack which triggered CNN models to recognize particular embedded patterns with a target label in a covert manner without compromising the accuracy of the victim models.

In the year 2021, at the time of the literature review, there were [39,58,70] papers. Zhou et al. [39] presented a unique optimization-based model poisoning attack on federated learning by injecting adversarial neurons in a neural network's redundant space using the regularization term in the objective function. The results demonstrated that the proposed attack mechanism outperformed backdoor attacks in terms of performance effectiveness, durability, and robustness. Furthermore, Xue et al. [58] introduced another hidden backdoor attack method known as BHF2N, which also conceals the created backdoors in facial features (eyebrows and beard). The proposed methods provide the hiding of backdoor attacks and can be used in cases requiring strong identity authentication. As a result, the suggested invisible backdoor attacks are stealthy and possible for more rigorous face recognition scenarios, emphasizing the importance of creating strong defenses against DNN-based biometric recognition systems. Guo et al. [70] proposed a backdoored attack against a face verification system that matches a master face to give a positive answer against any other face. They consider a white-box scenario where the adversary can have full knowledge of the target system such as in MLaaS systems. The experiments showed that the attack is effective to the siamese network that conforms images of two faces belonging to the same person.

5.3. Trojans

As the effort required to train deep neural network models dramatically increased in recent years due to the high hardware requirement and time consumption, machine-learning-as-a-service becomes the option for many users. Early trojan papers [40,66,71,74,96–98] in the systematic review were published between 2017-2019. Liu et al. [96] demonstrated that malicious samples can be embedded in the training dataset of the deep neural networks. These malicious samples might be activated in later stages and this hidden malicious functionality is known as neural Trojans. Training data poisoning-based attacks Zou et al. [97] proposed a Neural-Level Trojans named PoTrojan. The authors claim that PoTrojan was the first proposal of an attack on pre-trained neural network models. Attackers can insert additional malicious neurons into a pre-trained DNN that would remain inactive until they are triggered to cause malfunction of the deep neural network. However, this might be hard to achieve in practice since the attackers need to have full knowledge of the target deep neural networks. Binary-level attack Trojans can also be embedded using the binary code of neural networks. Liu et al. [98] proposed a low-cost modular binary code attack namely "SIN2" Stealth Infection on Neural Network. The rising dangers in this intelligent supply chain scenario enable a novel practical neural Trojan attack that has no effect on the quality of intelligent services. Li et al. [74] proposed a hardware-software collaborative neural Trojans attack in which neural Trojans are hidden into a well-structured subnet during the training process and triggered by hardware Trojans at the appropriate time.

Liu et al. [66] proposed a low-cost neural Trojan attack named SIN2 (Stealth Infection on Neural Network) that can threaten an intelligent supply chain system by performing malicious payloads to cause misclassification in such a service. The trigger is injected into the victim DNN model through a proposed embedding method to replace selected weight parameters. The infected DNN model performs a normal service until the malicious payloads are extracted and executed at the victim side on the runtime which can violate the confidentiality, integrity, and efficiency of such a system.

Clements and Lao [71] proposed a hardware neural Trojans attack. This novel technique injects hardware Trojans into computational blocks of a DNN implementation in order to achieve an adversarial aim by triggering the logic operation that targets the activation's functions. The results show that the different scenarios of this hardware Trojans can be worthwhile. Davaslioglu and Sagduyu [40] proposed a Trojan attack against a deep wireless communications classifier model. They use raw (I/Q) samples as features and modulation types as labels to categorize wireless signals. The trigger injects into a few training data samples by changing their phases and labeling them with a target label. During testing, the adversary emits signals with the identical phase shift that was added as a trigger during training. The Trojans attack remains hidden until the trigger that bypasses a signal classifier, such as an authentication, is activated.

In the systematic review of Trojan attacks, papers that were published between 2020-2021 are [44,45,49,51,62,64,65,80]. Tang et al. [51] proposed a training-free Trojan attack strategy in which a little Trojan module named TrojanNet is inserted into the target model. The authors claim that this method differs from past similar attacks in that the Trojan behaviors are injected by the retraining model on a poisoned dataset. TrojanNet is an all-label Trojan attack that can inject the Trojan into all labels at the same time and achieves a 100% attack success rate without impacting model accuracy on original tasks. The results have shown that the proposed approach can mislead cutting-edge Trojan detection algorithms, causing TrojanNet attacks to go undetected.

Lin et al. [45] introduced a composite attack, a more adaptable and covert Trojan attack that avoids backdoor detectors by utilizing Trojan triggers constructed of existing benign subjects/features from several labels. In this attack, training is outsourced to a malicious agent with the goal of providing the user with a pre-trained model that contains a backdoor. The infected model performs well on normal inputs but predicts the target label when the inputs match particular composition rules and meet attacker-chosen attributes. The attack demonstrates that a DNN with a constructed backdoor may achieve accuracy comparable to its original version on benign data while misclassifying when the composite trigger is present. Chen et al. [49] presented SPA, an unique stealthy Trojan attack technique. A generative adversarial network is used to generate the poisoned samples. Experiments have shown that SPA may obtain a high success rate for a Trojan attack with only a few poisoned samples in well-known datasets such as LFW and CASIA.

Costales et al. [62] presented a live Trojan attack that can exploit a DNN model's parameters in memory at run-time to achieve predefined malicious behavior on a certain set of inputs. They demonstrate the feasibility of this attack by using a method of performing patches with clean and Trojaned images after retraining against multiple real-world deep learning systems. Liu et al. [64] proposed a hardware Trojan attack that leverages a specific sequence of normal images as a trigger. The result shows that this triggering technique is more robust to the pixel-bit triggering to evade the image pre-processing and is also invisible to human beings.

Rakin et al. [65] proposed a neural Trojan attack named Targeted Bit Trojan (TBT), which inserts a neural Trojan in the main memory such as a DRAM to target the DNN weights using a bit-flip attack. At the inference phase, the DNN operates as normal with the normal accuracy until the trigger is activated which violates availability because the network is forced to classify all inputs to a certain target class. Pan [80] proposed a black box Trojan attack that can induce malicious inferences to any deep neural network model with simple steps. The attack operates in different modes from normal behavior to false positives and false negative errors which alter the integrity and availability of any deep learning image classification system. The trigger vector could manipulate the weights to cause specific types of errors and change of the modes. Hu et al. [44] proposed a hardware Trojans attack on DNN by accessing to merely the memory bus data. The attack injects malicious logics into the memory controllers of DNN systems without the need for toolchain manipulation or access to the victim model, making it viable for usage.

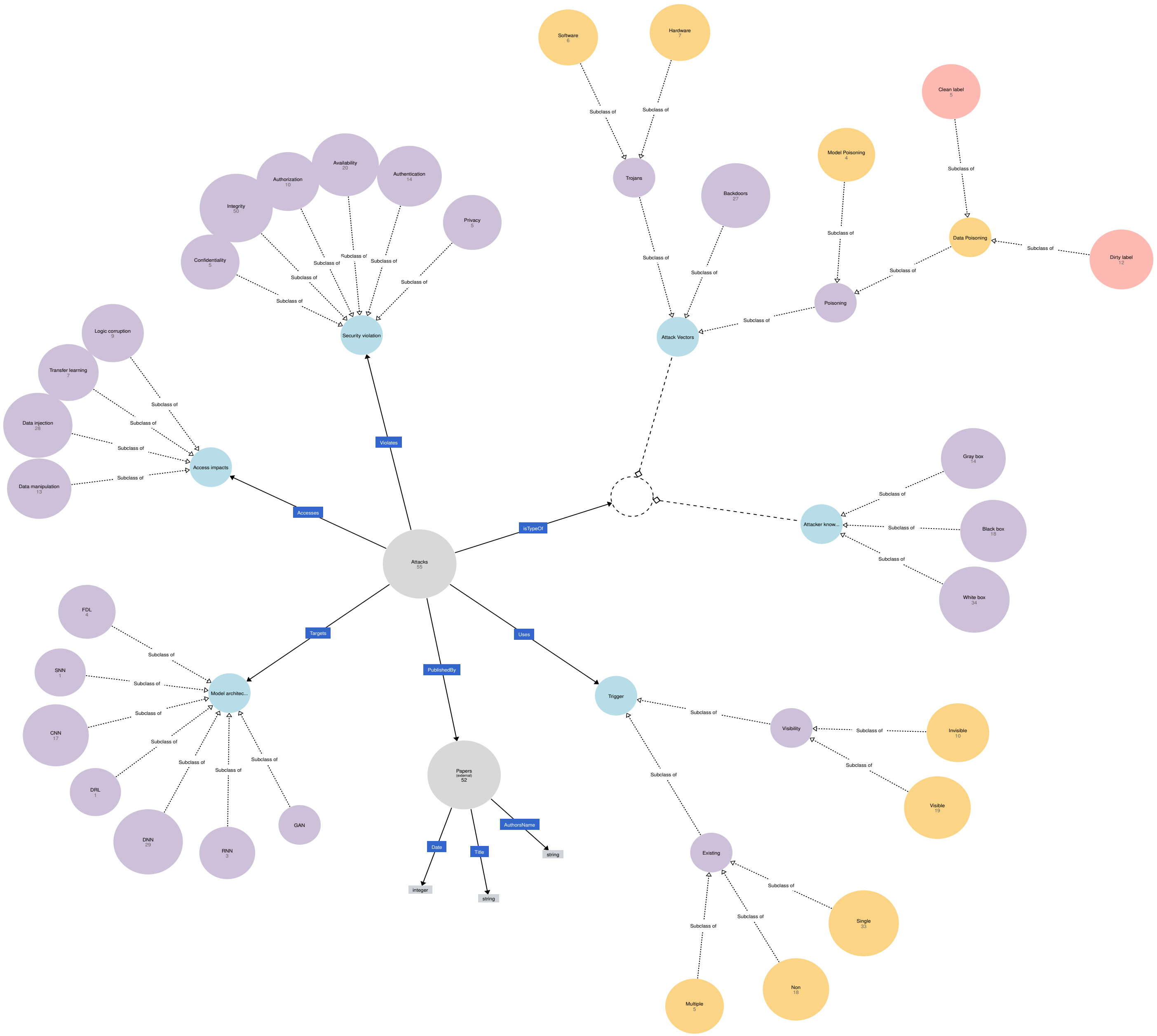


Figure 7. The DNNPAO ontology including classes and properties

6. DNN Poisoning Attacks Ontology (DNNPAO) and Knowledge Base Construction

In this section, an ontology and a knowledge base is presented in order to obtain semantic information about the poisoning attacks. To determine the domain and scope of the ontology we used our extracted taxonomy as a scheme. The DNN Poisoning Attacks Ontology (DNNPAO) maps all the classes that were introduced in the poisoning attack taxonomy in Section 4 into a single concept, and maps all the attacks that were systematically reviewed as individuals with their relationships. The basic aim of the proposed ontology is to express the complex knowledge of poisoning attacks against DNNs. Thus, we built a knowledge base using the DNNPAO as a framework.

