Hindawi
Mobile Information Systems
Volume 2020, Article ID 6535834, 18 pages
https://doi.org/10.1155/2020/6535834

Review Article

A Review of Deep Learning Security and Privacy Defensive Techniques

Muhammad Imran Tariq,[1] **Nisar Ahmed Memon**,[2] **Shakeel Ahmed**,[2] **Shahzadi Tayyaba**,[3] **Muhammad Tahir Mushtaq**,[4] **Natash Ali Mian**,[5] **Muhammad Imran**,[6] **and Muhammad W. Ashraf**[6]

[1]*Department of Computer Science, Superior University, Lahore, Pakistan*
[2]*College of Computer Science and Information Technology (CCSIT), King Faisal University, Al-Ahsa, Saudi Arabia*
[3]*Department of Computer Engineering, The University of Lahore, Lahore, Pakistan*
[4]*School of Systems and Technology, The University of Management and Technology (UMT), Lahore, Pakistan*
[5]*School of Computer and Information Technology, Beaconhouse National University, Lahore, Pakistan*
[6]*Department of Physics (Electronics), Government College University, Lahore, Pakistan*

Correspondence should be addressed to Muhammad Imran Tariq; imrantariqbutt@yahoo.com

Received 21 November 2019; Revised 24 January 2020; Accepted 12 February 2020; Published 7 April 2020

Guest Editor: Fawad Zaman

In recent past years, Deep Learning presented an excellent performance in different areas like image recognition, pattern matching, and even in cybersecurity. The Deep Learning has numerous advantages including fast solving complex problems, huge automation, maximum application of unstructured data, ability to give high quality of results, reduction of high costs, no need for data labeling, and identification of complex interactions, but it also has limitations like opaqueness, computationally intensive, need for abundant data, and more complex algorithms. In our daily life, we used many applications that use Deep Learning models to make decisions based on predictions, and if Deep Learning models became the cause of misprediction due to internal/external malicious effects, it may create difficulties in our real life. Furthermore, the Deep Learning training models often have sensitive information of the users and those models should not be vulnerable and expose security and privacy. The algorithms of Deep Learning and machine learning are still vulnerable to different types of security threats and risks. Therefore, it is necessary to call the attention of the industry in respect of security threats and related countermeasures techniques for Deep Learning, which motivated the authors to perform a comprehensive survey of Deep Learning security and privacy security challenges and countermeasures in this paper. We also discussed the open challenges and current issues.

1. Introduction

Deep Learning is also called hierarchical learning and deep-structured learning, and it is comprised of supervised or unsupervised machine learning techniques. The idea of Deep Learning derived from the structure and functionality of the human brain and also the processing of signals through neurons in the human mind. Deep Learning is also taking the benefits of artificial neural networks, and it also consists of input, output, and many hidden layers. Each layer of Deep Learning relies upon the nonlinear response based on the data provided through the input layer. For the last few years, the Deep Learning technique has been mostly and widely used in the signal processing of voice recognition, graphic recognition, discovery of the thing, and so numerous other areas, such as the discovery of the medicine for diseases and genomics [1]. Deep Learning developed a structure to deal with big data sets through a backpropagation algorithm to highlight in what way the device changes its core parameters that are being opted to calculate the representation in each rendering layer in the previous layer [2].

Despite their enormous size, successful Deep Neural Networks can make a very minor difference between training and test presentation. Traditional wisdom attributes the error of small circularization to the typical characteristics of the family or to the organizational techniques used during training [3].

The crucial problem of the DL is its encrypted data that flows from training and interface modules. The security and privacy issues are very important due to mostly adopted DL models in many applications as mentioned above. Further, actually Deep Learning prevailing in all models for training part relies upon a huge number of big data, sensitive, and confidential data of the user particularly training data. Keeping this in view, DL models must not disclose confidential and sensitive data. In this paper, systematic literature reviewed was conducted about the Deep Learning security threats, privacy threats regarding private data, and their corresponding developed defense techniques. The paper also included most secured techniques that use cryptographic primitives without the indulgence of the third party and the summary of the future challenges and opportunities.

1.1. Application of Deep Learning. Deep learning has introduced new ways to look at technologies. Artificial Intelligence (AIT) and its branches ML and Deep Learning have a lot of excitements. It is a reality that Deep Learning changed the ways of living and will also affect life in the near future. DL is grabbing market space day by day and we are sure, in coming five to ten years, the tools, techniques, and libraries of DL will include in every development toolkit.

Here, we will discuss the Deep Learning applications that captured the marked in 2019 and beyond.

1.1.1. Self-Driving Car. Many of the car manufacturing companies have built self-driving cars with the help of digital sensor systems. It is accomplished through training algorithms through the huge unstructured amount of data.

1.1.2. DL in Healthcare. Deep learning is also used to bring improvement in the field of Healthcare especially in breast cancer diagnostics and monitoring apps. It is also used to predict personalized medicine keeping in view the Biobank data. Deep learning completely reshaped the healthcare industry as well as life sciences. The key features of Deep Learning are advancing the future of health management.

1.1.3. DL in Voice Search. The most famous utilization of Deep Learning is voice recognition, searching, and activation. This facility is already available in every smartphone since 2011. Google and Apple are already offering these services, and now Microsoft Cortana has also launched a voice activation assistant.

1.1.4. Automatic Machine Translation. The google translator is the main example of the translation of one language into another language. The user entered the word, sentences, paragraphs, and phrases of one language, and it easily converts to another language. Although this facility is available for a long time, DL is getting improvement in the results with the passage of time, and now machine translation is also translating images. Image to text conversion is an example of machine translation and is the innovation of Deep Learning.

1.1.5. Automatic Handwriting Generation. Deep Learning has also played a vital role in the automatic handwriting generation. The system automatically captures the movement of the pen and the letters to learn. The DL also facilitates the generation of new writing styles.

Also, there are numerous applications of the Deep Learning that cannot be covered in one paper, and the more applications of Deep Learning are as follows:

(i) Image recolonization
(ii) Face recolonization
(iii) Automatic colorization
(iv) Image captioning
(v) Advertising
(vi) Earthquake prediction
(vii) Brain cancer detection
(viii) Price forecasting
(ix) Natural Language Processing
(x) Gamming
(xi) Cybersecurity

1.2. Innovative Contributions of Deep Learning. Deep learning has contributed to every field of science and brought innovative changes. Deep learning also uplifts every area of life by solving routine problems and also introduced new dimensions of research. The outstanding performance of Deep Learning is in the area of modern security systems. It is a very critical problem that today every small- and large-scale organization is facing; millions of new malware and virus threats are created, and large organizations like banks and government institutions are attacked by finding grey areas in the tools. Although many security solutions exist, security is an ongoing area in research. Deep learning presented new dimensions in the area of cybersecurity by detecting network attacks, removing malware, identifying vulnerabilities, and securing the system.

1.3. Organization of Study. Section 2 of the paper is related to background/literature review, Section 3 discusses Deep Learning private data frameworks, and Deep Learning treats and attacks are discussed in Section 4 of the paper, and defense techniques against security issues in Deep Learning briefly explained in Section 5 of the paper. The final conclusion of the paper is also discussed in Section 6 of the paper.

2. Background

2.1. Deep Learning. Deep learning permits high computational models that consist of multiple layers of processing to learn the depiction of data at multiple levels of abstraction layers. These techniques have vastly improved the state of the art in voice recognition, visual recognition, discovery of the object, and so many other areas, such as the discovery of the medicine for diseases and genomics. Deep learning artificial neural networks regularly contain additional trainable model parameters as compared with the number of samples in which they have been trained [4]. However, some of these models show a significantly lower circular error, that is, the difference between the training error and the test error. It is certainly easy to reach normal typical structures with little circulation [5]. What then distinguishes neural networks that generalize well from those that do not? A satisfactory answer to this question will not only help make neural networks more interpretable but can also lead to a more reliable and reliable architectural design. To answer this question, the theory of statistical learning proposed several different measures of complexity capable of controlling the error of generalization. These include the VC dimension, Rademacher complexity, and uniform stability. Also, when the number of parameters is large, the theory suggests that some type of regulation is needed to guarantee a small circular error. The regulation may be implicit as with the early suspension [6].

Machine learning technology operates many sides of current society like from online research to content filtering on social networks to recommendations on e-commerce sites and are increasingly present in consumer products such as cameras and smartphones. Machine learning systems are used to identify objects in pictures, convert voice into text, relate news items, publications or products with user interests, and identify relevant search results. Increasingly, all these applications are using Deep Learning [7].

According to [8], traditional machine learning techniques have not completed the ability to manipulate natural network data in its original shape. For decades, the establishment of a machine learning system requires precise engineering and substantial experience in the field to design a feature extractor that transforms raw information into an appropriate internal representation [9].

2.2. Deep Neural Networks (DNNs). This greater use of Deep Learning creates incentives for opponents to approach Deep Neural Networks (DNNs) to impose a poor classification of inputs. For example, Deep Learning applications use image workstations to differentiate themselves from inappropriate content, textures, and images to distinguish spam from nonintrusive mail [10]. An adversary capable of formulating erroneous inputs would benefit from the evasion of detection; even today, these attacks occur in classification systems other than Deep Learning. In the real world, consider a driverless car system that uses deep learning to identify traffic signals. If a change in the "stop" marks causes the Deep Neural Networks to be incorrectly classified, the vehicle will not stop [11].

The neural network basically consists of 03 elements, one is called the input layer, which is basically the data that the user wants to analyze [12]. The second layer is actually hidden layers; it may consist of one node or maybe more than more nodes; the primary function of this node is to complete the computation in the light of the Deep Learning algorithm. The last layer is always the output layer, which calculates the result. Figure 1 illustrates the basic neural network, and Figure 2 illustrates the Deep Learning Neural Network.

For classification tasks, higher representation layers amplify important entry aspects of discrimination and suppress irrelevant differences. For example, the image comes in the form of an array of pixel values, and the features learned in the first rendering layer generally represent the presence or absence of edges in certain directions and locations in the image. The second layer usually discovers the motifs by detecting a certain arrangement of the edges, regardless of the small differences in the positions of the edges. The third layer can group shapes into larger groups that correspond to parts of familiar objects, and the following layers will discover the objects as groups of these parts.

