A Survey on Deep Learning for Cybersecurity: Progress, Challenges, and Opportunities

Mayra Macas[a,b,**], Chunming Wu[a], Walter Fuertes[b]

[a] College of Computer Science and Technology, Zhejiang University No. 38 Zheda Road, Hangzhou 310027, China
[b] Department of Computer Science, Universidad de las Fuerzas Armadas ESPE, Av. General Rumiñahui S/N, P.O. Box 17-15-231B, Sangolquí, Ecuador

Abstract

As the number of Internet-connected systems rises, cyber analysts find it increasingly difficult to effectively monitor the produced volume of data, its velocity and diversity. Signature-based cybersecurity strategies are unlikely to achieve the required performance for detecting new attack vectors. Moreover, technological advances enable attackers to develop sophisticated attack strategies that can avoid detection by current security systems. As the cyber-threat landscape worsens, we need advanced tools and technologies to detect, investigate, and make quick decisions regarding emerging attacks and threats. Applications of artificial intelligence (AI) have the potential to analyze and automatically classify vast amounts of Internet traffic. AI-based solutions that automate the detection of attacks and tackle complex cybersecurity problems are gaining increasing attention. This paper comprehensively presents the promising applications of deep learning, a subfield of AI based on multiple layers of artificial neural networks, in a wide variety of security tasks. Before critically and comparatively surveying state-of-the-art solutions from the literature, we discuss the key characteristics of representative deep learning architectures employed in cybersecurity applications, we introduce the emerging trends in deep learning, and we provide an overview of necessary resources like a generic framework and suitable datasets. We identify the limitations of the reviewed works, and we bring forth a vision of the current challenges of the area, providing valuable insights and good practices for researchers and developers working on related problems. Finally, we uncover current pain points and outline directions for future research to address them.

Keywords: Cybersecurity, artificial intelligence, machine learning, deep learning, cyber-threat, botnets, intrusion detection, spam filtering, encrypted traffic analysis

1. Introduction

Cybersecurity is a set of technologies and methods that strive to safeguard computer networks, end systems, programs, and data from attacks, unauthorized access, changes, or harm. Cyber-defense mechanisms exist at the host, network, application, and data level. A plethora of tools, such as firewalls, antivirus software, intrusion detection systems (IDS) and intrusion protection systems (IPS), work stealthily to avoid attacks and detect security breaches. However, adversaries are still at an advantage because they solely need to find one vulnerability on the systems under protection. As the number of systems connected to the Internet rises, cyber-attacks are also increasing in size, sophistication, and cost. During the years 2019 and 2020, the attack surface expanded due to the accelerated digitization and the increasing dependence on the digital infrastructure (e.g., cloud-based services, remote work). According to Symantec's report, over 60 million malicious attempts were blocked in the second quarter of 2020, representing a 74.6% increase over the previous months [1]. In the beginning of 2021, one of the most bizarre and terrifying cyberattacks happened. A cybercriminal gained unlawful access to the water treatment system of the City of Oldsmar in Florida and attempted to make the water unsafe to consume by changing specific chemical levels [2]. Following this trend, in the last report of threat landscape compiled by the European Union Agency for Network and Information Security (ENISA) [3], it is predicted that a secure and reliable cyberspace will become even more important in the new social and economic norm established after the COVID-19 pandemic and the consequent transformation of the digital environment.

Organizations face an urgent need to ramp up and improve their cybersecurity due to the continuous growth of the number of end-user devices, networks, and user interfaces, combined with the implicated increasing quantities of data transmitted over the Internet brought by the advances in cloud and fog/edge computing, the Internet of Things (IoT), Industry 4.0/5.0, and 5G/6G [4, 5]. In such a context, signature-based cybersecurity strategies have become time-consuming and make security protection systems behave reactively rather than proactively. Further-

*The research work of Mrs. Mayra Macas was partially supported by the Chinese government scholarship CSC Reg. No.: 2017GBJ005834.

**Corresponding author
Email address: mayramacas@ieee.org (Mayra Macas)

more, technological advances are also benefiting attackers who are developing new, complex, and sophisticated attack strategies that can evade detection by existing security systems [6]. As the cyber-threat landscape continues to extend, advanced technologies and tools for predicting, detecting, and making decisions faster to address emerging attacks and threats are required.

In the past few years, cybersecurity researchers have started to explore AI because it can potentially intelligently analyze and automatically classify large amounts of Internet traffic [4, 5]. It is estimated that the market for AI in cybersecurity will grow from US$8.8 billion in 2019 to a US$38.2 billion net worth by 2026 [7]. Deep learning, a subfield of AI based on multiple layers of artificial neural networks, has established a key role in solving complicated cybersecurity problems due to its ability to manage complex data structures, its automatic feature extraction, and its efficiency in recognizing patterns and correlations. In practice, the application of deep learning in cybersecurity offers three strategic advantages:

• *Simplicity:* In contrast to traditional machine learning (ML) techniques, deep learning considerably simplifies feature handcrafting, replacing brittle, complex, engineering-heavy pipelines with straightforward, end-to-end trainable models allowing to offload a lot of work [8, 9]. Within the cybersecurity context, feature handcrafting for each type of attack requires outstanding human effort because of the ever-changing and growing cyber-threat landscape [9]. Deep learning techniques, on the other hand, can be trained for learning the features, which makes them an excellent choice for security tasks because they can detect previously unknown intrusions (e.g., zero-day malware) [10].

• *Scalability:* Classical ML learning algorithms often require storing all data points in memory, something that is computationally non-viable under big data scenarios. Besides, the traditional ML algorithms do not significantly improve their performance with a massive volume of data. Thus, they do not provide scalability [8]. In contrast, deep learning models can be trained on datasets of varying size, since they can iterate over small batches of data (e.g., using Stochastic Gradient Descent-SGD [11]). Furthermore, using a vast amount of data in the training process of the deep learning techniques has the additional benefit of preventing model over-fitting. These properties are compatible with the security domain, where vast amounts of heterogeneous data are produced from sensors, logs, endpoint agents, and distributed directory systems.

• *Reusability:* Unlike many traditional ML approaches, deep learning models can be trained on additional data without starting again from scratch. Thus, they are suitable for continuous online training, a desirable property for huge production models [8]. Moreover, trained deep learning models are repurposable and, therefore, reusable via transfer learning, allowing to reinvest previous work into increasingly sophisticated and robust models. This is essential in the cybersecurity domain because it decreases the computational and memory requirements of cyber defense systems when performing multi-task learning applications, e.g., collaborative spam filtering [12, 13].

1.1. Survey Scope

This survey aims to present a comprehensive analysis of state-of-the-art deep learning practices in the cybersecurity domain. By doing this, we aim to answer the following key questions:

• *Q1. What are the cutting-edge deep learning techniques significant to the cybersecurity domain?*

• *Q2. How can cyber analysts, researchers, or engineers apply deep learning to specific cybersecurity problems?*

• *Q3. What are the available datasets for training, validating and testing deep learning-based cyber-defense systems?*

• *Q4. What are the latest successful deep learning-based systems in cybersecurity?*

• *Q5. Which are the most important and promising directions for further study?*

To that end, we explore the application of deep learning techniques to cybersecurity tasks like intrusion detection in IoT and Software Defined Networks (SDNs), IoT malware analysis, botnet detection and domain generation algorithms (DGAs), cyber-physical system security, spam filtering, fraud detection, and encrypted traffic analysis.

1.2. Previous Surveys

Cybersecurity and deep learning problems have been researched mostly independently. Only recently, a crossover between the two areas has emerged. In [14, 15, 16, 17], machine learning applications to cybersecurity problems have been studied without deep learning techniques. Other authors describe deep learning methods for a narrow set of cybersecurity applications. Xin et al. [18] analyze only the attacks related to intrusion detection and the deficiencies of the existing datasets. Similarly, the authors in [5] review different deep learning techniques focusing on their application to intrusion detection, along with the respective datasets. Shaukat et al. [19] analyze cyber-attacks related solely to malware analysis, intrusion detection, and spam detection and do not cover other areas. In [20], the authors summarize and present works that aim to secure cyber physical systems (CPSs). Several deep learning-based anomaly detection approaches for CPSs are also analyzed and evaluated in [21]. The studies [22, 23] review machine learning and deep learning methods for securing IoT. Rodriguez et al. [24] compile deep learning techniques in mobile networks security.

More recent surveys like [25, 26, 10] present deep learning models that have been used to automate various security tasks such as intrusion detection and malware analysis, but they suffer from the following shortcomings: (i) they do not include the most cutting-edge deep learning methods; (ii) deep learning model selection directions for solving security problems are not provided; (iii) they focus only on unencrypted traffic and traditional networks, neglecting emerging paradigms like IoT and SDNs; (iv) spam filtering is examined exclusively within the context of e-mail, ignoring online social networks and user reviews; (v) credit card and telecomunication fraud detection is overlooked; (vi) intelligent transportation systems are not examined; and (vii) they do not provide any guidelines regarding the use of the most appropriate dataset per case.

1.3. Contributions

This paper comprehensively reviews recent advances and state-of-the-art DL-based solutions in a wide range of cybersecurity applications, in order to bridge the gaps and limitations identified in previous surveys. The contributions of our work can be summarized as follows:

• We examine cutting-edge deep learning methods from the perspective of the security domain, focusing on their applicability to this area, while giving less attention to conventional deep learning models that may be out-of-date. Moreover, we provide insights about selecting the most appropriate deep learning architecture for each security task (answering *Q1* in Section 2).

• We introduce a step-wise deep learning framework customized for cybersecurity applications. At every constituent stage of the framework, we discuss the related challenges, and we provide insights (answering *Q2* in Section 3.1).

• We present an exhaustive list and a brief overview of the available datasets that can be used for each cybersecurity application (answering *Q3* in Section 3.2).

• We explore applications previously overlooked in other related surveys, such as malware detection in IoT devices, network intrusion detection in IoT and SDNs, credit card and telecomunication fraud detection, spam filtering in online social networks and users reviews, cyber-physical systems security, and encrypted traffic analysis. Furthermore, we discuss the advantages and limitations of each reviewed study in order to highlight best practices and caution against potential pitfalls in developing deep learning-based cyber-defense systems (answering *Q4* in Section 4).

• We identify the open issues and current challenges of the crossover between deep learning and cybersecurity, and we provide a comprehensive list of promising research directions to address and overcome them (answering *Q5* in Section 5).

1.4. Literature Search Methodology

Initially, we meticulously investigated research contributions that addressed various aspects of the most prominent threats to security and privacy in recent times. The aim was to extract relevant, common, and impactful cyberthreats. We then confirmed their consistency with the last report of threat landscape compiled by ENISA [3]. Next, for each topic, we searched prominent academic publication repositories (e.g., Google Scholar, IEEE Xplore, and ACM Digital Library) to check whether there are works that applied deep learning methods to detect/identify such threats. Additionally, we looked for publications from the same authors and articles citing or being cited by already found works. The publications are presented in chronological order from 2016 up to 2021, inclusively. We did not include papers published prior to 2016, except for seminal and highly relevant works. The numbers of citations assessed by Google Scholar and Scopus were used to filter the most influential research papers. Overall, we reviewed 78 research works, whose distribution per year is illustrated in Fig. 1.

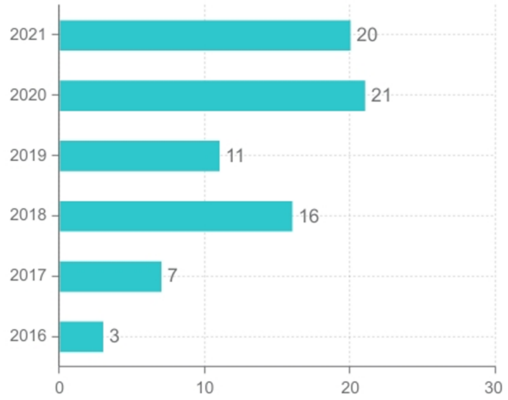

Figure 1: Distribution of analyzed research works by year.

1.5. Paper Organization

The remainder of this article is structured as follows. Section 2 introduces and compares the most relevant deep learning models, also providing guidelines for model selection towards solving cybersecurity problems. Furthermore, it analyzes the most recent advances of deep learning (e.g., attention and transformers) and their influence in the cybersecurity domain. The proposed framework and the datasets used for training, validation, and testing of DL-based defense systems are overviewed in Section 3. In Section 4, state-of-the-art studies applying deep learning techniques to specific cybersecurity applications are reviewed, and the related advantages and shortcomings are discussed. Lessons learned, future research directions and open challenges are presented in Section 5. Finally, the paper is concluded in Section 6 with a summary of its main take-away messages.

2. Deep Learning Background

AI is an approach striving to build intelligent machines that mimic or even surpass human intelligence. Many techniques fall under this broad umbrella, such as expert systems, evolutionary algorithms, and machine learning. Machine learning enables the artificial process to absorb knowledge from data and make decisions without being explicitly programmed. Generally, machine learning algorithms are categorized into supervised, unsupervised, and reinforcement learning. Deep learning (DL) is a subfield of machine learning that carries out representation learning through multilayer transformation, thereby generating more accurate results for detection and prediction tasks. Particularly in cybersecurity, DL-based defense systems are being used to automate the detection of cyber-attacks while evolving and improving their capabilities over time. With the goal of answering *Q1 (What are the cutting-edge deep learning techniques significant to the cybersecurity domain?)*, this section summarizes the most representative models and emerging trends in deep learning for cybersecurity applications.

2.1. Deep Learning models

DL is a set of prediction models that are based on Artificial Neural Networks (ANNs) [27, 28]. An ANN is a generic term that encompasses any structure of interconnected neurons sending information to each other. The key difference between Deep Neural Networks (DNNs) and the simpler single-hidden-layer neural networks is the large depth of the former; that is, the high number of hidden layers participating in the multistep process of pattern recognition. Specifically, a DNN consists of an input layer, a suitable number (more than one) of hidden layers, and an output layer. Each layer of a DNN is composed of neurons that are capable of producing nonlinear outputs from their inputs. Data is propagated to the hidden layers by the input layer neurons that initially receive it. Then, the neurons in the hidden layers generate weighted sums of the input data on which they apply specific activation functions (e.g., ReLU or tanh). Subsequently, these outputs are propagated to the output layer, where the final results are presented. Fig. 2(a) shows an example of a DNN.

2.1.1. Convolutional Neural Networks (CNNs):

CNNs, also called ConvNets, are a specialized kind of neural network destined to process data that come in the form of multiple arrays. Many data structures are organized in multiple arrays, such as 1D arrays for signals and sequences, 2D arrays for digital images or spectrograms of audio, and 3-D arrays for volumetric images and videos. To fully use the 2D structure of input data, local connections and shared weights in the network are employed in place of the traditional fully connected networks. This process results in significantly fewer parameters, making the network faster to train. In a typical CNN, a series of convolutional layers are followed by polling (subsampling) layers, while in the final stage fully connected layers (similar to multilayer perceptron - MLP) are generally employed. An example of image classification using CNN is illustrated in Fig. 2(c). The CNN and its variants (e.g., ResNet [29], DenseNet [30], SqueezeNet [31], MobileNets [32], and YOLO [33]) have been investigated for a variety of cybersecurity applications such as user authentication, fraud, and malware detection.

2.1.2. Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM)

RNNs [34] are robust sequence learners, arranged to capture the temporal dependencies in data by including memory. Fig. 2(b) illustrates a conventional RNN unfolded in time that can be trained across many times steps using backpropagation through time (BPTT) [35]. Despite their efficiency in modelling sequential data, RNNs suffer from the so-called vanishing gradient problem that occurs when the output at any given time step depends on inputs much earlier in time. The LSTM [36, 37] architecture has been developed in order to address this issue. As shown in Fig. 2(e), each unit in the LSTM model has a cell memory with a state that stores information about the input sequences across time steps. The reading and modifying access to the memory units are controlled through sigmoid gates. LSTM models perform better than RNN models when data is characterized by a long dependency on time [38]. Such long dependency can be observed in data generated by IoT networks or complex systems (e.g., Cyber-Physical Systems - CPSs). Overall, the use of LSTM and its variants (e.g., Convolutional Long Short-Term Memory - ConvLSTM [39]) shows promise in improving the attack detection and prediction accuracy in settings where data is time-dependent.

2.1.3. Autoencoder (AE)

In AEs, the desired output is set to be equal to the input [27]. As shown in Fig. 2(d), AEs generally include two parts, namely the encoder and the decoder, which are non-linear mapping functions implemented through NNs. The encoder maps the input data into the low-dimensional latent space, whereas the decoder maps the latent representation into the output layer to reconstruct the input. The encoder and decoder can be implemented by different types of NNs, including recurrent neural networks or feed-forward non-recurrent neural networks. This type of models has been mainly employed to solve unsupervised learning problems and transfer learning [27]. Depending on the size of the hidden layer, an autoencoder is classified as undercomplete or overcomplete. Moreover, based on the constraints imposed on the loss function, there exist various types of AEs, such as sparse AEs [27], denoising AEs [40], Stacked Contractive AEs [41, 42], Adversarial AEs [43], and Variational Autoencoder (VAEs) [44, 45, 46]. Due to reconstructing the input at the output layer, AEs are typically employed for network intrusion and spam detection

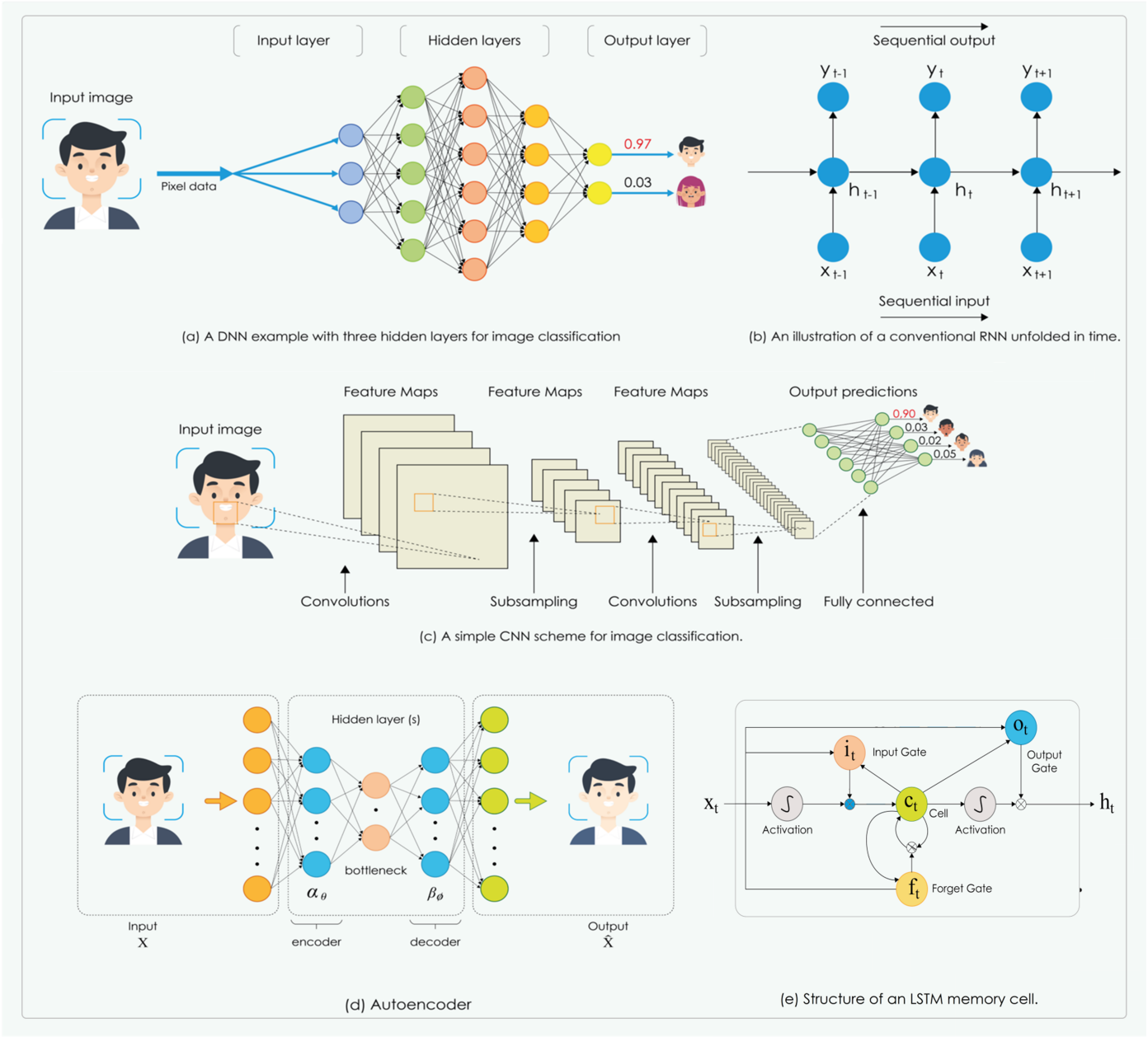

Figure 2: DNN, RNN, CNN, and LSTM architectures.

tasks. This architecture has received much attention for enabling applications of Industrial IoT (IIoT), such as fault diagnosis in machines and hardware devices, and physical-based anomaly detection mechanisms.

2.1.4. Deep Belief Network (DBN)

As can be seen in Fig. 3(a), DBNs [47] are a type of generative ANNs that resembles a composition of several stacked Restricted Boltzmann Machines (RBMs) [48]. The RBM is an energy-based model that has a single layer of hidden units without connection to each other and an undirected connection to a layer of visible units. Multiple hidden layers can be trained using the hidden layer output of one RBM as the training data for the next higher-level RBM [49, 50, 51]. This method of training with multiple layers in a greedy layer-by-layer manner allows to build a hybrid probabilistic generative model, termed DBN, that involves both undirected connections between its top two layers and downward directed connections between the rest layers [50, 51]. The lower visible layer represents the states of the input layer as a data vector. A DBN learns to probabilistically reconstruct its inputs in an unsupervised manner, while the layers act as further detectors on the inputs. Moreover, an additional supervised training process provides the ability to perform classification tasks. Many applications can benefit from the use of DBNs, such as false data injection attack detection in industrial environments and the detection of anomalies in IoT networks.

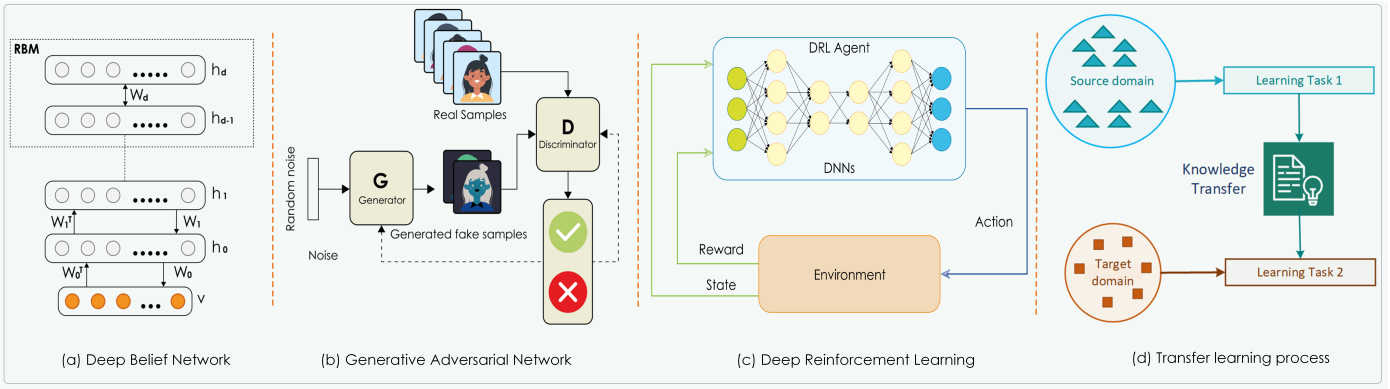

Figure 3: DBN, GAN, DRL, TL architectures.