6.1. DNNPAO Construction

An ontology is a piece of semantic information that is written using a semantic language to describe things and the relationships between these things. There are several semantic languages; in this paper, we use the Web Ontology Language (OWL) to describe our ontology. To develop the ontology using the OWL semantic language the Protégé tool [20] and webprotege.stanford.edu [20] were used.

Table 9. The definition of the Main Classes in DNNPAO

Class Label	Definition	Class Type
Attacks	Represent poisoning attacks against DNNs	super class(Thing)
Attack Vectors	Represent the mechanisms that attackers use to exploit the DNNs' vulnerabilities.	super class(Thing)
Security Violation	A violation on the DNNs caused by an attack.	super class(Thing)
Triggers	A hidden malicious functionality added to DNNs to activate Trojans and Backdoors.	super class(Thing)
Attackers knowledge	Define the knowledge that an adversary has of the targeted model.	super class(Thing)
Access Impacts	The adversaries' capability of gaining access to DNNs.	super class(Thing)
Models Architectures	The architecture of DNNs' target models.	super class(Thing)
Papers	Represent the published research papers that proposed those attacks.	super class(Thing)

Table 10. Definition of the main relationships in DNNPAO

Object Properties	Definition	From -> To
Subclass of	A sub division of a class	Class -> Sub Class
Violates	A relationship of an attack to violate the security of a DNN.	Attacks -> Security Violation
Uses	The relationship between an attack and a trigger that, if exist, activates the attack.	Attacks -> Triggers
isTypeOf	Is a relationship indicates that an attack is a type of an attack vector or an attackers knowledge scenario.	Attacks -> Attack Vectors Attacks -> Attackers knowledge
Accesses	Represent the relation between an attack and the access possibility.	Attacks -> Access Impacts
Targets	Represent the connection of an attack and its DNN targeted model.	Attacks -> Models Architectures
PublishedBy	Is a relationship between the attacks and their reference.	Attacks -> Papers

Table 11. The main facets of the ontology.

Metrics	Results
Class count	45
Object property count	6
Data property count	4
Individual count	107
SubClassOf	44
ObjectPropertyDomain	6
ObjectPropertyRange	7
DataPropertyDomain	3
DataPropertyRange	3

We have mapped the 55 extracted attacks as individuals which were added into the DNNPAO ontology with their relations to the corresponding classes and sub-classes these attacks connected with their papers as shown in Table 12 on Page 27. As shown in Figure 8 on Page 26 these three examples of individuals (attacks) appear in the DNNPAO ontology at webprotege.stanford.edu [20]. Attack1, Attack3, and Attack34 are individuals that have relations with multiple classes and sub-classes; in other words, they are instances of these classes. In addition, they have a "PublishedBy" relationship with other individuals that are the papers where these attacks were published in. For a full list of Individuals (attacks) in DNNPAO please refer to Appendix A on Page 32.

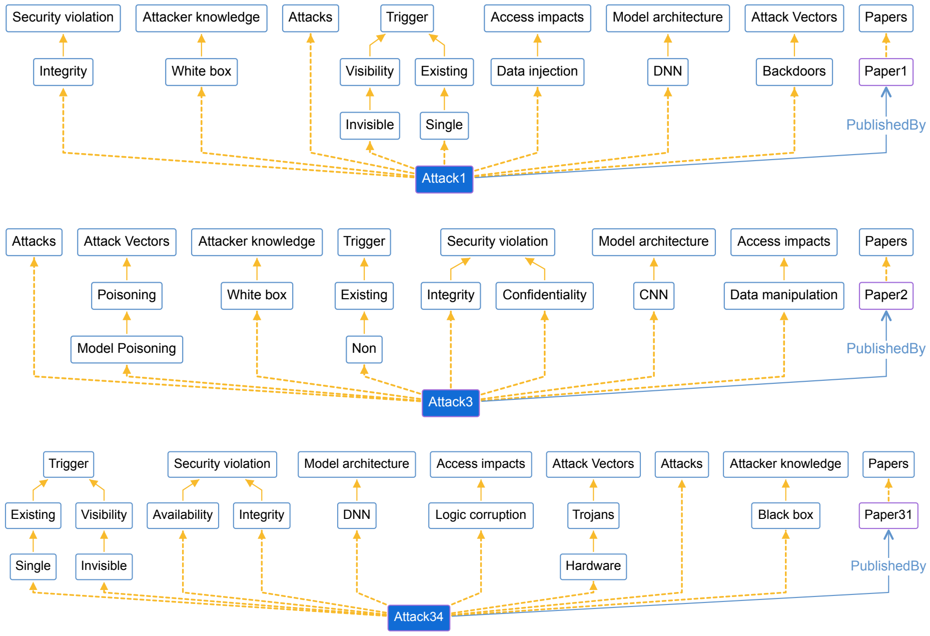


Figure 8. Three individuals (i.e., attacks), Attack1, Attack3, and Attack34, as instances of DNNPAO classes and sub-classes.

Figure 8 provides examples of three individuals, that is, attacks, Attack1, Attack3, and Attack34, as instances of DNNPAO classes and sub-classes. For instance, Attack1 is an instance of the class "Integrity" which is a sub-class of the super-class "Security violation". In addition, Attack1 is connected to Paper1 which is another individual or instance of the class "Papers". Because Attack1 and Paper1 are both individuals they have the relationship "PublishedBy". The other examples Attack3 and Attack4 are created similarly. For a full list of Individuals (attacks) in DNNPAO please refer to Appendix A on Page 32.

Table 12. A List of Individuals in DNNPAO.

Attacks	Papers	Reference
Attack1	Paper1	Li et al. [34]
Attack2	Paper1	Li et al. [34]
Attack3	Paper2	Lee et al. [36]
Attack4	Paper3	Dumford et al. [37]
Attack5	Paper4	Bhagoji et al. [38]
Attack6	Paper5	Zhou et al. [39]
Attack7	Paper6	Davaslioglu et al. [40]
Attack8	Paper7	Zhong et al. [41]
Attack9	Paper8	Huai et al. [42]
Attack10	Paper9	Xue et al. [30]
Attack11	Paper9	Xue et al. [30]
Attack12	Paper10	Liu et al. [43]
Attack13	Paper11	Hu et al. [44]
Attack14	Paper12	Lin et al. [45]
Attack15	Paper13	Dai et al. [46]
Attack16	Paper14	Zhao et al. [47]
Attack17	Paper15	Liu et al. [48]
Attack18	Paper16	Chen et al. [49]
Attack19	Paper17	Tan et al. [50]
Attack20	Paper18	Tang et al. [51]
Attack21	Paper19	Wu et al. [52]
Attack22	Paper20	Barni et al. [53]
Attack23	Paper21	Xiong et al. [54]
Attack24	Paper22	Kwon et al. [55]
Attack25	Paper23	Chen et al. [56]
Attack26	Paper23	Chen et al. [57]
Attack27	Paper24	He et al. [31]
Attack28	Paper25	Xue et al. [58]
Attack29	Paper26	Yao et al. [59]
Attack30	Paper27	Quiring et al. [60]
Attack31	Paper28	Bhalerao et al. [61]
Attack32	Paper29	Costales et al. [62]
Attack33	Paper30	Kwon et al. [63]
Attack34	Paper31	Liu et al. [64]
Attack35	Paper32	Rakin et al. [65]
Attack36	Paper33	Liu et al. [66]
Attack37	Paper34	Zhou et al. [67]
Attack38	Paper35	Zhu et al. [68]
Attack39	Paper36	Gu et al. [69]
Attack40	Paper37	Guo et al. [70]
Attack41	Paper38	Clements et al. [71]
Attack42	Paper39	Munoz et al. [72]
Attack43	Paper40	Li et al. [73]
Attack44	Paper41	Li et al. [74]
Attack45	Paper42	Xu et al. [75]
Attack46	Paper43	Venceslai et al. [76]
Attack47	Paper44	Kwon et al. [77]
Attack48	Paper45	Cole et al. [78]
Attack49	Paper46	Zeng et al. [79]
Attack50	Paper47	Pan [80]
Attack51	Paper48	Garofalo et al. [81]
Attack52	Paper49	Tolpegin et al. [82]
Attack53	Paper50	Li et al. [83]
Attack54	Paper51	Zhang et al. [84]
Attack55	Paper52	Lovisotto et al. [85]

6.2. Knowledge Base Construction

Our main topic is DNN security but it is useful for the field and community to get to share complex knowledge using an ontology and knowledge base. Thus, we have built the knowledge base using DNNPAO as a framework. To do so, we chose the Neo4j database [21] with Neosemantics plugin [22] that enables an OWL ontology to be imported into Neo4j database.