The main feature of DL layers is that these layers are not designed by the human; actually, it has been learned from the data through a general-purpose learning procedure. Deep learning is making great progress in solving problems that have withstood the best efforts of the AI community for many years. It has proven to be very good at detecting complex structures in high-dimensional data and, therefore, is applicable to many fields of science, business, and government addition to multiply the registers in picture recognition and voice recognition; other machine learning methods have been overcome by actively predicting possible drug molecules, analyzing particle accelerator data, reconstructing cerebral circuits, and predicting the effects of mutations in noncoding DNA on gene expression and disease. Perhaps, most surprising thing is that Deep Learning has yielded very promising results for several tasks in the understanding of natural language, the classification of the particular topic, the analysis of morals, the answer to questions, and the translation of the language [13].

It is pertinent to add here that weaknesses in DL systems have recently been discovered in a big number of publications. It is very dangerous that these applications are based on a small understanding of security and privacy in DL systems [14].

Although many research studies have been published on attacks and the defense of the security and privacy of Deep Learning, they are still fragmented. Here, we review recent attempts to secure Artificial Intelligence and Private Data of Artificial Intelligence.

In order to meet the requirement for strong AI systems in information security and private data, we need to develop a take Secured Artificial Intelligence system. That secure Artificial Intelligence system should provide security

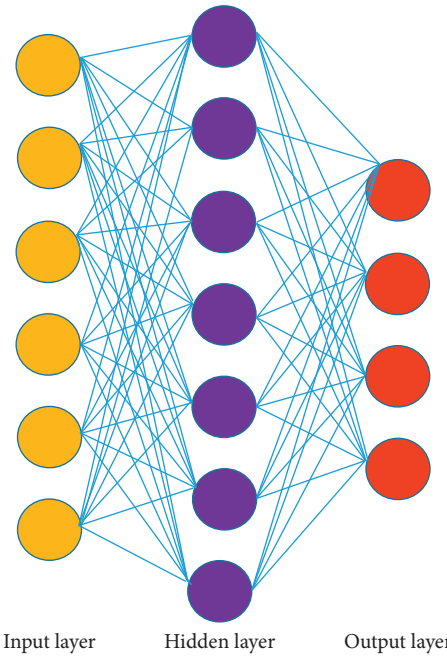

Figure 1: Basic neural network.

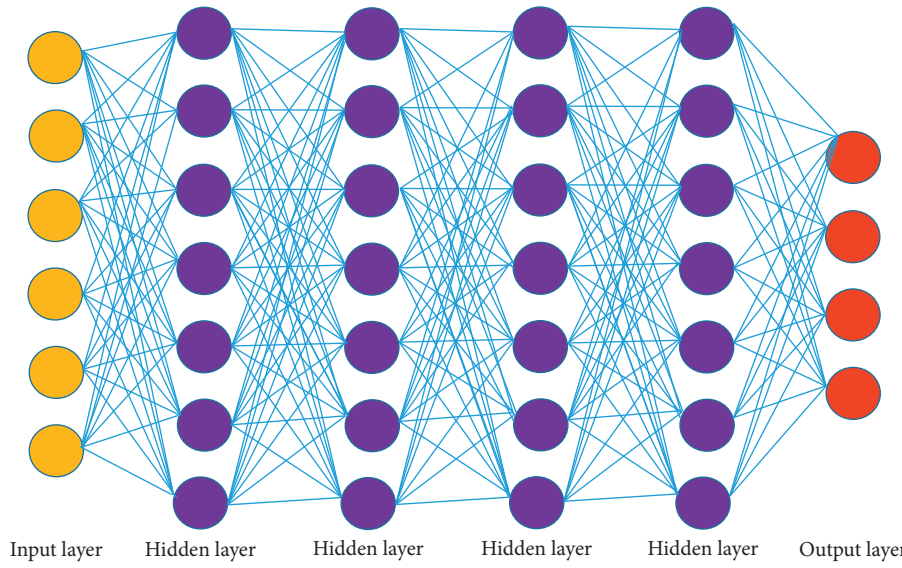

Figure 2: Deep Learning neural network.

guarantee, and Private Data Artificial Intelligence should maintain data privacy of the system [15].

The Secure Artificial Intelligence always focuses on attacks, threats, vulnerabilities, and accordingly defense of Artificial Intelligence systems, in respect of Deep Learning, which is a more effective model. The attacks on Deep Learning generate false predications by injecting wrong samples, such types of attacks are called white-box attacks, and it includes gradient-based techniques to compromise the system. In contrast, attacks from the black-box cause the suspect system to make fake predictions, without getting some information about the system. It has been observed that almost every attack exploits the predictive confidence of the system without getting information about the structure and parameters of the system [16].

In order to develop defense against these attacks, various methods have been proposed such as adversarial training, generative adversarial network, statistical approach, and recurrent neural network.

The input data of the user contains sensitive data to the Deep Learning machines for recognition. The more secure option for the user is to install the Deep Learning model on its platform and execute it and obviously; it is not feasible for the user because the Deep Learning model always consists of

massive data and it processed them [17]. Every organization desires to keep their data confidential, and their competitors may not use it for their business purposes.

The upshot, the Deep Learning machine, should meet three main requirements while preserving privacy:

(i) The data stored in the training model should not be disclosed to the cloud server

(ii) The user request should not be disclosed to the cloud server

(iii) The configurations of the cloud server should not be disclosed to the user

It is highly needed for the organizations using Deep Learning to establish privacy frameworks in which neither any intruder nor any attacker discloses information during the shared computation or modify it. In order to strengthen privacy computation in respect of Deep Learning, it is critically significant to plan new privacy-specific techniques that can minimize the complexity of secure function evaluation protocols [18].

The purpose of this research is to study the recent development of deep leering on private data and security issues attached to Deep Learning in different domains. Furthermore, we describe different types of Deep Learning possible security and privacy attacks along with different defense methods.

The core part of the Deep Neural Network is called Artificial Neuron. Artificial Neurons purely calculate the weighted amount of inputs and output, according to the following equation:

$$y = \sigma \sum_{i=1}^{n} w_i x_i, \tag{1}$$

where y is denoted as the output, x is for the input, σ is denoted as the activation function which is actually a nonlinear function, and w is called the weights. Artificial Neurons are basically used to develop construct layer (details are given in below figures), and if these layers are piled up, then it constructs DNN. The nonlinearity of the σ piles up the number of DNN layers that cultivates and allows the Deep Neural Networks to estimate the objective functions without any manmade feature selections.

2.3. Artificial Intelligence in Deep Learning. Figure 3 is a high-level group diagram of the learning process to develop a stereotype Deep Learning model. The performance of the DL model depends on the size of the existing available training data.

Nevertheless, training samples are typically gathered from the content of users stored on cloud machines that hold sensitive information, like photographs, video, sound, and location records. The privacy of the user is a major concern in Deep Learning during training and inference [19]. Internet service providing organizations are providing Deep Learning as a service where users can insert input to the cloud machines and obtain the result based on prediction.

2.4. Architectures of DNNs. The DNN model has different types of architectures that are briefly explained below.

2.4.1. Feed-Forward Neural Network (FNN). This is the fundamental and core building block of the Deep Neural Network. It consists of different types of the multiple layers, and these middle layers are completely connected with each other while the nodes within the layer are not linked to each other [20]. Figures 1 and 2 are examples of Feed-Forward Neural Network.

2.4.2. Convolutional Neural Network. This architecture is demonstrated in Figure 4. A CNN architecture consists of many convolutional and pooling layers. These layers use convolutional operations to compute and generate layerwise outcomes. The convolutional and pooling layer's operation permits the DNN network to get more knowledge about spatial. Hence, the CNN architecture shows exceptional results particularly on image applications [21, 22].

2.4.3. Recurrent Neural Network. It is extensively opted to process sequential information. As illustrated in Figure 5, the RNN calculates the output after updating the currently hidden units, past hidden units, and presently available input data [23]. The RNN also faces problems like gradient vanishing problem and long short-term memory. To solve these problems, the gated recurrent unit is used.

2.4.4. Generative Adversarial Network. This architecture of DNNs is basically comprised of two modules, one is called Discriminator (D) and the other is known as Generator (G). The Generator generates false data in the architecture while Discriminator is used in the architecture to inform whether the Generator's data are real or not? as illustrated in Figure 6. The Generator and Discriminator are usually used in DNNs, and it has many types of structures based upon the application of the network [24]. Generative Adversarial Networks are opted by many fields like image processing, voice recognizing, and domain adaptation.

2.5. Deep Learning Privacy Preserving Techniques. In the forthcoming section, the prevailing cryptographic primitives that are presently opted by the organizations for privacy preserving both for training and interface of the Deep Neural Networks (DNNs) are discussed.

2.5.1. Homomorphic Encryption (HE). Homomorphic Encryption (HE) is primitive encryption that allows a party to encrypt data and send it to another party that can then perform certain operations on the encrypted version of the data [25]. An encryption system that allows arbitrary calculations to be encoded on encrypted data without decryption or access to any symmetric cryptographic decryption key is known HE [26]. When the account ends, the encrypted version of the result is sent to the first party that can decrypt and get the result in plain text.

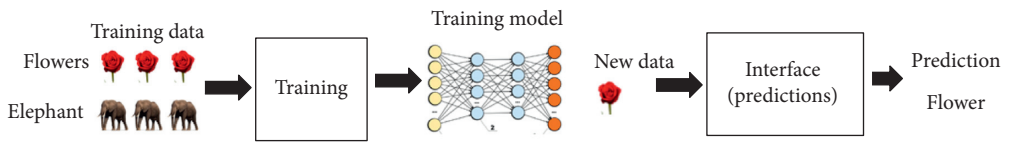

Figure 3: Training and interface in Deep Learning.