2.1.5. Generative Adversarial Network (GAN)

GAN was developed by Goodfellow et al. [52] and consists of a generative network and a discriminative network. The generator captures the distributions of the real data and tries to produce samples with the same characteristics in order to fool and confuse the discriminator, which in turn attempts to distinguish the real data from the fake/generated. Typically, the training process of conventional GANs is highly sensitive to model structures, learning rates, and other hyper-parameters. Thus, numerous ad hoc "tricks" are usually needed for achieving convergence and improving the fidelity of generated data. In order to mitigate this problem, several variants of GANs have been introduced, such as Wasserstein Generative Adversarial Network (WGAN) [53], BigGAN [54], and Loss-Sensitive Generative Adversarial Network (LS-GAN) [55]. In the cybersecurity domain, GANs are typically applied to overcome the data imbalance problem by using artificial variations to minimize any bias in data collection [56, 57]. An example of GAN is depicted in Fig. 3(b).

2.1.6. Deep Reinforcement Learning (DRL)

DRL is composed of Reinforcement Learning (RL) and DNNs. It aims to create an intelligent agent that can carry out efficient policies for maximizing the rewards of long term tasks with controllable actions (see Fig. 3(c)). Typically, RL searches for the optimal policy of actions over states from the environment, and the DNN represents a large number of states and approximates the action values to estimate the quality of the action for any given state. Representative DRL methods comprise Deep Q-Networks (DQNs) [58], Deep Deterministic Policy Gradient (DDPG) algorithm [59], Asynchronous Advantage Actor-Critic (A3C) [60], deep policy gradient methods [61], Rainbow [62], Distributed Proximal Policy Optimization (DPPO) [63], adaptive deep Q-learning (ADQL) algorithm [64], and content based deep reinforcement learning (C-DRL) [65]. By incorporating DL into traditional RL, DRL is highly efficient in solving dynamic, complex, and especially high-dimensional security problems, including DRL-based security methods for CPSs, multiagent DRL-based game theory simulations for cyber-defense strategies against cyber-attacks, and autonomous intrusion detection approaches.

Lessons learned. More recent architectures, including VAE and GAN, are expected to have a significant impact on cybersecurity applications since they cover semi-supervised/unsupervised learning. Those are more favorable for cybersecurity applications, particularly in IIoT-based security, where only a small fraction of the vast amount of generated data can be annotated for supervised ML [66]. Emerging machine learning architectures such as DRL can support autonomous intrusion detection approaches and DRL-based security methods used for CPSs.

2.2. Emerging Trends in Deep Learning

Herein, we briefly review the recent advances and emerging trends in deep learning.

2.2.1. Transfer learning (TL)

Ideally, in machine learning, there is a considerable volume of labeled training data that follows the same distribution as the test data [67, 68]. Nevertheless, collecting relevant and sufficient training data is often time-consuming, expensive, or even unrealistic in some scenarios. Particularly in the cybersecurity domain, where new types of attacks (e.g., zero-day attacks) appear daily [67, 69]. Semi-supervised learning can partly alleviate this problem by relaxing the requirement of a large volume of labeled data. However, in many cases, unlabeled instances are also complicated to collect. To solve the above problem, a promising machine learning methodology that focuses on transferring knowledge across domains is TL [27]. TL aims at improving the performance of target learners on target domains by transferring the knowledge contained in different but related source domains (see Fig. 3(d)) [68]. In this way, target learners can be constructed without depending on the existence of a large amount of target domain data.

TL is an attractive potential solution for many cybersecurity applications where gathering training data is not an easy task (e.g., data from IoT devices [67]). Deep learning models are a good match for TL due to their capability of learning both low-level and abstract representations from input data [70]. In particular, stacked denoising autoencoders [71, 67] and other variants of autoencoders [72] have been shown to perform very well in this area. More information about TL can be found in the survey [68].

2.2.2. One-shot/few-shot learning.

Two extreme TL paradigms are one-shot learning and zero-shot learning. The former involves a pre-trained model and only one or a handful of samples per category, whereas the latter does not require any sample [73]. Instead, it leverages the meta description of the category and the correlations with existing training data. Even though research regarding deep one-shot learning and deep zero-shot learning [74, 75, 76] is in its infancy, both paradigms are very promising in detecting new threats or intrusions. Some initial works in cybersecurity leveraged ANNs such as Triplet Networks [77] and Siamese Networks [76] for one-shot/few-shot learning, alleviating in this way the need to gather and train with a large dataset. A Siamese neural network (SNN) (a.k.a. twin NN) [78] is composed of two "twin" networks that are trained simultaneously to learn the similarity of two instances called a pair. Triplet networks [79] are comprised of parallel and identical sub-networks that share the same weights and hyperparameters. The networks are trained using three different inputs called triplets. During training, each input is individually fed to its corresponding sub-network.

2.2.3. DL with attention.

Broadly speaking, attention is used to focus on the important parts of the input data, while ignoring other irrelevant information. To that end, an attention-mechanism examines the input sequence and decides at each step which other parts of the sequence are important. In the cybersecurity domain, attention mechanisms (e.g., temporal and spatial) have been demonstrated to achieve outstanding accuracy in predicting intrusion/attacks. Usually, such attention mechanisms are used in conjunction with a recurrent network (e.g., GRU, LSTM, Bi-LSTM, and so on) [80, 81, 82, 83, 84, 85]. However, recurrent architectures are computationally inefficient because they rely on the sequential processing of input at the encoding step, prohibiting parallelization. A novel architecture called Transformer (Tr) [86, 87, 88] addresses this issue by relying only on a self-attention mechanism to capture global dependencies between input and output. For further details about attention models, see reference [89].

2.2.4. DL with non-Euclidean data.

Deep learning techniques have been very successful in processing grid-like data such as image, sound, video or text. However, it is important to explore DL in non-Euclidean domains (e.g., graphs and manifolds), which are becoming increasingly present in real life. Within the context of cybersecurity, recent works [90, 91, 92, 93] have started employing Graph Neural Networks (GNNs) due to their ability to capture complex relationships between objects and make inferences based on data described by graphs [94].

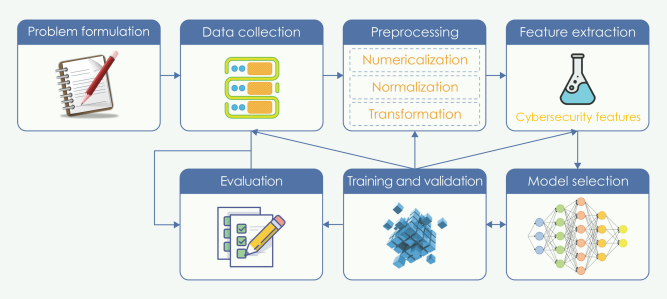

Figure 4: Deep learning framework for cybersecurity applications.

3. Deep learning framework and Datasets for Cybesecurity

In order to answer *Q2 (How can cyber analysts, researchers, or engineers apply deep learning to specific cybersecurity problems?)* and *Q3. (What are the available datasets for training, validating and testing deep learning-based cyber-defense systems?)*, this section introduces a deep learning framework for cybersecurity applications and briefly overviews the datasets and testbeds that can be used for training, validating, and testing deep learning-based cyber-defense solutions.

3.1. Deep learning framework

Herein, we propose a generic **d**eep **l**earning **f**ramework for **c**ybersecurity **a**pplications (DLF-CA) that is built upon the knowledge distilled from the papers examined in this survey. The conceptual model of our DLF-CA is illustrated in Fig. 4. As can be seen, the first step in building a cyber defense system (i.e., classifier/predictor) is the *problem formulation.* In this step, we should clearly define the goal of the classifier/predictor. Typically, goals include malware detection/classification, botnet detection, cyber-physical system security, network intrusion detection, spam filtering, fraud detection, and encrypted traffic analysis. Then, it is necessary to *collect a vast amount of data.* Sufficient data quality and quantity are crucial for solving complex and challenging security problems. In practice, researchers often collect a dataset specific to their classifier's/predictor's goal. To do this, the first step is to determine the data collection location(s), which can drastically affect the capability to learn general trends, as well as the available features, the granularity, and the reliability of the data. In particular, data collection can happen

in various locations, namely the client or server-side of the communication channel, the edge of the network, or any place in between. After collecting an appropriate and sufficient amount of raw data, it is necessary to contemplate how to store it for carrying out further processing. Typically, physical devices and cloud storage services are used to keep the data into databases or files [95].

Subsequently, in *the preprocessing step,* the previously collected data should be cleaned, mapped into a common schema, merged, and converted into suitable formats and types. Feature engineering refers to *extracting and selecting suitable features,* which are of paramount importance in machine learning since they are pivotal in defining and enriching the predictors. Nevertheless, in practice, coming up with appropriate features is usually challenging and requires a lot of labor, time, and technical expertise. An alternative approach is representation learning, whose fundamental concept is recognizing and disengaging the latent explanatory factors present in the data [96]. In that way, the extraction of valuable information in the form of appropriate features is facilitated via learning representations of the data.

Table 1: Common evaluation metrics for deep learning models.

Metric	Description	Equation
False Positive Rate (FPR) or False Acceptance Rate (FAR)	The proportion of the elements that were wrongly determined as positive among the actual negatives.	$\frac{FP}{FP+TN}$
Recall, Sensitivity or True Positive Rate (TPR)	The proportion of actual positives that were correctly identified.	$\frac{TP}{FN+TP}$
False Negative Rate (FNR) or False Rejection Rate (FRR)	The proportion of the elements that were wrongly determined as negatives among the actual positives.	$\frac{FN}{FN+TP}$
Specificity or True Negative Rate (TNR)	The proportion of actual negatives that were correctly identified.	$\frac{TN}{FP+TN}$
Precision	The ratio of actual positives over all the elements predicted as positives.	$\frac{TP}{TP+FP}$
Accuracy	The ratio of correctly predicted items over the total number of items.	$\frac{TP+TN}{TP+TN+FP+FN}$
F1-Score	The harmonic mean of precision and recall. Also known as F-Score or F-measure.	$\frac{2\times Pr\times Rec}{(Pr+Rec)}$.
Area Under Curve (AUC)	The area covered by the plot of TPR and FPR (ROC Curve) at different threshold values between 0 and 1.	$\int ROC$

Selecting the "right" deep learning model depends greatly on the input features, which directly determine the model's accuracy. The modeling process is iterative, providing critical insights regarding the refinement of data preparation and model specification at each repetition. In order to find the optimal model, it is necessary to try several algorithms with specific parameters (and hyper-parameters) in a trial-and-error fashion. *Typically, a dataset comprises three parts: training, validation, and test set.* The first subset is employed during training, while the second is used to measure the prediction accuracy. This validation accuracy is one of the principal criteria for deciding whether to accept or reject the trained model. When we settle on the chosen model type and hyper-parameters, we proceed to train a new model with the entire set of available data using the best hyper-parameters found. This should include any data that was previously held aside for validation. The last step is *periodic evaluation* over updated test sets, which is essential for verifying that the model can recognize and predict zero-day attacks. Table 1 describes the most common evaluation metrics for deep learning models.

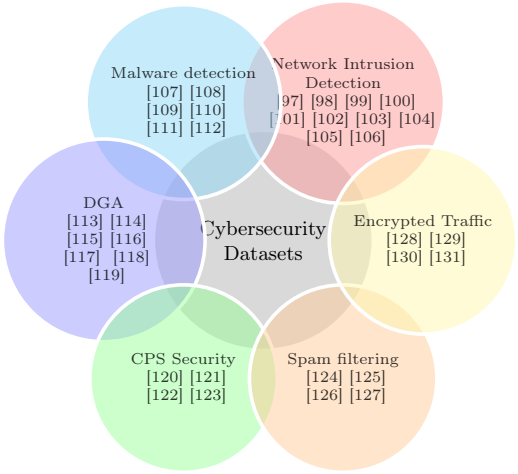

Figure 5: Cybersecurity datasets per application/category.

3.2. Cybersecurity Datasets

Benchmark datasets allow researchers to train, validate, and test the proposed DL-based security solutions. Furthermore, they are essential for reproducing experiments and comparatively evaluating the achieved performance using the same data. In this subsection, we provide an overview of the most prominent datasets employed in cybersecurity, organized per specific application/category (Fig. 5). We have also created a GitHub repository[1] with an extended curated list of available datasets, which we aim to continuously update.

The most commonly used datasets for *network intrusion detection* are KDD Cup 1999 [97] and its evolution NSL-KDD [132]. Both datasets include traffic records that are labeled in five classes: Normal, Probe, Remote to Local (R2L) attacks, User to Root (U2R) attacks, and Denial of Service (DoS) attacks. However, it should be noted that that the underlying network traffic dates back to the year 1998. NSL-KDD was created to mitigate the redundancy and bias problems of its predecessor. Another popular dataset is CTU-13 [98], which contains the raw packet data (PCAP files) and considers 13 scenarios with different botnet attacks. UNB ISCX 2012 [99] was created by the Canadian Institute for Cybersecurity of the University of New Brunswick (UNB), in 2012. Traffic was captured in

[1] https://github.com/mmacas11/Cybersecurity-Datasets

an emulated network environment over a period of 7 days. AWID [100] is a labeled dataset that focuses on 802.11 networks. A small network environment with 10 clients was designed to capture the WLAN traffic in a packet-based format, while 15 specific attacks (e.g., Probe Request, and CTS Flooding) were executed. The CIC-IDS2017 [101] dataset was also created by UNB and contains different types of user profiles (generating background traffic) and multistage attacks (e.g., Heartbleed and DDoS). The CICFlowMeter tool was used to extract 80 traffic features.

CSE-CIC-IDS2018 [102] contains 6 different types of network attacks (i.e., DoS, DDoS, Botnet, brute-force, infiltration, and web attacks) and was generated by using synthetic user profiles to capture abstract representations of network events and behaviors. A victim infrastructure composed of 420 computers and 30 servers was attacked by fifty network nodes. The CICFlowMeter-V3 tool was employed to extract the 84 network traffic features of the dataset. The CIC-DDoS2019 dataset [103] includes many different DDoS attacks carried out through application layer protocols using TCP/UDP. IoT-23 [106] was created in 2019 and consists of 20 captures with malware activity and three captures of benign IoT traffic. TON_IoT was generated in 2019 by Abdullah et al. [105] and includes various types of IoT data (e.g., operation system logs, telemetry data), as well as IoT traffic gathered from a medium-scale network at the Cyber Range and IoT Labs of the UNSW Canberra, Australia. Finally, the LITNET-2020 dataset [104] contains feature vectors generated during 12 attacks on general computers deployed on an academic network. It was collected for a period of 10 months.

Regarding *malware detection and analysis in IoT*, N-BaIoT [107] contains real traffic (115 numerical features) obtained from 9 commercial IoT devices. For each device, the data was collected under normal operating conditions and when several different attacks were launched by BASHLITE. Furthermore, IoTPOT [108] contains 500 IoT malware samples (belonging to four major families) collected by an IoT honeypot. Finally, VirusShare [109] is a repository of malware samples that is continuously updated, providing access to the latest threats discovered.

To detect spam in mobile devices, Genome [110] has 1242 malicious applications gathered from unofficial Chinese marketplaces in 2010 and 2011. The samples are divided into 49 malware families comprising almost all malware categories: Rootkit, Botnet, SMS Trojans, Trojan, Installer, and Spyware. Contagio-Mobile [111] is a blog-like website operating since 2012 that aims to collect malware for different mobile operating systems. Contrary to Genome which contains several samples of each malware family, Contagio provides only a few samples (most commonly one). Finally, AndroZoo [112] currently contains more than 16 million apps collected from several sources. Each app has been analyzed by tens of different AntiVirus products in order to determine if it is malware or not.

The most significant data sources for *domain generation algorithms detection* are several dynamic lists such as the global "Top Sites" published by Alexa Internet [113] for generating benign domain names and OSINT [114], DGArchive [115], 360netlab [116], AmritaDGA [118], and UMUDGA [119] for producing malicious domain names. In particular, Bambenek Consulting provides more than 800 thousand malicious domain names from 50 different families via the OSINT DGA feed. The DGArchive contains more than 30 reverse-engineered DGAs that can be leveraged to generate malicious domain names on an internal network. Similarly, 360netlab and AmritaDGA include 52 and 20 DGA families, respectively. Finally, UMUDGA offers a collection of over 30 million manually-labeled algorithmically generated domains, sorted in 50 malware classes. Apart from the above, the list of the most queried domains based on passive DNS usage provided by Cisco Umbrella [117] is also used for DNS-based detection.

The following four datasets are widely utilized in *industrial control systems.* BATADAL [120] regards a water distribution network composed of seven storage tanks with eleven pumps and five valves, controlled via nine Programmable Logic Controllers (PLCs). The network was created using epanetCPA [133] and the dataset contains 8761 records of 43 variables. SWaT [121] concerns a scaled-down, fully operational water treatment plant and comprises a six-stage process. The entire dataset has 946,722 labeled (attack or normal) records, containing 51 attributes corresponding to the sensor and actuator data. WADI [122] is also available upon request and regards several large water tanks supplying water to consumer tanks. It contains 15 attacks, aiming to stop the water supply to the consumer tanks. It is significantly larger than the SWaT and BATADAL datasets, with 1,221,372 records and 126 features. Finally, the HAI Dataset (HIL-based Augmented ICS) [123] is a collection of data from three physical control systems (i.e., a GE's turbine, an Emerson's boiler, and a FESTO's water treatment system), combined through the dSPACE HIL simulator. On the other hand, the studies focusing on smart grids most frequently employ simulations. To that end, the IEEE X-bus (e.g., 39-bus, 123-bus) is an effective evaluation platform.

The most frequent dataset used for *web spam detection* is WEBSPAM-UK2007 [124]. It is based on crawls of the .uk Web domain performed in May 2007 and includes 105.9 million pages, over 3.7 billion links, and about 114,529 hosts. In order to detect spam in the Twitter social network, the following datasets can be employed: Social honeypot [126] and UtkMI [125]. Social honeypot has 22,223 content polluters together with their number of followings and 2,353,473 tweets, as well as 19,276 legitimate users with the corresponding number of followings and 3,259,693 tweets, collected over 7 months. The UtkMl's Twitter dataset contains 11968 tweets and 8 features (i.e., id, tweet, following, followers, action, is_tweet, location and type), with the 49% of the tweets being spam. Regarding spam detection in short message service (SMS), the SMS Spam Collection v.1 [127] dataset is employed. It is a collection of 5574 spam and legitimate English text

messages distributed in 4827 legitimate messages and 747 spam messages.

The available datasets for *encrypted internet traffic classification* include ISCX VPN-nonVPN [128] and ISCX Tor-nonTor [129]. The former covers 15 popular applications such as Facebook, YouTube, Netflix, etc., which are encrypted using various protocols. The latter contains eight types of traffic—-namely, VOIP, chat, audio-streaming, video-streaming, mail, P2P, browsing, and File Transfer—-from more than 18 representative applications. It uses benign traffic from a VPN project created by Draper-Gill [128]. The Open HTTPS [130] and QUIC [131] datasets can be employed for the performance analysis of dedicated encryption protocols. The first contains full HTTPS raw PCAP files from crawling the top 779 accessed HTTPS websites. The second is a self-collected encrypted dataset of the newly established QUIC protocol.

Cybersecurity research can be challenging due to the continuous technological advances and the required interdisciplinary collaboration. Going beyond simulations, which are useful during the development of the initial proof-of-concepts, researchers need to experimentally test and evaluate their prototypes in suitable platforms. To that end, Table 2 provides some indicative testbeds for experimentally evaluating the proposed systems with real pilots and experiments over bare-metal hardware.

Table 2: Some open access testbeds for cybersecurity.

Testbed	Focus area
DETER [134]	Networked or distributed cyber and cyber-physical systems.
FIT Lab [135]	Wireless sensors and IoT
NITOS [136]	The Outdoor Testbed, the Indoor RF Isolated Testbed, and the Office Testbed.
ORBIT [137]	Cognitive radio networks, future Internet architecture, WiFi networks, inter-layer wireless security and cloud computing.
Fed4FIRE+ project [138]	5G technology, IoT, OpenFlow, Cloud computing, Big Data, wired, and wireless networks.
DRAKVUF [139]	Malware detection
Emulab [140]	Networking and distributed systems
COSMOS [141]	Real-world experimentation on next-generation wireless technologies and applications
EdgeNet [142]	Distributed edge cloud

4. DL applications to Cybersecurity

Deep Learning is playing an increasingly important role in the cybersecurity domain, enabling and facilitating a wide range of applications (Fig. 6). In this section, we address *Q4 (What are the latest successful deep learning-based systems in cybersecurity?)* by critically reviewing state-of-the-art DL-based defense systems for each main application category.

4.1. Network Intrusion Detection

By 2023, it is anticipated that the number of IP-connected devices will be three times larger than the global population, producing up to 4.8 ZB of IP traffic annually, as pointed out by Cisco [117]. This accelerated increase raises overwhelming security challenges. Identifying network attacks is a critical function that should not be overlooked. IDS is widely used for monitoring a network or system for malicious attacks as well as policy violations and can be implemented on a misuse or anomaly basis. Misuse-based techniques search for specific patterns or signatures of attacks in system calls, network traffic, and so forth. However, these techniques demand regular updates to databases storing rules and signatures and, as a consequence, are not useful against zero-day attacks. Anomaly-based techniques focus on knowing normal behavior in order to identify abnormalities (i.e., significant differences from normal traffic). Such techniques can detect unknown attacks, and are difficult to be avoided by the attackers because the normal activity is customized for the particular user, applications, and network being monitored. Anomaly-based techniques, however, may produce high FPRs because they flag any previously unseen traffic or system as a possible malicious attack, even though it might actually be benign. They also must be trained individually for every deployment.

Although most traditional learning techniques, such as NNs, fuzzy logic, and Hidden Markov Models (HMMs), have been successfully applied for intrusion detection, they suffer from their shallow architecture. This leads to some limitations in dealing with the emergence of new technologies and the increased Internet traffic that produces large-scale and multidimensional data making the attack scenarios increasingly more sophisticated. In contrast, DL techniques have demonstrated an outstanding performance in heterogeneous and non-linearity data analysis. Moreover, deep networks can automatically reduce the network traffic complexity to find the correlations among data without human intervention. Unlike shallow ML algorithms, DL approaches can be designed to perform feature extraction and classification tasks together. Different IDS deployment types are shown in Fig. 7.