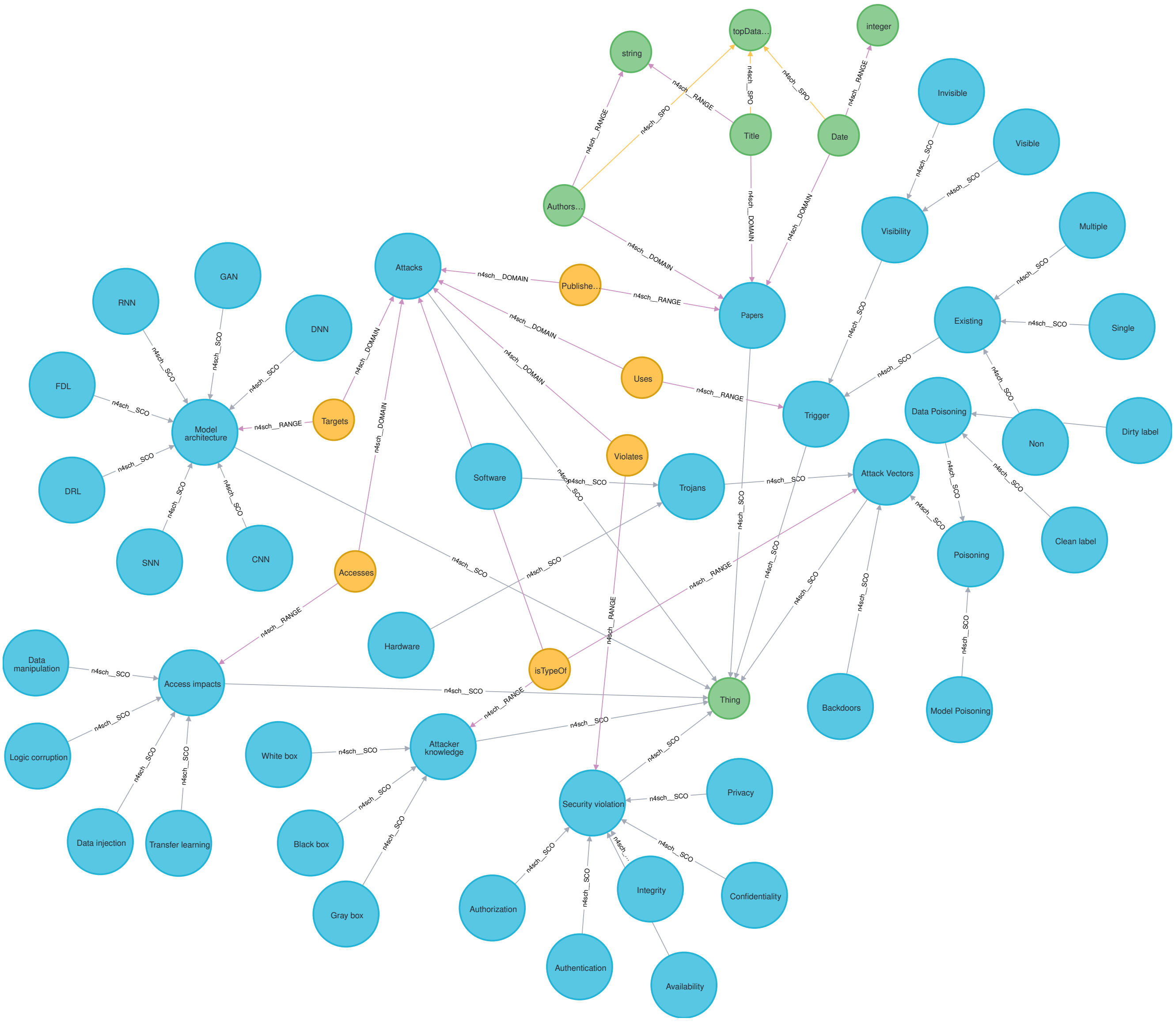


Figure 9. DNNPAO classes and sub-classes with their relationships in Neo4j database.

doi:10.20944/preprints202208.0197.v1

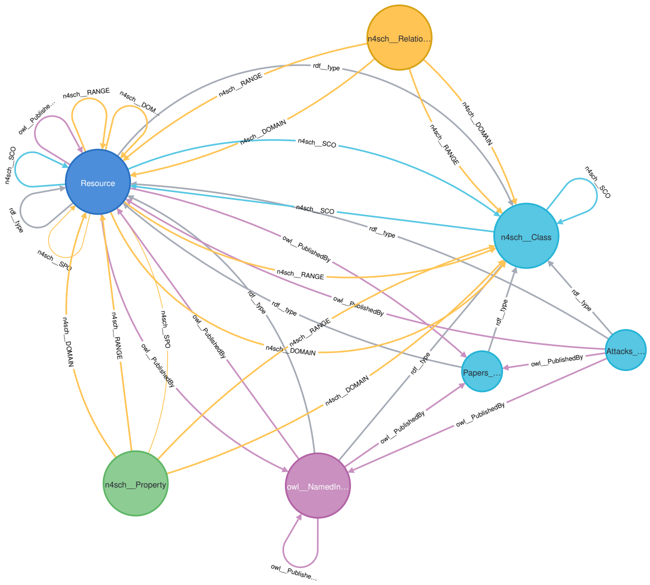


Figure 10. Meta-graph of the DNNPAO in Neo4j database.

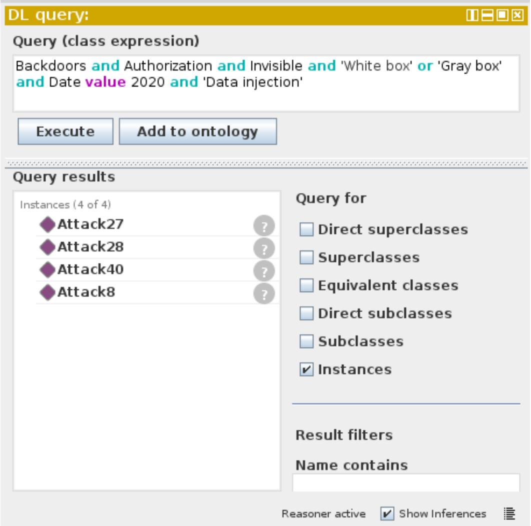
7. Discussion, Conclusion, and Future Work

DNNs have brought innovative changes and introduced new dimensions in many fields. They rely upon their input data, structure, and parameters and any change might mislead the DNNs and this sensitivity of DNNs makes them brittle against adversarial attacks. Attackers can mislead the DNNs by corruption parameters, maximizing error functions, or manipulating the datasets. Recent research demonstrates that the robustness of deep neural networks is a critical weakness against malicious attacks.

The robustness of DNNs is also vital due to the emerging concepts of responsible and green artificial intelligence (AI) that aim to preserve ethics, fairness, democracy, and explainability of AI-based decision systems.

Although there are a growing number of research on adversarial attacks, only a few of them focus on poisoning attacks. Meanwhile, with the rapid use of DNNs, evaluating the robustness of DNNs involves exploring weaknesses in their models. Traditional security management and threat analysis lack methods for intelligent responses to new threats.

A semantic knowledge representation of security attacks is a significant means of retrieving data for analysts whether they are human or AI agents. In addition, there is still a lack of semantic knowledge graphs and intelligent reasoning technologies for emerging attacks. The purpose of an ontological knowledge base is to derive knowledge that has not been clearly expressed. For instance, DNNPAO was queried in order to deduce some knowledge of the attacks, as seen in Figure 11 on Page 30. The intention of the queries is to demonstrate the viability of DNNPAO rather than to provide definitive answers to all passable questions. As illustrated in Figure 11 the query was carried out in Protégé using the DL Query and the HermiT reasoner. It shows the attacks, Attack8, Attack27, Attack28, and Attack40 as a result of the question of what are the invisible backdoor attacks that violate the authorization of the system and can be injected on the dataset from adversaries in white-box or gray-box scenarios. To restrict the reuse, we select the 2020 published attack; however, more complex queries can be conducted using SPARQL or the Neo4j database [21]. Furthermore, more relevant graph exploration tools, such as SemSpect [99], can be used to represent the data from Neo4j using a different visual interaction manner. Updating such a knowledge base with new attacks and countermeasures can result in more autonomous DNN security threat assessments.



Backdoors and Authorization and Invisible and 'White box' or 'Gray box' and Date value 2020 and 'Data injection'

Instances (4 of 4)
Attack27
Attack28
Attack40
Attack8

☐ Direct superclasses
☐ Superclasses
☐ Equivalent classes
☐ Direct subclasses
☐ Subclasses
☑ Instances

☑ Show Inferences

Figure 11. Using the DL Query and the HermiT reasoner to query the DNNPAO in Protégé.

This paper proposed an ontology of poisoning and backdoor attacks in DNNs, called DNN Poisoning Attacks Ontology (DNNPAO) that will enhance knowledge sharing and enable further advancements in the field. We performed a systematic review of the relevant literature to identify the current state, collected 28,469 papers and, through a rigorous process, 55 poisoning attacks in DNNs were identified and classified. We extracted a taxonomy of the poisoning attacks as a scheme to develop DNNPAO. Subsequently, we used DNNPAO as a framework to create a knowledge base.

Considering on the literature, we can highlight some essential directions toward robust DNNs. As can be seen in Table 7 on Page 15, the CNN is the architecture of DNNs that has been attacked the most, however, that does not mean it is the most vulnerable architecture but it is the most used architecture, especially for computer vision. In fact, the DRL and DFL might be at more risk if we consider the environment and participants as part of the training process.

Backdoors and Trojans are usually based on the poisoning injections of the triggers, thus, filtering the data sets is an essential process in detecting the triggers. However, as shown in some of the attacks, triggers could be embedded in hardware, which means that scanning devices to detect these triggers are needed. In addition, triggers if injected might not affect the accuracy until they have been activated. Some of the attacks show that even a physical tool can be used to activate the triggers at the inference phase. Therefore, creating an effective trigger detection tool at this phase is also necessary. Such a tool should consider the invisible and multiple triggers.

Although there are growing intentions to better understand the DNNs security threats, only a few works deal with poisoning and backdoor attacks. Taxonomies are incapable of representing complex relationships, either between attacks and their characteristics or among other attacks. One approach to addressing this complexity is using an ontology knowledge base. An ontology can cope with this information while enabling scalability, so that more complex information may be added. Thus in this paper, we proposed a seed ontology knowledge base. Our ontology was generated according to a taxonomic scheme that we developed on the basis of a systematic literature review. For further work, we will consider other inference phase attacks such as adversarial example attacks and we will add more attack characteristics. In addition, we plan to use the complete ontology to train a DNN model to detect attacks toward better robustness of DNNs. Finally, we have mentioned several applications where DNNs are used, some of these are developed as part of our broader work in DNNs. As part of this strand of research, we plan to explore various DNN attacks on these applications in the future. We envision that the work presented in this paper will open new directions within the field of AI security.