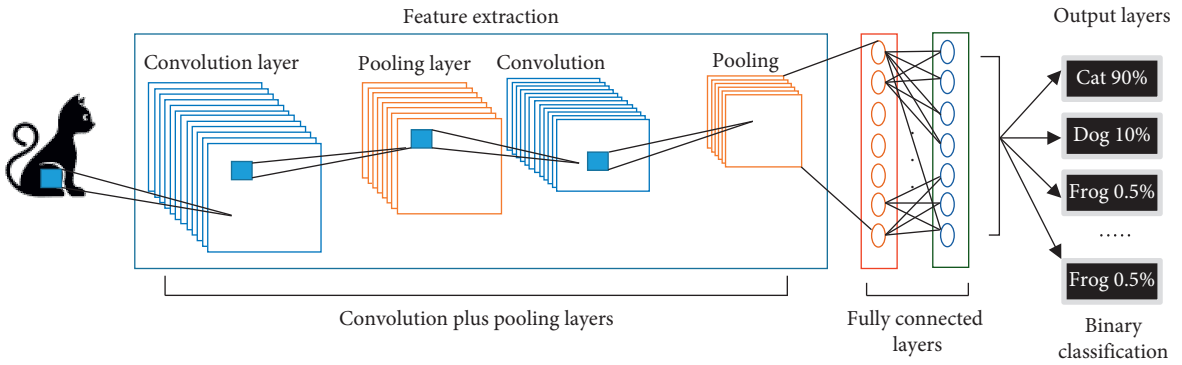

Figure 4: Structure of convolutional neural network.

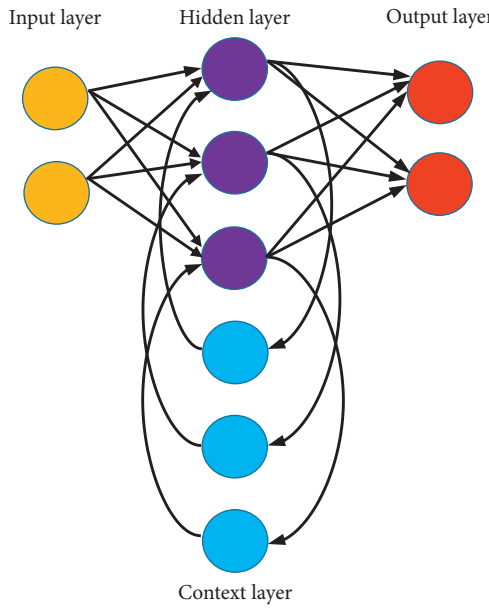

Figure 5: Structure of recurrent neural network.

Homomorphic encryption methods can be partially divided into completely Homomorphic Encryption and partially Homomorphic Encryption [27]. For example, the Paillier encryption system only supports adding to the two-digit encrypted version, which is partially Homomorphic Encryption. In contrast, a fully symmetric encryption system supports arbitrary functional logic. The Homomorphic Encryption scheme (Enc) follows the following equation:

$$\text{Enc}\,(a)\Delta\text{Enc}\,(b) = \text{Enc}\,(a * b). \tag{2}$$

where Enc: $X \longrightarrow Y$ is a Homomorphic Encryption scheme wherein X is used for a set of messages and Y is used for ciphertext. Furthermore, a and b are messages in X and Δ, $*$ are linear operations. At the beginning when Homomorphic Encryption used partial scheme and with the passage of time, researchers developed a full Homomorphic Encryption scheme which allowed complete computation on any type of data.

2.5.2. Garbled Circuits (GCs). Yao's garbled circuit method provides a general mechanism for building a secure two parties x and y, respectively, to develop an arbitrary Boolean function $f(x, y)$ without disclosing information regarding

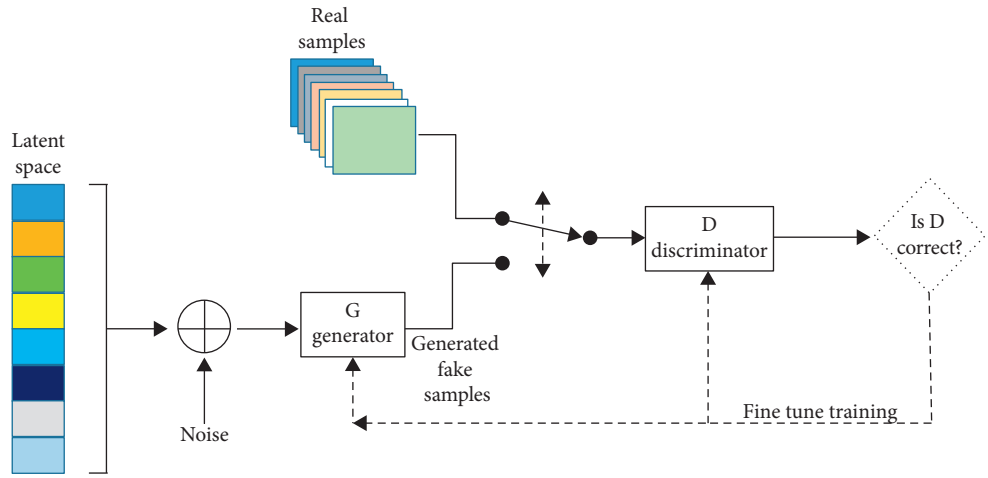

Figure 6: Generative adversarial network.

inputs irrespective of output of the function. The basic idea behind this algorithm is that one party will prepare the encrypted version of the circuit by computing f and the second party will obviously compute the output of the garbled circuit without knowing any value and information of the first circuit [28]. For example, in the 1st step, the first party will assign random keys to each wire of the circuit. The mentioned circuit has gates, and the first party shall encrypt output keys of the gates by using the associated input key and generate a garbled table [29]. The first party will then send the developed tables to the second party along with the associated input keys. On the other hand, the second party will get generated garbled tables and input keys. The second party then decrypts each gate that was encrypted by the first party until they find the output keys of the circuit [30]. The first party after decryption of the circuit will map the output keys to generate the plain text of the circuit.

2.5.3. Goldreich Micali Wigderson (GMW). It is also a generic secure function evaluation protocol, and it was developed in the year 1987 with the idea to evaluate the circuit through wire values by using secure linear secret sharing. This is like the Garbled Circuit protocol; this also requires the function that designates as a Boolean circuit [31, 32]. However, unlike Garbled Circuits, two users are required to cooperate for each AND gate. Thus, all AND gates are handled independently and in parallel, and the linear complexity is used in respect of the circuit. This technique is only used in short-level communication.

2.5.4. Differential Privacy (DP). DP is a metric that determines how much information about one entry in a database is exposed when a query is made to the database [33]. To preserve the privacy of database entries, carefully selected noise is added to the database so that the statistical properties of the database are retained while each data point is changed due to added noise [34]. Equally, DP can be considered as a way to reduce the dependency between the query result and individual data points in the database, thus reducing the leakage of information. It ensures that the attacker cannot infer any high confidence information from the databases or forms that have been released [35].

2.5.5. Share Secret (SS). It is a way to distribute the secret to two or more parties where each share does not give any information/data about the secret, but the secret can be reconstructed from the posts. One of the utmost famous Share Secret variants is Share Secret additive. In this case, the secret is shared by taking random samples and creating the last post so that collecting all the shares gets the secret value [36]. The secret of the algorithm can be reconstructed by inserting all the shares.

3. Deep Learning Private Data Frameworks

In this section, we will briefly describe the most efficient private data security frameworks for Deep Learning. All the given below frameworks are highly protected in the light of the Honest-but-Curious (HbC) adversary model. All parties adhering with this protocol are supposed to follow the protocol's instruction, but it is also observed that parties might infer more information. The said protocol is very secured as it stops the malicious attacks and also stops parties to deviate from protocol norms.

3.1. Shokri and Shmatikov. The authors suggested a method for maintaining privacy based on Differential Privacy (DP) for Deep Learning when the data are laid with different parties. In this situation, each party locally installs its own version of the neural network and selectively participates in some parameters updated with other parts. The authors proposed that the algorithm should be run on different machines in parallel, and then the results of the separate machines shall be aggregated to generate the final result. In order to protect the private data of the users, the Differential Privacy algorithm shall be applied when the parameters are shared instead of sharing the initial values. As a result, an

exchange is introduced between the precision of the trained neural network and the specificity of the data.

3.2. SecureML. It is a system to learn to maintain privacy in general and neural networks in particular. The system is based on the HE, GC, and SS protocols. Data owners secretly share their data with servers that do not comply with the rules and that train the particular neural network [37]. SecureML uses a more efficient custom activation feature to train a neural network using secure account protocols [31]. At the end of the account, the managed model is shared privately between the servers. In addition to training, SecureML also provides a conclusion to maintain privacy.

3.3. Google. A secure collection protocol was introduced for high-dimensional operators maintained by premium users. These protocols can be used in a unified education in which users maintain their databases and forms [38]. The core server recognizes the intelligent intelligence model by securely assembling the user's learning updates. The method is based on the covert exchange of the code and is powerful against users who exit the protocol at any time [39].

3.4. CryptoNets. CryptoNets, by applying ML to the problem regarding medical, educational, financial, or other kinds of confidential data, requires not only accurate forecasts but also careful cares to keep them safe and secure [40]. CryptoNets is basically developed by the Microsoft Research group, by introducing levelled Homomorphic Encryption (LHC). Due to nonlinear activation functions that cannot be achieved using LHE, the authors proposed that the activation functions are approached using polynomials of multiple degrees [41]. Therefore, the neural network must be retrained in plain text with the same activation function to maintain good prediction accuracy. Another disadvantage of this approach is that there is a certain limit on the number of serial multipliers imposed by LHE that makes the solution prohibitive. In addition, CryptoNets has an exchange of privacy/utility to achieve a higher level of privacy, and accuracy must be reduced within the same computing capabilities.

3.5. MiniONN. The authors observed that there are still privacy-preserving risks, and clients are still facing disclosure of sensitive information threats [42]. The MiniONN introduced the method for transmuting the existing DNN to the newly developed Oblivious Neural Network that addresses the privacy-preserving risks. It offers that the server does not know about the input of the client-side and the client also does not know about the model [42]. The performance of the MiniONN is better than CryptoNets and SecureML. It influences additive Homomorphic Encryption, Garbled Circuits, and secret sharing and also supports activation functions viz-a-viz pooling for CNN. It also has two main stages:

(i) An offline phase that supports additive Homomorphic Encryption that is not dependent on input

(ii) An online phase consists of GC and SS; nonlinear layers use GC and SS for processing

3.6. Chameleon. This protocol consists of mix frameworks regarding privacy preservation. This framework gets the benefits of the existing work of GMW protocol for in-depth analysis of the activation function and other Garbled Circuits for complicated activation functions and pooling layers. Chameleon uses secret sharing for arithmetic and addition functions. It has offline and online phases like in MiniONN [41]. The offline computation provided more fast computation for prediction instead of the online phase. Like SecureML, the Chameleon also requires two noncolluding machines, and unlike SecureML, it does not allow the involvement of the third party during the online phase. The Chameleon is more efficient as compared with all other discussed techniques.