4.1.1. Software Defined Networks (SDNs) and Wireless Local Area Networks (WLANs)

In SDN, the brain of the system is decoupled from the nodes comprising the network. It is located in a centralized and well-separated entity called the Controller. This entity has control of the entire network and can act at a higher level coordinating all the network nodes in order to evade potential intrusions. Tang et al. [143] used a DNN model consisting of one input layer, three hidden layers, and a softmax layer as an output layer in an SDN environment to detect intrusion activities by classifying traffic flows into normal and abnormal classes. After being trained on the NSL-KDD dataset [132], the model was loaded in the (logically) centralized SDN controller, which received the statistics of the traffic traversing the network, and applied the prediction scheme in order to decide whether a particular flow is affected by malicious

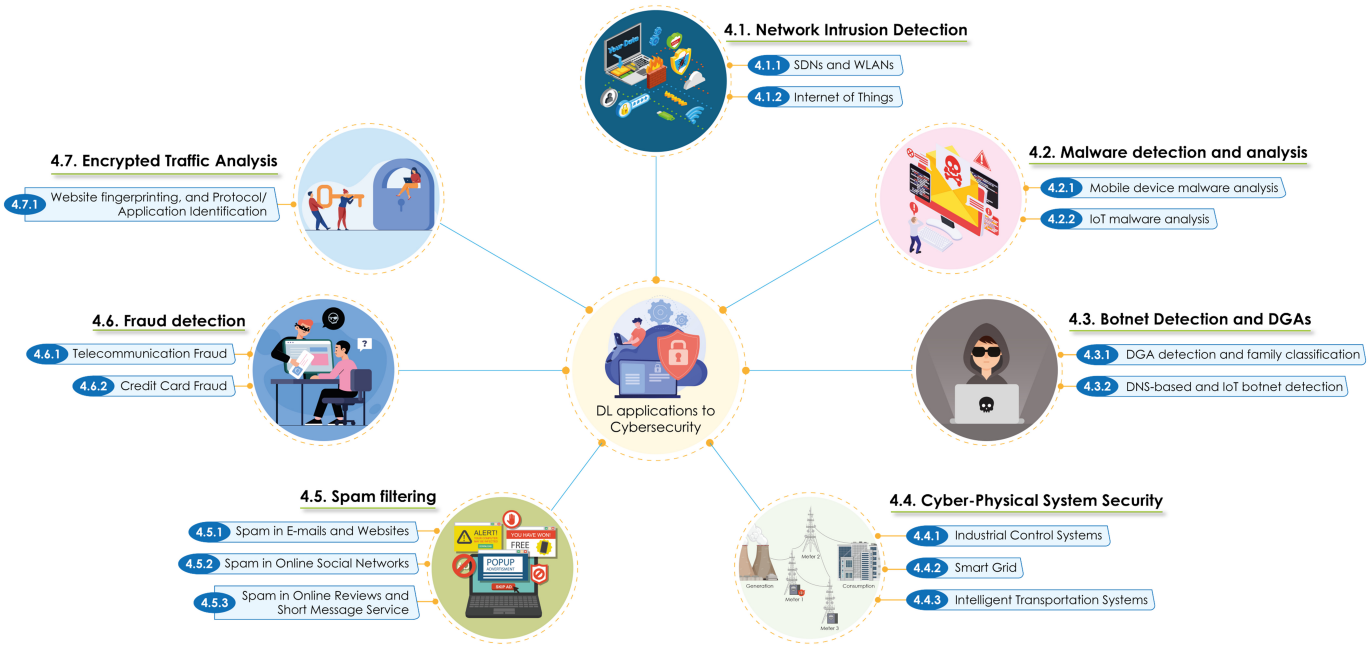

Figure 6: Overview of the most common cybersecuirty applications.

code or not. The experimental results showed that their model achieved a detection accuracy rate of 75.75%, utilizing only six basic flow features. Nonetheless, there was notable evidence of overfitting to the data, suggesting that regularization techniques could enhance results.

DDoS attacks can benefit from existing vulnerabilities in virtualization technologies (e.g., virtual machines and containers) and IoT devices, which can be harnessed as part of a botnet to launch attacks [144, 108]. Elsayed et al. [144] introduced an IDS to combat DDoS attacks in an SDN environment, combining an RNN with AE models. The RNN model was applied to address the loss in data information due to the sequential traffic, and the flow-based features from CICFlowMeter were exploited by employing the CIC-DDoS2019 dataset [103]. In most of the experiments, an AUC of 0.988 was achieved. Due to the limitations that RNNs suffer, the proposed method is unsuitable in scenarios when the network data is characterized by long dependency on time.

With the extensive popularization of WLAN technology used in hardware devices, the IEEE 802.11 protocol-based short-distance transmission wireless network confronts significant security challenges [145, 146]. The application of DL to recognize the attack features and perform wireless network intrusion detection on two imbalanced datasets (i.e., AWID [100] and LITNET [104]) is described by Yang et al. [146]. In order to handle the impact of the imbalanced dataset and the data redundancy on the detection accuracy, a window-based instance selection algorithm "SamSelect" was adopted to undersample the majority class data samples. Then, the stacked contractive AE algorithm [41, 42] was used to reduce the dimensionality of the data samples. Finally, Conditional DBN [146] performed the attack detection. The proposed approach achieved an accuracy of 97.40% and recall and F1 score of 97.60%, and 97.10%, respectively, on both datasets, outperforming LR and SVM. However, more experimental scenarios and the FPR/FNR should be examined and analyzed.

In [145], a Restricted Boltzmann Machine-based Clustered IDS (RBC-IDS) was employed for monitoring critical infrastructures by wireless sensor networks (WSNs). RBC-IDS slightly outperformed the adaptively supervised and clustered hybrid IDS (ASCH-IDS) [147], achieving 99.12% and 99.91% for detection rate and accuracy, respectively, over the KDD Cup 1999 dataset. The detection time of RBC-IDS was approximately twice that of ASCH-IDS, which additionally degrades the performance of IDS. However, the KDD Cup 1999 dataset was generated up a decade ago and may not depict the type of network attack traffic that would be expected in today's WSNs.

4.1.2. Internet of Things (IoT)

The number of client and IoT/mobile devices (i.e., edge devices) is continuously rising, leading to the rapid expansion of the digital substrate serving as a launchpad for automated attacks, which are expected to grow both in scope and in frequency [148]. However, it is necessary to consider that such devices are not able to support the same type of cybersecurity functionality found in enterprise data centers and clouds. The main challenge in the design and implementation of robust attack detection systems for IoT devices regards resource limitations, delay sensitivity, and distribution issues [149, 150]. The enterprise infrastructure and services deployed closer to the end-user devices (i.e., at the perimeter of the enterprise networks) comprise

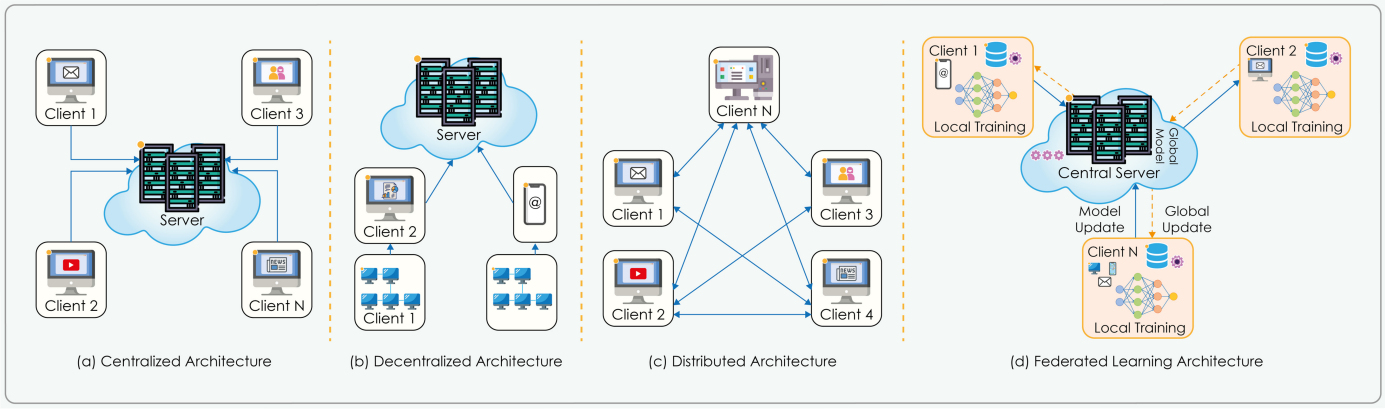

Figure 7: Various Architectures for IDS deployment. Centralized deployment is usually used in small networks due to the involved transmissions to the central server, whereas distributed and decentralized architectures employ inter-agent communication and hierarchical decision-making. Finally, federated learning takes advantage of edge computing.

a definite first line of defense for dealing with the scale and distribution of cyberattacks [151, 152]. The fog/edge infrastructure tier introduces new opportunities to support cybersecurity strategies, since it resides closer to the devices and therefore can see, detect, and react to events more quickly, thus forming a new security perimeter that can contain attacks faster and more efficiently.

In this regard, Abeshu et al. [149] used the self-learning abilities of DL methods to detect cyberattacks by employing fog nodes as data and control processing centers in IoT with a fog platform. Using the NSL-KDD dataset [132], they built stacked autoencoders (SAEs) with two hidden layers to extract hidden features. Then, the obtained features were applied to the test data to extract end features for softmax classification. The experimental results showed that their model reached an accuracy of 99.2% with a detection rate of 99.27% and a false alarm rate of 0.85% on a four-class problem, outperforming shallow learning models. Although the proposed approach seems to be useful to intrusion detection in the fog-to-things computing platform, the dataset used in the evaluation is out-of-date (i.e., it lacks the more recent types of attacks) and so its relevance is questionable.

The Industrial IoT (IIoT) regards the use of many interconnected smart sensors, instruments, and other devices in industrial sectors and applications, including manufacturing and energy management. Yao et al. [153] proposed a hybrid IDS architecture for edge-based IIoT leveraging ML techniques. They divided the IIoT scenario into a central network component and an edge component. Devices with reliable computing power and enough resources such as edge routers were considered as the master nodes, while the industrial equipment of the edge part was regarded as edge nodes. Due to the restricted computing power and resources of edge nodes, they applied the LightGBM algorithm on them, and performed the first intrusion detection task. On master nodes, they employed a CNN structure to perform the second intrusion detection task and enhance the detection accuracy of the overall network. The authors claimed that their proposed IDS could improve detection accuracy and reduce detection time and network resource consumption. Nevertheless, not enough information about the employed datasets (e.g., size and date of creation) and the evaluation metrics for the entire system is included.

Ferdowsi and Saad [154] introduced a distributed GAN-based IDS that enabled the IIoT devices to monitor their neighbors in order to detect intrusions with a minimum dependence on a central unit. The objective of their distributed GAN [155] was to place a discriminator at every IoT device without sharing their local datasets. The principal difference between the distributed IDS and the standalone IDS is that the latter learns to compare a new data point with its own data distribution. Contrary to that, in the proposed distributed IDS, every IoT device could compare a new data point with the distribution of the total data. The SBHAR dataset [156] was used in the experiments. The study reported the achieved performance as 20% higher accuracy, 25% higher precision, and 60% lower false-positive rate than the standalone IDS. Nonetheless, the proposed distributed approach might be vulnerable to iterative generated attacks. Once the discriminator is trained, it is possible to find ways to generate instances capable of bypassing the detection system.

Notwithstanding the significant effort made in annotating IoT traffic records, the number of labeled records is still limited, raising the difficulty in identifying attacks and intrusions. To address this issue, Abdel-Basset et al. [157] introduced a semi-supervised DL approach for intrusion detection called SS-Deep-ID, which consists of a multi-scale residual temporal convolutional (MS-Res) module that finetunes the network capability in learning spatio-temporal representations. The key in the MS-Res module is the dilated causal convolutions (DC-Conv) [158], and a traffic attention module is incorporated to help the network emphasize the most significant features for detecting intrusions. The CIC-IDS2017 [101] and CSE-CIC-IDS2018 [102] datasets were used in the experiments. Despite its efficiency, the proposed model suffers from the

following limitations: (i) It is not clear whether it can preserve its effectiveness in scenarios with a massive amount of IoT traffic data; (ii) Distributed training, which is essential in intelligent IoT applications, is not analyzed.

Given that the fifth generation (5G) mobile communications are still in their infancy, several security gaps that can be exploited in intrusion attempts are expected. Rezvy et al. [159] introduced a DL model for intrusion classification and prediction in 5G and IoT networks. In the proposed model, an AE was used to provide a compressed representation of the input space and a dense NN functioned as the supervised classifier to distinguish intrusive events (i.e., impersonation, flooding, and injection) from the normal ones. The study reported its five-fold cross-validation performance over the imbalanced AWID dataset [100] as overall 99.9% accuracy with unknown variance. The lowest performance of 99.42% corresponded to the flooding category. Although the work achieved promising results, it, unfortunately, did not present how the multi-class imbalanced problem was addressed.

Energy-efficient IoT is known as Green IoT. With the development of 5G mobile communications, Green IoT has attracted considerably more attention. Nie et al. [160] developed an IDS based on the DDPG algorithm [59]. Their method first extracts the statistical features of prior network traffic to capture the trends of traffic flows and perform traffic prediction. Then, the developed traffic predictors are employed in combination with a suitable threshold to enable intrusion detection. The CIC-DDoS2019 dataset [103] was used to evaluate the proposed model. The achieved performance included 99% precision with an FPR of 1.21%, outperforming PCA and Sparse Regularized Matrix Factorization (SRMF) [161]. With the 5G cellular applications, traffic will show more complex features, raising a more significant challenge to network traffic prediction [162]. How to identify and extract significant features for improving the accuracy must be analyzed.

Social IoT (SIoT), which combines users' social behaviors and physical IoT [163], can provide ubiquitous Internet access for users. As a strategy to mitigate the rapid increase of resource congestion, collaborative edge computing (CEC) has become a paradigm for covering the demands of IoT. To ensure the security of CEC, the authors in [163] proposed a GAN-based IDS for extracting low-dimensional features from original network flows. The CIC-DDoS2019 [103] and CSE-CIC-IDS2018 [102] datasets were used to evaluate the proposed methods for binary and multi-class classification, showing a high achieved precision. However, it is unclear how they handle the spatio-temporal patterns present in networks data, given that the generator and the discriminator network of the employed GAN were constructed using FFNNs.

There are scenarios in which it is useful or even mandatory to isolate different subsets of training data from each other [164, 165]. Federated learning systems fully embrace this principle [166, 164, 167]. Wang et al. [168] proposed a federated anomaly detection system employing DRL to enable multiple parties to jointly learn an accurate deep model, while preserving the data itself local and confidential. Another significant advantage of using federated distribution instead of a centralized architecture is that unexpected intrusions in one or even several client systems do not affect the whole system. However, since federated learning employs secure aggregation to protect the confidentiality of the local models, it cannot detect anomalies in the participants' contributions to the joint model. For instance, a compromised participant can submit a malicious model that, apart from being trained for the task at hand, has been also tampered with to include backdoor functionality [169]. Designing robust federated learning systems is an exciting and vital topic for future research.

Findings. Table 3 presents a summary of the DL-based cybersecurity systems for network intrusion detection. We note that CNNs and AEs are the most commonly employed deep neural networks. This can be attributed to the former's capability of processing data from multiple arrays and the latter's suitability for semi-supervised/unsupervised learning. Although a large part of the analyzed works focuses on algorithms that improve detection results, most studies have neglected to evaluate the reliability of the benchmark datasets. Furthermore, some features are less effective in the detection of new variants of attacks. As a consequence, the reported performance evaluation that is based on detecting older attack patterns present in well-known datasets can be misleading. Thus, there is a need for the construction and usage of more current and up-to-date datasets.

4.2. Malware detection and analysis

Techniques for malware (short for malicious software) analysis can be divided into three groups [170]: 1) Static, 2) Dynamic, and 3) Hybrid. Static malware analysis is conducted by reverse-engineering the malware binary to its assembly code and then examining the included instructions without actually executing it. Nevertheless, such a technique can be effortlessly defeated by evasion techniques like obfuscation and embedding of syntactic code errors. Dynamic malware analysis is carried out by executing the malware in a controlled sandbox environment in order to observe its behavior and effect on the host system. Although resource-intensive, it is effective against malware obfuscation. However, sophisticated malware can avoid dynamic analysis via detecting whether it is being run inside a sandbox or a controlled environment and, based on this, deciding not to exhibit any malicious behavior. The hybrid technique combines the respective advantages of both the static and dynamic analysis methods.

In practice, these techniques are time-consuming and involve a manual component, which makes them hard to

Table 3: Selected studies focusing on DL-based security methods for Network Intrusion Detection.

Software Defined Networks and Wireless Networks (Section 4.1.1)				
Year	**Authors**	**Deep Model**	**Dataset**	**Advantages and Limitations**
2016	Tang et al. [143]	- DNN	- NSL-KDD [132]	- The model is overfitting.
2019	Rezvy et al. [159]	- AE - Dense NN	- AWID [100]	- No explanation about how the multi-class imbalanced problem was addressed.
2019	Otoum et al. [145]	- RBM	- KDD Cup 1999 [97]	- Outdated dataset.
2020	Yang et al. [146]	- Contractive AE - Conditional DBN	- AWID [100] - LITNET [104]	- Needs more experimental scenarios and the FPR/FNR should be analyzed.
2020	Elsayed et al. [144]	- RNN - AE	- CIC-DDoS2019 [103]	- Not suitable for scenarios when the network data is characterized by long dependency on time.
Internet of Things (Section 4.1.2)				
Year	**Authors**	**Deep Model**	**Dataset**	**Advantages and Limitations**
2018	Abeshu et al. [149]	- SAEs	- NSL-KDD [132]	- The dataset lacks more recent types of attacks.
2018	Diro et al. [150]	- LSTM	- UNB ISCX 2012 [99] - AWID [100]	- The resource limitations of IoT devices are addressed.
2019	Yao et al. [153]	- LightGBM - CNN	- Unknown	- Not enough information about the datasets used.
2019	Ferdowsi et al. [154]	- Distributed GAN	- SBHAR [156]	- This approach might be prone to iterative generated attacks.
2021	Abdel-Basset et al. [157]	- DC-Conv - C-Conv - Attention module	- CIC-IDS2017 [101] - CSE-CIC-IDS2018 [102]	- Experiments with a massive amount of IoT traffic data are needed. - Distributed training is not analyzed.
2021	Nie et al. [160]	- DDPG	- CIC-DDoS2019 [103]	- 99% precision with an FPR of 1.21%.
2021	Nie et al. [163]	- GAN	- CIC-DDoS2019 [103] - CSE-CIC-IDS2018 [102]	- It is unclear how the spatio-temporal patterns present in network data are handled.
2021	Wang [168]	- DDPG	- Simulated IoT environment	- In order to prevent privacy leakage, abnormal actions of users were detected on time.

scale as the number, sophistication, and complexity of the malware increase. Many ML-based techniques have been proposed that address scalability by automating various steps of the malware detection and categorization processes. Nevertheless, their effectiveness is restricted by high FPRs that render them inaccurate. To address this issue, researchers have recently focused on DL-based systems. DL-based methods have been demonstrated to classify malware at a much better speed than human analysis with high accuracy rates. These methods can be used in malware analysis to comb out the suspicious binaries that can then be examined and validated by an expert human.

4.2.1. Mobile malware analysis

Globally, the total number of mobile devices is expected to grow from 8.8 billion in 2018 to 13.1 billion by 2023, 6.7 billion of which are anticipated to be smartphones [117]. Smartphones have become attractive due to the accessibility of office applications, Internet, vehicle guidance employing location-based services, and games along with traditional services such as voice calls, SMS texts, and multimedia services. The Android smartphone OS has captured a significant market share because of its open architecture and the high acceptance of its application programming interface (APIs) in the developer community, leaving its major competitor iOS far behind [171]. The increasing popularity of Android devices and associated monetary gains have drawn malware developers' attention, raising a notable increase in Android malware apps. Recent studies reveal that a new malicious application for Android is introduced every ten seconds [172]. The common types of Android malware apps include, but are not limited to, trojans, spyware, backdoors, worms, botnets, adware, and ransomware, which use different techniques to infiltrate the users' systems. Dynamic payload, code obfuscation, drive-by download, and repackaging popular applications are some examples of abundant methods leveraged by malware authors to bypass the existing protection mechanisms.

Consequently, the exclusion of those malicious Android applications is highly requested by app markets. Commonly, the employed methods utilize ML algorithms to detect malware. However, their performance relies heavily on human-engineered features, which limits their generalizability. On the contrary, DL removes the necessity of a domain expert selecting features because it automatically selects features by training. Thus, it is no wonder that DL methods have been widely applied in Android malware detection and categorization, recently.

Starting from the raw sequences of the app's API method calls derived from the DEX (Dalvik Executable) file, Karbab et al. [173] introduced MalDozer that aimed to detect malware and its associated class using a CNN architecture that received as input a sequence of vectors obtained using the Word2Vec word embedding technique [174]. The experiments employed several different datasets, including Malgenome [175] (1K data points), Drebin [176] (5.5K data points), the MalDozer dataset (20K data points) that contained data collected from the Internet (e.g., Virusshare [109] and Contagio Minidump [111]), and a merged dataset of 33K malware data samples. Apart from that, around 38K of benign apps were downloaded from Google Play Store [177] and were used during evaluation. The experimental results showed that the F1 scores achieved for the class attribution and detection tasks were between 96%-98% and 96%–99% respectively, whereas the false positive rate was in the range of 0.06%–2%. Although MalDozer appears to be useful in malware detection and class classification, the Drebin [176] and MalGenome [175] datasets are out-of-date and could negatively influence the resilience of the proposed approach facing new, sophisticated and obfuscated malware families.

MobiDroid [178] is a lightweight Android malware detection system that can be executed on the users' mobile devices in real-time and relies on the information from APK files. The first stage of the system is feature preparation which is based on decoding each APK file into original resources and .smali files. The resulted feature vector is then fed to a CNN classifier. Finally, the Android app prediction is performed with the help of a migrated and quantized detection model. The study used a dataset of 50,000 Android apps, 29,010 of which were malware samples collected from various sources including Drebin [176], Genome [110], Contagio [179], Pwnzen, and VirusShare [109], whereas the rest were benign apps crawled from Google Play Store [177]. The authors reported a performance of 97% accuracy and demonstrated the fast reactive (less than 10 seconds) detection service provided directly on mobile services. A similar approach was proposed in [180], but instead of decompiling APK into source code, like smali code, the manifest properties and API calls were extracted and vectorized directly from the binary code.

Multivector malware usually hides under legitimate third-party software and can be easily turned into an executable file extension, making its detection extremely challenging. Haq et al. [181] proposed a hybrid DL system that leverages CNN and Bi-LSTM to identify persistent malware. The data used consists of 30831 legitimate Android APK's extracted from Androzoo [112] and 8011 Malware APK's obtained from AMD dataset [182]. Although the constructed dataset was highly imbalanced, the proposed model achieved the best performance with a precision rate of 99.39% and an FPR of 1.9% using tenfold cross-validation and 905 features.