Author Contributions: Conceptualization, M.A. and R.M.; methodology, M.A. and R.M.; software, M.A.; validation, M.A. and R.M.; formal analysis, M.A., R.M., F.A., T.Y. and J.M.C.; investigation, M.A., R.M., F.A., T.Y., and J.M.C.; resources, R.M. and F.A.; data curation, M.A.; writing—original draft preparation, M.A.; writing—review and editing, R.M., F.A., T.Y. and J.M.C.; visualization, M.A.; supervision, R.M. and F.A.; project administration, R.M.; funding acquisition, R.M. All authors have read and agreed to the published version of the manuscript.

Funding: The authors acknowledge with thanks the technical and financial support from the Deanship of Scientific Research (DSR) at the King Abdulaziz University (KAU), Jeddah, Saudi Arabia, under Grant No. RG-11-611-38.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data used in this paper will be made publicly available at https://github.com/MajedCS/DNNPAO

Acknowledgments: The work carried out in this paper is supported by the HPC Center at the King Abdulaziz University.

Conflicts of Interest: The authors declare no conflict of interest.

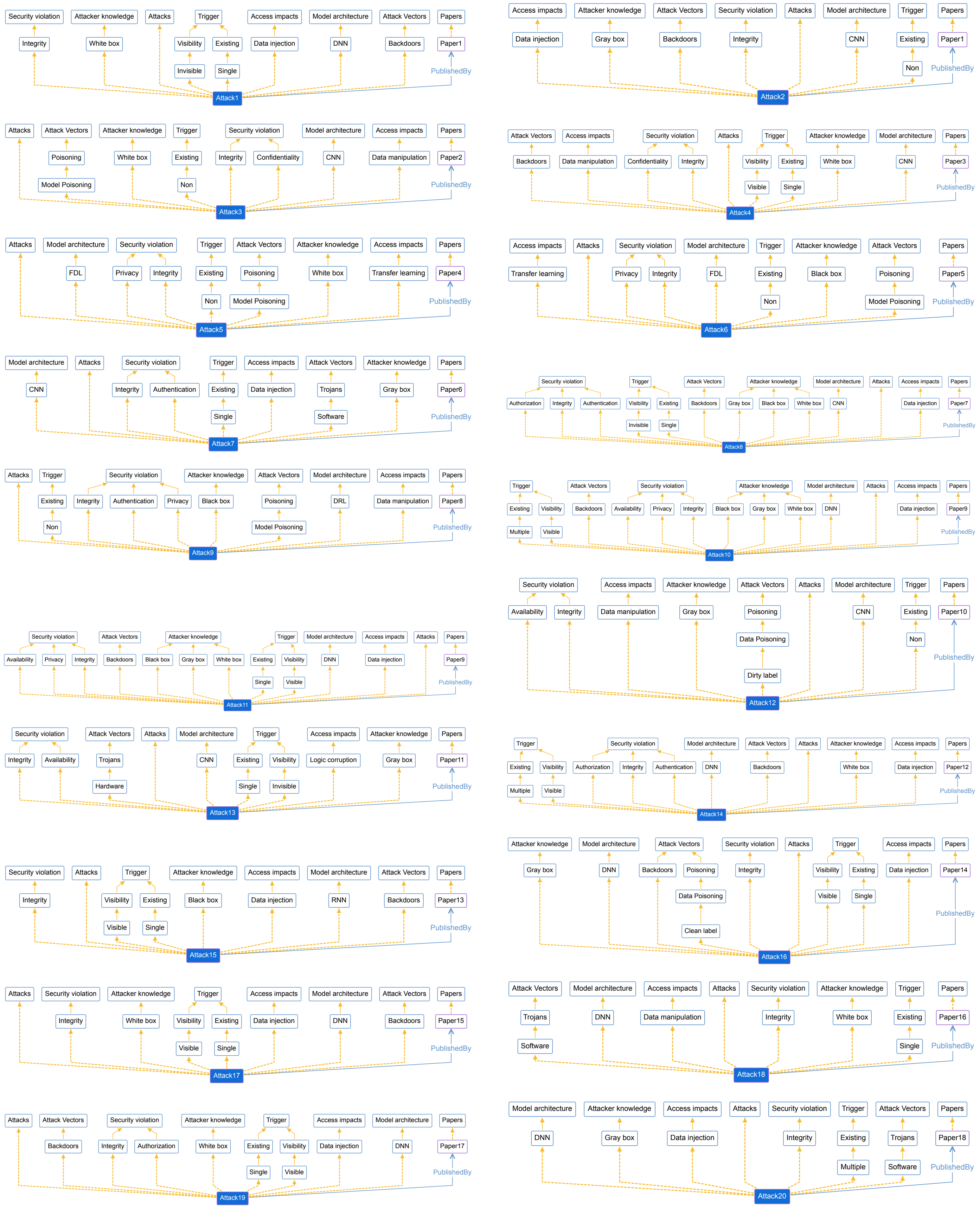
Abbreviations

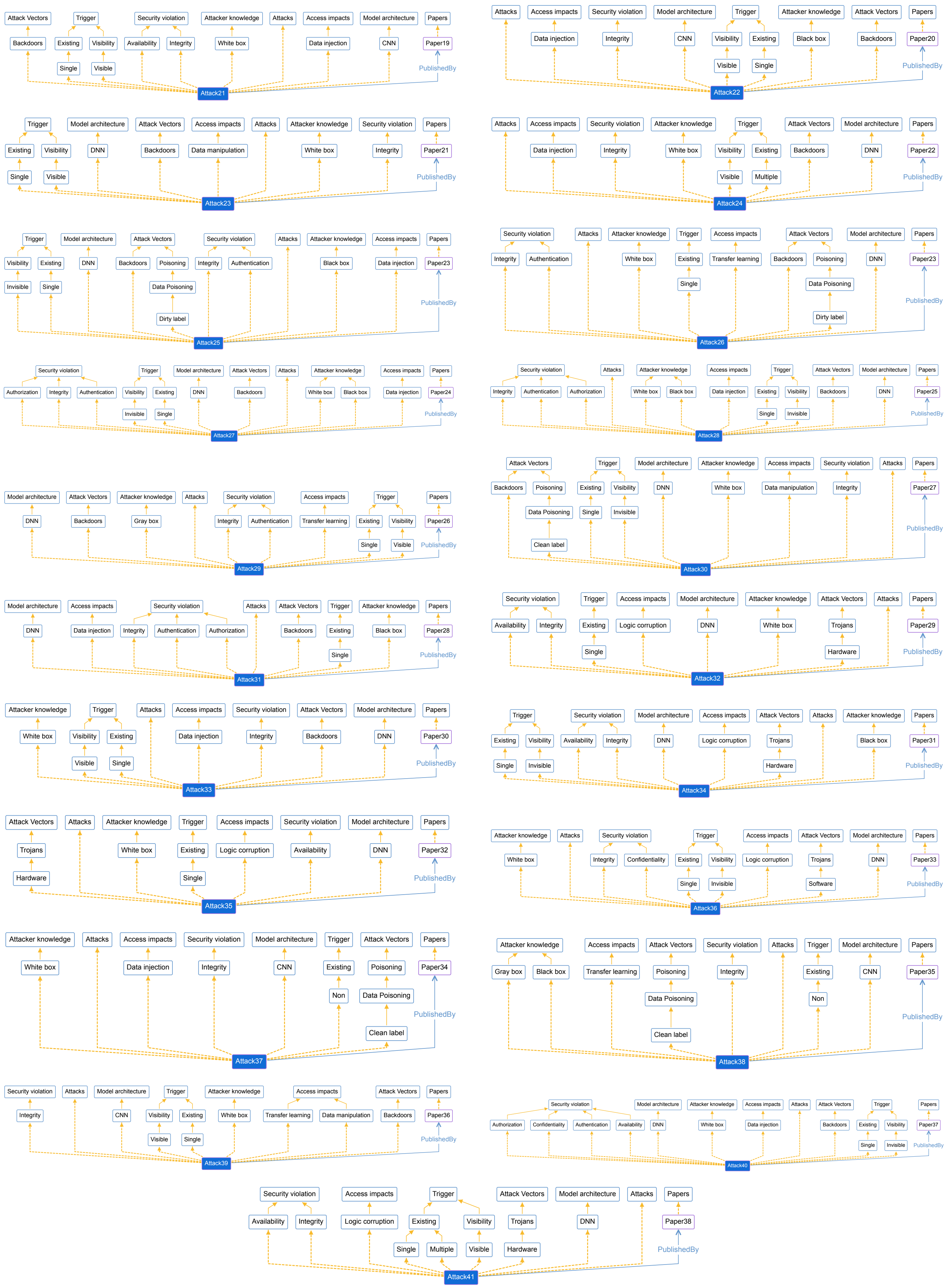
The following abbreviations are used in this manuscript:

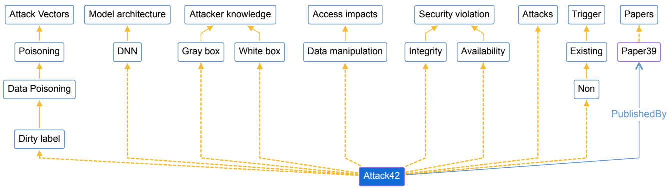
DNN	Deep neural networks
DNNPAO	DNN Poisoning Attacks Ontology
NLP	Natural Language Processing
SLR	Systematic Literature Review
AI	Artificial Intelligence
MRQs	Main Research Questions
SRQ	Sub Research Questions
S1	Search String 1
S2	Search String 2
S3	Search String 3
S4	Search String 4
S5	Search String 5
S6	Search String 6
S7	Search String 7
S8	Search String 8
S9	Search String 9
NN	Neural Network
CNN	Convolutional Neural Network
RNN	Recurrent Neural Network
GAN	Generative Adversarial network
DCGAN	Deep Convolutional GAN
RL	Reinforcement Learning
SNN	Spiking Neural Network
DFL	Deep Federated Learning
OWL	Web Ontology Language

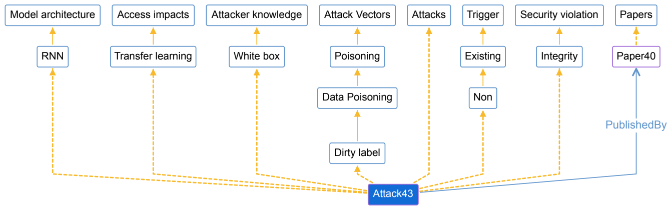
doi:10.20944/preprints202208.0197.v1

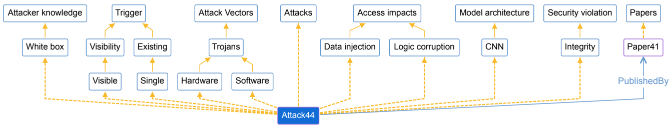
Appendix A. List of Individuals (attacks) in DNNPAO.