3.7. DeepSecure. It is one of the modern frameworks based on the Garbled Circuit protocol. Since garbled circuit is a generic function evaluation protocol, the framework supports all nonlinear activation functions. DeepSecure offers the idea of decreasing the size of the data and the network before the implementation of the Garbled Circuits, thus compressing the account and connecting up to two things in size [43]. The preprocessing phase is independent of the basic encryption protocol and can be adopted by any other backend engine for its inference. DeepSecure also supports secure outsourcing of the account to a secondary server when the client has restricted resources.

4. Deep Learning Threats and Attacks

Deep learning faces various types of threats and attacks, and all famous threats and attacks are listed below.

4.1. Security Attack Taxonomy. Ji et al. [44] proposed classification of security threats for Deep Learning in 3 different angles, which influence classifieds, security breaches, and privacy of attacks.

In the view of impact, security risks and threats of Deep Learning are characterized into two categories.

4.1.1. Causative Attack. In the causative attack used to decrease the performance and reliability of the training processes, the machine learning algorithm provided incorrect training data after modification in the labels of the samples that are not covered under the decision limit. Many researchers performed causative attacks on the images and revealed that it expressively decreases the performance of the training phase.

This means that the opponents have the ability to change the input of training data, which becomes the cause of changes in the parameters of the learning models during

recycling, resulting in a substantial reduction in the presentation of jobs in succeeding taxonomy tasks.

4.1.2. Exploratory Attack. The exploratory attacks basically do not influence on a training dataset. The key objective of the exploratory attacks is to get knowledge with respect to the learning algorithm as much as it can about the basic system. Model invasion attack, model extraction, and membership inference are the examples of the exploratory attacks.

In a security break viewpoint, threats to Deep Learning may be characterized into 3 groups:

(1) Integrity Attack. The integrity attack occurs and then the Deep Learning models failed to trace the negative cases when categorizing harmful samples. The output of the system will clearly show that the integrity of the learning machine has been compromised. Suppose, we used spam filter to stop unwanted/harm messages, and if the attacker sends a message that has unwanted/harm words then, the filter does not get it. The integrity attack is tested through exploratory testing.

(2) Availability Attack. The availability attack is the opposite of an integrity attack in which the Deep Learning models filtered out the legitimate cases during the categorization of the unwanted/harmful samples. The output of the system will clearly show that the availability of the learning machine has been compromised and it is no more available and hacked. The DoS attack is one of the examples of availability wherein legitimate cases failed to cross the filters and ultimately the system becomes compromised.

(3) Privacy Violation Attack. In the privacy violation attack, the attacker becomes successful to get the sensitive/confidential information of the system from both training and learning models. In terms of attack privacy, security threats for Deep Learning have further 02 categories.

4.1.3. Targeted Attack. It is highly dangerous, and it directly decreases the performance of the classifier in a single specific sample or set of one of the samples.

4.1.4. Indiscriminate Attack. An indiscriminate attack is the subtype of the poisoning attack. The attacker's key goals are to increase the general classification error. Further, the indiscriminate attack always chooses a random value from the training sample. It randomly fails the classifier.

4.2. Deep Learning Attack Types. Although Deep Learning becomes successful to get draw the attention of the industry its security and privacy challenges, unfortunately, it could not get full attention as it should have. Here, we discuss the attack surface of the machine learning and discuss the weaknesses in the implementation of Deep Learning.

During the research, numerous types of attacks targeting DL applications and containing DoS attacks, evasion attacks, and organic termination attacks are revealed. Though all these attacks are different in its nature and in terms of their offensive objectives, the attacker's attack sources in Deep Learning applications are essentially from the following three angles.

4.2.1. Deep Learning Attack Surface Type-I. Deep learning application after trained mostly works on input data of the user for its classification. The attacker planned a malformed input attack on the input files or sometimes the network [24]. This type of attack applies to image recognition application which uses files on input and also applied to the applications that use sensors and cameras on the input. Due to the input type of the application, this risk can be reduced to implement risk mitigation techniques but the risk cannot be eliminated.

4.2.2. Deep Learning Attack Surface Type-II. This surface attack is also called a poisoning attack. The earlier surface type attack is due to the contaminated input data type of the application. This type of attack is not dependent on the application flaws or software breaches. However, defects in applications can become the reason of data poisoning easier. Suppose we observed variation in the procedure of analyzing the image in the frame and in common desktop applications. This variation allows the contamination of confidential data without being observed by the people who monitor the training process.

4.2.3. Deep Learning Attack Surface Type-III. It is a great chance of an attack on the Deep Learning applications if the developer will opt the model developed by the experts. Even though many programmers plan and create models from the beginning, many templates of the models exist for programmers who do not sufficient knowledge of machine learning. In this scenario, the attacker has also access to the template of the models. Like attacks of data poisoning, an attacker can easily attack all those applications and can get access to the private data that uses external models without any barrier. However, implementation flaws, such as a security vulnerability in the form analysis code, help attackers hide damaged models.

The readers should keep in mind that there are many types of attack surfaces and differ from each other, and it depends on the particular application, but above these 03 types of attack, surfaces cover most of the attack area. The comparison of attacking techniques against Deep Learning is given in Table 1.

4.3. Types of Threats. During the literature review, the authors studied many types of threats that affect the functionality of Deep Learning, and these threats targets different stages of Deep Learning. Here, in this paper, we are going to present the threat caused by the malformed input with the assumption that Deep Learning applications are taking input from files or networks.

Table 1: Comparison of attacking techniques against Deep Learning.

Attacking technique	Advantages	Disadvantages	Countermeasure technique
Causative attack [45, 46]	Influence on training data and exploits misclassifications	Time consuming Not fit for large dataset	[45, 47–49]
Exploratory attack [47]	Changes the discriminant results Misclassifies positive sample	Resource consuming	[50–52]
Integrity attack [53]	False negative passes through the system	Easily detected	[54–56]
Availability attack [57]	False positive results in blocking records	Time and resource consuming	[58–60]
Privacy violation attack [61]	Easily exploit the training dataset	Its performance is not reliable as it based on iterations	[62–64]
Targeted attack [65]	Misclassified to any arbitrary class	It does not provide assurance about the generated samples	[66–68]
Indiscriminate attack [69]	Good trade-off Highly efficient	Perturbation is high	[70, 71]

4.3.1. Deep Learning Threat Type-I. The most common weaknesses in Deep Learning frameworks are program errors that which cause software crashes, an infinite loop, or full memory depletion. The immediate threat of these errors is the denial of service attacks for applications running at the top of the window [72].

4.3.2. Deep Learning Threat Type-II. Deep Neural Networks are vulnerable to attacks at the time of its testing [45–48]. For example, in image recognition, an attacker may insert little noise to test a sample so that the error is classified as a DNN [73]. An example of a noise test is called an adversarial example. The noise is usually so small for a human. The benign is the alternate name of the adversarial example.

Evasion attacks are one of the Deep Learning attacks that restrict sensitive security and protection applications, like vehicles that drive on their own. Examples of self-driving adversaries can make unwanted decisions [74–78]. For example, one of the basic capabilities of autonomous cars is to automatically identify stop signals and traffic lights of the road.

Let us say, the adversary generates an adversarial stop, which means that the adversarial adds many imperceptible points to the stop, so that the vehicle that is driving alone is not recognized as a stop. As a result, vehicles that drive on their own will not stop at the stop sign and may collide with other vehicles, which could lead to serious traffic accidents.

There are many memory corruption-related bug in Deep Learning framework which may be a cause of wrong output. The evasion can be achieved through exploiting bugs in the Deep Learning framework by overwriting classification and control flow. In order to develop an effective defense against evasion attack, Goodfellow et al. [79] proposed adversarial training and adversarial example by introducing training of a DNN through augmenting training dataset. In order to train a DNN, the system generates training adversarial example through evasion attacks. The learner understands both the original training example and relating adversarial examples.

The adversarial training is weak as compared with adversarial examples that cannot be seen during training. Papernot et al. [80] developed a decontamination based technique to train Deep Neural Networks and Carlini and Wagner [81] revealed that their generated attacks have maximum success for Deep Neural Networks trained with concentration. Furthermore, Carlini and Wagner [81] determined that all measures must be assessed against the taxonomy of evasion attacks.

4.3.3. Deep Learning Threat Type-III. The software bugs of the systems that hosted Deep Learning applications on its operating system can be hijacked due to remote compromise and application bugs [44, 82]. This mostly happens when the system is connected with the cloud system and the Deep Learning applications are also running on that cloud-based system. All the input to the Deep Learning system is received through the network.

5. Defense Techniques against Security Issues in Deep Learning

During the literature review, many defense techniques against security concerns of Deep Learning were found, and we categorized these techniques into two major categories known as evasion and poisoning. Further, there are many evasion attack mitigation techniques, but in this chapter, only well-known and effective types are explained herein. Whereas, in a similar faction, the defense techniques against the poisoning attack proposed by the researcher are also given in Section 5.1. These defense techniques cannot 100% overcome the attacks, but these techniques can improve the prediction of the results.

5.1. Defense against Evasion Attacks. The most effective method of defense against evasion attack is to augment the adversarial examples and detect adversarial examples, adversarial training, and defensive distillation.

5.1.1. Detecting Adversarial Examples. The researchers [81, 83, 84] proposed different techniques to detect adversarial examples in the input and to create different benign and adversarial examples. As we mentioned earlier, the target of the attacker is to add more noise to formulate

effective adversarial examples. According to [83], it is not easy to detect such adaptive attacks, and some detection techniques effectively work while some ineffective. The main problem in the detection of adversarial examples is that it is unclear, and it is very hard to manage the testing example that is used to predict the adversarial example. Therefore, the expert should label the test examples manually. We give the above example of an automated/self-driving car which automatically takes decisions; it is not possible for the human to mark the label manually to detect adversarial example [75, 85–90].

Meng and Chen [84] proposed an approach to verify adversarial examples through testing examples and also the template of the testing example. According to the authors, if during verification of the adversarial example, it is proved through testing examples, then there is no need to label the classifier; otherwise, in the case of not predicted, the testing examples are required to be reformed through the reformer by removing unwanted noise from the testing example. After the completion of this task, the classifier shall label the example of testing to the Deep Neural Network and will consider it a genuine testing example. The experiments of MagNet show that it successfully presented defense against the evasion attacks.