4.2.2. IoT malware analysis

Existing vulnerabilities in IoT devices that could be employed for malware injection are related to application security, authorization, and authentication. Apart from these, physically tampering with the IoT devices for software modification and misconfiguration of security parameters could also enable attackers to inject malicious code. Traditional approaches, such as classical ML-based malware detection mechanisms, have been applied during the last decades. However, it has already been proven that they have low accuracy and limited scalability for malware detection and analysis in IoT devices [183, 184, 67]. DL is a promising approach for IoT devices due to some of their specific properties. For example, IoT devices produce a sheer amount of data required by DL techniques to bring intelligence to the systems. Furthermore, the heterogeneous data generated by IoT devices is better utilized by DL techniques, which enable the IoT systems to make informed, fast, and intelligent decisions.

A DL-based method to detect the Internet of Battlefield Things (IoBT) malware via the device's OpCode sequence was proposed by Azmoodeh et al. [183]. The Class-Wise Information Gain technique was used to select the top 82 features and, at the same time, overcome the problem of the imbalanced dataset. Each sample's selected features were converted into a Control Flow Graph which was used to classify IoT malware and goodware applications applying deep eigenspace learning and CNN techniques. The experimental results showed that the proposed system achieved 99.68% accuracy in detecting malware samples, with precision and recall rates of 98.59% and 98.37%, respectively. Unfortunately, the study does not report any information about the hyper-parameters used during the training phase of the network.

In order to detect IoT malware in embedded Linux-based IoT devices, Jeon et al. [184] introduced DAIMD that performed dynamic analysis in nested cloud-based VM environments and learned behavior images compressed with behavior data based on CNNs. The study stated a performance of 99.28% with 0.63% FPR. However, the dataset used in the experiments includes only 1,401 samples which are not cross-validated. Moreover, sending data to the cloud for inference or training may incur additional queuing and propagation delays from the network and cannot satisfy strict end-to-end low-latency requirements needed for real-time.

To address the "lack of labeled information" for training the detection model in pervasive IoT devices, Vu et al. [67] developed a system based on deep TL named MMD-AE. The labeled and unlabeled data were fitted into two AE models with the same network structure. Moreover, the Maximum Mean Discrepancy (MMD) metric was used to transfer knowledge from the first AE to the second AE. This study used nine IoT attacks from N-BaIoT [107] for the evaluation. Overall, the IoT attack detection task in the target domain achieved an AUC score of 0.937. A limitation in this approach is the excessive time for training the model compared to baseline methods.

In [185], Dib et al. defined a multi-dimensional classification approach employing LSTM and CNN models to first extract and then combine the features of string- and image-based representations of the executable binaries towards accurate IoT malware classification and family attribution. The proposed model was evaluated over 74,429 IoT malware binaries from well-known online malware repositories together with a special-purpose IoT honeypot (IoTPOT [108]). The results were 99.78% accuracy and 99.57% F-score, outperforming classifiers based on a single data modality.

Table 4: Selected studies focusing on DL-based security methods for Malware detection and analysis.

Mobile devices (Section 4.2.1)				
Year	**Authors**	**Deep Model**	**Dataset**	**Advantages and Limitations**
2018	Karbab et al. [173]	- CNN	- Drebin [176] - MalGenome [175] - Maldozer [173]	- Datasets are out-of-date. - This work introduced MalDozer dataset
2018	Kim et al. [186]	- DNN (Multimodal)	- VirusShare [109] - Malgenome [175] - Google Play Store [177]	- 98% accuracy and 99% F1 score.
2019	Feng et al. [178]	- CNN - Parameter Quantization	- Drebin [176], Genome [110] - Contagio [179], Pwnzen - VirusShare [109] - Google Play Store [177]	- 97% accuracy and fast reactive (less than 10 seconds) detection service provided directly on mobile services.
2020	Feng et al. [180]	- LSTM/GRU - Bi LSTM/GRU, CNN - Parameter quantization	- Drebin [176], Genome [110] - Contagio [179], Pwnzen - VirusShare [109] - Google Play Store [177]	- Extension of previous work.
2021	Haq et al. [181]	- CNN, Bi-LSTM	- Androzoo [112], AMD [182]	- Highly imbalanced dataset.
IoT devices (Section 4.2.2)				
Year	**Authors**	**Deep Model**	**Dataset**	**Advantages and Limitations**
2018	Azmoodeh et al. [183]	- CNN	- IoT App store [187]	- Does not report any information about the hyper parameters used during the training phase.
2020	Jeon et al. [184]	- CNN	- Live	- Limited set of samples considered. - May not achieve a good accuracy and time response for real-time IoT malware detection.
2020	Vu et al. [67]	- AEs, TL	- N-BaIoT [107]	- Excessive training time compare to baseline methods.
2021	Dib et al. [185]	- LSTM, CNN	- IoTPOT [108], VirusShare [109]	- 99.78% accuracy and 99.57% F-score.

Findings. Since deep learning can automatically find correlations in the data, it is a promising technology for developing novel malware detection methods, particularly for detecting zero-day malware. Note that the final stage in an ML-based malware detection system's life cycle is evaluating and validating the learned model via large-scale deployments in real-world environments and settings. This is a challenging task that requires manual analysis and human intervention, consuming considerable human effort by malware analysts. Most existing works in the literature do not proceed to this type of experimental validation, relying on performance evaluation using one or more well-known datasets. However, for the proposed systems to find practical commercial applications in the industry, this type of trial deployments must not be overlooked. Table 4 summarizes the examined deep learning-based research in malware detection.

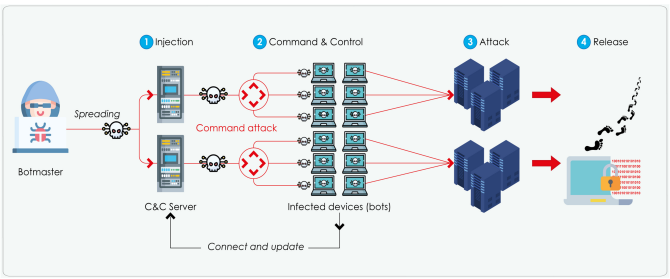

Figure 8: The basic botnet life cycle contains four phases: 1) injection (i.e., spreading bots by injection), 2) Command & Control (i.e., bots are ready to receive commands and launch an attack), 3) attack (i.e., the botmaster send the command for launching an attack), and 4) release (i.e., the botmaster removes his footprint or distributed source code).

4.3. Botnet detection and DGAs

According to [188], the global Botnet Detection market size is expected to reach US$965.6 million by 2027, from US$207.4 million in 2020, at a compound annual growth rate (CAGR) of 24.0% over the period 2021-2027. A *botnet* is a software program that manages computers (or other devices) for malicious intentions. Bots are small scripts created for carrying out particular automated tasks [189] and are operated by one or a small group of collaborating attackers known as "botmaster(s)" [190]. Fig. 8 shows a typical botnet's life-cycle. Botnets rely on DGAs to connect to their command and control (C&C or C2) server. These DGAs periodically produce many algorithmically generated domains (AGDs), which serve as trust points for the botnet [191, 192, 84, 193]. Although the bots query all of the AGDs, only the ones registered by the botmaster in advance resolve to valid IP addresses. In this way, blocking the bot's connection attempts to its C2 server is more complex than using fixed IP addresses or fixed domain names, and conventional techniques such as blacklisting or sinkholing are becoming less efficient. Traditional ML methods for DGA detection based on domain name strings rely on the extraction of predefined, human-engineered lexical features [194, 195]. As a consequence, the maintenance of such ML systems is rather labor-intensive. Today, the DL techniques for detecting DGAs learn features automatically, thereby potentially bypassing the human effort of feature engineering.

4.3.1. DGA detection and family classification

To perfom DGA detection and family classification, Woodbridge et al. [193] proposed a data-driven approach using an LSTM model trained on data collected from the Alexa (top 1 million) whitelist [113] and approximately 750,000 DGA domain names from the Bambenek Consulting blacklist [114], including thirty DGA malware families. The LSTM network for binary classification (DGA or

not-DGA) was extended to an LSTM network with a softmax layer to perform multi-class classification (i.e., detect the DGA class) and was compared with an One-versus-All Random Forest (OVA-RF) approach, including Alexa [113] as a benign family. The experimental results showed that the LSTM approach outperformed the OVA-RF approach for all families; however, it failed to identify some of the families correctly due to their insufficient representation in the dataset. As a potential solution, the authors proposed to train a DGA classifier that assigns domain names to super-families, effectively making the multi-class classification problem more straightforward and resulting in higher predictive accuracy scores.

The study by Tran et al. [192] proposed the use of a cost-sensitive learning algorithm to train an LSTM network for DGA family classification that takes class imbalances into account. Specifically, they employed a hierarchical classifier architecture. The paper used the data collected from Alexa (whitelist) [113] and Bambenek Consulting (blacklist). The authors claimed that their approach allows them to provide an improvement of at least 7% in terms of macro averaging recall, precision and F1-score compared to the CS-NN, CS-SVM, CS-C4.5, Weighted Extreme Learning Machine, HMM, C5.0 DT and the earlier study of Woodbridge et al. [193]. Still, some families are not accurately classified at all by this approach, such as the *locky* family(ransomware malware), which was not included in the work of Woodbridge et al. [193].

Many DGAs use English wordlists to generate plausibly clean-looking domain names, making automatic detection difficult. Curtin et al. [191] applied RNN architecture for DGA domain detection using a smashword score as a measure of domain resemblance to English words. The proposed DGA detection model split the input data into subdomain, domain, and top-level domain (TLD). The subdomain and domain were fed into individual RNN (with LSTM cells) models to predict the next character in them and these predictions were combined via a generalized likelihood ratio test (GLRT). Then, the output was combined with WHOIS data and the one-hot encoded TLD into the final logistic regression model to predict whether the input domain is malicious or not. The employed dataset included 41 DGA families plus not-DGA data, totaling 2.29 million domain names (1.01 million were not-DGA, and 1.28 million were DGA), which were used in several experiments. The experimental results demonstrated that the proposed approach provides better performance at lower false-positive rates on DGA families with high *smashword* scores, such as the difficult *matsnu* and *suppobox* families.

A heterogeneous DNN framework for extracting the local features of domain names and a self-attention based Bi-LSTM (SA-Bi-LSTM) for extracting further global features were presented by Yang et al. [84]. The authors used benign samples from the top 1 million domain names dataset collected by Cisco [196], whereas the DGA samples were from the dataset collected by 360netlab [116]. The focal loss function [197] was introduced to mitigate the imbalance of the samples' quantity in the training phase. Although HDNN achieved a better DGA detection, the average accuracy rate was still not higher than 90%. According to the authors, the detection of DGA domain names can be further improved by incorporating more perspectives, such as the side information of DNS request behaviors. Moreover, the proposed approach is computationally intensive, which limits its application in the real world.

With the development of beyond 5G (B5G) mobile networks, issues that threaten security and privacy have risen. To detect the malicious domain names, Xu et al. [198] used a hierarchical Bidirectional LSTM (H-BiLSTM) model, which was trained with data collected from Alexa [113] and 360netlab [116]. The experimental results showed that the proposed model with three layers in the hierarchy achieved 96.1% precision, outperforming traditional algorithms (i.e., SVM, DT, LR, and NB). However, when the number of layers increases, the computational complexity of the model increase too, decreasing the precision rate.

4.3.2. DNS-based and IoT botnet detection

DNS data carry rich traces of Internet activities and is a powerful resource in fighting against malicious domains. DNS-based detection techniques rely on particular DNS information generated by a botnet. In order to access the C&C server that is typically hosted by a Dynamic DNS (DDNS) provider, the bots send DNS queries. Therefore, it is possible to identify botnet DNS traffic and, ultimately, traffic anomalies via DNS monitoring. To detect whether domain names and IP addresses are benign, malicious, or sinkholes, Lison et al. [194] used a DNN architecture. The model was trained on a large passive DNS database provided by Mnemonic [199] (not freely available to the public). Rather than employing the domain name as the only input to the NN, numerical (e.g., the lifespan of the DNS record, and number of TTL changes) and categorical (e.g., ISP associated with the IP address or the TLD) features were also used as inputs. While the domain names were fed into a RNN with Gated Recurrent Units, the categorical features were fed into an embedding layer and finally all three inputs were fed into dense feed-forward layers. The authors claimed that the model was capable of detecting 95% of the malicious hosts with a false positive rate of 0.1%. Nevertheless, their model takes much time to converge, which can be a problem in a real environment.

To detect attacks launched from IoT bots, Meidan et al. [107] proposed a network-based approach that employs a data collection, feature extraction, and deep AEs stage, combined with continuous monitoring. In the first stage, real traffic data (in PCAP format) was collected from IoT devices that were connected via Wi-Fi to several access points. In the second stage, whenever a packet arrived, a behavioral snapshot of the hosts and protocols that communicated this packet was taken. The snapshot obtained the packet's context by extracting 115 traffic statistics over several temporal windows. The behavioral snapshots of benign IoT traffic were used in the third step to train a

deep autoencoder for each IoT device. The main idea is that after the model is trained with normal traffic, it is expected to reconstruct it efficiently. However, when the model receives anomalous input, it should not be able to reconstruct it equally well, thus leading to higher reconstruction errors. The experimental results showed that the method succeeded in detecting every single attack launched by every compromised IoT device with a TPR of 100% and an FPR of 0.007±0.01. Although this approach achieved a very high detection rate, its scalability is questionable given that it requires a separate NN for modeling the behavior of each IoT device-type.

In [189], Kim et al. proposed a flow-based botnet detection system using DL to cope with the periodicity of traffic flows, consisting of three stages: data pre-processing, anomaly scoring, and anomaly detection. In the first stage, every flow sorted in chronological order is aggregated to obtain statistic features within the windows, which are then fed to the second stage, an RVAE. The RVAE model produces anomaly scores by comparing the input with the model's output (i.e., the reconstructed input). Lastly, based on the calculated anomaly scores, the anomaly detection function classifies individual connections into either Malicious or Non-malicious. The CTU-13 dataset [98] was employed during evaluation, showing good detection performance and demonstrating generalization capabilities. However, the sequences used in this work were created by aggregating NetFlows on their source IP and then each sequence was reduced by summarizing NetFlows, thus inducing a non-negligible loss of data which may break the underlying temporal patterns.

To detect domain name spoofing attacks in smart cities from DNS traffic, Vinayakumar et al. [190] proposed cost-sensitive deep learning architectures (i.e., RNN, LSTM, GRU, IRNN, B-RNN, B-LSTM, B-GRU, and B-IRNN) combined with Siamese Neural Networks (SNNs) [200]. Although the experimental results revealed substantial improvements in terms of F1-score, speed of detection, and false alarm rate compared to other DL-based DGA approaches (i.e., LSTM, RNN, and GRU), the proposed models were not able to correctly recognize some DGA families.

DNS homograph (DNSH) attacks are a form of phishing that is used by an attacker to make the domain name look very similar to a trusted domain name (e.g., netflixlife.com → netflixlife.com). In order to detect randomly generated domain names and domain name system homograph attacks, Ravi et al. [201] focused on several DL-based SNNs and DL-based cost-sensitive models. The SNNs accept a pair of domain names and have identical DL subnetworks for each input. The Euclidean distance between the fully-connected layer outputs is computed and then passed through a sigmoid activation function to determine similarity (i.e., similar or spoof and dissimilar or not-spoof). Moreover, cost-sensitive classification using several DL models is performed for DGA domain name categorization. Experimental validation used four datasets (i.e., HDN [202], HPN [202], IDFC [192], and AmritaDGA [118]). The best performance was achieved using the Bi-LSTM with an accuracy of 99%. However, a major limitation of the proposed approach appears when an attacker uses a spoofed domain name that has more than one unknown character, falsely resulting in greater distance than the predefined threshold.

Findings. RNNs (e.g., LSTM, GRU, and Bi-LSTM) can efficiently handle sequential, time-dependent, and high-dimensional massive data. For that reason, the majority of the works in this category employed this type of NNs. Additionally, the employed dataset plays an essential role in developing DGA domain detection systems using ML/DL. By accurately selecting the data involved in the analysis, it is possible to boost the accuracy and generally increase the performance. Thus, it is necessary to use more representative and up-to-date DGA datasets, e.g., UMUDGA [119]. Table 5 presents the DL-based security systems for Botnet Detection and DGAs.

4.4. Cyber-Physical System (CPS) Security

CPSs comprise a new generation of complex systems whose normal operation depends mainly on the robust communications between their cyber and physical components. The CPS market is expected to grow by 9.7% annually, reaching US$9,563 million by 2025 [204]. These systems have been proven instrumental to various sectors and have been widely implemented in several industrial environments, such as electrical power grids, oil refineries, water treatment & distribution plants, and public transportation systems. As the deployment of IoT and 5G/6G mobile communications is undergoing an exponential increase, the rise in the use of CPSs comes as no surprise. Simultaneously, CPSs, IoT and 5G/6G also increase the likelihood of cybersecurity incidents and vulnerabilities [205]. From the cybercriminals' perspective, attacking CPSs is a unique opportunity to provoke maximum damage with minimum effort [206]. As a result, many elaborate techniques have been applied to stealthily exploit the CPSs of essential sectors, including smart cities [207], energy networks [208], and supply chain management [209].

Since the services provided by such systems are essential for the well-being of the community, CPSs can be classified as Critical Infrastructures [205]. Therefore, they must be robust and flexible against cyber-attacks. Any attack that could compromise or interrupt the provided services would result in severe consequences for the public safety and order, the economy and the environment. Hence, the ability to detect sophisticated cyber-attacks on the increasingly heterogeneous nature of the CPSs, amplified by the arrival of IoT and 5G/6G, has become a critical task. Regrettably, developing a well-defined security model of the complex physical process is not a trivial task [83, 210]. It

Table 5: Selected studies focusing on DL-based security methods for Botnet Detection and Domain Generation Algorithms.

DGA detection and family classification (Section 4.3.1)				
Year	**Authors**	**Deep Model**	**Dataset**	**Advantages and Limitations**
2016	Woodbridge et al. [193]	- LSTM	Alexa top 1M domain names [113] and 750,000 DGA domain names from the Bambenek Consulting blacklist.	- Pioneer work in applying DL for DGA detection and family classification.
2018	Tran et al. [192]	- LSTM	88,347 benign domains collected from Alexa [113], and 81,484 DGA domain names from the Bambenek.	- Suitable for DGA families with a limited representation in the data - The collected dataset is publicly available.
2019	Curtin et al. [191]	- LSTM	1.02M benign domains collected from Alexa [113] as well as OpenDNS and approximately 1M malware domains spread over 22 distinct types of malware obtained from DGArhive, Andrey Abakumovs DGA repository, Johannes Baders DGA implementations and Sinkholed domains collected from public WHOIS registration.	- Variety of experiments were performed.
2020	Yang et al. [84]	- SA-Bi-LSTM	Benign samples from the top 1 million domain names dataset collected by Cisco [196] and the DGA samples were from the dataset collected by 360netlab [116].	- Computationally intensive.
2021	Xu et al. [198]	- H-Bi-LSTM	Alexa top 1M domain names [113] and 360netlab [116].	- With more than three layers in the hierarchy, the precision rate falls drastically.
DNS-based and IoT botnet detection (Section 4.3.2)				
Year	**Authors**	**Deep Model**	**Dataset**	**Advantages and Limitations**
2017	Lison et al. [194]	- GRU	171 million distinct domain names and 17 million IP addresses extracted from Mnemonic.	- The model takes a lot of time to converge.
2018	Spaulding et al. [195]	- CNN - LSTM	Around 1,5M benign samples from Alexa [113], DMOZ directory, and Agten et al. [203]'s dataset, and around 40k malicious records from Openphish, malwaredomain-list.com, and Agten et al. [203]'s dataset.	- AUC value of 0.950.
2018	Meidan et al. [107]	- AEs	Simulated IoT environment	- The scalability of the model is questionable
2020	Kim et al. [189]	- RVAE	CTU-13 dataset [98]	- The data preprocesing stage induces a non-negligible loss of data.
2020	Vinayakumar et al. [190]	- SNNs - RNNs	AmritaDGA [118]	- High model complexity.
2021	Ravi et al. [201]	- SNNs - RNNs - FNNs	HDN [202], HPN [202], IDFC [192], AmritaDGA [118]	- It is not suitable when an attacker uses a spoofed domain name that has more than one unknown character.

demands in-depth knowledge of the system and its implementation, which takes significant time and cannot scale properly to large-scale and complex systems. An alternative strategy that has recently received attention uses deep learning techniques to build more intelligent and powerful methods that leverage big data to identify intrusions and anomalies.

4.4.1. Industrial Control Systems (ICSs)

In ICSs, different methods based on DL have been proposed for detecting both attacks and faults [211, 212, 213, 214, 83, 215]. The attack types include injecting false control commands, spoofing sensor values, and altering communicating traffic packets. Goh et al. [211] applied LSTM and Cumulative Sum (CUSUM) to detect anomalies on the first stage of the SWAT dataset [121]. They reported the recognition of nine out of ten attacks with four false positives. Unfortunately, this study did not provide a comprehensive analysis of the stability of the results. In [212], a comparative study of two anomaly detection techniques, namely DNN and OC-SVM, was conducted. The study was carried out on all six stages of the SWaT dataset and achieved 92% and 98% precision for SVM and DNN, respectively. However, 23 out of the 36 attacks had a detection recall of zero for DNN and only slightly better for OC-SVM, leading to a low F1 score for both models. Additionally, the proposed framework is resource-demanding and complex.

Similarly, Kravchik et al. [213] used two deep learning models, namely 1D-CNN and LSTM, for detecting attacks on the SWAT testbed [121]. They reported that the proposed system reached average rates of 96.8%, 79.1%, and 87.1% for precision, recall, and F1 score, respectively. However, the attack detection was performed separately at each stage, with no way of learning inter-stage dependencies. Contrary to that, Macas et al. [83] applied an Attention-based Convolutional LSTM Encoder-Decoder (ConvLSTM-ED) model on the SWaT testbed [121] in its entirety (i.e., all six subsystems). Thus, their framework was able to model both inter-sensor correlation and temporal dependencies of multivariate time series. This study reported 96.0%, 81.5% and 88.0% for precision, recall, and F1 scores respectively. In [214], the authors applied 1D convolution and autoencoders to detect anomalies (cyber-attacks) using the physical state of the system as measured by the sensors. Among the three employed datasets (namely, SWaT [121], BATDAL [120], and WADI [122]), the experimental evaluation of the proposed model reported higher performance results for the SWaT dataset (89.0%, 80.3%, and 84.4% for precision, recall, and F1 score). A drawback of this research is that it requires the manual setting of a threshold to detect attacks. Xie et al. [215] proposed a hybrid NN architecture that relies on CNN and RNN for anomaly detection in CPSs on the

SWaT dataset [121]. Although the performance of their framework reported a high precision, they do not consider that many SWaT features do not have the same distribution in training and testing data [121]. Features like this create a lot of false positives and, therefore, should be excluded from modeling.

Lu et al. [216] proposed the population extremal optimization-based deep belief network detection method (PEO-DBN), where the PEO algorithm [217] was employed to determine the DBN's parameters. Furthermore, the performance in cyber-attack detection was enhanced by introducing a majority voting scheme for aggregating the proposed PEO-DBN, leading to the creation of EnPEO-DBN. The study used two SCADA network datasets of a gas pipeline system [218] and a water storage tank system [218] in several experiments. Although simulation results showed that PEO-DBN and EnPEO-DBN outperform the accuracy of other methods such as SVM, DT, the ensemble of SVM, and the ensemble of DBN [219], the whole process of the PEO algorithm has high computational complexity. The authors suggest using a surrogate-assisted model to address this problem.