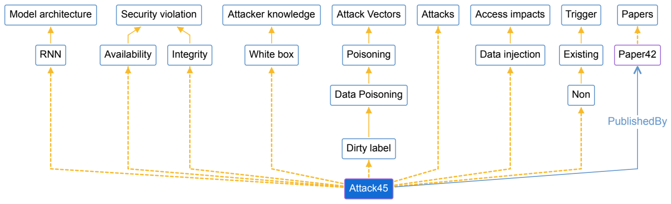


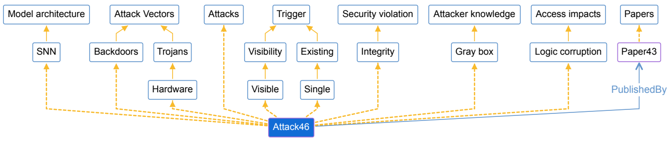

Attack Vectors
Trigger
Security violation
Attacker knowledge
Attacks
Access impacts
Model architecture
Papers
Backdoors
Existing
Visibility
Availability
Integrity
White box
Data injection
CNN
Paper19
Single
Visible
PublishedBy
Attack21
Attacks
Access impacts
Security violation
Model architecture
Trigger
Attacker knowledge
Attack Vectors
Papers
Data injection
Integrity
CNN
Visibility
Existing
Black box
Backdoors
Paper20
Visible
Single
PublishedBy
Attack22
Trigger
Model architecture
Attack Vectors
Access impacts
Attacks
Attacker knowledge
Security violation
Papers
Existing
Visibility
DNN
Backdoors
Data manipulation
White box
Integrity
Paper21
Single
Visible
PublishedBy
Attack23
Attacks
Access impacts
Security violation
Attacker knowledge
Trigger
Attack Vectors
Model architecture
Papers
Data injection
Integrity
White box
Visibility
Existing
Backdoors
DNN
Paper22
Visible
Multiple
PublishedBy
Attack24
Trigger
Model architecture
Attack Vectors
Security violation
Attacks
Attacker knowledge
Access impacts
Papers
Visibility
Existing
DNN
Backdoors
Poisoning
Integrity
Authentication
Black box
Data injection
Paper23
Invisible
Single
Data Poisoning
Dirty label
PublishedBy
Attack25
Security violation
Attacks
Attacker knowledge
Trigger
Access impacts
Attack Vectors
Model architecture
Papers
Integrity
Authentication
White box
Existing
Transfer learning
Backdoors
Poisoning
DNN
Paper23
Single
Data Poisoning
Dirty label
PublishedBy
Attack26
Security violation
Trigger
Model architecture
Attack Vectors
Attacks
Attacker knowledge
Access impacts
Papers
Authorization
Integrity
Authentication
Visibility
Existing
DNN
Backdoors
White box
Black box
Data injection
Paper24
Invisible
Single
PublishedBy
Attack27
Security violation
Attacks
Attacker knowledge
Access impacts
Trigger
Attack Vectors
Model architecture
Papers
Integrity
Authentication
Authorization
White box
Black box
Data injection
Existing
Visibility
Backdoors
DNN
Paper25
Single
Invisible
PublishedBy
Attack28
Model architecture
Attack Vectors
Attacker knowledge
Attacks
Security violation
Access impacts
Trigger
Papers
DNN
Backdoors
Gray box
Integrity
Authentication
Transfer learning
Existing
Visibility
Paper26
Single
Visible
PublishedBy
Attack29
Attack Vectors
Trigger
Model architecture
Attacker knowledge
Access impacts
Security violation
Attacks
Papers
Backdoors
Poisoning
Existing
Visibility
DNN
White box
Data manipulation
Integrity
Paper27
Data Poisoning
Single
Invisible
Clean label
PublishedBy
Attack30
Model architecture
Access impacts
Security violation
Attacks
Attack Vectors
Trigger
Attacker knowledge
Papers
DNN
Data injection
Integrity
Authentication
Authorization
Backdoors
Existing
Black box
Paper28
Single
PublishedBy
Attack31
Security violation
Trigger
Access impacts
Model architecture
Attacker knowledge
Attack Vectors
Attacks
Papers
Availability
Integrity
Existing
Logic corruption
DNN
White box
Trojans
Paper29
Single
Hardware
PublishedBy
Attack32
Attacker knowledge
Trigger
Attacks
Access impacts
Security violation
Attack Vectors
Model architecture
Papers
White box
Visibility
Existing
Data injection
Integrity
Backdoors
DNN
Paper30
Visible
Single
PublishedBy
Attack33
Trigger
Security violation
Model architecture
Access impacts
Attack Vectors
Attacks
Attacker knowledge
Papers
Existing
Visibility
Availability
Integrity
DNN
Logic corruption
Trojans
Black box
Paper31
Single
Invisible
Hardware
PublishedBy
Attack34
Attack Vectors
Attacks
Attacker knowledge
Trigger
Access impacts
Security violation
Model architecture
Papers
Trojans
White box
Existing
Logic corruption
Availability
DNN
Paper32
Hardware
Single
PublishedBy
Attack35
Attacker knowledge
Attacks
Security violation
Trigger
Access impacts
Attack Vectors
Model architecture
Papers
White box
Integrity
Confidentiality
Existing
Visibility
Logic corruption
Trojans
DNN
Paper33
Single
Invisible
Software
PublishedBy
Attack36
Attacker knowledge
Attacks
Access impacts
Security violation
Model architecture
Trigger
Attack Vectors
Papers
White box
Data injection
Integrity
CNN
Existing
Poisoning
Paper34
Non
Data Poisoning
Clean label
PublishedBy
Attack37
Attacker knowledge
Access impacts
Attack Vectors
Security violation
Attacks
Trigger
Model architecture
Papers
Gray box
Black box
Transfer learning
Poisoning
Integrity
Existing
CNN
Paper35
Data Poisoning
Non
Clean label
PublishedBy
Attack38
Security violation
Attacks
Model architecture
Trigger
Attacker knowledge
Access impacts
Attack Vectors
Papers
Integrity
CNN
Visibility
Existing
White box
Transfer learning
Data manipulation
Backdoors
Paper36
Visible
Single
PublishedBy
Attack39
Security violation
Model architecture
Attacker knowledge
Access impacts
Attacks
Attack Vectors
Trigger
Papers
Authorization
Confidentiality
Authentication
Availability
DNN
White box
Data injection
Backdoors
Existing
Visibility
Paper37
Single
Invisible
PublishedBy
Attack40
Security violation
Access impacts
Trigger
Attack Vectors
Model architecture
Attacks
Papers
Availability
Integrity
Logic corruption
Existing
Visibility
Trojans
DNN
Paper38
Single
Multiple
Visible
Hardware
PublishedBy
Attack41