5.1.2. Adversarial Training. Goodfellow et al. [79] presented a technique to train a Deep Neural Network through expanding dataset of training along with several adversarial examples and named it as adversarial training. In order to handle the evasion attack, the author proposed training benign examples against each training adversarial example. The learner of the system will use the backpropagation algorithm to get the knowledge of the Deep Neural Network through the original benign example and the attack adversarial example. The following authors also proposed the variants of the adversarial training. The authors used robust optimization techniques to solve min-max optimization problems. The core issue in the adversarial training is accuracies in the benign example.

5.1.3. Defensive Distillation. Sethi et al. [50] projected a method dependent on distillation for Deep Neural Network Training. The Deep Neural Network is trained first using a typical method. For each training example, Deep Neural Network produces a set of confidence levels. Confidence levels are treated as a soft mark for the training example. Due to software labels and training examples, Deep Neural Network weights are retrained. The named T parameter is used for the distillation temperature in the soft top layer during both training sessions to control confidence levels. In addition, noise is added to good example when hostile examples generated are slightly higher in distilled Deep Neural Network than in non-Deep Neural Network.

5.2. Defense against Poisoning Attack. The framework suggested in [91] takes the method of eliminating extreme values that fall outside the relevant group. In the binary grouping, they seek to discover the midpoints of the positive and negative categories. Then, the authors eliminate the points that are not near to the relevant focal point. To get information about these points, they use the defense field that eliminates points outside the radius of the ball, and a slab defense ignores points away from the line in a complementary manner.

Sun et al. [57] selected to rename the data points that are external values instead of deleting them. Attack flipping label is a distinct item for data poisoning that permits an attacker/hacker to control the appointment of a trifling number of training points. The author further describes a mechanism that studies points beyond the limits of the resolution to be harmful and reclassifies them. The procedure resets the label of every case.

Paudice et al. [92] also propose a protection mechanism to alleviate the intensity of poisoning attacks through remote sensing. The label tries to have the utmost influence on the protector with an inadequate number of poison points. The external detection process computes the external result of every x in the original data set. Further, there are many and different methods to calculate the external result.

It is stated that the impact functions are used to trail the predictions of the model and find the best persuasive data points that are accountable for the given forecast. It shows that the approximation of functions is still able to provide important materials that are nontransferable and nondiscriminatory models where the theory collapses [93]. The authors also assert that by using impact functions; the protector can verify the priority data only by the degree of impact. This method is superior to the previous methods to determine the greatest loss of training to eliminate contaminated samples.

The authors of this paper, to convince of the researchers, compared the advantages and disadvantages of existing countermeasure methods of Deep Learning, as presented in Table 2.

Various Deep Learning security attacks and corresponding countermeasures have drawn the attention of the industry and researchers. Table 3 presents comparative results and qualitative analysis of attacks and corresponding defensive techniques.

6. Observations and Recommendations

Deep Learning is providing new techniques to solve security problems. It introduced significant improvements over stereotype techniques and classical ML algorithms. Table 4 is a list of Deep Learning papers related to Deep Learning that we reviewed during the literature review. This table consists of methods used to solve the problems and citations of each paper. The authors reviewed 41 papers in this survey; the majority of the researchers conducted their study on malware detection and intrusion detection. During the survey, we also noticed some new areas of health security and vehicle security wherein Deep Learning techniques can be applied. Autoencoder technique is the most favorite one for the researchers to detect malware; thereafter, the Recurrent Neural Networks (RNNs) are also used for the same purpose

Table 2: Comparison of countermeasure techniques of Deep Learning.

Countermeasure methods	Advantages	Disadvantages
Adversarial training [94]	Very easy to understand and implement Scalable and have the ability to handle the complex dataset	It depends upon the sample size in the training phase
Defense distillation [80]	Sample and have the defense ability	Difficult to converge and high complexity
Ensemble method [95]	Model-independent, good generalization	Do not rebut the training data and computation overhead
Differential Privacy [96]	Preserves the privacy of training and learning data Low overhead, low complexity	It also affects legitimate data and model-independent
Homomorphic Encryption [97]	Maintains security and privacy of data and simple	It increases the data size and extensive computation overhead

Table 3: Comparison of attacking and defensive techniques in Deep Learning.

Attack/ defense	Technique	Training/testing	Taxonomy
Attack	Adversarial label flips	Training	Confidentiality, integrity, and reliability
Attack	Enchanting	Training	Integrity and reliability Exploratory attack
Attack	Obfuscation	Training	Exploratory attack Targeted attack Integrity and reliability
Attack	Poisoning	Training	Confidentiality, integrity, and reliability Causative attack Indiscriminate attack
Attack	Impersonate	Training	Exploratory attack Integrity and reliability
Defense	Adversarial training	Training	Creates a fool-proof system, improves the safety and security of the system, and defeats security attacks
Defense	Defense distillation	Training	It ensures the integrity, availability, reliability, and authenticity. Smooth classifier
Defense	Ensemble method	Training	Detects anomalies in the network Boosts data mining and intrusion detection
Defense	Differential Privacy	Training and testing	It protects the privacy of the data
Defense	Homomorphic Encryption	Training and testing	It protects the privacy of the data to ensure confidentiality

Table 4: Survey of Deep Learning approaches, methods, and security applications.

DL method	Citation	No of times cited (as of 17.01.2020)	Security application
Autoencoder	Hardy et al. [98]	59	Malware detection
Autoencoder	Rhode et al. [99]	50	Malware detection
Autoencoder	Kalash et al. [100]	35	Malware classification
Autoencoder	Wang and Yiu [101]	17	Malware classification
Autoencoder	Chalapathy and Chawla [102]	61	Anomaly detection
Autoencoder	Chen and Ye [103]	7	Adversarial malware attacks
Autoencoder	Maniath et al. [104]	10	Ransomware detection
Autoencoder	Zakaria [105]	-	Ransomware detection
Autoencoder	Demetrio [106]	10	Adversarial malware binaries
Autoencoder	James and Aimone [107]	18	File Type Identification
Autoencoder	Wang [108]	132	Traffic Identification
Autoencoder	Fadlullah et al. [109]	238	Network traffic control systems
Autoencoder	Aminanto et al. [110]	46	Wi-Fi impersonation detection
Autoencoder	Aceto et al. [111]	34	Mobile encrypted traffic
Autoencoder	Mi et al. [112]	15	Spam identification
Autoencoder	Shi et al. [113]	57	User authentication
Autoencoder	Catak and Yazi [114]	—	Malware classification
CNN	Gibert [115]	33	Malware classification

Table 4: Continued.

DL method	Citation	No of times cited (as of 17.01.2020)	Security application
CNN	Cha et al. [116]	517	Crack damage detection
CNN	Murata and Yamanishi [117]	01	Download attack
CNN	Vinayakumar et al. [118]	58	Network intrusion detection
CNN	Wang et al. [119]	109	Malware traffic classification
RNN	Yin et al. [120]	225	Intrusion detection
CNN RNN	Maleh [121]	-	Malware classification
CNN RNN	Kolosnjaji et al. [122]	179	Malware detection
CNN RNN	Tobiyama et al. [123]	98	Malware detection
CNN RNN	Yu et al. [124]	33	Intrusion detection
CNN (dynamic)	Hill and Bellekens [125]	03	Malware detection
DNN	M.-J. Kang and J.-W. Kang [126]	224	Intrusion detection
DNN	Potluri and Diedrich [127]	67	Intrusion detection
DNN	Dahl et al. [128]	283	Malware classification
DNN	Sebastián et al. [129]	146	Massive malware labeling
DNN RNN	Mi et al. [130]	15	Insider threat
DNN RNN	Mi et al. [131]	03	Spam detection
GAN	Anderson et al. [132]	67	Intrusion detection
GAN	Yu et al. [133]	23	Character detection
GAN	Zhauniarovich et al. [134]	20	Malicious domains detection
RBM	Alrawashdeh and Purdy [135]	63	Intrusion detection
RBM	Yuan et al. [136]	202	Malware detection
RBM	Wang et al. [137]	216	Defect prediction
RBM	Chen et al. [138]	67	Detecting android malware

as well as to detect information security threats. Restricted Boltzmann Machines (RBMs) are also used for the same purpose, but we cannot find much study using this technique for security purposes. Different authors combined autoencoders and RNN techniques to train the unlabelled data. RBM is a popular technique due to its easy implementation and simplicity.

After studying the above techniques, it is very difficult for the authors to exactly define the performance of the techniques due to different datasets and metrics. It is pertinent to add here that the performance of these techniques/methods varies across security areas. The information security domain has a vast range of data collected through different sources to apply Deep Learning tests. The researches/studies could not be completed and generate accurate results because a large volume of datasets is not publically available. The majority of the dataset sources are small and old. To develop a security solution through the meaningful method, it is necessary to test the method on large, updated, and reliable datasets. The results of the methods should be compared with each other through real-time scenarios.

7. Conclusion

Deep learning has now become part of our daily lives, and when new technology invested, definitely security and privacy issues arise. In recent years, extensive research was carried out on the security and privacy preserving issues and its counter frameworks for Deep Learning and Deep Neural Network's training and interface modules. Therefore, security and privacy become very critical and important issues as in the other technologies that cannot be overlooked.

During the literature review, we found two basic types of security attacks: evasion and poisoning. We also presented the effective countermeasures of these two types of attacks. We explained both security and private attacks, frameworks, and countermeasure techniques.

These frameworks have cryptographic primitives and numerous characteristics. It should be noted that private interference frameworks have no complete capability to provide DNNs security and privacy. We outline the details of different types of security attacks on Deep Learning. There are many types of attacks that are invested to exploit the Deep Learning results so that model information may be extracted or get the knowledge about the training data like model inversion, model extraction, and membership inference. The said attacks steel training data and generate expected results. The private training section of Deep Learning has more computation overhead as compared with the interface. Therefore, more concentration and research are required in this direction to develop a more efficient solution for the privacy preservation of the data while maintaining models.

Privacy risks always persist due to various characteristics of the Deep Neural Networks which is actually relying upon a huge amount of input training data. In this chapter, we also discussed possible privacy threats on sensitive and confidential Deep Learning model's data. Various studies have been conducted on privacy preserving attacks by using Deep Learning.