Considering that with the advent of 5G and the increased use of CPSs in the industry the attack surface will become broader, the authors in [220] developed a framework based on the ResNet-50 model to mitigate such attacks. This work used the Telecom Italia's dataset [221] for experimentation. Even though the proposed model achieved higher than 91% attack detection accuracy, its complexity is not analyzed. Therefore, it is not known how well it could scale well to a real-time environment.

4.4.2. Smart Grid (SG)

Smart grids take advantage of CPSs to provide services with high reliability and efficiency, focusing on consumer needs. They can adapt to energy demands in real time, allowing for increased functionality [25]. However, these grids depend on information technology, which is vulnerable to cyber-attacks. One such attack is false data injection (FDI) [222, 223, 224, 225]. Generally, FDI attacks inject malicious packets with the goal of creating small measurement errors that corrupt the component of the smart grid that performs state estimation. To overcome this problem, He et al. [222] used Conditional Deep Belief Network (CDBN) to efficiently reveal the high-dimensional temporal behavior features of the unobservable FDI attacks. Their approach was evaluated on the IEEE 118-bus power test system and the IEEE 300-bus system and was able to achieve an accuracy surpassing 93% on several tests. However, the number of examined experimental scenarios was rather limited. According to [223], attackers could inject multivariate malicious data points in a time period (contextual or collective anomalies). Since such FDI attacks are stealthier, inspecting measurement data alone may fail to detect them. The authors proposed a hybrid anomaly detection model based on 1D-CNN and RNN that combined sensor measurements and network packets to address this issue. Data points that generated large prediction errors were classified as anomalies. The proposed framework was evaluated on an IEEE 39-bus system, where it achieved an accuracy above 90%. By considering anomaly detection as a binary classification problem, Wang et al. [225] applied an RNN model to FDI attacks. Simulations over the IEEE 39-bus system indicate that their model can achieve an acceptable FDI attack detection accuracy. However, the performance of their framework was compared only with shallow architectures, and the settings of the RNN model were not specified.

Given that collecting large-scale labeled data can be excessively expensive and time-consuming, Zhang et al. [226] proposed a semi-supervised learning approach by integrating the AEs into a GAN framework for detecting unobservable FDI attacks in distribution systems. The AEs serve for dimension reduction and feature extraction of measurement datasets and the GAN is employed in the attack detection task. The authors evaluated the proposed approach on three-phase unbalanced benchmarks: IEEE 13-bus and 123-bus distribution systems [227]. The experimental results showed that their method has a high and robust detection accuracy compared (around 97% of accuracy on both systems) to other semi-supervised learning techniques. Similar to [226], the authors in [228] combined two DNNs, namely AE and GAN, to develop an anomaly detection model capable of (i) detecting anomalies and (ii) classifying Modbus/TCP and DNP3 cyber-attacks. The proposed model was validated on three SG evaluation environments originating from the SPEAR project [229]: (i) SG lab, (ii) hydropower plant, and (iii) power plant. The performance of the parallel detection of both anomalies and particular cyberattacks was 95% accuracy with 3.6% FPR.

A phasor measurement unit (PMU) data manipulation attack (PDMA) can blind the control centers to the real-time operating conditions of power systems. To detect this type of attack, Wang et al. [224] used a deep AE. The input of the AE was 108 features extracted from PMU measurements (e.g., the three-phase magnitude, angles, and voltages). An attack was detected, if the reconstruction error was above a pre-defined threshold. The proposed model achieved a high detection performance reaching 94.1% accuracy, 99.6% precision, 88.6% recall, and 93.8% F1 score.

4.4.3. Intelligent Transportation Systems (ITSs)

By enabling the seamless exchange of information between vehicles and roadside infrastructure in real-time, connected and automated vehicles (CAVs) are expected to entirely and drastically change the transportation industry [230]. CAVs rely heavily on their sensor readings and on the information received from other vehicles and roadside units to navigate roadways. Therefore, anomalous sensor readings caused by either malicious cyber-attacks or faulty vehicle sensors can result in disruptive consequences and lead to fatal crashes. In this context, before the mass

implementation of CAVs, it is essential to develop strategies for the real-time and seamless detection of anomalies, including identifying their sources. Wyk et al. [230] created an anomaly detection for CAVs by combining CNN and Kalman filtering (KF). First, a CNN model that consisted of three CNN layers and two fully connected layers was employed to eliminate false sensor readings. Then, scrutinized data was fed to KF to remove further anomalies undetected by the CNN model. The method was validated on a two-year real-world dataset obtained from the Safety Pilot Model Deployment (SPMD) program [231]. The overall hybrid approach is promising, reaching up to 99.7% accuracy, 99.2% sensitivity, 99.8% precision, and 99.5% F1 score, outperforming the two baseline methods (standalone KF and CNN).

In-vehicle security is particularly challenging due to the controller area network (CAN) bus that does not have built-in security. Aiming to detect attacks/anomalies on the controller area network (CAN) bus, which is responsible for the communication between devices (e.g., airbags) and Electronic Control Units (ECUs) [232, 233], Hanselmann et al. [233] introduced a DL-based framework called CANet trained in an unsupervised manner. They designed CANet using LSTM to capture the CAN bus time series behavior, AE to learn the normal behavior, and Exponential Linear Unit (ELU) [234] to improve the classification of their framework. CANet was tested on high-dimensional real-work and synthetic data and achieved a True Positive (detection) Rate of around 99%.

In intelligent transportation infrastructure, safe and reliable intelligent charging stations are of paramount importance. Many smart charging stations have been deployed over the past few years, and most of them are online and connected, raising the potential risks of threats. Li et al. [88] proposed a DL-based anomaly detection method for in-vehicle power supply systems on real data (i.e., data collected by the author's institute). In particular, they used the Transformer architecture that considers the inherent correlations of traffic generated by ICSs. The results showed that their model achieved an accuracy rate and an F1 score of 99.80%, outperforming other traditional and deep architectures such as DT, RF, and CNN. The train Ethernet Consist Network (ECN) undertakes the task of transmitting critical train control instructions. In order to detect network attacks against the train ECN, Yue et al. [235] introduced an ensemble IDS based on CNN and RNN. In particular, they used three variants of CNN, namely LetNet5, AlexNet, and VGGNet, to capture the spatial patterns in the data. At the same time, three variants of recurrent NN called vanilla-RNN, LSTM, and GRU models were used to capture the temporal patterns. Thirty-four features of various protocol contents were extracted from the raw data produced by employing an ECN testbed to build a specific dataset. Although the proposed model achieved an accuracy rate of 98%, it is not possible to evaluate whether it could work in a real environment since its time complexity is not examined.

Findings. We can see that in ICSs, sensor time-series measurement data is usually collected. Often, the attacker's goal is to change the system's physical behavior, to which end she/he is spoofing the sensors' values, thus breaking time relations in the data. LSTM-based models and variants are used to capture such time relations. In contrast, FDI attacks are widespread in the SG. As can be observed, the majority of security methods are employed to aid conventional state estimator methods. AEs and LSTM techniques can both be adopted. In ITSs, attacks on the CAN bus system [233] are the most frequent. Thus, LSTM and CNN are applied to capture both time relations and context information (e.g., packet order and content). Lastly, DNNs coupled with attention mechanisms have been shown to improve the performance of DL-based security methods (e.g., [83], [88]), as they enable the model to learn and focus automatically on the essential features. Finally, Table 6 summarizes the representative works focusing on deep learning-based security methods for CPSs.

4.5. Spam filtering

Spam is a severe problem for most Internet users and the most common cyber-attack. It is estimated that spam e-mails constitute the majority proportion of global e-mails, around 73% [237]. According to [237], spam costs businesses US$20.5 billion annually in decreased productivity and technical expenses. Thus, it becomes evident that there is a need for reliable and intelligent anti-spam filters. Traditional ML methods are sometimes considered inadequate in capturing the variability of spamming behavior. Recently, DL has been adopted in developing anti-spam filters and was proven to be effective in this area [238, 239, 91, 240, 241, 242, 243, 244, 245].

4.5.1. Spam in E-mails and Websites

Seth et al. [243] proposed two multimodal architectures based on CNN to tackle spam e-mails based on images and spam content. These architectures combined both text and image classifiers to produce an output class (i.e., spam or non-spam). The first architecture employed feature fusion, whereas the other mined the rules between the two classifiers and employed class probabilities. Among the two approaches, the second architecture achieved the best accuracy with a rate of 98.11%. However, the size of the used dataset is limited (it includes only 1500 images), resulting in an overfitting problem. On the other hand, to detect spam websites in IoT environments, Makkar [241] introduced an LSTM-based model that was trained using the link features. This work used the WEBSPAM-UK2007 dataset [124] and achieved 95.25% accuracy in detecting spam hosts. Regrettably, the evaluation part does not

Table 6: Selected studies focusing on DL-based security methods for Cyber-physical Systems.

Industrial Control Systems (Section 4.4.1)				
Year	**Authors**	**Deep Model**	**Dataset**	**Advantages and Limitations**
2017	Goh et al. [121]	- LSTM	- SWAT [121]	- Does not provide a detailed analysis regarding the stability of the results.
2017	Inoue et al. [212]	- DNN	- SWAT [121]	- The detection recall for 23 out of the 36 attacks was 0% for DNN.
2018	Kravchik et al. [213]	- 1D-CNN, LSTM	- SWAT [121]	- Learning inter-stage dependencies was not examined.
2019	Macas et al. [83]	- Attention module - ConvLSTM-ED	- SWAT [121]	- Adaptively selects the most significant input features at each time step.
2019	Kravchik et al. [214]	- 1D-CNN, AE	- SWaT [121] - BATDAL [120] - WADI [122]	- A manually set up threshold is required.
2020	Xie et al. [215]	- 1D-CNN, RNN	- SWaT [121]	- Does not consider that several SWaT features do not have the same distribution in the training and test data.
2021	Lu et al. [216]	- PEO [217], DBN	- Gas Pipeline [218] - Water Storage Tank [218]	- High computational complexity.
2021	Hussain et al. [220]	- CNN (ResNet-50)	- Telecom Italia [221]	- The complexity of the model was not analyzed.
Smart Grid (Section 4.4.2)				
Year	**Authors**	**Deep Model**	**Dataset**	**Advantages and Limitations**
2017	He et al. [222]	- CDBN	- IEEE 118-bus, IEEE 300-bus	- The number of examined experimental scenarios was rather limited.
2018	Wang et al. [224]	- AE	- IEEE 9-bus, IEEE 30-bus	- 99.6% precision in detecting PDMA.
2019	Niu et al. [223]	- 1D-CNN, LSTM	- IEEE 39-bus	- Achieved higher than 90% accuracy in several experiments.
2019	Wang et al. [225]	- RNN	- IEEE 39-bus	- The settings (i.e., hyper-parameters) of the deep model are not specified.
2020	Zhang [226]	- AEs, GAN	- IEEE 13-bus, IEEE 123-bus	- A reduced amount of annotated data was used.
2021	Siniosoglou [228]	- AEs, GAN	- SPEAR testbed [229]	- A great number of scenarios in the experimental stage.
Intelligent Transportation Systems (Section 4.4.3)				
Year	**Authors**	**Deep Model**	**Dataset**	**Advantages and Limitations**
2018	Wyk et al. [230]	- CNN	- SPMD [231]	- Achieved 99.8% precision and 99.5% F1 score.
2020	Hanselmann et al. [233]	- LSTM, AE	- Real and synthetic CAN data	- True Positive (detection) Rate of around 99%. - Synthetic data is publicly available in [236].
2021	Li et al. [88]	- Transformer	- Real-World (DCS and substation system traffic)	- Accuracy and F1-Score of 99.80%.
2021	Yue et al. [235]	- CNN, RNN	- ECN testbed	- Time complexity is not examined.

provide any other measures such as precision, recall, or F1 score.

A DL-based scheme providing edge intelligence for web data filtration and web spam detection was proposed in [246]. The proposed solution comprised three layers. The bottom layer was responsible for collecting web data from various sources and delivering the processed data to the next layer. The middle layer was responsible for web spam detection at the edge, where the LSTM and CNN models were used for building the spam detector. Finally, the upper layer was in charge of storing efficient web information in the cloud environment. Although the proposed approach achieved an accuracy of 98.77%, a small-sized dataset of 300 samples for training and 300 samples for testing was used, which might cause an overfitting problem.

4.5.2. Spam in Online Social Networks (OSNs)

OSNs such as Twitter, Facebook, and Sina Weibo have become increasingly popular platforms where users can post their messages and share ideas worldwide. Unfortunately, spammers are also active on these social networks using many social techniques to spread spam, such as sending unsolicited messages to legitimate users, posting malicious links, displaying aggressive behavior to obtain attention, and repeatedly posting duplicate updates. In this regard, the study [244] applied a DL technique based on Word2Vec [174] and MLP to detect spam in Twitter at the tweet level. The proposed model was evaluated on four real-world datasets and achieved an accuracy higher than 90% on several tests, outperforming traditional classifiers such as Naive Bayes, Random Forest, and Decision Tree. However, this study does not include explanations of the technical details of the individual methods. A similar approach was developed in [240] but using the HSpam14 [247], and 1KS10KN [248] datasets.

A multistage spam detector for mobile social networks using DL techniques was developed by Feng et al. [238]. The authors followed an edge-computing architecture, where initial detection occurred at a mobile terminal and results were then forwarded to the cloud server for further calculation. The study used the Sina Weibo dataset and reported an accuracy of 88.62%, with 9.04% FPR. Specialized strategies (e.g., parameter pruning, parameter quantization, and knowledge distillation) for compressing deep networks to accommodate their high resource requirements on less powerful mobile terminals could improve the accuracy and reduce the FPR of the proposed framework. Guo et al. [91] investigated spamming problems in IoT-based social media applications. They proposed a spammer detection mechanism named Co-Spam composed of three NNs: Bi-AE, GCN, and LSTM for IoT applications. A series of experiments were performed on Twitter [245] and Weibo datasets. The results revealed that Co-Spam achieved a high precision rate, but many hyper-parameters were introduced to construct a more fine-grained feature space.

Taking into account that deep reinforcement learning

algorithms converge slower when looking for an optimal sequence of actions to reach out a goal state, Lingam et al. [249] introduced a particle swarm optimization (PSO) based deep Q learning algorithm (P-DQL) for detecting social spam bots by integrating PSO with a Q-value function. The performance of the proposed P-DQL algorithm was evaluated on two Twitter datasets, namely the Social Honeypot dataset [126] and the Fake Project dataset [250]. The results demonstrated an improvement of up to 15% on precision over other existing algorithms such as FFNN, ADQL [64], and C-DRL [65]. This approach works in offline settings, thus an interactive environment with online experiments can be embedded in the Twitter network as future work.

Attackers exploit OSNs by using abusive accounts to perform malicious actions for personal or political gain. To address this issue, Xu et al. [251] introduced a Deep Entity Classification (DEC) framework that leverages DNNs and the multi-stage multi-task learning (MS-MTL) paradigm to detect abusive accounts in OSNs. DNNs are used to extract the accounts' features based on the properties and behavioral features observed in their social graph neighbors, whereas MS-MTL allows DEC to learn the common underlying representations of different abuse types (i.e., fake, compromised, spam, and scam). During its deployment for a period larger than two years at Facebook, DEC detected hundreds of millions of abusive accounts. The authors estimate that DEC is responsible for a 27% reduction in the platform's volume of active abusive accounts. The major limitation of DEC is that it is computationally expensive, mainly due to deep features.

4.5.3. Spam in Online Reviews and Short Message Service

With the prevalence of the Internet, online reviews have become a valuable information resource for people. However, the authenticity of online reviews remains a concern, and deceptive reviews have become one of the security problems to be solved. The study [239] introduced an unsupervised spam detection model based on DL for online reviews using a modified version of Conditional GAN. The proposed model was evaluated on reviews collected from Douban (a Chinese online community where users share their reviews to express their feelings about movies) and reported 87.03% accuracy, outperforming several popular deep and traditional unsupervised classifiers (e.g., VAE, LOF, and OC-SVM).

Considering that short message service [242] usage has been rising over the last decade, Roy et al. conducted a comparative analysis of two DL models (namely, LSTM and CNN) and several traditional systems for classifying spam and not-spam text messages [242]. The proposed models were based only on text data, and the SMS Spam Collection v.1 dataset [127] was used for their evaluation. The study reports that CNN outperformed LSTM and other traditional ML models, achieving a rate of 98,5%, 97,6%, and 98,0% for precision, recall, and F1-score, respectively. However, the dataset used is not big enough, with only 747 spam messages and 4827 not-Spam messages, leading to overfitting. Moreover, the text messages examined in this study are written exclusively in English.

To address the computational efficiency limitation of RNN variants such as LSTM, the authors in [87] proposed a modified version of the Transformer model [252], which uses only a multi-head attention mechanism instead of RNN variants as encoders and decoders. In the experiments, two different datasets were utilized: the SMS Spam Collection v.1 [127] and UtkMl's Twitter [125] datasets. Although the results revealed that the proposed model outperforms both traditional and deep learning models (e.g., LR, NB, RF, SVM, LSTM, and CNN-LSTM) on the two datasets, there are some improved models based on the Transformer with more complex architecture such as GPT-3 [253] and BERT [254] that could be examined in the future.

Findings. Looking at Table 7, it is interesting to note that the majority of deep learning-enabled methods for Spam detection employ RNNs and CNNs, as they can efficiently extract the underlying spatial and temporal correlations. Futhermore, we observe that it is necessary to introduce new tools and extend novel approaches for analyzing and handling multilingual spam detection systems (eg., [255]). Finally, another frequently encountered flaw of spam detectors is their inability to detect new spam variants caused by training with relatively small and severely outdated datasets.

4.6. Fraud detection

In technological systems, fraudulent actions occur in several areas of daily life, e.g., in telecommunication networks, online banking, mobile communications, and e-commerce [256]. These frauds lead to considerable financial loss for individuals, businesses, and the government. According to [257], the global market for Fraud Detection and Prevention is projected to reach US$51.3 billion by 2027, from US$19.5 Billion in 2020, growing at a CAGR of 14.8% over the period 2020-2027. Fraud detection refers to promptly recognizing fraud as soon as possible after it has been committed [92, 258, 259, 260]. Detection methods are under a constant development to confront criminals by adapting to their strategies, usually leveraging data mining, statistics, and machine learning. In particular, DL techniques allow the extraction of complex information from the data and are more capable to explore deeper implicit fraud patterns.

4.6.1. Telecommunication Fraud

Fraud is expensive for a network carrier both in terms of lost income and wasted capacity. Aiming to detect fraudulent activities at a city-wide telecommunication network,

Table 7: Selected studies focusing on DL-based security methods for Spam filtering.

Spam in E-mails and Websites (Section 4.5.1)				
Year	**Authors**	**Deep Model**	**Dataset**	**Advantages and Limitations**
2017	Seth et. al [243]	- CNN	- Real-world (images)	- Dataset with only 1500 images.
2020	Makkar et al. [241]	- LSTM	- WEBSPAM-UK2007 [124]	- Only accurancy rate was reported.
2021	Makkar et al. [246]	- LSTM, CNN	- Real-world (images)	- Small-size dataset (only 600 images).
Spam in Online Social Networks (Section 4.5.2)				
Year	**Authors**	**Deep Model**	**Dataset**	**Advantages and Limitations**
2017	Wu et al. [244]	- WordVector, MLP	- Real-world (tweets)	- Comprehensive technical details were not reported.
2018	Madisetty et al. [240]	- CNN	- HSpam14 [247], 1KS10KN [248]	- Several experiments were executed.
2018	Feng et al. [238]	- CNN	- Real-world (Sina Weibo)	- Multistage Spam Detection (mobile terminal and cloud server).
2020	Guo et al. [91]	- Co-Spam (Bi-AE, GCN, LSTM)	- Yang et al.'s dataset [245] - Live (Sina Weibo)	- High accuracy requires high complexity.
2021	Lingam et al. [249]	- P-DQL	- Social Honeypot [126] - Fake Project [250]	- This approach works in offline settings.
2021	Xu et al. [251]	- DNN, MS-MTL	- Real-world (Facebook)	- Computationally expensive.
Spam in Online Reviews and Short Message Service (Section 4.5.3)				
Year	**Authors**	**Deep Model**	**Dataset**	**Advantages and Limitations**
2020	Gong et al. [239]	- Improved Conditonal GAN	- Real-world (Douban)	- Unsupervised learning.
2020	Roy el al. [242]	- CNN, LSTM	- SMS Spam Collection v.1 [127]	- English only text messages.
2021	Liu el al. [87]	- Modified Transformer [252]	- SMS Spam Collection v.1 [127] - UtkMl's Twitter [125]	- A multihead attention mechanism is used for improving accuracy.

Ji et al. [92] proposed the Multi-Range Gated Graph Neural Network (MRG-GNN) for learning latent features from social networks. First, a social network was modeled as a directed graph whose vertices represent subscribers and edges represent activities between them. Then, a graph convolution block was used to capture content information and related information between users, with convolutions based on efficient short walks and node-merging. A real-world dataset collected from Shanghai, China, was used for experimental evaluation. The proposed model achieved an AUC of 0.948, outperforming traditional models (e.g., SVM). Unfortunately, the study examines a limited number of scenarios for the evaluation of the proposed model.

Robocalling systems affect millions of people daily. Regrettably, traditional approaches in detecting such activities rely on the construction of blacklisting number systems. Nevertheless, criminals can easily masquerade their phone numbers. In order to address the above challenge, Yu et al. [260] introduced a DL-based approach for blacklisting unwanted phone numbers, while keeping a high detection rate through distributed crowdsourcing. The system comprises two parts: (i) a semi-automated caller ID tagging system leveraging the predictions of an LSTM model and (ii) a blacklist-based crowdsourcing and aggregation edge system. The experiments were performed against real incoming calls on Android phones. Although the results showed that the system design could attain decent detection rates, the study used a small dataset with just 1488 not cross-validated data points.

RobocallGuard, a DNN-based virtual assistant, was introduced by Pandit et al. [258] to stop current robocalls. In particular, RobocallGuard mimics a human call screener who picks up an incoming phone call and makes the user aware of the call only after confirming that the call is not a robocall or other type of spam. RobocallGuard was tested over a corpus of 8,081 real robocalls and managed to correctly label 97.8% of robocalls without negatively impacting legitimate calls. However, the dataset is not recent, and as such, it does not reflect the behavior of more sophisticated robocalls.