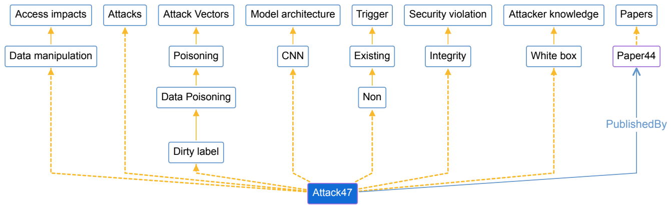


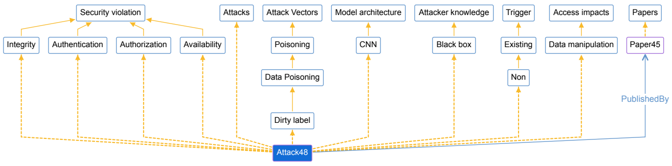


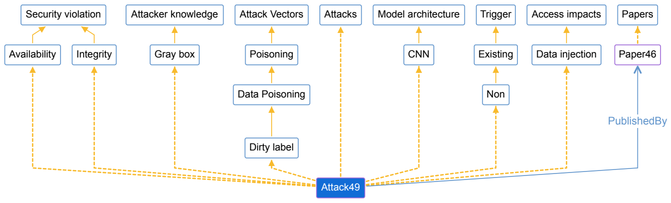


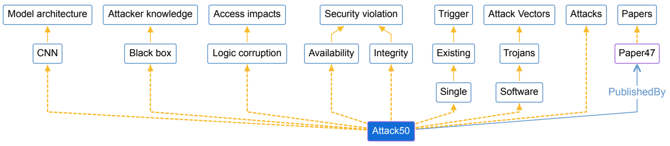


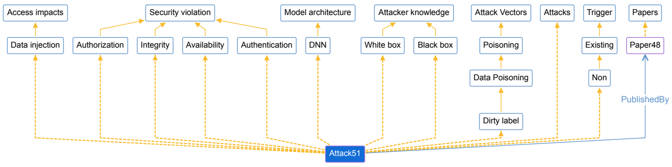


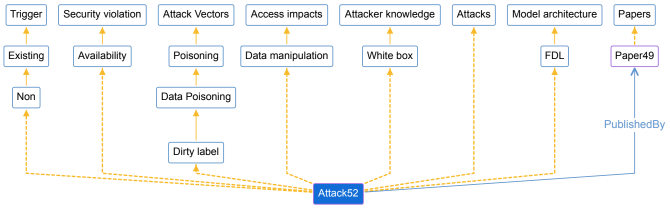


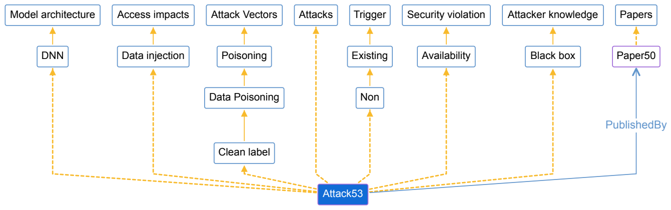


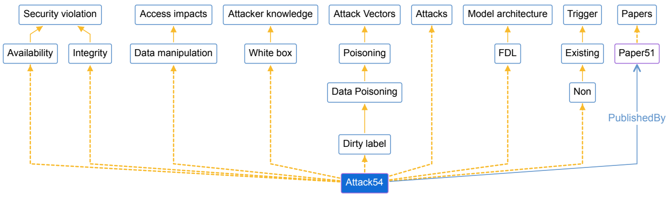


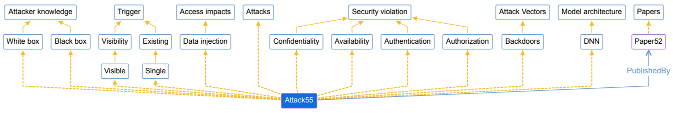












References

1. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. *arXiv e-prints* **2013**, p. arXiv:1311.2524, [arXiv:cs.CV/1311.2524].
2. Zhao, Z.Q.; Zheng, P.; Xu, S.t.; Wu, X. Object Detection with Deep Learning: A Review. *arXiv e-prints* **2018**, p. arXiv:1807.05511, [arXiv:cs.CV/1807.05511].
3. Mohammed, T.; Albeshri, A.; Katib, I.; Mehmood, R. DIESEL: A Novel Deep Learning-Based Tool for SpMV Computations and Solving Sparse Linear Equation Systems. *J. Supercomput.* **2021**, *77*, 6313–6355. https://doi.org/10.1007/s11227-020-03489-3.
4. Aqib, M.; Mehmood, R.; Alzahrani, A.; Katib, I.; Albeshri, A.; Altowaijri, S. Rapid Transit Systems: Smarter Urban Planning Using Big Data, In-Memory Computing, Deep Learning, and GPUs. *Sustainability* **2019**, *11*, 2736.

5. Fathi, E.; Maleki Shoja, B. Chapter 9 - Deep Neural Networks for Natural Language Processing. In *Computational Analysis and Understanding of Natural Languages: Principles, Methods and Applications*; Gudivada, V.N.; Rao, C., Eds.; Elsevier, 2018; Vol. 38, *Handbook of Statistics*, pp. 229–316. https://doi.org/https://doi.org/10.1016/bs.host.2018.07.006.
6. Ahmad, I.; Alqurashi, F.; Abozinadah, E.; Mehmood, R. Deep Journalism and DeepJournal V1.0: A Data-Driven Deep Learning Approach to Discover Parameters for Transportation. *Sustainability* **2022**, *14*. https://doi.org/10.3390/su14095711.
7. Alkhayat, G.; Mehmood, R. A review and taxonomy of wind and solar energy forecasting methods based on deep learning. *Energy and AI* **2021**, *4*, 100060. https://doi.org/https://doi.org/10.1016/j.egyai.2021.100060.
8. Piccialli, F.; Somma, V.D.; Giampaolo, F.; Cuomo, S.; Fortino, G. A survey on deep learning in medicine: Why, how and when? *Information Fusion* **2021**, *66*, 111–137. https://doi.org/https://doi.org/10.1016/j.inffus.2020.09.006.
9. Janbi, N.; Mehmood, R.; Katib, I.; Albeshri, A.; Corchado, J.M.; Yigitcanlar, T. Imtidad: A Reference Architecture and a Case Study on Developing Distributed AI Services for Skin Disease Diagnosis over Cloud, Fog and Edge. *Sensors* **2022**, *22*, 1854.
10. Yigitcanlar, T.; Butler, L.; Windle, E.; Desouza, K.C.; Mehmood, R.; Corchado, J.M. Can Building "Artificially Intelligent Cities" Safeguard Humanity from Natural Disasters, Pandemics, and Other Catastrophes? An Urban Scholar's Perspective. *Sensors* **2020**, *20*, 2988. https://doi.org/10.3390/s20102988.
11. Alotaibi, H.; Alsolami, F.; Abozinadah, E.; Mehmood, R. TAWSEEM: A Deep-Learning-Based Tool for Estimating the Number of Unknown Contributors in DNA Profiling. *Electronics* **2022**, *11*. https://doi.org/10.3390/electronics11040548.
12. Pawlicki, M.; Kozik, R.; Choraś, M. A survey on neural networks for (cyber-) security and (cyber-) security of neural networks. *Neurocomputing* **2022**, *500*, 1075–1087. https://doi.org/https://doi.org/10.1016/j.neucom.2022.06.002.
13. Muhammed, T.; Mehmood, R.; Albeshri, A.; Katib, I. UbeHealth: A Personalized Ubiquitous Cloud and Edge-Enabled Networked Healthcare System for Smart Cities. *IEEE Access* **2018**, *6*, 32258–32285. https://doi.org/10.1109/ACCESS.2018.2846609.
14. Mohammed, T.; Albeshri, A.; Katib, I.; Mehmood, R. UbiPriSEQ—Deep Reinforcement Learning to Manage Privacy, Security, Energy, and QoS in 5G IoT HetNets. *Applied Sciences 2020, Vol. 10, Page 7120* **2020**, *10*, 7120. https://doi.org/10.3390/APP10207120.
15. Yigitcanlar, T.; Corchado, J.M.; Mehmood, R.; Li, R.Y.M.; Mossberger, K.; Desouza, K. Responsible urban innovation with local government artificial intelligence (Ai): A conceptual framework and research agenda. *Journal of Open Innovation: Technology, Market, and Complexity* **2021**, *7*. https://doi.org/10.3390/joitmc7010071.
16. Yigitcanlar, T.; Mehmood, R.; Corchado, J.M. Green Artificial Intelligence: Towards an Efficient, Sustainable and Equitable Technology for Smart Cities and Futures. *Sustainability* **2021**, *13*. https://doi.org/10.3390/su13168952.
17. Kitchenham, B.; Brereton, P. A systematic review of systematic review process research in software engineering. *Information and Software Technology* **2013**, *55*, 2049–2075. https://doi.org/https://doi.org/10.1016/j.infsof.2013.07.010.
18. Ouzzani, M.; Hammady, H.; Fedorowicz, Z.; Elmagarmid, A. Rayyan—a web and mobile app for systematic reviews. *Systematic Reviews* **2016**, *5*, 210. https://doi.org/10.1186/s13643-016-0384-4.
19. van de Schoot, R.; de Bruin, J.; Schram, R.; Zahedi, P.; de Boer, J.; Weijdema, F.; Kramer, B.; Huijts, M.; Hoogerwerf, M.; Ferdinands, G.; et al. ASReview: Open Source Software for Efficient and Transparent Active Learning for Systematic Reviews. *CoRR* **2020**, *abs/2006.12166*, [2006.12166].
20. Musen, M.A. The protégé project: a look back and a look forward. *AI Matters* **2015**, *1*, 4–12. https://doi.org/10.1145/2757001.2757003.
21. . Neo4j - Graph Data Platform. https://neo4j.com/, accessed on 14.05.2021.
22. Jesús Barrasa. Neosemantics - a plugin that enables the use of RDF in Neo4j. https://github.com/neo4j-labs/neosemantics, accessed on 15.03.2022.
23. Pitropakis, N.; Panaousis, E.; Giannetsos, T.; Anastasiadis, E.; Loukas, G. A taxonomy and survey of attacks against machine learning. *Comput. Sci. Rev.* **2019**, *34*, 100199. https://doi.org/10.1016/j.cosrev.2019.100199.
24. Dang, T.K.; Truong, P.T.T.; Tran, P.T. Data Poisoning Attack on Deep Neural Network and Some Defense Methods. IEEE, 2020, pp. 15–22. https://doi.org/10.1109/ACOMP50827.2020.00010.
25. Jere, M.S.; Farnan, T.; Koushanfar, F. A Taxonomy of Attacks on Federated Learning. *IEEE Security & Privacy* **2021**, *19*, 20–28. https://doi.org/10.1109/MSEC.2020.3039941.
26. Isakov, M.; Gadepally, V.; Gettings, K.M.; Kinsy, M.A. Survey of Attacks and Defenses on Edge-Deployed Neural Networks. IEEE, 2019, pp. 1–8. https://doi.org/10.1109/HPEC.2019.8916519.
27. Chen, D.; Yu, N.; Zhang, Y.; Fritz, M. GAN-Leaks: A Taxonomy of Membership Inference Attacks against Generative Models. ACM, 2020, pp. 343–362. https://doi.org/10.1145/3372297.3417238.
28. Liu, Y.; Mondal, A.; Chakraborty, A.; Zuzak, M.; Jacobsen, N.; Xing, D.; Srivastava, A. A Survey on Neural Trojans. *Proc. - Int. Symp. Qual. Electron. Des. ISQED* **2020**, *2020-March*, 33–39. https://doi.org/10.1109/ISQED48828.2020.9137011.
29. Xu, Q.; Arafin, M.T.; Qu, G. Security of Neural Networks from Hardware Perspective. ACM, 2021, pp. 449–454. https://doi.org/10.1145/3394885.3431639.
30. Xue, M.; Yuan, C.; Wu, H.; Zhang, Y.; Liu, W. Machine Learning Security: Threats, Countermeasures, and Evaluations. *IEEE Access* **2020**, *8*, 74720–74742. https://doi.org/10.1109/ACCESS.2020.2987435.
31. He, Y.; Meng, G.; Chen, K.; Hu, X.; He, J. Towards Security Threats of Deep Learning Systems: A Survey. *IEEE Trans. Softw. Eng.* **2020**, [1911.12562]. https://doi.org/10.1109/TSE.2020.3034721.