For future work, it is essential for the researchers to deeply investigate different cryptographic primitive's solutions for DNNs. A mixed protocol technique can reduce the computation overhead on the security and privacy preserving solutions. Furthermore, customization of the privacy and security protocols for DNNs is also an interesting and open research area to develop a viable solution. The authors are also intended to perform their research in the application

of Deep Learning especially in the area of astrophysics, plasma physics, atomic physics, thermodynamics, electromagnetic, machines, nanotechnology, fluid mechanics, electro hydrodynamics, signal processing, power, energy, bioinformatics, economy, and finance.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this article.

References

[1] B. Shickel, P. J. Tighe, A. Bihorac, and P. Rashidi, "Deep EHR: a survey of recent advances in deep learning techniques for electronic health record (EHR) analysis," *IEEE Journal of Biomedical and Health Informatics*, vol. 22, no. 5, pp. 1589–1604, 2018.

[2] A. L. Buczak and E. Guven, "A survey of data mining and machine learning methods for cyber security intrusion detection," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 2, pp. 1153–1176, 2016.

[3] J. Schmidhuber, "Deep learning in neural networks: an overview," *Neural Networks*, vol. 61, pp. 85–117, 2015.

[4] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[5] M. A. Nielsen, *Neural Networks and Deep Learning*, Vol. 25, Determination press, San Francisco, CA, USA, 2015.

[6] R. Miikkulainen, "Evolving deep neural networks," in *Artificial Intelligence in the Age of Neural Networks and Brain Computing*, pp. 293–312, Elsevier, Amsterdam, Netherlands, 2019.

[7] Y.-G. Jiang, Z. Wu, J. Wang, X. Xue, and S.-F. Chang, "Exploiting feature and class relationships in video categorization with regularized deep neural networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 2, pp. 352–364, 2018.

[8] V. Kotu and B. Deshpande, "Deep learning," in *Data Science*, pp. 307–342, Elsevier, Amsterdam, Netherlands,, 2019.

[9] S. Gollapudi, "Deep learning for computer vision," in *Learn Computer Vision Using OpenCV*, pp. 51–69, Apress, Berkeley, CA, USA, 2019.

[10] T. N. Sainath, A. Mohamed, B. Kingsbury, and B. Ramabhadran, "Deep convolutional neural networks for LVCSR," in *Proceedings of the 2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 8614–8618, Vancouver, Canada, May 2013.

[11] T. N. Sainath, O. Vinyals, A. Senior, and H. Sak, "Convolutional, long short-term memory, fully connected deep neural networks," in *Proceedings of the 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4580–4584, South Brisbane, Australia, April 2015.

[12] T. P. Huster, C. J. Chiang, R. Chadha, and A. Swami, "Towards the development of robust deep neural networks in adversarial settings," in *Proceedings of the 2018 IEEE Military Communications Conference (MILCOM)*, pp. 419–424, Los Angeles, CA, USA, October 2018.

[13] U. Shaham, A. Cloninger, and R. R. Coifman, "Provable approximation properties for deep neural networks," *Applied and Computational Harmonic Analysis*, vol. 44, no. 3, pp. 537–557, 2018.

[14] P. Mohamed Shakeel, S. Baskar, V. R. Sarma Dhulipala, S. Mishra, and M. M. Jaber, "Maintaining security and privacy in health care system using learning based deep-Q-networks," *Journal of Medical Systems*, vol. 42, no. 10, p. 186, 2018.

[15] R. Shokri and V. Shmatikov, "Privacy-preserving deep learning," in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security—CCS'15*, pp. 1310–1321, Denver, Colorado, USA, 2015.

[16] R. Vinayakumar, K. P. Soman, and P. Poornachandran, "Detecting malicious domain names using deep learning approaches at scale," *Journal of Intelligent & Fuzzy Systems*, vol. 34, no. 3, pp. 1355–1367, 2018.

[17] B. Feng, Q. Fu, M. Dong, D. Guo, and Q. Li, "Multistage and elastic spam detection in mobile social networks through deep learning," *IEEE Network*, vol. 32, no. 4, pp. 15–21, 2018.

[18] J. Li, L. Sun, Q. Yan, Z. Li, W. Srisa-an, and H. Ye, "Significant permission identification for machine-learning-based android malware detection," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 7, pp. 3216–3225, 2018.

[19] W. W. Stead, "Clinical implications and challenges of artificial intelligence and deep learning," *JAMA*, vol. 320, no. 11, pp. 1107-1108, 2018.

[20] D. Cohen, J. Foley, H. Zamani, J. Allan, and W. B. Croft, "Universal approximation functions for fast learning to rank," in *Proceedings of the 41st International ACM SIGIR Conference on Research & Development in Information Retrieval—SIGIR'18*, pp. 1017–1020, New York, NY, USA, 2018.

[21] X. Zhang, X. Zhou, M. Lin, and J. Sun, "ShuffleNet: an extremely efficient convolutional neural network for mobile devices," in *Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6848–6856, Long Beach, CA, USA, 2018.

[22] A. Hassan, M. Kamran, A. Illahi, and R. M. A. Zahoor, "Design of cascade artificial neural networks optimized with the memetic computing paradigm for solving the nonlinear Bratu system," *The European Physical Journal Plus*, vol. 134, no. 3, p. 122, 2019.

[23] X.-Y. Zhang, F. Yin, Y.-M. Zhang, C.-L. Liu, and Y. Bengio, "Drawing and recognizing Chinese characters with recurrent neural network," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 4, pp. 849–862, 2018.

[24] Q. Yang, P. Yan, Y. Zhang et al., "Low-dose CT image denoising using a generative adversarial network with wasserstein distance and perceptual loss," *IEEE Transactions on Medical Imaging*, vol. 37, no. 6, pp. 1348–1357, 2018.

[25] A. Acar, H. Aksu, A. S. Uluagac, and M. Conti, "A survey on homomorphic encryption schemes," *ACM Computing Surveys*, vol. 51, no. 4, pp. 1–35, 2018.

[26] D. Boneh, "Threshold cryptosystems from threshold fully homomorphic encryption," in *Advances in Cryptology—CRYPTO 2018*, pp. 565–596, Springer, Berlin, Germany, 2018.

[27] S. Halevi, Y. Polyakov, and V. Shoup, "An improved RNS variant of the BFV homomorphic encryption scheme," in *Topics in Cryptology—CT-RSA 2019*, pp. 83–105, Springer, Berlin, Germany, 2019.

[28] Q. Yang, G. Peng, P. Gasti et al., "MEG: memory and energy efficient garbled circuit evaluation on smartphones," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 4, pp. 913–922, 2019.

[29] A. Saleem, A. Khan, F. Shahid, M. Masoom Alam, and M. K. Khan, "Recent advancements in garbled computing: how far have we come towards achieving secure, efficient and reusable garbled circuits," *Journal of Network and Computer Applications*, vol. 108, pp. 1–19, 2018.

[30] A. Dupin, D. Pointcheval, and C. Bidan, "On the leakage of corrupted garbled circuits," in *Proceedings of the Provable Security*, pp. 3–21, Jeju, South Korea, October 2018.

[31] S. Sharma and K. Chen, "Privacy-preserving boosting with random linear classifiers," in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, pp. 2294–2296, New York, NY, USA, 2018.

[32] H. Ahmad, L. Wang, H. Hong et al., "Primitives towards verifiable computation: a survey," *Frontiers of Computer Science*, vol. 123, pp. 451–478, 2018.

[33] J. Wang, J. Zhang, W. Bao, X. Zhu, B. Cao, and P. S. Yu, "Not just privacy," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 2407–2416, New York, NY, USA, 2018.

[34] N. C. Abay, Y. Zhou, M. Kantarcioglu, B. Thuraisingham, and L. Sweeney, "Privacy preserving synthetic data release using deep learning," in *Machine Learning and Knowledge Discovery in Databases*, pp. 510–526, Springer, Berlin, Germany, 2019.

[35] N. Hynes, D. Dao, D. Yan, R. Cheng, and D. Song, "A demonstration of sterling," *Proceedings of the VLDB Endowment*, vol. 11, no. 12, pp. 2086–2089, 2018.

[36] L. T. Phong and T. T. Phuong, "Privacy-preserving deep learning via weight transmission," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 11, pp. 3003–3015, 2019.

[37] P. Mohassel and Y. Zhang, "SecureML: a system for scalable privacy-preserving machine learning," in *Proceedings of the 2017 IEEE Symposium on Security and Privacy (SP)*, pp. 19–38, San Jose, CA, USA, 2017.

[38] G. Lin, N. Sun, S. Nepal, J. Zhang, Y. Xiang, and H. Hassan, "Statistical twitter spam detection demystified: performance, stability and scalability," *IEEE Access*, vol. 5, pp. 11142–11154, 2017.

[39] K. Bonawitz, V. Ivanov, B. Kreuter et al., "Practical secure aggregation for privacy-preserving machine learning," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pp. 1175–1191, New York, NY, USA, 2017.

[40] B. Hitaj, G. Ateniese, and F. Perez-Cruz, "Deep models under the GAN," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security—CCS'17*, pp. 603–618, New York, NY, USA, 2017.

[41] M. S. Riazi, C. Weinert, O. Tkachenko et al., "Chameleon," in *Proceedings of the 2018 on Asia Conference on Computer and Communications Security—ASIACCS'18*, pp. 707–721, New York, NY, USA, 2018.

[42] J. Liu, M. Juuti, Y. Lu, and N. Asokan, "Oblivious neural network predictions via MiniONN transformations," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security—CCS'17*, pp. 619–631, New York, NY, USA, 2017.

[43] B. D. Rouhani, M. S. Riazi, and F. Koushanfar, "Deepsecure," in *Proceedings of the 55th Annual Design Automation Conference—DAC'18*, pp. 2:1–2:6, New York, NY, USA, 2018.

[44] Y. Ji, X. Zhang, S. Ji, X. Luo, and T. Wang, "Model-reuse attacks on deep learning systems," in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, pp. 349–363, New York, NY, USA, 2018.

[45] C. Burkard and B. Lagesse, "Analysis of causative attacks against SVMs learning from data streams," in *Proceedings of the 3rd ACM on International Workshop on Security And Privacy Analytics—IWSPA'17*, pp. 31–36, Scottsdale, AZ, USA, March 2017.