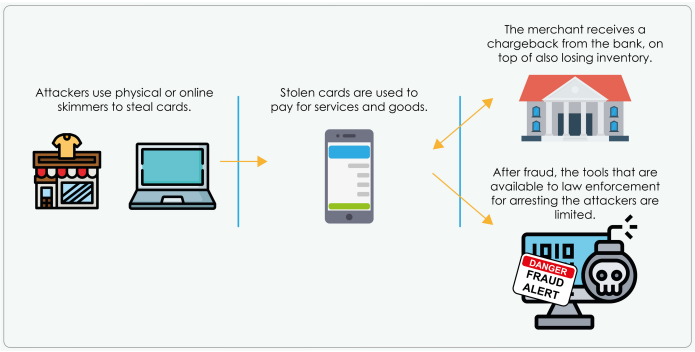

Figure 9: The comprehensive lifecycle of credit card fraud. It begins with attackers stealing card numbers and using them to purchase services and/or goods, leading to chargebacks for merchants, while law enforcement has limited or no routes of action.

4.6.2. Credit Card Fraud

Generally, credit card fraud activities can happen both online and offline (see Fig. 9). Online fraud is performed via phone shopping, the web, or cardholder-not-present. The criminals solely require the card information and it is not necessary to have the card in hand or simulate the cardholder's signature. In order to detect the most relevant fraudulent behavior patterns in an online detection system, Cheng et al. [93] proposed a spatial-temporal attention-based graph network (STAGN), where the temporal and location-based transaction graph features were learned by a GNN. The attentional weights were jointly learned in an end-to-end manner with 3D convolution and detection networks. STAGN was evaluated on a real-world card transaction dataset from a commercial bank. Although a mean AUC of 0.88 to 0.90 was achieved in most of

the conducted experiments, human interaction is required and there is no way to block the fraudulent transactions in real-time.

Regarding finding the online fraud transactions, Cao et al. [85] proposed a two-level attention model to capture the deep representation of features of online transaction behaviors. This is achieved by integrating two data embeddings at the data sample level (tree-based model) and the feature level (bidirectional GRU). Finally, the embeddings learned by the two attention mechanisms were combined for the training of a fraud detection model. The proposed model was evaluated on four public datasets and a private dataset (card transaction records) provided by a financial company in China. The results showed that the proposed method achieved an accuracy, precision, recall and F1 of over 85% in the conducted experiments outperforming traditional techniques such as Gradient boosting.

Findings. We note that the challenge in credit card fraud detection is that frauds have no consistent patterns. The typical approach in credit card fraud detection is to maintain a usage profile for each user and monitor the user profiles to detect any deviations [85, 93]. Since there are billions of credit card users, this technique of user profiling is not very scalable. Thus, the majority of the studies used deep anomaly detection methods. On the other hand, most of the fraud detection methods in telecommunications focused on mobile cellular networks due to their rapid deployment and evolution. In this context, using GNNs for developing security frameworks against telecommunication frauds (e.g., [92]) is becoming a trend due to their ability to capture complex relationships between objects and make inferences based on data described by graphs. The summary of the analyzed works in this subsection can be found in Table 8.

4.7. Encrypted Traffic Analysis

Encryption protocols provide security guarantees for data confidentiality and integrity, reducing as a side effect the network administrators' ability to monitor their infrastructure for malicious traffic and sensitive data exfiltration. Attackers have shifted to using encryption and cryptographic methods in their attacks, extending from ransomware to HTTPS for protecting communications with infected devices and avoiding detection. Accordingly, the main objective of security professionals is to find a balance between end-to-end security and the ability to gather valuable information from the traffic to detect possible threats and better allocate and protect resources.

Traffic classification refers to categorizing network traffic into suitable classes, which is essential for several applications including malware/intrusion detection. The first and more straightforward approach employs port numbers. Nevertheless, its accuracy has deteriorated because new applications use well-known port numbers for disguising their traffic or evading standard registration port numbers [261]. Deep packet inspection (DPI) is the next generation of traffic classification, which focuses on payloads. However, this technique can be applied only to unencrypted traffic and has a high computational cost. As a consequence, new methods that depend on statistical or time-series features enabling them to handle both encrypted and unencrypted traffic came forth. They usually use traditional ML techniques such as RF and K-NN and their performance depends mainly on human-engineered features, which limits their generalizability. To address this, employing DL techniques can eliminate the disadvantages of manually constructing features.

4.7.1. Website fingerprinting, and Protocol/Application Identification

Identifying network traffic of visited websites through privacy-enhancing technologies like Tor is also known as **Website Fingerprinting (WF)**. In [262], the authors studied the efficiency of DL-based classifiers in WF for the first time. They demonstrated that stacked denoising AEs are useful in detecting websites using only the Tor packets' direction (incoming or outgoing) and inter-arrival times with an 86% success rate. Rimmer et al. [263] showed that DL is a helpful tool for automating the features engineering process, and their SDAE model achieved a 95.3% success rate using only the Tor packets' direction. In [264], a deeper CNN classifier was built to outperform earlier studies, improving the success rate to 98%.

Application protocol classification is closely related to **application type classification**. The former identifies protocols such as HTTP or SSH, while the latter recognizes individual applications (such as Skype and Google Talk). A mobile traffic classifier based on DL was introduced in [265], demonstrating that deep NNs can handle encrypted traffic and represent its complex patterns. The experimental results showed that DL-based solutions (e.g., MLPs, CNNs, and LSTMs) achieved superior accuracy over RF in classifying IOS, Android, and Facebook traffic. However, the experiment settings were not completely fair and equal because the input features used for RF and the DL methods were different.

A convolutional neural network (CNN) model is proposed in [266], whose input consists of each Internet flow represented as a picture. A LeNet-5 style architecture was used for classifying Internet traffic into five categories (VoIP, video, file transfer, chat, and browsing). For each flow, an image was constructed based on the packet sizes and arrival times. The ISCX VPN-nonVPN [128] and ISCX Tor-nonTor [129] datasets were used for evaluating the model. The authors demonstrated that their model can classify traffic to a category with an accuracy of over 96%, except for browsing. Furthermore, the proposed method is able to classify traffic that passes through a VPN

Table 8: Selected studies focusing on DL-based security methods for Fraud detection.

Telecomunication Fraud (Section 4.6.1)				
Year	**Authors**	**Deep Model**	**Dataset**	**Advantages and Limitations**
2020	Ji et al. [92]	- MRG-GNN	- Real-world (call detail records in a city)	- Limited number of scenarios for the model evaluation.
2020	Yu et al. [260]	- LSTM	- Real-world (incoming calls on Android phones)	- Small dataset.
2021	Pandit et al. [258]	- DNN	- Real-world (robocalls records)	- The dataset is not recent.
Credit Card Fraud (Section 4.6.1)				
Year	**Authors**	**Deep Model**	**Dataset**	**Advantages and Limitations**
2020	Cheng et al. [93]	- STAGN	- Real-world (card transaction)	- Human interaction is necessary.
2021	Cao et al. [85]	- Attention-based bidirectional GRU	- Custom - Real-world (card transaction records)	- There is no detailed information about the used datasets.

with an accuracy of over 99.2% and also achieves good results for traffic that traverses Tor (over 89% except for file transfer). However, it requires the recording of the size and the timestamping of the packets of each flow, causing additional overhead to the classification time.

To reduce the number of parameters and shorten the running time of a DL architecture for performing real-time traffic classification, Cheng et al. [86] developed a DL model based on multi-head attention and 1D-CNN. The model automatically extracts high-order flow-level and packet-level features. The experimental results demonstrated that the number of parameters was reduced to the 1.8% and 2.7% of the number used for 1D-CNN and CNN with LSTM, respectively. Similarly, the training time was the 49.7% and the 6.8% of the time needed by 1D-CNN and CNN with LSTM. Although the proposed model reached a precision and recall of around 100% on both datasets, outperforming the 1D-CNN and CNN with LSTM, the employed datasets only include HTTPS and VPN traffic, without involving other protocols.

In [90], an encryption traffic classification method based on Graph Convolutional Network (GCN) and AE was proposed. The authors used a two-layer GCN architecture for flow feature extraction and encrypted traffic classification. Furthermore, an autoencoder was employed to learn the representation of the flow data itself and integrate it into the GCN-learned representation to form a complete feature representation. The experimental results demonstrated that their method achieved an accuracy of 85.82% and 94.33% on ISCX VPNnonVPN [128] and USTC-TFC2016 [267] datasets respectively, outperforming traditional methods like RF.

In order to avoid the need for an extensive labeled dataset, Rezaei et al. [268] introduced a multi-task learning model architecture using an 1D-CNN. They employed a large unlabeled dataset and a small labeled dataset to train a model that predicts three tasks: application, bandwidth, and duration of flows. The experimental results revealed that the proposed model significantly outperforms both single-task and transfer learning approaches. To examine a larger number of classes, Wang et al. [269] proposed a GAN-based semi-supervised learning encrypted traffic classification method named ByteSGAN. The ISCX VPN-nonVPN traffic data set [128] was used in the experiments. When the number of labeled samples is 1000, the classification accuracy of ByteSGAN is 99.18%. However, this performance was achieved using only 7 of the total 15 classes and gets worse when more classes are employed (drops to 92.15% with 15 classes).

Findings. As deep learning techniques can perform automatic feature extraction, they become a strong candidate for encrypted traffic classification and analysis. Among them, we find systems based on either CNNs, which are effective at processing data coming in the form of multiple arrays and capturing the spatial patterns, or AEs, which can be pre-trained on unlabeled data and fine-tuned on a small amount of labeled data. Although some initial works like [270], and [268] have demonstrated the potential of DL-based methods over more robust encryption protocols (e.g., TLS 1.3 and QUIC), the employed datasets are not large and diverse enough to represent real-world settings. Therefore, the scalability of such methods remains an open issue and needs to be further investigated. Table 9 presents the reviewed DL-based security methods in encrypted traffic analysis.

5. Lessons learned and Future directions

To address *Q5 (Which are the most important and promising directions for further study?)*, this section reviews the lessons learned and outlines future directions.

5.1. Lessons learned

This section provides useful insights and outlines recommendations to developers, researchers, and practitioners of the security domain who intend to use DL to solve security problems of interest. Table 10 presents an overview of the research in each cybersecurity application, focusing on the employed DL model. Fig. 10 also depicts the frequency with which the different research works have utilized the different models. About 28% of the papers have used RNNs and variants (e.g., sequence models) for constructing the proposed systems, while RBMs are the least used models (about 1%) overall. This highlights that defense methods based on deep learning are increasingly evolving,

Table 9: Selected studies focusing on DL-based security methods for Encrypted Traffic Analysis.

Year	Authors	Deep Model	Dataset	Advantages and Limitations
2016	Abe et al. [262]	- SDAE	- Wang's dataset [271]	- Pioneer study in implementing DL-based classifiers for WF.
2018	Rimmer et al. [263]	- SDAE	- Real-world (page visits)	- Larger datasets are used in the experimental phase. - The employed dataset is available upon request in [272].
2018	Sirinam et al. [264]	- CNN	- Real-world (page visits)	- Accuracy rate of 98%
2018	Aceto et al. [265]	- MLPs, CNNs, LSTMs	- Real-world (mobile user activity)	- Features used for the RF and the DL methods are different..
2019	Shapira et al. [266]	- CNN (LeNet-5)	- ISCX VPN-nonVPN [128] - ISCX Tor-nonTor [129]	- Additional processing and overhead to the classification time.
2020	Cheng et al. [86]	- Multi-head Attention - 1D-CNN	- ISCX VPN-nonVPN [128] - Open HTTPS [130]	- The dataset only includes HTTPS and VPN traffic, but it does not involve other protocols.
2020	Sun et al. [90]	- GCN, AE	- ISCX VPNnonVPN [128] - USTC-TFC2016 [267]	- Addressing the encrypted traffic classification problem with the traffic of unknown application types should be analyzed.
2020	Rezaei et al. [268]	- CNN (Multi-task learning)	- QUIC [131] - ISCX VPN-nonVPN [128]	- Only a few classes are considered in the experiments.
2021	Wang et al. [269]	- DCGAN [273]	- ISCX VPN-nonVPN [128]	- With 15 classes, the accuracy drops to 92.15%.

Table 10: The use of different DL models in Cybersecurity applications.

Domain	DL techniques										
	DNN	AE	CNN	DBN	RBM	RNN	GAN	DRL	SNN	TR	GNN
Network Intrusion Detection	[143] [159]	[159], [146] [144], [149]	[274], [153] [157]		[145]	[144], [150]	[154] [163]	[160] [168]			
Malware detection and analysis	[186]	[67]	[183], [184] [185], [173], [178], [181]			[185], [180], [181]					
Botnet Detection and DGAs		[107], [189]	[195]			[193], [192] [191], [84] [198], [194] [195], [190] [201]			[190] [201]		
Cyber-Physical System Security	[212]	[83], [214] [224], [226] [228], [233]	[213], [214] [215], [220] [223], [230] [235]	[216] [222]		[121], [213] [83], [215] [223], [225] [233], [235]	[226] [228]			[88]	
Spam filtering	[244] [251]	[91]	[243], [246] [240], [238] [242]			[241], [246] [91], [242]	[239]	[249]		[87]	[91]
Fraud detection	[258]					[260], [85]					[92] [93]
Encrypted Traffic Analysis	[265]	[262], [263] [90]	[264], [265] [266], [86] [268]			[265]	[269]			[86]	[90]

since researchers no longer focus on traditional DL models (e.g., RBM, DBN). The table also emphasizes that around one-quarter of the cybersecurity applications are related to time-series, text streams, or serial data. Moreover, another one-quarter of the works used CNNs due to their capability to extract high-level feature representations from spatial data such as images.

The main guidelines that we have identified via our analysis can be summarized as follows:

Obtain and use the right data. The data must be sufficient, representative, relevant to the current attack landscape, and correctly labeled if needed. Moreover, it is essential not to overlook other issues related to public benchmark datasets, such as repeated data, missing values, and incorrect labeling. Building models based on skewed and biased data produces systems that are unsuitable for exploitation. Therefore, obtaining valid, representative, and accurate data should be a priority and a primary objective of the research.

Combine detection methods in a multi-layered architecture. Due to the diversity of today's threat attack vectors, cybersecurity solutions should be organized in a multi-layered manner. In other words, deep learning-based detection should work in synergy with alternative kinds of detection, forming a multi-layered approach for achieving efficient cybersecurity protection.

Consider using online learning. The data used in the cybersecurity domain is increasing and evolving very quickly, so the data-driven attack detection models should be frequently retrained and updated. *Online learning* [275] is adapted to the constant change of the learning environments and as such can be used when deploying security models. Nevertheless, many challenges associated with such setups remain, like Catastrophic forgetting [276]. This challenge refers to a learning model that forgets its prior knowledge when fine-tuned with new data and is a severe problem for neural networks [277, 278]. Another aspect of real-time data handling, especially in traffic anomaly detection, is performing accurate traffic flow sampling. In large networks, it is not feasible to an-

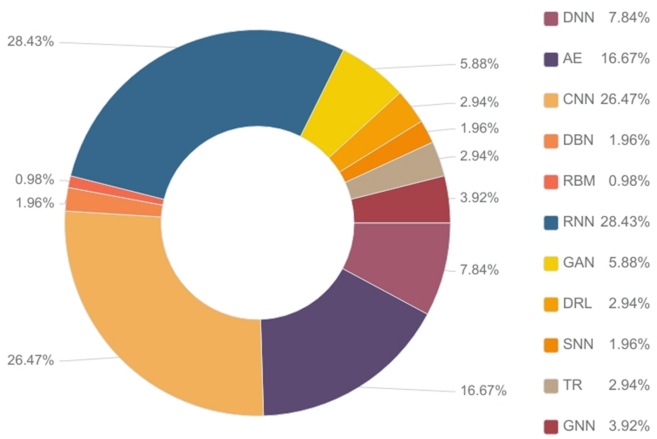

Figure 10: Use of DL models in the surveyed papers.

alyze all traffic flows. Therefore, it is crucial to explore methods proposed for traffic sampling [279, 280, 281] and incorporate them into the training process.

Beware of deep learning against rare attacks. ML/DL handles tasks efficiently when malicious and benign samples are numerously represented in the training set. However, some attacks are so rare that we have only a few sample data for training. This is typical for high-profile targeted attacks.

Decrease false-positive rates as much as possible. The objective should be to decrease the false positive rate as much as possible, ideally down to zero. To achieve this goal, it is necessary to impose stringent requirements for ML/DL models and metrics optimized during training, focusing on low FPR models. This still might be insufficient because new, previously unseen benign files may occasionally be falsely categorized as malicious. Thus, the goal should be to implement a flexible model design that allows fixing false positives on the fly without completely retraining the employed neural networks.

Do not overlook attack detection in resource-constrained platforms. The key challenge is supporting advanced ML/DL while still running on devices with a wide range of performance capabilities. For example, prediction times for the same ML/DL model with identical inputs and system architectures may vary by one to four orders of magnitude, depending on the underlying hardware. The core issue is that security-related ML/DL must be executed efficiently, or else it will exclude people using low-end devices.

Optimize the hyper-parameters. Often, the hyper-parameters of the deep models are not fine-tuned accurately and thoroughly. At first sight, it is impossible to know the optimal value for a model's hyper-parameter on every examined problem. Thus, we may follow heuristic guidelines [282, 283, 216], copy values from previous implementations, or search by trial and error to create a competent NN architecture. Another commonly used structured and systematic way of determining a model's optimal hyper-parameters is grid search [284]. Furthermore, AutoML techniques can also be used to automate this process [285].

5.2. Feature Research Directions

Construction of high quality and up to date datasets: Since the performance of deep learning-based methods strongly depends on the quantity and quality of the available data [286], the biases and limitations of the datasets used for training the models affect the reliability of the predictions. The majority of works currently focus on researching algorithms that can yield improved detection results, but very few studies are dedicated to evaluating the reliability of benchmark datasets. For instance, studies such as [287, 288] proposed a list of criteria for assessing the reliability of a dataset for intrusion detection. Among them, attack and traffic diversity play a significant role since a limited diversity or a high imbalance among the attack types might increase the bias of the detection approaches towards specific situations. The research community has not yet discovered a way to artificially generate adequately realistic cyber data [289]. On the other hand, data from real networks or the Internet usually contain sensitive information such as personal or company details and could potentially reveal security vulnerabilities of the network from which they originate if made publicly available. We can conclude that deep learning applications in the cybersecurity area can only advance if researchers and industry stakeholders release more realistic datasets, especially considering that attack patterns tend to evolve to better overcome the existing security systems. Finally, according to [290], some features in available benchmark datasets are less useful in detecting emerging attacks and serve better the detection of older attack patterns. Thus, it is essential to evaluate whether new features will be required for maintaining a high detection accuracy level.

Deep N-Shot Learning- Learning from Few Samples: Few-shot, low-shot, or n-shot learning allows a model to classify samples accurately based only on a few training examples (usually between zero and five) [73]. Although few-shot learning has received much research attention in computer vision, its application in cybersecurity remains mostly unexplored. Representative works include an IDS based on one-shot learning [76], a data augmentation method for the few-shot WF attack [74], and a behavioral biometrics-based user authentication scheme [75]. Nevertheless, n-shot learning is still an emerging area in cybersecurity, and extensive research must be conducted to understand how such methods can be effectively utilized.

Deep Lifelong Learning - Learning Continuously: Deep lifelong learning [291] aims to mimic human behavior and seeks to build a machine that can continuously adapt to new environments, while retaining as much knowledge as possible from previous learning experiences. This concept can be adopted in the dynamic IoT environment. Given IoT's nature, normal structures and patterns,

as well as threats and attacks can considerably change over time [23, 24]. Therefore, discerning between normal and abnormal IoT system behavior cannot be always predefined. As a consequence, security models must be frequently updated in order to handle and understand IoT modifications. This novel learning paradigm is in its infancy regarding the cybersecurity domain. Nevertheless, it shows great promise and potential in detecting new threats in dynamic environments.

Deep Active Learning: Active learning (AL) [292] strives to maximize a model's performance gain while annotating as few samples as possible. The main idea is to mitigate the cost of labeling without affecting performance by selecting only the most useful samples from the unlabeled dataset. This concept can also be applied in the cybersecurity domain, where the potent learning capabilities of DL can be retained while at the same time reducing the sample annotating cost [293, 294]. Deep AL is an emerging research area in cybersecurity, and we expect more efforts in this direction in the future.

Interpretability of Deep Neural Models: The black-box nature of DNNs constitutes one of the primary obstacles for their wide acceptance in mission-critical applications. Recent studies suggest that model interpretability and robustness are closely connected [295]. On the one hand, improvements made in a model's robustness also develop its interpretability. For example, a DNN that has been subjected to adversarial training shows better interpretability (with more accurate saliency maps) than the same model trained without adversarial examples. On the other hand, deeply understanding a model enables us to better determine its weaknesses and potential vulnerabilities, ultimately improving its accuracy and reliability. Apart from that, interpretability plays a vital role in the ethical use of DL [296]. In [297], the authors used SHAP [298] to provide the reasoning behind the IDSs' predictions and interpret the detected intrusions to the cybersecurity personnel. However, excluding this work, investigating the interpretability of DNNs in cybersecurity is currently rather scarce.

Deep Reinforcement Learning for Cybersecurity Deep reinforcement learning, which is created by incorporating deep learning into traditional reinforcement learning, can solve dynamic, complex, and mainly high-dimensional cybersecurity problems, as discussed in Section 4. However, current deep reinforcement learning applications to cybersecurity are usually limited by discretizing the action space, restricting in this way the achieved performance to real-world problems. Studying methods that can deal with continuous action spaces in cyber environments (e.g., policy gradient [299] and actor-critic [300] algorithms) is another promising research direction.

Adversarial attacks in the Cybersecurity domain: Although AI can help in cyber-defense, it can also facilitate dangerous attacks (i.e., offensive AI). Attackers can employ AI to make attacks smarter and more complex, avoiding detection methods to infiltrate computer systems or networks. Machine learning-based systems can mimic humans to craft convincing fake messages utilized in large-scale phishing attacks [301, 302]. Attackers can also inject or manipulate training data to either create a backdoor to use at inference time or to corrupt the training process [303]. Furthermore, hackers can manipulate the states or policies and falsify part of RL's reward signals for fooling the agent into taking sub-optimal actions [304]. These types of attacks are hard to prevent, detect, and fight against as they are part of a battle between AI systems. In the cybersecurity domain, techniques for evasion attacks have been widely adopted. Nevertheless, there are not enough studies dealing with feature-targeted attacks (a.k.a. Trojan neural network attacks [305]), backdoor attacks [306, 307], or attacks against deep reinforcement learning and deep unsupervised models [308]. In this context, further research is required to construct defense methods against these emergent types of attacks.

Deep Learning at the edge: The edge infrastructure tier can offer new opportunities for supporting cybersecurity strategies, mainly because it is closer to the data sources and it can detect and respond to events more rapidly. The characteristics of edge computing devices indicate that they cannot support the same cybersecurity functionality found in enterprise data centers and clouds [151]. The limited scale of the edge compared to the elastic cloud is one significant difference. Another difference is the localized context in which the edge-deployed functionality operates. These differences have several effects on the interplay of deep learning and security. First, concerning cybersecurity methods based on deep learning, the reduced resource footprint available at the edge dictates the types of deep learning models that can be successfully employed. Second, the mismatch of capabilities between the edge and the data centers calls for the construction and the deployment of different, more lightweight cybersecurity techniques at the former. Strategies and techniques for making cybersecurity more effective and more scalable are an active research topic.