32. J, Z.M.X.K.G.D. Adversarial Learning Targeting Deep Neural Network Classification: A Comprehensive Review of Defenses Against Attacks. *Proceedings of the IEEE* **2020**, *108*, 402–433.
33. Chen, Y.; Gong, X.; Wang, Q.; Di, X.; Huang, H. Backdoor Attacks and Defenses for Deep Neural Networks in Outsourced Cloud Environments. *IEEE Network* **2020**, *34*, 141–147. https://doi.org/10.1109/MNET.011.1900577.
34. Li, S.; Xue, M.; Zhao, B.; Zhu, H.; Zhang, X. Invisible Backdoor Attacks on Deep Neural Networks via Steganography and Regularization. *IEEE Trans. Dependable Secur. Comput.* **2020**, pp. 1–1, [1909.02742]. https://doi.org/10.1109/tdsc.2020.3021407.
35. Pitropakis, N.; Panaousis, E.; Giannetsos, T.; Anastasiadis, E.; Loukas, G. A taxonomy and survey of attacks against machine learning. *Comput. Sci. Rev.* **2019**, *34*, 100199. https://doi.org/10.1016/j.cosrev.2019.100199.
36. Lee, D.; Kim, H.; Ryou, J. Poisoning attack on show and tell model and defense using autoencoder in electric factory. In Proceedings of the Proc. - 2020 IEEE Int. Conf. Big Data Smart Comput. BigComp 2020. IEEE, 2020, pp. 538–541. https://doi.org/10.1109/BigComp48618.2020.000-9.
37. Dumford, J.; Scheirer, W. Backdooring convolutional neural networks via targeted weight perturbations. In Proceedings of the IJCB 2020 - IEEE/IAPR Int. Jt. Conf. Biometrics. IEEE, 2020, pp. 1–9, [1812.03128]. https://doi.org/10.1109/IJCB48548.2020.9304875.
38. Bhagoji, A.N.; Chakraborty, S.; Mittal, P.; Calo, S. Analyzing federated learning through an adversarial lens. In Proceedings of the 36th Int. Conf. Mach. Learn. ICML 2019, 2019, Vol. 2019-June, pp. 1012–1021, [1811.12470].
39. Zhou, X.; Xu, M.; Wu, Y.; Zheng, N. Deep Model Poisoning Attack on Federated Learning. *Futur. Internet* **2021**, *13*, 73. https://doi.org/10.3390/fi13030073.
40. Davaslioglu, K.; Sagduyu, Y.E. Trojan Attacks on Wireless Signal Classification with Adversarial Machine Learning. In Proceedings of the 2019 IEEE Int. Symp. Dyn. Spectr. Access Networks, DySPAN 2019. IEEE, 2019, pp. 1–6, [1910.10766]. https://doi.org/10.1109/DySPAN.2019.8935782.
41. Zhong, H.; Liao, C.; Squicciarini, A.C.; Zhu, S.; Miller, D. Backdoor Embedding in Convolutional Neural Network Models via Invisible Perturbation. *CODASPY 2020 - Proc. 10th ACM Conf. Data Appl. Secur. Priv.* **2020**, pp. 97–108, [1808.10307]. https://doi.org/10.1145/3374664.3375751.
42. Huai, M.; Sun, J.; Cai, R.; Yao, L.; Zhang, A. Malicious Attacks against Deep Reinforcement Learning Interpretations. In Proceedings of the Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.; ACM: New York, NY, USA, 2020; pp. 472–482. https://doi.org/10.1145/3394486.3403089.
43. Liu, K.; Tan, B.; Karri, R.; Garg, S. Poisoning the (Data) Well in ML-Based CAD: A Case Study of Hiding Lithographic Hotspots. In Proceedings of the 2020 Des. Autom. Test Eur. Conf. Exhib. IEEE, 2020, pp. 306–309. https://doi.org/10.23919/DATE48585.2020.9116489.
44. Hu, X.; Zhao, Y.; Deng, L.; Liang, L.; Zuo, P.; Ye, J.; Lin, Y.; Xie, Y. Practical Attacks on Deep Neural Networks by Memory Trojaning. *IEEE Trans. Comput. Des. Integr. Circuits Syst.* **2021**, *40*, 1230–1243. https://doi.org/10.1109/TCAD.2020.2995347.
45. Lin, J.; Xu, L.; Liu, Y.; Zhang, X. Composite Backdoor Attack for Deep Neural Network by Mixing Existing Benign Features. In Proceedings of the Proc. 2020 ACM SIGSAC Conf. Comput. Commun. Secur.; ACM: New York, NY, USA, 2020; pp. 113–131. https://doi.org/10.1145/3372297.3423362.
46. Dai, J.; Chen, C.; Li, Y. A backdoor attack against LSTM-based text classification systems. *IEEE Access* **2019**, *7*, 138872–138878, [1905.12457]. https://doi.org/10.1109/ACCESS.2019.2941376.
47. Zhao, S.; Ma, X.; Zheng, X.; Bailey, J.; Chen, J.; Jiang, Y.G. Clean-Label Backdoor Attacks on Video Recognition Models. In Proceedings of the Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. IEEE, 2020, pp. 14431–14440, [2003.03030]. https://doi.org/10.1109/CVPR42600.2020.01445.
48. Liu, Y.; Ma, X.; Bailey, J.; Lu, F. Reflection Backdoor: A Natural Backdoor Attack on Deep Neural Networks. In Proceedings of the Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), 2020, Vol. 12355 LNCS, pp. 182–199, [2007.02343]. https://doi.org/10.1007/978-3-030-58607-2_11.
49. Chen, J.; Zhang, L.; Zheng, H.; Xuan, Q. SPA: Stealthy Poisoning Attack. In Proceedings of the ACM Int. Conf. Proceeding Ser.; ACM: New York, NY, USA, 2020; pp. 303–309. https://doi.org/10.1145/3444370.3444589.
50. Tan, T.J.L.; Shokri, R. Bypassing Backdoor Detection Algorithms in Deep Learning. In Proceedings of the Proc. - 5th IEEE Eur. Symp. Secur. Privacy, Euro S P 2020. IEEE, 2020, pp. 175–183, [1905.13409]. https://doi.org/10.1109/EuroSP48549.2020.00019.
51. Tang, R.; Du, M.; Liu, N.; Yang, F.; Hu, X. An Embarrassingly Simple Approach for Trojan Attack in Deep Neural Networks. In Proceedings of the Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.; ACM: New York, NY, USA, 2020; pp. 218–228. https://doi.org/10.1145/3394486.3403064.
52. Wu, J.; Lin, X.; Lin, Z.; Tang, Y. A security concern about deep learning models. In Proceedings of the Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), 2018, Vol. 11287 LNCS, pp. 199–206. https://doi.org/10.1007/978-3-030-03026-1_15.
53. Barni, M.; Kallas, K.; Tondi, B. A New Backdoor Attack in CNNS by Training Set Corruption Without Label Poisoning. In Proceedings of the 2019 IEEE Int. Conf. Image Process. IEEE, 2019, pp. 101–105. https://doi.org/10.1109/ICIP.2019.8802997.
54. Xiong, Y.; Xu, F.; Zhong, S.; Li, Q. Escaping Backdoor Attack Detection of Deep Learning. In Proceedings of the IFIP Adv. Inf. Commun. Technol., 2020, Vol. 580 IFIP, pp. 431–445. https://doi.org/10.1007/978-3-030-58201-2_29.
55. Kwon, H.; Roh, J.; Yoon, H.; Park, K.W. TargetNet Backdoor: Attack on Deep Neural Network with Use of Different Triggers. In Proceedings of the ACM Int. Conf. Proceeding Ser.; ACM: New York, NY, USA, 2020; pp. 140–145. https://doi.org/10.1145/3385209.3385216.