[46] Y. Li and D. S. Yeung, "A causative attack against semi-supervised learning," in *Proceedings of the International Conference on Machine Learning and Cybernetics*, pp. 196–203, Qingdao, China, July 2014.

[47] L. Huang, A. D. Joseph, B. Nelson, B. I. Rubinstein, and J. D. Tygar, "Adversarial machine learning," in *Proceedings of the 4th ACM workshop on Security and artificial intelligence*, pp. 43–58, Chicago, IL, USA, October 2011.

[48] Y. Shi and Y. E. Sagduyu, "Evasion and causative attacks with adversarial deep learning," in *Proceedings of the MILCOM 2017-2017 IEEE Military Communications Conference (MILCOM)*, pp. 243–248, Baltimore, MD, USA, October 2017.

[49] B. Miller, "Adversarial active learning," in *Proceedings of the 2014 Workshop on Artificial Intelligent and Security Workshop*, pp. 3–14, Angers,, February 2014.

[50] T. S. Sethi, M. Kantardzic, and J. W. Ryu, "'Security theater': on the vulnerability of classifiers to exploratory attacks," in *Proceedings of the Pacific-Asia Workshop on Intelligence and Security Informatics*, pp. 49–63, Jeju Island, South Korea, May 2017.

[51] T. S. Sethi and M. Kantardzic, "Data driven exploratory attacks on black box classifiers in adversarial domains," *Neurocomputing*, vol. 289, pp. 129–143, 2018.

[52] L. Halawi, R. McCarthy, and N. Muoghalu, "Student approaches to learning: an exploratory study," *Issues in Information Systems*, vol. 10, no. 1, p. 13, 2009.

[53] M. Barreno, B. Nelson, R. Sears, A. D. Joseph, and J. D. Tygar, "Can machine learning be secure?" in *Proceedings of the 2006 ACM Symposium on Information, computer and communications security*, pp. 16–25, Taipei Taiwan, March 2006.

[54] S. Ntalampiras, "Automatic identification of integrity attacks in cyber-physical systems," *Expert Systems with Applications*, vol. 58, pp. 164–173, 2016.

[55] B. Biggio and F. Roli, "Wild patterns: ten years after the rise of adversarial machine learning," *Pattern Recognition*, vol. 84, pp. 317–331, 2018.

[56] X. Yuan, P. He, Q. Zhu, and X. Li, "Adversarial examples: attacks and defenses for deep learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 309, pp. 2805–2824, June 2019.

[57] L. Sun, J. Wang, P. S. Yu, and B. Li, "Adversarial attack and defense on graph data: a survey," 2018, https://arxiv.org/abs/1812.10528.

[58] A. Erba, "Real-time evasion attacks with physical constraints on deep learning-based anomaly detectors in industrial control systems," 2019, https://arxiv.org/abs/1907.07487.

[59] M. Jagielski, A. Oprea, B. Biggio, C. Liu, C. Nita-Rotaru, and B. Li, "Manipulating machine learning: poisoning attacks and countermeasures for regression learning," in *Proceedings of the 2018 IEEE Symposium on Security and Privacy (SP)*, pp. 19–35, San Francisco, CA, USA, May 2018.

[60] M. Sun, "Data poisoning attack against unsupervised node embedding methods," 2018, https://arxiv.org/abs/1810.12881.

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

[62] W. Li, Y. Wang, H. Li, and X. Li, "Leveraging Memory PUFs and PIM-based encryption to secure edge deep learning systems," in *Proceedings of the 2019 IEEE 37th VLSI Test Symposium (VTS)*, pp. 1–6, Monterey, CA, USA, April 2019.

[63] A. Siddiqi, "Adversarial security attacks and perturbations on machine learning and deep learning methods," 2019, https://arxiv.org/abs/1907.07291.

[64] Z. E. Mrabet, N. Kaabouch, H. E. Ghazi, and H. E. Ghazi, "Cyber-security in smart grid: survey and challenges," *Computers & Electrical Engineering*, vol. 67, pp. 469–482, 2018.

[65] K. Eykholt, "Robust physical-world attacks on deep learning models," 2017, https://arxiv.org/abs/1707.08945.

[66] J. Su, D. V. Vargas, and K. Sakurai, "One pixel attack for fooling deep neural networks," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 5, pp. 828–841, 2019.

[67] J. Rauber, W. Brendel, and M. Bethge, "Foolbox v0. 8.0: a python toolbox to benchmark the robustness of machine learning models," vol. 5, 2017, https://arxiv.org/abs/1707.04131.

[68] S. Shen, S. Tople, and P. Saxena, "A uror: defending against poisoning attacks in collaborative deep learning systems," in *Proceedings of the 32nd Annual Conference on Computer Security Applications*, pp. 508–519, New York, NY, USA, December 2016.

[69] M. Barreno, B. Nelson, A. D. Joseph, and J. D. Tygar, *The Security of Machine Learning*, p. 26, Springer, Berlin, Germany.

[70] S. L. Wang, K. Shafi, C. Lokan, and H. A. Abbass, "Robustness of neural ensembles against targeted and random Adversarial Learning," in *Proceedings of the International Conference on Fuzzy Systems*, pp. 1–8, Barcelona, Spain, July 2010.

[71] J. Peng and P. P. K. Chan, "Revised Naive Bayes classifier for combating the focus attack in spam filtering," in *Proceedings of the 2013 International Conference on Machine Learning and Cybernetics*, vol. 2, pp. 610–614, Tianjin, China, July 2013.

[72] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, "The limitations of deep learning in adversarial settings," in *Proceedings of the 2016 IEEE European Symposium on Security and Privacy (EuroS P)*, pp. 372–387, London, UK, April 2016.

[73] C. Szegedy, "Intriguing properties of neural networks," 2013, https://arxiv.org/abs/1312.6199.

[74] S. A. Butt, M. I. Tariq, T. Jamal, A. Ali, J. L. Diaz Martinez, and E. De-La-Hoz-Franco, "Predictive variables for agile development merging cloud computing services," *IEEE Access*, vol. 7, pp. 99273–99282, 2019.

[75] M. I. Tariq, "Towards information security metrics framework for cloud computing," *International Journal of Cloud Computing and Services Science*, vol. 1, no. 4, p. 209, 2012.

[76] M. I. Tariq, *Providing Assurance to Cloud Computing through ISO 27001 Certification: How Much Cloud is Secured after Implementing Information Security Standards*, CreateSpace, Scotts Valley, CA, USA, 2015.

[77] M. I. Tariq, "Analysis of the effectiveness of cloud control matrix for hybrid cloud computing," *International Journal of Future Generation Communication and Networking*, vol. 11, no. 4, pp. 1–10, 2018.

[78] M. I. Tariq, "Agent based information security framework for hybrid cloud computing," *KSII Transactions on Internet & Information Systems*, vol. 13, no. 1, 2019.

[79] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," 2014, https://arxiv.org/abs/1412.6572.

[80] N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami, "Distillation as a defense to adversarial perturbations against deep neural networks," in *Proceedings of the 2016 IEEE Symposium on Security and Privacy (SP)*, pp. 582–597, San Jose, CA, USA, May 2016.

[81] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *Proceedings of the 2017 IEEE Symposium on Security and Privacy (SP)*, pp. 39–57, San Jose, CA, USA, May 2017.

[82] J. Saxe, R. Harang, C. Wild, and H. Sanders, "A deep learning approach to fast, format-agnostic detection of malicious web content," in *Proceedings of the 2018 IEEE Security and Privacy Workshops (SPW)*, pp. 8–14, San Francisco, CA, USA, May 2018.

[83] N. Carlini and D. Wagner, "Adversarial examples are not easily detected," in *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security—AISec'17*, pp. 3–14, New York, NY, USA, 2017.

[84] D. Meng and H. Chen, "MagNet: a two-pronged defense against adversarial examples," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security—CCS'17*, pp. 135–147, New York, NY, USA, 2017.

[85] M. I. Tariq, S. Tayyaba, H. Rasheed, and M. W. Ashraf, "Factors influencing the cloud computing adoption in higher education institutions of Punjab, Pakistan," in *Proceedings of the 2017 International Conference on Communication, Computing and Digital Systems (C-CODE)*, pp. 179–184, Islamabad, Pakistan, March 2017.

[86] M. I. Tariq, D. Haq, and J. Iqbal, "SLA based information security metric for cloud computing from COBIT 4.1 framework," *International Journal of Computer Networks and Communications Security*, vol. 1, no. 3, pp. 95–101, 2013.

[87] M. I. Tariq, S. Tayyaba, M. U. Hashmi, M. W. Ashraf, and N. A. Mian, "Agent based information security threat management framework for hybrid cloud computing," *International Journal of Computer Science and Network Security*, vol. 17, no. 12, p. 57, 2017.

[88] M. I. Tariq, S. Tayyaba, M. W. Ashraf, and V. E. Balas, "8—deep learning techniques for optimizing medical big data," in *Deep Learning Techniques for Biomedical and Health Informatics*, B. Agarwal, V. E. Balas, L. C. Jain, R. C. Poonia, and Manisha, Eds., pp. 187–211, Academic Press, Cambridge, MA, USA, 2020.

[89] M. I. Tariq, S. Tayyaba, M. W. Ashraf, and H. Rasheed, "Risk based NIST effectiveness analysis for cloud security," *Bahria University Journal of Information & Communication Technologies (BUJICT)*, vol. 10, no. Special Is, 2017.

[90] M. I. Tariq, S. Tayyaba, M. W. Ashraf, H. Rasheed, and F. Khan, "Analysis of NIST SP 800-53 rev. 3 controls effectiveness for cloud computing," in *Proceedings of the 1st National Conference on Emerging Trends and Innovations in Computing & Technology*, pp. 88–92, Karachi, Pakistan, 2016.

[91] J. Steinhardt, P. W. Koh, and P. Liang, "Certified defenses for data poisoning attacks," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pp. 3520–3532, Long Beach, CA, USA, December 2017.

[92] A. Paudice, L. Mu. ñoz-Gonźalez, A. György, and E. C. Lupu, "Detection of adversarial training examples in poisoning attacks through anomaly detection," 2018, https://arxiv.org/abs/1802.03041.

[93] P. W. Koh and P. Liang, "Understanding black-box predictions via influence functions," 2017, https://arxiv.org/abs/1703.04730.

[94] F. Tramèr, A. Kurakin, N. Papernot, I. Goodfellow, D. Boneh, and P. McDaniel, "Ensemble adversarial training: attacks and defenses," 2018, https://arxiv.org/abs/1705.07204.