Secure and Privacy-Preserving Deep Learning: Traditional AI data processing systems often involve simple models of data transactions, in which one party collects and transfers data to another party, which is responsible for the cleaning and merging processes. Finally, a third party takes the embedded data and creates models for other parties. This traditional procedure faces challenges with new regulations for the protection of data security and privacy like GDPR [309]. How to legally resolve data isolation and fragmentation is a significant challenge for researchers and AI professionals today. Federated learning is a possible solution to these challenges, as privacy is one of its essential properties [166, 164, 167]. When data cannot be directly aggregated due to intellectual property rights, privacy protection, and data security, federated learning can be employed as a promising solution.

Federated Averaging (FedAvg) proposed in [310] is the most commonly used method for FL. According to it, a

client locally updates model weights and sends the local weights to a server for model aggregation, collaboratively training a global model together with other clients. In [311], a framework for multi-task learning that allows multiple clients for training different tasks was presented. Its primary advantage is mitigating communication costs and stragglers through the training stage. In [312], homomorphic encryption was applied to a horizontal FL framework for protecting the gradients. Gao et al. [313] introduced a heterogeneous Federated Transfer Learning (FTL) framework for feature space training among multiple clients. The conducted experiments demonstrated that it outperforms local training schemes and homogeneous FL schemes. Finally, an FTL framework called FedSteg was proposed by Yang et al. [314] to train a personalized and distributed model for secure image steganalysis. Interested readers can refer to other works that surveyed the privacy-preserving federated learning in-depth, such as [315].

A significant challenge when working with federated machine learning models in settings with several different actors is building a highly distributed, secure, and reliable platform that enables the scalable cooperation of participants who do not fully trust each other. The blockchain and smart contract technologies can create an immutable audit trail for federated models to achieve higher trustworthiness in tracking and proving provenance. As a representative example, a blockchain-based framework using a DL approach to detect intrusion attacks while preserving data privacy was introduced by Alkadi et al. [316]. Moreover, the authors in [317] presented a privacy-aware DL method, which allows the collaboration of multiple nodes for training DNNs.

Another way to guarantee user privacy in DL-based services is to train the network on encrypted data. Cryptographic techniques, like fully homomorphic encryption [318, 319], enable the processing of encrypted data. Nevertheless, they are too slow for training DNN models due to the computational complexity and the arduous operations involved. Gilad-Bachrach et al. [318] presented Crypto-Nets, which perform the inference phase of a NN on encrypted data. Despite its originality and novelty, this work has much room for improvement, particularly in terms of achieved throughput and latency. Indeed, Nandakumar et al. [319] extended the aforementioned approach and built the first fully homomorphic computationally efficient DL service for training on encrypted data. Other works in the literature address privacy vulnerabilities using dummy approaches. A dummy approach refers to constructing a group of dummy requests for each user service request and then submitting them with the real one in random order to the server-side [320, 321]. The goal is to make it difficult for the untrusted server to obtain the users' real requests, and thus protect the users' privacy in recommendation services [322], digital libraries (book search [320] and browsing privacy [323, 321]), and location services [324, 325].

Emerging Deep Learning Applications in Cybersecurity

Unmanned aerial vehicles may lead to significant security breaches in terms of hardware, software, and communication channels [326]. AI-based detection systems can help predict future events more accurately. *Extended reality (XR)* is an umbrella term covering augmented, virtual, and mixed reality [327]. XR includes cameras, microphones, and sensors that help users interface with the real world, which means that information about their environment can also be collected and shared. DL can be applied to identify anomalies in the high-dimensional data generated by the XR technology applications. Finally, a booming increase in *human-robot interactions* is expected in the following years, according to which humans will share workspaces, collaborative tasks, and, eventually, significant parts of their daily living environments with robots. In this context, we can employ DL techniques to yield more intelligent and powerful physical-based anomaly detection mechanisms that leverage big data [83].

6. Conclusions

Deep Learning is playing an increasingly important role in the cybersecurity domain. In this paper, we provided a comprehensive survey of recent work regarding deep learning for cybersecurity applications. We summarized both fundamental concepts and advanced principles of various deep learning models, along with necessary resources like a generic framework and datasets. We reviewed the state-of-the-art DL-based cybersecurity systems across different application scenarios. Finally, we concluded by pinpointing several open research issues and promising directions, leading to valuable future research suggestions.

We hope that this article will become a guide to developers/researchers and security practitioners interested in applying artificial intelligence (particularly deep learning) to complex problems in cyber environments. Lastly, we would like to caution against a potential pitfall. Despite its impressive capabilities, Deep Learning is not a panacea and should not be used indiscriminately in every use case. Instead, it should be employed in problems characterized by large-scale data and complex non-linear hypotheses with many features and high-order polynomial terms.

Acknowledgment

The authors would like to thank the Universidad de las Fuerzas Armadas-ESPE of Sangolquí, Ecuador, for the resources granted to develop the research project entitled: "Design and Implementation of the IT infrastructure and service management system for the ESPE Academic CERT", coded as PIC-2020-ESPE-CERT.

References

[1] T. H. T. Symantec. Threat landscape trends - q2 2020. [online] (2020). [Accessed: Sep 5, 2021].

[2] S. Magazine. Hacker breaks into florida water treatment facility, changes chemical levels [online]. [Accessed: Sep 5, 2021].

[3] E. U. A. for Cybersecurity. Enisa threat landscape - the year in review. [online] (2020). [Accessed: Sep 5, 2021].

[4] C. R. Institute. Reinventing cybersecurity with artificial intelligence. [online] (2019). [Accessed: Sep 5, 2021].

[5] D. Gumusbas, T. Yldrm, A. Genovese, F. Scotti, A comprehensive survey of databases and deep learning methods for cybersecurity and intrusion detection systems, IEEE Syst. J. (2020) 1–15doi:10.1109/jsyst.2020.2992966.
URL https://doi.org/10.1109/jsyst.2020.2992966

[6] S. Zeadally, E. Adi, Z. Baig, I. A. Khan, Harnessing artificial intelligence capabilities to improve cybersecurity, IEEE Access 8 (2020) 23817–23837. doi:10.1109/access.2020.2968045.
URL https://doi.org/10.1109/access.2020.2968045

[7] M. Research. Artificial intelligence in cybersecurity market [online]. [Accessed: Sep 5, 2021].

[8] F. Chollet, Deep Learning mit Python und Keras: Das Praxis-Handbuch vom Entwickler der Keras-Bibliothek, MITP-Verlags GmbH & Co. KG, 2018.

[9] J. Saxe, H. Sanders, Malware Data Science: Attack Detection and Attribution, No Starch Press, 2018.

[10] A. Singla, E. Bertino, How deep learning is making information security more intelligent, IEEE Secur. Privacy 17 (3) (2019) 56–65. doi:10.1109/msec.2019.2902347.
URL https://doi.org/10.1109/msec.2019.2902347

[11] L. Bottou, Stochastic gradient descent tricks, in: Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2012, pp. 421–436. doi:10.1007/978-3-642-35289-8_25.
URL https://doi.org/10.1007/978-3-642-35289-8_25

[12] K. Weinberger, A. Dasgupta, J. Langford, A. Smola, J. Attenberg, Feature hashing for large scale multitask learning, in: Proceedings of the 26th Annual International Conference on Machine Learning - ICML '09, CEAS, ACM Press, 2009. doi:10.1145/1553374.1553516.
URL https://doi.org/10.1145/1553374.1553516

[13] S. Ruder, An overview of multi-task learning in deep neural networks, arXiv preprint arXiv:1706.05098.

[14] A. L. Buczak, E. Guven, A survey of data mining and machine learning methods for cyber security intrusion detection, IEEE Commun. Surv. Tutorials 18 (2) (2016) 1153–1176. doi:10.1109/comst.2015.2494502.
URL https://doi.org/10.1109/comst.2015.2494502

[15] P. A. A. Resende, A. C. Drummond, A survey of random forest based methods for intrusion detection systems, ACM Comput. Surv. 51 (3) (2018) 1–36. doi:10.1145/3178582.
URL https://doi.org/10.1145/3178582

[16] J. M. Torres, C. I. Comesaña, P. J. García-Nieto, Machine learning techniques applied to cybersecurity, International Journal of Machine Learning and Cybernetics 10 (10) (2019) 2823–2836.

[17] S. X. Wu, W. Banzhaf, The use of computational intelligence in intrusion detection systems: A review, Appl. Soft Comput. 10 (1) (2010) 1–35. doi:10.1016/j.asoc.2009.06.019.
URL https://doi.org/10.1016/j.asoc.2009.06.019

[18] Y. Xin, L. Kong, Z. Liu, Y. Chen, Y. Li, H. Zhu, M. Gao, H. Hou, C. Wang, Machine learning and deep learning methods for cybersecurity, IEEE Access 6 (2018) 35365–35381. doi:10.1109/access.2018.2836950.
URL https://doi.org/10.1109/access.2018.2836950

[19] K. Shaukat, S. Luo, V. Varadharajan, I. A. Hameed, M. Xu, A survey on machine learning techniques for cyber security in the last decade, IEEE Access 8 (2020) 222310–222354. doi:10.1109/access.2020.3041951.
URL https://doi.org/10.1109/access.2020.3041951

[20] C. S. Wickramasinghe, D. L. Marino, K. Amarasinghe, M. Manic, Generalization of deep learning for cyber-physical system security: A survey, in: IECON 2018 - 44th Annual Conference of the IEEE Industrial Electronics Society, IEEE, IEEE, 2018, pp. 745–751. doi:10.1109/iecon.2018.8591773.
URL https://doi.org/10.1109/iecon.2018.8591773

[21] Y. Luo, Y. Xiao, L. Cheng, G. Peng, D. D. Yao, Deep learning-based anomaly detection in cyber-physical systems: Progress and opportunities, ACM Comput. Surv. 54 (5). doi:10.1145/3453155.
URL https://doi.org/10.1145/3453155

[22] F. Hussain, R. Hussain, S. A. Hassan, E. Hossain, Machine learning in IoT security: Current solutions and future challenges, IEEE Commun. Surv. Tutorials 22 (3) (2020) 1686–1721. doi:10.1109/comst.2020.2986444.
URL https://doi.org/10.1109/comst.2020.2986444

[23] M. A. Al-Garadi, A. Mohamed, A. K. Al-Ali, X. Du, I. Ali, M. Guizani, A survey of machine and deep learning methods for internet of things (IoT) security, IEEE Commun. Surv. Tutorials 22 (3) (2020) 1646–1685. doi:10.1109/comst.2020.2988293.
URL https://doi.org/10.1109/comst.2020.2988293

[24] E. Rodriguez, B. Otero, N. Gutierrez, R. Canal, A survey of deep learning techniques for cybersecurity in mobile networks, IEEE Communications Surveys & Tutorials 23 (3) (2021) 1920–1955. doi:10.1109/comst.2021.3086296.
URL https://doi.org/10.1109/comst.2021.3086296

[25] D. Berman, A. Buczak, J. Chavis, C. Corbett, A survey of deep learning methods for cyber security, Information 10 (4) (2019) 122. doi:10.3390/info10040122.
URL https://doi.org/10.3390/info10040122

[26] S. Mahdavifar, A. A. Ghorbani, Application of deep learning to cybersecurity: A survey, Neurocomputing 347 (2019) 149–176. doi:10.1016/j.neucom.2019.02.056.
URL https://doi.org/10.1016/j.neucom.2019.02.056

[27] I. Goodfellow, Y. Bengio, A. Courville, Deep learning, MIT press, 2016.

[28] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (7553) (2015) 436–444. doi:10.1038/nature14539.
URL https://doi.org/10.1038/nature14539

[29] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2016, pp. 770–778. doi:10.1109/cvpr.2016.90.
URL https://doi.org/10.1109/cvpr.2016.90

[30] G. Huang, Z. Liu, L. Van Der Maaten, K. Q. Weinberger, Densely connected convolutional networks, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2017, pp. 4700–4708. doi:10.1109/cvpr.2017.243.
URL https://doi.org/10.1109/cvpr.2017.243

[31] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, K. Keutzer, SqueezeNet: Alexnet-level accuracy with 50x fewer parameters and< 0.5 MB model size, arXiv preprint arXiv:1602.07360.

[32] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, H. Adam, Mobilenets: Efficient convolutional neural networks for mobile vision applications, arXiv preprint arXiv:1704.04861.

[33] J. Redmon, A. Farhadi, YOLO9000: Better, faster, stronger, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2017, pp. 7263–7271. doi:10.1109/cvpr.2017.690.
URL https://doi.org/10.1109/cvpr.2017.690

[34] D. E. Rumelhart, G. E. Hinton, R. J. Williams, Learning representations by back-propagating errors, Nature 323 (6088) (1986) 533–536. doi:10.1038/323533a0.
URL https://doi.org/10.1038/323533a0

[35] P. Werbos, Backpropagation through time: What it does and how to do it, Proc. IEEE 78 (10) (1990) 1550–1560. doi:10.1109/5.58337.
URL https://doi.org/10.1109/5.58337

[36] A. Graves, Generating sequences with recurrent neural networks, arXiv preprint arXiv:1308.0850.

[37] R. Pascanu, C. Gulcehre, K. Cho, Y. Bengio, How to construct deep recurrent neural networks, arXiv preprint arXiv:1312.6026.
[38] J. Chung, C. Gulcehre, K. Cho, Y. Bengio, Empirical evaluation of gated recurrent neural networks on sequence modeling, arXiv preprint arXiv:1412.3555.
[39] S. Xingjian, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, W.-c. Woo, Convolutional LSTM network: A machine learning approach for precipitation nowcasting, in: Advances in neural information processing systems, 2015, pp. 802–810.
[40] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, P.-A. Manzagol, L. Bottou, Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion., J. Mach. Learn. Res. 11 (12).
[41] S. Rifai, X. Muller, X. Glorot, G. Mesnil, Y. Bengio, P. Vincent, Learning invariant features through local space contraction, arXiv preprint arXiv:1104.4153.
[42] S. Rifai, P. Vincent, X. Muller, X. Glorot, Y. Bengio, Contractive auto-encoders: Explicit invariance during feature extraction, in: Icml, 2011.
[43] A. Makhzani, J. Shlens, N. Jaitly, I. Goodfellow, B. Frey, Adversarial autoencoders, arXiv preprint arXiv:1511.05644.
[44] G. Kakkavas, M. Kalntis, V. Karyotis, S. Papavassiliou, Future network traffic matrix synthesis and estimation based on deep generative models, in: 2021 International Conference on Computer Communications and Networks (ICCCN), IEEE, 2021. doi:10.1109/icccn52240.2021.9522222.
URL https://doi.org/10.1109/icccn52240.2021.9522222
[45] D. P. Kingma, M. Welling, Auto-encoding variational bayes, arXiv preprint arXiv:1312.6114.
[46] D. J. Rezende, S. Mohamed, D. Wierstra, Stochastic backpropagation and approximate inference in deep generative models, arXiv preprint arXiv:1401.4082.
[47] G. E. Hinton, S. Osindero, Y.-W. Teh, A fast learning algorithm for deep belief nets, Neural Comput. 18 (7) (2006) 1527–1554. doi:10.1162/neco.2006.18.7.1527.
URL https://doi.org/10.1162/neco.2006.18.7.1527
[48] P. Smolensky, Information processing in dynamical systems: Foundations of harmony theory, Tech. rep., Colorado Univ at Boulder Dept of Computer Science (1986).
[49] L. Deng, A tutorial survey of architectures, algorithms, and applications for deep learning, APSIPA trans. signal inf. process. 3. doi:10.1017/atsip.2013.9.
URL https://doi.org/10.1017/atsip.2013.9
[50] G. E. Hinton, Learning multiple layers of representation, Trends Cogn. Sci. 11 (10) (2007) 428–434. doi:10.1016/j.tics.2007.09.004.
URL https://doi.org/10.1016/j.tics.2007.09.004
[51] G. E. Hinton, To recognize shapes, first learn to generate images, Prog Brain Res 165 (2007) 535–547.
[52] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, in: Advances in neural information processing systems, 2014, pp. 2672–2680.
[53] M. Arjovsky, S. Chintala, L. Bottou, Wasserstein GAN, ArXiv abs/1701.07875.
[54] A. Brock, J. Donahue, K. Simonyan, Large scale gan training for high fidelity natural image synthesis, arXiv preprint arXiv:1809.11096.
[55] G.-J. Qi, Loss-sensitive generative adversarial networks on lipschitz densities, Int J Comput Vis 128 (5) (2019) 1118–1140. doi:10.1007/s11263-019-01265-2.
URL https://doi.org/10.1007/s11263-019-01265-2
[56] A. Ali-Gombe, E. Elyan, MFC-GAN: Class-imbalanced dataset classification using multiple fake class generative adversarial network, Neurocomputing 361 (2019) 212–221. doi:10.1016/j.neucom.2019.06.043.
URL https://doi.org/10.1016/j.neucom.2019.06.043
[57] A. Antoniou, A. Storkey, H. Edwards, Data augmentation generative adversarial networks, arXiv preprint arXiv:1711.04340.
[58] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, D. Hassabis, Human-level control through deep reinforcement learning, Nature 518 (7540) (2015) 529–533. doi:10.1038/nature14236.
URL https://doi.org/10.1038/nature14236
[59] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, D. Wierstra, Continuous control with deep reinforcement learning, arXiv preprint arXiv:1509.02971.
[60] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, K. Kavukcuoglu, Asynchronous methods for deep reinforcement learning, in: International conference on machine learning, 2016, pp. 1928–1937.
[61] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, Y. Chen, T. Lillicrap, F. Hui, L. Sifre, G. van den Driessche, T. Graepel, D. Hassabis, Mastering the game of go without human knowledge, Nature 550 (7676) (2017) 354–359. doi:10.1038/nature24270.
URL https://doi.org/10.1038/nature24270
[62] M. Hessel, J. Modayil, H. Van Hasselt, T. Schaul, G. Ostrovski, W. Dabney, D. Horgan, B. Piot, M. Azar, D. Silver, Rainbow: Combining improvements in deep reinforcement learning, arXiv preprint arXiv:1710.02298.
[63] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, O. Klimov, Proximal policy optimization algorithms, arXiv preprint arXiv:1707.06347.
[64] G. Lingam, R. R. Rout, D. V. Somayajulu, Adaptive deep q-learning model for detecting social bots and influential users in online social networks, Applied Intelligence 49 (11) (2019) 3947–3964.
[65] N. Zhou, J. Du, X. Yao, W. Cui, Z. Xue, M. Liang, A content search method for security topics in microblog based on deep reinforcement learning, World Wide Web 23 (1) (2020) 75–101.
[66] J. Gantz, D. Reinsel, The digital universe in 2020: Big data, bigger digital shadows, and biggest growth in the far east, IDC iView: IDC Analyze the future 2007 (2012) (2012) 1–16.
[67] L. Vu, Q. U. Nguyen, D. N. Nguyen, D. T. Hoang, E. Dutkiewicz, Deep transfer learning for IoT attack detection, IEEE Access 8 (2020) 107335–107344. doi:10.1109/access.2020.3000476.
URL https://doi.org/10.1109/access.2020.3000476
[68] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, Q. He, A comprehensive survey on transfer learning, Proc. IEEE 109 (1) (2021) 43–76. doi:10.1109/jproc.2020.3004555.
URL https://doi.org/10.1109/jproc.2020.3004555
[69] J. Zhao, S. Shetty, J. W. Pan, C. Kamhoua, K. Kwiat, Transfer learning for detecting unknown network attacks, EURASIP J. on Info. Security 2019 (1) (2019) 1. doi:10.1186/s13635-019-0084-4.
URL https://doi.org/10.1186/s13635-019-0084-4
[70] M. Mohammadi, A. Al-Fuqaha, S. Sorour, M. Guizani, Deep learning for IoT big data and streaming analytics: A survey, IEEE Commun. Surv. Tutorials 20 (4) (2018) 2923–2960. doi:10.1109/comst.2018.2844341.
URL https://doi.org/10.1109/comst.2018.2844341
[71] Y. Bengio, Deep learning of representations for unsupervised and transfer learning, in: Proceedings of ICML workshop on unsupervised and transfer learning, 2012, pp. 17–36.
[72] J. Deng, R. Xia, Z. Zhang, Y. Liu, B. Schuller, Introducing shared-hidden-layer autoencoders for transfer learning and their application in acoustic emotion recognition, in: 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, IEEE, 2014, pp. 4818–4822. doi:10.1109/icassp.2014.6854517.
URL https://doi.org/10.1109/icassp.2014.6854517
[73] N. Bendre, H. T. Marín, P. Najafirad, Learning from few samples: A survey (2020). arXiv:2007.15484.
[74] M. Chen, Y. Wang, Z. Qin, X. Zhu, Few-shot website fingerprinting attack (2021). arXiv:2101.10063.
[75] Y. Gu, H. Yan, M. Dong, M. Wang, X. Zhang, Z. Liu, F. Ren,

WiONE: One-shot learning for environment-robust device-free user authentication via commodity wi-fi in man–machine system, IEEE Transactions on Computational Social Systems 8 (3) (2021) 630–642. doi:10.1109/tcss.2021.3056654.
URL https://doi.org/10.1109/tcss.2021.3056654

[76] H. Hindy, C. Tachtatzis, R. Atkinson, D. Brosset, M. Bures, I. Andonovic, C. Michie, X. Bellekens, Leveraging siamese networks for one-shot intrusion detection model (2021). arXiv:2006.15343.

[77] P. Sirinam, N. Mathews, M. S. Rahman, M. Wright, Triplet fingerprinting: More practical and portable website fingerprinting with n-shot learning, in: Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, ACM, 2019. doi:10.1145/3319535.3354217.
URL https://doi.org/10.1145/3319535.3354217

[78] J. BROMLEY, J. W. BENTZ, L. BOTTOU, I. GUYON, Y. LECUN, C. MOORE, E. SÄCKINGER, R. SHAH, SIGNATURE VERIFICATION USING a "SIAMESE" TIME DELAY NEURAL NETWORK, International Journal of Pattern Recognition and Artificial Intelligence 07 (04) (1993) 669–688. doi:10.1142/s0218001493000339.
URL https://doi.org/10.1142/s0218001493000339

[79] E. Hoffer, N. Ailon, Deep metric learning using triplet network, in: Similarity-Based Pattern Recognition, Springer International Publishing, 2015, pp. 84–92. doi:10.1007/978-3-319-24261-3_7.
URL https://doi.org/10.1007/978-3-319-24261-3_7

[80] W. Yao, Y. Ding, X. Li, Deep learning for phishing detection, in: 2018 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Ubiquitous Computing & Communications, Big Data & Cloud Computing, Social Computing & Networking, Sustainable Computing & Communications (ISPA/IUCC/BDCloud/SocialCom/SustainCom), IEEE, IEEE, 2018, pp. 645–650. doi:10.1109/bdcloud.2018.00099.
URL https://doi.org/10.1109/bdcloud.2018.00099

[81] R. Agrawal, J. W. Stokes, K. Selvaraj, M. Marinescu, Attention in recurrent neural networks for ransomware detection, in: ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2019. doi:10.1109/icassp.2019.8682899.
URL https://doi.org/10.1109/icassp.2019.8682899

[82] Y. Huang, Q. Yang, J. Qin, W. Wen, Phishing URL detection via CNN and attention-based hierarchical RNN, in: 2019 18th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/13th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE), IEEE, IEEE, 2019, pp. 112–119. doi:10.1109/trustcom/bigdatase.2019.00024.
URL https://doi.org/10.1109/trustcom/bigdatase.2019.00024

[83] M. Macas, C. Wu, An unsupervised framework for anomaly detection in a water treatment system, in: 2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA), IEEE, IEEE, 2019, pp. 1298–1305. doi:10.1109/icmla.2019.00212.
URL https://doi.org/10.1109/icmla.2019.00212

[84] L. Yang, G. Liu, Y. Dai, J. Wang, J. Zhai, Detecting stealthy domain generation algorithms using heterogeneous deep neural network framework, Ieee Access 8 (2020) 82876–82889.