56. Chen, J.; Zheng, H.; Su, M.; Du, T.; Lin, C.; Ji, S. Invisible Poisoning: Highly Stealthy Targeted Poisoning Attack. In Proceedings of the Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), 2020, Vol. 12020 LNCS, pp. 173–198. https://doi.org/10.1007/978-3-030-42921-8_10.
57. Chen, J.; Zheng, H.; Su, M.; Du, T.; Lin, C.; Ji, S. Invisible Poisoning: Highly Stealthy Targeted Poisoning Attack. In Proceedings of the Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), 2020, Vol. 12020 LNCS, pp. 173–198. https://doi.org/10.1007/978-3-030-42921-8_10.
58. Xue, M.; He, C.; Wang, J.; Liu, W. Backdoors hidden in facial features: a novel invisible backdoor attack against face recognition systems. *Peer-to-Peer Netw. Appl.* **2021**, *14*, 1458–1474. https://doi.org/10.1007/s12083-020-01031-z.
59. Yao, Y.; Zheng, H.; Li, H.; Zhao, B.Y. Latent backdoor attacks on deep neural networks. In Proceedings of the Proc. ACM Conf. Comput. Commun. Secur.; ACM: New York, NY, USA, 2019; pp. 2041–2055. https://doi.org/10.1145/3319535.3354209.
60. Quiring, E.; Rieck, K. Backdooring and poisoning neural networks with image-scaling attacks. In Proceedings of the Proc. - 2020 IEEE Symp. Secur. Priv. Work. SPW 2020. IEEE, 2020, pp. 41–47, [2003.08633]. https://doi.org/10.1109/SPW50608.2020.00024.
61. Bhalerao, A.; Kallas, K.; Tondi, B.; Barni, M. Luminance-based video backdoor attack against anti-spoofing rebroadcast detection. In Proceedings of the 2019 IEEE 21st Int. Work. Multimed. Signal Process. IEEE, 2019, pp. 1–6. https://doi.org/10.1109/MMSP.2019.8901711.
62. Costales, R.; Mao, C.; Norwitz, R.; Kim, B.; Yang, J. Live Trojan Attacks on Deep Neural Networks. In Proceedings of the 2020 IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Work. IEEE, 2020, pp. 3460–3469. https://doi.org/10.1109/CVPRW50498.2020.00406.
63. KWON, H.; YOON, H.; PARK, K.W. Multi-Targeted Backdoor: Indentifying Backdoor Attack for Multiple Deep Neural Networks. *IEICE Trans. Inf. Syst.* **2020**, *E103.D*, 883–887. https://doi.org/10.1587/transinf.2019EDL8170.
64. Liu, Z.; Ye, J.; Hu, X.; Li, H.; Li, X.; Hu, Y. Sequence Triggered Hardware Trojan in Neural Network Accelerator. In Proceedings of the 2020 IEEE 38th VLSI Test Symp. IEEE, 2020, pp. 1–6. https://doi.org/10.1109/VTS48691.2020.9107582.
65. Rakin, A.S.; He, Z.; Fan, D. TBT: Targeted Neural Network Attack With Bit Trojan. In Proceedings of the 2020 IEEE/CVF Conf. Comput. Vis. Pattern Recognit. IEEE, 2020, pp. 13195–13204. https://doi.org/10.1109/CVPR42600.2020.01321.
66. Liu, T.; Wen, W.; Jin, Y. SIN2: Stealth infection on neural network — A low-cost agile neural Trojan attack methodology. In Proceedings of the 2018 IEEE Int. Symp. Hardw. Oriented Secur. Trust. IEEE, 2018, pp. 227–230. https://doi.org/10.1109/HST.2018.8383920.
67. Zhou, Q.; Ren, Y.; Xia, T.; Yuan, L.; Chen, L. Data Poisoning Attacks on Graph Convolutional Matrix Completion; 2020; Vol. 11945, pp. 427–439. https://doi.org/10.1007/978-3-030-38961-1_38.
68. Zhu, C.; Ronny Huang, W.; Shafahi, A.; Li, H.; Taylor, G.; Studer, C.; Goldstein, T. Transferable clean-label poisoning attacks on deep neural nets. In Proceedings of the 36th Int. Conf. Mach. Learn. International Machine Learning Society (IMLS), 2019, Vol. 2019, pp. 13141–13154.
69. Gu, T.; Liu, K.; Dolan-Gavitt, B.; Garg, S. BadNets: Evaluating Backdooring Attacks on Deep Neural Networks. *IEEE Access* **2019**, *7*, 47230–47244. https://doi.org/10.1109/ACCESS.2019.2909068.
70. Guo, W.; Tondi, B.; Barni, M. A Master Key backdoor for universal impersonation attack against DNN-based face verification. *Pattern Recognit. Lett.* **2021**, *144*, 61–67. https://doi.org/10.1016/j.patrec.2021.01.009.
71. Clements, J.; Lao, Y. Hardware Trojan Design on Neural Networks. In Proceedings of the 2019 IEEE Int. Symp. Circuits Syst. IEEE, 2019, pp. 1–5. https://doi.org/10.1109/ISCAS.2019.8702493.
72. Muñoz-González, L.; Biggio, B.; Demontis, A.; Paudice, A.; Wongrassamee, V.; Lupu, E.C.; Roli, F. Towards poisoning of deep learning algorithms with back-gradient optimization. *AISec 2017 - Proc. 10th ACM Work. Artif. Intell. Secur. co-located with CCS 2017* **2017**, pp. 27–38, [1708.08689]. https://doi.org/10.1145/3128572.3140451.
73. Li, M.; Sun, Y.; Lu, H.; Maharjan, S.; Tian, Z. Deep Reinforcement Learning for Partially Observable Data Poisoning Attack in Crowdsensing Systems. *IEEE Internet Things J.* **2020**, *7*, 6266–6278. https://doi.org/10.1109/JIOT.2019.2962914.
74. Li, W.; Yu, J.; Ning, X.; Wang, P.; Wei, Q.; Wang, Y.; Yang, H. Hu-Fu: Hardware and Software Collaborative Attack Framework Against Neural Networks. In Proceedings of the 2018 IEEE Comput. Soc. Annu. Symp. VLSI. IEEE, 2018, pp. 482–487. https://doi.org/10.1109/ISVLSI.2018.00093.
75. Xu, H.; Ma, Y.; Liu, H.C.; Deb, D.; Liu, H.; Tang, J.L.; Jain, A.K. Adversarial Attacks and Defenses in Images, Graphs and Text: A Review. *International Journal of Automation and Computing* **2020**, *17*, 151–178. https://doi.org/10.1007/s11633-019-1211-x.
76. Venceslai, V.; Marchisio, A.; Alouani, I.; Martina, M.; Shafique, M. NeuroAttack: Undermining Spiking Neural Networks Security through Externally Triggered Bit-Flips. In Proceedings of the 2020 Int. Jt. Conf. Neural Networks. IEEE, 2020, pp. 1–8. https://doi.org/10.1109/IJCNN48605.2020.9207351.
77. Kwon, H.; Yoon, H.; Park, K.W. Selective Poisoning Attack on Deep Neural Network to Induce Fine-Grained Recognition Error. In Proceedings of the 2019 IEEE Second Int. Conf. Artif. Intell. Knowl. Eng. IEEE, 2019, pp. 136–139. https://doi.org/10.1109/AIKE.2019.00033.
78. Cole, D.; Newman, S.; Lin, D. A New Facial Authentication Pitfall and Remedy in Web Services. *IEEE Trans. Dependable Secur. Comput.* **2021**, pp. 1–1. https://doi.org/10.1109/TDSC.2021.3067794.
79. Zeng, Y.; Qiu, M.; Niu, J.; Long, Y.; Xiong, J.; Liu, M. V-PSC: A Perturbation-Based Causative Attack Against DL Classifiers' Supply Chain in VANET. In Proceedings of the 2019 IEEE Int. Conf. Comput. Sci. Eng. IEEE Int. Conf. Embed. Ubiquitous Comput. IEEE, 2019, pp. 86–91. https://doi.org/10.1109/CSE/EUC.2019.00026.

80. Pan, J. Blackbox Trojanising of Deep Learning Models: Using Non-Intrusive Network Structure and Binary Alterations. In Proceedings of the 2020 IEEE Reg. 10 Conf. IEEE, 2020, pp. 891–896. https://doi.org/10.1109/TENCON50793.2020.9293933.
81. Garofalo, G.; Rimmer, V.; van Hamme, T.; Preuveneers, D.; Joosen, W. Fishy faces: Crafting adversarial images to poison face authentication. In Proceedings of the 12th USENIX Work. Offensive Technol., 2018.
82. Tolpegin, V.; Truex, S.; Gursoy, M.E.; Liu, L. Data Poisoning Attacks Against Federated Learning Systems; 2020; Vol. 12308, pp. 480–501. https://doi.org/10.1007/978-3-030-58951-6_24.
83. Li, P.; Zhao, W.; Liu, Q.; Liu, X.; Yu, L. Poisoning Machine Learning Based Wireless IDSs via Stealing Learning Model; 2018; Vol. 10874, pp. 261–273. https://doi.org/10.1007/978-3-319-94268-1_22.
84. Zhang, J.; Chen, J.; Wu, D.; Chen, B.; Yu, S. Poisoning Attack in Federated Learning using Generative Adversarial Nets. In Proceedings of the 2019 18th IEEE Int. Conf. Trust. Secur. Priv. Comput. Commun. IEEE Int. Conf. Big Data Sci. Eng. IEEE, 2019, pp. 374–380. https://doi.org/10.1109/TrustCom/BigDataSE.2019.00057.
85. Lovisotto, G.; Eberz, S.; Martinovic, I. Biometric Backdoors: A Poisoning Attack Against Unsupervised Template Updating. In Proceedings of the 2020 IEEE Eur. Symp. Secur. Priv. IEEE, 2020, pp. 184–197. https://doi.org/10.1109/EuroSP48549.2020.00020.
86. Fofanov, G.A. Problems of Neural Networks Training. In Proceedings of the 2018 19th International Conference of Young Specialists on Micro/Nanotechnologies and Electron Devices (EDM), 2018, pp. 6403–6405. https://doi.org/10.1109/EDM.2018.8434935.
87. Lecun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE* **1998**, *86*, 2278–2324. https://doi.org/10.1109/5.726791.
88. Elman, J.L. Finding Structure in Time. *Cognitive Science* **1990**, *14*, 179–211, [https://onlinelibrary.wiley.com/doi/pdf/10.1207/s15516709cog1 https://doi.org/https://doi.org/10.1207/s15516709cog1402_1.
89. Goodfellow, I.J.; Shlens, J.; Szegedy, C. Explaining and harnessing adversarial examples. *3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc.* **2015**, pp. 1–11, [1412.6572].
90. Radford, A.; Metz, L.; Chintala, S. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. *arXiv e-prints* **2015**, p. arXiv:1511.06434, [arXiv:cs.LG/1511.06434].
91. Sutton, R.; Barto, A. Reinforcement Learning: An Introduction. *IEEE Transactions on Neural Networks* **1998**, *9*, 1054–1054. https://doi.org/10.1109/TNN.1998.712192.
92. Maass, W. Networks of spiking neurons: The third generation of neural network models. *Neural Networks* **1997**, *10*, 1659–1671. https://doi.org/https://doi.org/10.1016/S0893-6080(97)00011-7.
93. Brendan McMahan, H.; Moore, E.; Ramage, D.; Hampson, S.; Agüera y Arcas, B. Communication-efficient learning of deep networks from decentralized data. *Proc. 20th Int. Conf. Artif. Intell. Stat. AISTATS 2017* **2017**, *54*, [1602.05629].
94. Ji, Y.; Zhang, X.; Ji, S.; Luo, X.; Wang, T. Model-reuse attacks on deep learning systems. In Proceedings of the Proc. ACM Conf. Comput. Commun. Secur. ACM, 2018, Vol. 15, pp. 349–363, [1812.00483]. https://doi.org/10.1145/3243734.3243757.
95. Chen, X.; Liu, C.; Li, B.; Lu, K.; Song, D. Targeted Backdoor Attacks on Deep Learning Systems Using Data Poisoning. *arXiv1712.05526 [cs]* **2017**, [1712.05526].
96. Liu, Y.; Xie, Y.; Srivastava, A. Neural trojans. *Proc. - 35th IEEE Int. Conf. Comput. Des. ICCD 2017* **2017**, pp. 45–48, [1710.00942]. https://doi.org/10.1109/ICCD.2017.16.
97. Zou, M.; Shi, Y.; Wang, C.; Li, F.; Song, W.Z.; Wang, Y. PoTrojan: Powerful neuron-level trojan designs in deep learning models, 2018, [1802.03043].
98. Liu, T.; Wen, W.; Jin, Y. SIN2: Stealth infection on neural network - A low-cost agile neural Trojan attack methodology. *Proc. 2018 IEEE Int. Symp. Hardw. Oriented Secur. Trust. HOST 2018* **2018**, pp. 227–230. https://doi.org/10.1109/HST.2018.8383920.
99. . SemSpect - Scalable Graph Exploration Tool for Neo4j. https://www.semspect.de/, accessed on 20.07.2022.