[95] X. Qiu, L. Zhang, Y. Ren, P. N. Suganthan, and G. Amaratunga, "Ensemble deep learning for regression and time series forecasting," in *Proceedings of the 2014 IEEE Symposium on Computational Intelligence in Ensemble Learning (CIEL)*, pp. 1–6, Orlando, FL, USA, December 2014.

[96] M. Abadi, A. Chu, I. Goodfellow et al., "Deep learning with differential privacy," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security—CCS'16*, pp. 308–318, Vienna, Austria, 2016.

[97] L. T. Phong, Y. Aono, T. Hayashi, L. Wang, and S. Moriai, "Privacy-preserving deep learning via additively homomorphic encryption," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 5, pp. 1333–1345, 2018.

[98] W. Hardy, L. Chen, S. Hou, Y. Ye, and X. Li, "DL4MD: a deep learning framework for intelligent malware detection," in *Proceedings of the International Conference on Data Mining (DMIN)*, p. 61, Las Vegas, NE, USA, July 2016.

[99] M. Rhode, P. Burnap, and K. Jones, "Early-stage malware prediction using recurrent neural networks," *Computers & Security*, vol. 77, pp. 578–594, 2018.

[100] M. Kalash, M. Rochan, N. Mohammed, N. D. Bruce, Y. Wang, and F. Iqbal, "Malware classification with deep convolutional neural networks," in *Proceedings of the 2018 9th IFIP International Conference on New Technologies, Mobility and Security (NTMS)*, pp. 1–5, Paris, France, February 2018.

[101] X. Wang and S. Yiu, "A multi-task learning model for malware classification with useful file access pattern from API call sequence," 2016, https://arxiv.org/abs/1610.05945.

[102] R. Chalapathy and S. Chawla, "Deep learning for anomaly detection: a survey," 2019, https://arxiv.org/abs/1901.03407.

[103] L. Chen and Y. Ye, "SecMD: make machine learning more secure against adversarial malware attacks," in *Proceedings of the Australasian Joint Conference on Artificial Intelligence*, pp. 76–89, Melbourne, Australia, August 2017.

[104] S. Maniath, A. Ashok, P. Poornachandran, V. Sujadevi, A. P. Sankar, and S. Jan, "Deep learning LSTM based ransomware detection," in *Proceedings of the 2017 Recent Developments in Control, Automation & Power Engineering (RDCAPE)*, pp. 442–446, Noida, India, October 2017.

[105] W. Z. Zakaria, M. F. Abdollah, and A. F. Mohd Ariffin, "On Ransomware Detection," in *Proceedings of the Seventh International Conference on Informatics and Applications (ICIA2018)*, pp. 12–17, Takamatsu, Japan, November 2018.

[106] L. Demetrio, B. Biggio, G. Lagorio, F. Roli, and A. Armando, "Explaining vulnerabilities of deep learning to adversarial malware binaries," 2019, https://arxiv.org/abs/1901.03583.

[107] C. D. James and J. B. Aimone, *A Signal Processing Approach for Cyber Data Classification with Deep Neural Networks*, Sandia National Lab.(SNL-NM), Albuquerque, NM, USA, 2015.

[108] Z. Wang, *The Applications of Deep Learning on Traffic Identification*, Vol. 24, TechRepublic, Louisville, KY, USA, 2015.

[109] Z. M. Fadlullah, F. Tang, B. Mao et al., "State-of-the-Art deep learning: evolving machine intelligence toward tomorrow's intelligent network traffic control systems," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2432–2455, 2017.

[110] M. E. Aminanto, R. Choi, H. C. Tanuwidjaja, P. D. Yoo, and K. Kim, "Deep abstraction and weighted feature selection for Wi-Fi impersonation detection," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 3, pp. 621–636, 2017.

[111] G. Aceto, D. Ciuonzo, A. Montieri, and A. Pescapé, "Mobile encrypted traffic classification using deep learning," in *Proceedings of the 2018 Network Traffic Measurement and Analysis Conference (TMA)*, pp. 1–8, Vienna, Austria, June 2018.

[112] G. Mi, Y. Gao, and Y. Tan, "Apply stacked auto-encoder to spam detection," in *Proceedings of the International Conference in Swarm Intelligence*, pp. 3–15, Beijing, China, June 2015.

[113] C. Shi, J. Liu, H. Liu, and Y. Chen, "Smart user authentication through actuation of daily activities leveraging WiFi-enabled IoT," in *Proceedings of the 18th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, p. 5, Chennai, India, July 2017.

[114] F. O. Catak and A. F. Yazı, "A benchmark API call dataset for windows PE malware classification," 2019, https://arxiv.org/abs/1905.01999.

[115] D. Gibert, *Convolutional Neural Networks for Malware Classification*, University Rovira i Virgili, Tarragona, Spain, 2016.

[116] Y.-J. Cha, W. Choi, and O. Büyüköztürk, "Deep learning-based crack damage detection using convolutional neural networks," *Computer-Aided Civil and Infrastructure Engineering*, vol. 32, no. 5, pp. 361–378, 2017.

[117] M. Murata and K. Yamanishi, *Detecting Drive-By Download Attacks from Proxy Log Information Using Convolutional Neural Network*, Osaka University, Osaka, Japan, 2017.

[118] R. Vinayakumar, K. P. Soman, and P. Poornachandran, "Applying convolutional neural network for network intrusion detection," in *Proceedings of the 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pp. 1222–1228, Manipal, India, September 2017.

[119] W. Wang, M. Zhu, X. Zeng, X. Ye, and Y. Sheng, "Malware traffic classification using convolutional neural network for representation learning," in *Proceedings of the 2017 International Conference on Information Networking (ICOIN)*, pp. 712–717, Da Nang, Vietnam, January 2017.

[120] C. Yin, Y. Zhu, J. Fei, and X. He, "A deep learning approach for intrusion detection using recurrent neural networks," *IEEE Access*, vol. 5, pp. 21954–21961, 2017.

[121] Y. Maleh, "Malware classification and analysis using convolutional and recurrent neural network," in *Handbook of Research on Deep Learning Innovations and Trends*, pp. 233–255, IGI Global, Harrisburg, PA, USA, 2019.

[122] B. Kolosnjaji, A. Zarras, G. Webster, and C. Eckert, "Deep learning for classification of malware system call sequences," in *Proceedings of the AI 2016: Advances in Artificial Intelligence*, pp. 137–149, Cham, Switzerland, December 2016.

[123] S. Tobiyama, Y. Yamaguchi, H. Shimada, T. Ikuse, and T. Yagi, "Malware detection with deep neural network using process behavior," in *Proceedings of the 2016 IEEE 40th Annual Computer Software and Applications Conference (COMPSAC)*, vol. 2, pp. 577–582, Atlanta, GA, USA, June 2016.

[124] Y. Yu, J. Long, and Z. Cai, "Network intrusion detection through stacking dilated convolutional autoencoders," *Security and Communication Networks*, vol. 2017, Article ID 4184196, 10 pages, 2017.

[125] G. D. Hill and X. J. A. Bellekens, "Deep learning based cryptographic primitive classification," 2017, https://arxiv.org/abs/1709.08385.

[126] M.-J. Kang and J.-W. Kang, "Intrusion detection system using deep neural network for in-vehicle network security," *PLoS One*, vol. 11, no. 6, Article ID e0155781, 2016.

[127] S. Potluri and C. Diedrich, "Accelerated deep neural networks for enhanced Intrusion Detection System," in *Proceedings of the 2016 IEEE 21st International Conference on*

Emerging Technologies and Factory Automation (ETFA), pp. 1–8, Berlin, Germany, September 2016.

[128] G. E. Dahl, J. W. Stokes, L. Deng, and D. Yu, "Large-scale malware classification using random projections and neural networks," in *Proceedings of the 2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 3422–3426, Vancouver, Canada, May 2013.

[129] M. Sebastián, R. Rivera, P. Kotzias, and J. Caballero, "AVclass: a tool for massive malware labeling," in *Research in Attacks, Intrusions, and Defenses*, pp. 230–253, Springer, Cham, Switzerland, 2016.

[130] G. Mi, Y. Gao, and Y. Tan, "Apply stacked auto-encoder to spam detection," in *Advances in Swarm and Computational Intelligence*, pp. 3–15, Springer, Cham, Switzerland, 2015.

[131] G. Mi, Y. Gao, and Y. Tan, "Term space partition based ensemble feature construction for spam detection," in *Data Mining and Big Data*, pp. 205–216, Springer, Cham, Switzerland, 2016.

[132] H. S. Anderson, J. Woodbridge, and B. Filar, "DeepDGA: adversarially-tuned domain generation and detection," in *Proceedings of the 2016 ACM Workshop on Artificial Intelligence and Security—ALSec'16*, pp. 13–21, Vienna, Austria, 2016.

[133] B. Yu, J. Pan, J. Hu, A. Nascimento, and M. De Cock, "Character level based detection of DGA domain names," in *Proceedings of the 2018 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, Rio de janeiro, Brazil, 2018.

[134] Y. Zhauniarovich, I. Khalil, T. Yu, and M. Dacier, "A survey on malicious domains detection through DNS data analysis," *ACM Computing Surveys*, vol. 51, no. 4, pp. 1–36, 2018.

[135] K. Alrawashdeh and C. Purdy, "Toward an online anomaly intrusion detection system based on deep learning," in *Proceedings of the 2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pp. 195–200, Anaheim, CA, USA, December 2016.

[136] Z. Yuan, Y. Lu, Z. Wang, and Y. Xue, "Droid-sec: deep learning in android malware detection," in *Proceedings of the 2014 ACM conference on SIGCOMM—SIGCOMM'14*, pp. 371-372, Chicago, IL, USA, 2014.

[137] S. Wang, T. Liu, and L. Tan, "Automatically learning semantic features for defect prediction," in *Proceedings of the 2016 IEEE/ACM 38th International Conference on Software Engineering (ICSE)—ICSE'16*, pp. 297–308, Austin TX, USA, May 2016.

[138] S. Chen, M. Xue, Z. Tang, L. Xu, and H. Zhu, "StormDroid: a streaminglized machine learning-based system for detecting android malware," in *Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security—ASIA CCS'16*, pp. 377–388, Xi'an, China, 2016.