[85] R. Cao, G. Liu, Y. Xie, C. Jiang, Two-level attention model of representation learning for fraud detection, IEEE Transactions on Computational Social Systems.

[86] J. Cheng, R. He, E. Yuepeng, Y. Wu, J. You, T. Li, Real-time encrypted traffic classification via lightweight neural networks, in: GLOBECOM 2020-2020 IEEE Global Communications Conference, IEEE, 2020, pp. 1–6.

[87] X. Liu, H. Lu, A. Nayak, A spam transformer model for sms spam detection, IEEE Access 9 (2021) 80253–80263. doi:10.1109/ACCESS.2021.3081479.

[88] Y. Li, L. Zhang, Z. Lv, W. Wang, Detecting anomalies in intelligent vehicle charging and station power supply systems with multi-head attention models, IEEE Transactions on Intelligent Transportation Systems 22 (1) (2021) 555–564. doi:10.1109/TITS.2020.3018259.

[89] S. Chaudhari, V. Mithal, G. Polatkan, R. Ramanath, An attentive survey of attention models (2021). arXiv:1904.02874.

[90] B. Sun, W. Yang, M. Yan, D. Wu, Y. Zhu, Z. Bai, An encrypted traffic classification method combining graph convolutional network and autoencoder, in: 2020 IEEE 39th International Performance Computing and Communications Conference (IPCCC), IEEE, 2020, pp. 1–8.

[91] Z. Guo, Y. Shen, A. K. Bashir, M. Imran, N. Kumar, D. Zhang, K. Yu, Robust spammer detection using collaborative neural network in internet of thing applications, IEEE Internet Things J. (2020) 1–1doi:10.1109/jiot.2020.3003802.
URL https://doi.org/10.1109/jiot.2020.3003802

[92] S. Ji, J. Li, Q. Yuan, J. Lu, Multi-range gated graph neural network for telecommunication fraud detection, in: 2020 International Joint Conference on Neural Networks (IJCNN), IEEE, 2020, pp. 1–6.

[93] D. Cheng, X. Wang, Y. Zhang, L. Zhang, Graph neural network for fraud detection via spatial-temporal attention, IEEE Transactions on Knowledge and Data Engineering.

[94] B. Bowman, H. H. Huang, Towards next-generation cybersecurity with graph ai, SIGOPS Oper. Syst. Rev. 55 (1) (2021) 61–67. doi:10.1145/3469379.3469386.
URL https://doi.org/10.1145/3469379.3469386

[95] N. Sun, J. Zhang, P. Rimba, S. Gao, L. Y. Zhang, Y. Xiang, Data-driven cybersecurity incident prediction: A survey, IEEE Communications Surveys & Tutorials 21 (2) (2019) 1744–1772. doi:10.1109/comst.2018.2885561.
URL https://doi.org/10.1109/comst.2018.2885561

[96] Y. Bengio, I. Goodfellow, A. Courville, Deep learning, Vol. 1, Citeseer, 2017.

[97] Kdd. KDD cup task presentation [online] (Sep. 1999). [Accessed: Dec 9, 2020].

[98] S. García, M. Grill, J. Stiborek, A. Zunino, An empirical comparison of botnet detection methods, Comput. Secur. 45 (2014) 100–123. doi:10.1016/j.cose.2014.05.011.
URL https://doi.org/10.1016/j.cose.2014.05.011

[99] A. Shiravi, H. Shiravi, M. Tavallaee, A. A. Ghorbani, Toward developing a systematic approach to generate benchmark datasets for intrusion detection, Comput. Secur. 31 (3) (2012) 357–374. doi:10.1016/j.cose.2011.12.012.
URL https://doi.org/10.1016/j.cose.2011.12.012

[100] C. Kolias, G. Kambourakis, A. Stavrou, S. Gritzalis, Intrusion detection in 802.11 networks: Empirical evaluation of threats and a public dataset, IEEE Commun. Surv. Tutorials 18 (1) (2016) 184–208. doi:10.1109/comst.2015.2402161.
URL https://doi.org/10.1109/comst.2015.2402161

[101] I. Sharafaldin, A. H. Lashkari, A. A. Ghorbani. Intrusion detection evaluation dataset (cic-ids2017). [online]. [Accessed: May 25, 2021].

[102] I. Sharafaldin, A. H. Lashkari, A. A. Ghorbani. Cse-cic-ids2018 on aws. [online]. [Accessed: May 25, 2021].

[103] I. Sharafaldin, A. H. Lashkari, S. Hakak, A. A. Ghorbani, Developing realistic distributed denial of service (ddos) attack dataset and taxonomy, in: 2019 International Carnahan Conference on Security Technology (ICCST), IEEE, 2019, pp. 1–8.

[104] R. Damasevicius, A. Venckauskas, S. Grigaliunas, J. Toldinas, N. Morkevicius, T. Aleliunas, P. Smuikys, Litnet-2020: An annotated real-world network flow dataset for network intrusion detection, Electronics 9 (5) (2020) 800.

[105] A. Alsaedi, N. Moustafa, Z. Tari, A. Mahmood, A. Anwar, TON_IoT telemetry dataset: A new generation dataset of IoT and IIoT for data-driven intrusion detection systems, IEEE Access 8 (2020) 165130–165150. doi:10.1109/access.2020.3022862.
URL https://doi.org/10.1109/access.2020.3022862

[106] S. Laboratory. A labeled dataset with malicious and benign iot network traffic [online]. [Accessed: Sep 2, 2021].

[107] Y. Meidan, M. Bohadana, Y. Mathov, Y. Mirsky, A. Shab-

tai, D. Breitenbacher, Y. Elovici, N-BaIoT—Network-based detection of IoT botnet attacks using deep autoencoders, IEEE Pervasive Comput. 17 (3) (2018) 12–22. doi:10.1109/mprv.2018.03367731.
URL https://doi.org/10.1109/mprv.2018.03367731

[108] Y. M. P. Pa, S. Suzuki, K. Yoshioka, T. Matsumoto, T. Kasama, C. Rossow, IoTPOT: A novel honeypot for revealing current IoT threats, Journal of Information Processing 24 (3) (2016) 522–533. doi:10.2197/ipsjjip.24.522.
URL https://doi.org/10.2197/ipsjjip.24.522

[109] Virusshare. Virustotal [online] (Jun. 2020). [Accessed: Feb 2, 2021].

[110] Y. Zhou, X. Jiang, Dissecting android malware: Characterization and evolution, in: 2012 IEEE Symposium on Security and Privacy, IEEE, 2012, pp. 95–109. doi:10.1109/sp.2012.16.
URL https://doi.org/10.1109/sp.2012.16

[111] C. mobile. Contagio mobile, mobile malware mini dump [online] (Jun. 2019). [Accessed: Dec 20, 2020].

[112] K. Allix, T. F. Bissyandé, J. Klein, Y. Le Traon, AndroZoo, in: Proceedings of the 13th International Conference on Mining Software Repositories, IEEE, ACM, 2016, pp. 468–471. doi:10.1145/2901739.2903508.
URL https://doi.org/10.1145/2901739.2903508

[113] A. Internet. Alexa top sites. [online]. [Accessed: Nov 9, 2021].

[114] B. Consulting. Free osint tools [online]. [Accessed: Sep 2, 2021].

[115] P. Daniel. Dgarchive [online]. [Accessed: Sep 2, 2021].

[116] N. S. R. L. at 360. Netlab dga project. [online]. [Accessed: May 25, 2021].

[117] Cisco. Cisco annual internet report [online] (Jan. 2020). [Accessed: Dec 20, 2020].

[118] R. Vinayakumar, K. Soman, P. Poornachandran, M. Alazab, S. Thampi, Amritadga: a comprehensive data set for domain generation algorithms (dgas) based domain name detection systems and application of deep learning, in: Big Data Recommender Systems-Volume 2: Application Paradigms, Institution of Engineering and Technology (IET), 2019, pp. 455–485.

[119] M. Zago, M. G. Pérez, G. M. Pérez, UMUDGA: A dataset for profiling DGA-based botnet, Computers & Security 92 (2020) 101719. doi:10.1016/j.cose.2020.101719.
URL https://doi.org/10.1016/j.cose.2020.101719

[120] M. Aghashahi, R. Sundararajan, M. Pourahmadi, M. K. Banks, Water distribution systems analysis Symposium–Battle of the attack detection algorithms (BATADAL), in: World Environmental and Water Resources Congress 2017, American Society of Civil Engineers, 2017, pp. 101–108. doi:10.1061/9780784480595.010.
URL https://doi.org/10.1061/9780784480595.010

[121] J. Goh, S. Adepu, K. N. Junejo, A. Mathur, A dataset to support research in the design of secure water treatment systems, in: International Conference on Critical Information Infrastructures Security, Springer, 2016, pp. 88–99.

[122] C. M. Ahmed, V. R. Palleti, A. P. Mathur, Wadi, in: Proceedings of the 3rd International Workshop on Cyber-Physical Systems for Smart Water Networks, ACM, ACM, 2017, pp. 25–28. doi:10.1145/3055366.3055375.
URL https://doi.org/10.1145/3055366.3055375

[123] H.-K. Shin, W. Lee, J.-H. Yun, H. Kim, HAI 1.0: Hil-based augmented ICS security dataset, in: 13th USENIX Workshop on Cyber Security Experimentation and Test (CSET 20), USENIX Association, 2020.
URL https://www.usenix.org/conference/cset20/presentation/shin

[124] C. Castillo, D. Donato, L. Becchetti, P. Boldi, S. Leonardi, M. Santini, S. Vigna, A reference collection for web spam, in: ACM Sigir Forum, Vol. 40, ACM New York, NY, USA, 2006, pp. 11–24.

[125] kaggle. Utkml's twitter spam detection competition [online] (2019). [Accessed: May 25, 2021].

[126] K. Lee, B. Eoff, J. Caverlee, Seven months with the devils: A long-term study of content polluters on twitter, in: Proceedings of the International AAAI Conference on Web and Social Media, Vol. 5, 2011.

[127] T. A. Almeida, J. M. G. Hidalgo, A. Yamakami, Contributions to the study of sms spam filtering: new collection and results, in: Proceedings of the 11th ACM symposium on Document engineering, 2011, pp. 259–262.

[128] G. Draper-Gil, A. H. Lashkari, M. S. I. Mamun, A. A. Ghorbani, Characterization of encrypted and vpn traffic using time-related, in: Proceedings of the 2nd international conference on information systems security and privacy (ICISSP), 2016, pp. 407–414.

[129] A. H. Lashkari, G. Draper-Gil, M. S. I. Mamun, A. A. Ghorbani, Characterization of tor traffic using time based features., in: ICISSp, 2017, pp. 253–262.

[130] S. Wazen, C. Thibault, F. Jerome, C. Isabelle. Https websites dataset [online] (2016). [Accessed: May 25, 2021].

[131] S. Rezaei, X. Liu, How to achieve high classification accuracy with just a few labels: A semi-supervised approach using sampled packets, arXiv preprint arXiv:1812.09761.

[132] M. Tavallaee, E. Bagheri, W. Lu, A. A. Ghorbani, A detailed analysis of the KDD CUP 99 data set, in: 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications, IEEE, IEEE, 2009, pp. 1–6. doi:10.1109/cisda.2009.5356528.
URL https://doi.org/10.1109/cisda.2009.5356528

[133] R. Taormina, S. Galelli, H. Douglas, N. Tippenhauer, E. Salomons, A. Ostfeld, A toolbox for assessing the impacts of cyber-physical attacks on water distribution systems, Environ. Modell. Softw. 112 (2019) 46–51. doi:10.1016/j.envsoft.2018.11.008.
URL https://doi.org/10.1016/j.envsoft.2018.11.008

[134] T. U. of Southern California. The DETER project [online] (Jan. 2012). [Accessed: Jun 20, 2021].

[135] S. Université. FIT future internet testing facility [online] (Jan. 2019). [Accessed: Jun 20, 2021].

[136] NITlab. Nitos facility [online] (Jan. 2019). [Accessed: Jun 20, 2021].

[137] Orbit. Open-access research testbed for next-generation wireless networks (ORBIT) [online] (Jan. 2016). [Accessed: Jun 20, 2021].

[138] F. Consortium. Open-access research testbed for next-generation wireless networks (ORBIT) [online] (Jan. 2017). [Accessed: Jun 20, 2021].

[139] T. K. Lengyel, S. Maresca, B. D. Payne, G. D. Webster, S. Vogl, A. Kiayias, Scalability, fidelity and stealth in the DRAKVUF dynamic malware analysis system, in: Proceedings of the 30th Annual Computer Security Applications Conference on - ACSAC '14, ACM Press, 2014. doi:10.1145/2664243.2664252.
URL https://doi.org/10.1145/2664243.2664252

[140] T. U. of Utah. Emulab [online] (Jan. 2019). [Accessed: Jun 20, 2021].

[141] D. Raychaudhuri, I. Seskar, G. Zussman, T. Korakis, D. Kilper, T. Chen, J. Kolodziejski, M. Sherman, Z. Kostic, X. Gu, H. Krishnaswamy, S. Maheshwari, P. Skrimponis, C. Gutterman, Challenge, in: Proceedings of the 26th Annual International Conference on Mobile Computing and Networking, ACM, 2020, pp. 1–13. doi:10.1145/3372224.3380891.
URL https://doi.org/10.1145/3372224.3380891

[142] J. Cappos, M. Hemmings, R. McGeer, A. Rafetseder, G. Ricart, EdgeNet: A global cloud that spreads by local action, in: 2018 IEEE/ACM Symposium on Edge Computing (SEC), IEEE, IEEE, 2018, pp. 359–360. doi:10.1109/sec.2018.00045.
URL https://doi.org/10.1109/sec.2018.00045

[143] T. A. Tang, L. Mhamdi, D. McLernon, S. A. R. Zaidi, M. Ghogho, Deep learning approach for network intrusion detection in software defined networking, in: 2016 International Conference on Wireless Networks and Mobile Communications (WINCOM), IEEE, IEEE, 2016, pp. 258–263. doi:10.1109/wincom.2016.7777224.
URL https://doi.org/10.1109/wincom.2016.7777224

[144] M. S. Elsayed, N.-A. Le-Khac, S. Dev, A. D. Jurcut, Ddosnet: A deep-learning model for detecting network attacks, in: 2020 IEEE 21st International Symposium on" A World of Wireless, Mobile and Multimedia Networks"(WoWMoM), IEEE, 2020, pp. 391–396.

[145] S. Otoum, B. Kantarci, H. T. Mouftah, On the feasibility of deep learning in sensor network intrusion detection, IEEE Netw. Lett. 1 (2) (2019) 68–71. doi:10.1109/lnet.2019.2901792.
URL https://doi.org/10.1109/lnet.2019.2901792

[146] L. Yang, J. Li, L. Yin, Z. Sun, Y. Zhao, Z. Li, Real-time intrusion detection in wireless network: A deep learning-based intelligent mechanism, IEEE Access 8 (2020) 170128–170139.

[147] S. Otoum, B. Kantarci, H. Mouftah, Adaptively supervised and intrusion-aware data aggregation for wireless sensor clusters in critical infrastructures, in: 2018 IEEE International Conference on Communications (ICC), IEEE, 2018, pp. 1–6. doi:10.1109/icc.2018.8422401.
URL https://doi.org/10.1109/icc.2018.8422401

[148] M. Antonakakis, T. April, M. Bailey, M. Bernhard, E. Bursztein, J. Cochran, Z. Durumeric, J. A. Halderman, L. Invernizzi, M. Kallitsis, D. Kumar, C. Lever, Z. Ma, J. Mason, D. Menscher, C. Seaman, N. Sullivan, K. Thomas, Y. Zhou, Understanding the mirai botnet, in: 26th USENIX Security Symposium (USENIX Security 17), USENIX Association, Vancouver, BC, 2017, pp. 1093–1110.
URL https://www.usenix.org/conference/usenixsecurity17/technical-sessions/presentation/antonakakis

[149] A. Abeshu, N. Chilamkurti, Deep learning: The frontier for distributed attack detection in fog-to-things computing, IEEE Commun. Mag. 56 (2) (2018) 169–175. doi:10.1109/mcom.2018.1700332.
URL https://doi.org/10.1109/mcom.2018.1700332

[150] A. Diro, N. Chilamkurti, Leveraging LSTM networks for attack detection in fog-to-things communications, IEEE Commun. Mag. 56 (9) (2018) 124–130. doi:10.1109/mcom.2018.1701270.
URL https://doi.org/10.1109/mcom.2018.1701270

[151] K. Bresniker, A. Gavrilovska, J. Holt, D. Milojicic, T. Tran, Grand challenge: Applying artificial intelligence and machine learning to cybersecurity, Computer 52 (12) (2019) 45–52. doi:10.1109/mc.2019.2942584.
URL https://doi.org/10.1109/mc.2019.2942584

[152] Y. Xiao, Y. Jia, C. Liu, X. Cheng, J. Yu, W. Lv, Edge computing security: State of the art and challenges, Proceedings of the IEEE 107 (8) (2019) 1608–1631.

[153] H. Yao, P. Gao, P. Zhang, J. Wang, C. Jiang, L. Lu, Hybrid intrusion detection system for edge-based IIoT relying on machine-learning-aided detection, IEEE Network 33 (5) (2019) 75–81. doi:10.1109/mnet.001.1800479.
URL https://doi.org/10.1109/mnet.001.1800479

[154] A. Ferdowsi, W. Saad, Generative adversarial networks for distributed intrusion detection in the internet of things, in: 2019 IEEE Global Communications Conference (GLOBECOM), IEEE, 2019, pp. 1–6. doi:10.1109/globecom38437.2019.9014102.
URL https://doi.org/10.1109/globecom38437.2019.9014102

[155] C. Hardy, E. Le Merrer, B. Sericola, MD-GAN: Multi-discriminator generative adversarial networks for distributed datasets, in: 2019 IEEE International Parallel and Distributed Processing Symposium (IPDPS), IEEE, IEEE, 2019, pp. 866–877. doi:10.1109/ipdps.2019.00095.
URL https://doi.org/10.1109/ipdps.2019.00095

[156] D. Anguita, A. Ghio, L. Oneto, X. Parra, J. L. Reyes-Ortiz, A public domain dataset for human activity recognition using smartphones., in: Esann, 2013.

[157] M. Abdel-Basset, H. Hawash, R. K. Chakrabortty, M. J. Ryan, Semi-supervised spatio-temporal deep learning for intrusions detection in iot networks, IEEE Internet of Things Journal.

[158] N. Ravi, S. M. Shalinie, Semisupervised-learning-based security to detect and mitigate intrusions in iot network, IEEE Internet of Things Journal 7 (11) (2020) 11041–11052.

[159] S. Rezvy, Y. Luo, M. Petridis, A. Lasebae, T. Zebin, An efficient deep learning model for intrusion classification and prediction in 5G and IoT networks, in: 2019 53rd Annual Conference on Information Sciences and Systems (CISS), IEEE, IEEE, 2019, pp. 1–6. doi:10.1109/ciss.2019.8693059.
URL https://doi.org/10.1109/ciss.2019.8693059

[160] L. Nie, W. Sun, S. Wang, Z. Ning, J. J. Rodrigues, Y. Wu, S. Li, Intrusion detection in green internet of things: A deep deterministic policy gradient-based algorithm, IEEE Transactions on Green Communications and Networking 5 (2) (2021) 778–788.

[161] L. Nie, Z. Ning, M. S. Obaidat, B. Sadoun, H. Wang, S. Li, L. Guo, G. Wang, A reinforcement learning-based network traffic prediction mechanism in intelligent internet of things, IEEE Transactions on Industrial Informatics 17 (3) (2020) 2169–2180.

[162] G. Kakkavas, A. Stamou, V. Karyotis, S. Papavassiliou, Network tomography for efficient monitoring in SDN-enabled 5G networks and beyond: Challenges and opportunities, IEEE Communications Magazine 59 (3) (2021) 70–76. doi:10.1109/mcom.001.2000458.
URL https://doi.org/10.1109/mcom.001.2000458

[163] L. Nie, Y. Wu, X. Wang, L. Guo, G. Wang, X. Gao, S. Li, Intrusion detection for secure social internet of things based on collaborative edge computing: A generative adversarial network-based approach, IEEE Transactions on Computational Social Systems.

[164] P. Vepakomma, T. Swedish, R. Raskar, O. Gupta, A. Dubey, No peek: A survey of private distributed deep learning, arXiv preprint arXiv:1812.03288.

[165] J. Verbraeken, M. Wolting, J. Katzy, J. Kloppenburg, T. Verbelen, J. S. Rellermeyer, A survey on distributed machine learning, ACM Comput. Surv. 53 (2) (2020) 1–33. doi:10.1145/3377454.
URL https://doi.org/10.1145/3377454

[166] H. B. McMahan, E. Moore, D. Ramage, B. A. y Arcas, Federated learning of deep networks using model averaging, CoRR abs/1602.05629. arXiv:1602.05629.
URL http://arxiv.org/abs/1602.05629

[167] X. Wang, Y. Han, V. C. Leung, D. Niyato, X. Yan, X. Chen, Convergence of edge computing and deep learning: A comprehensive survey, IEEE Commun. Surv. Tutorials 22 (2) (2020) 869–904. doi:10.1109/comst.2020.2970550.
URL https://doi.org/10.1109/comst.2020.2970550

[168] X. Wang, S. Garg, H. Lin, J. Hu, G. Kaddoum, M. J. Piran, M. S. Hossain, Towards accurate anomaly detection in industrial internet-of-things using hierarchical federated learning, IEEE Internet of Things Journal (2021) 1–1doi:10.1109/jiot.2021.3074382.
URL https://doi.org/10.1109/jiot.2021.3074382

[169] E. Bagdasaryan, A. Veit, Y. Hua, D. Estrin, V. Shmatikov, How to backdoor federated learning, in: International Conference on Artificial Intelligence and Statistics, PMLR, 2020, pp. 2938–2948.

[170] P. Faruki, A. Bharmal, V. Laxmi, V. Ganmoor, M. S. Gaur, M. Conti, M. Rajarajan, Android security: A survey of issues, malware penetration, and defenses, IEEE Commun. Surv. Tutorials 17 (2) (2015) 998–1022. doi:10.1109/comst.2014.2386139.
URL https://doi.org/10.1109/comst.2014.2386139

[171] I. Gartner. Gartner says worldwide smartphone sales will grow 3% in 2020 [online] (Jan. 2020). [Accessed: Dec 20, 2020].

[172] MalwarebytesLABS. 2019 state of malware [online] (Jan. 2019). [Accessed: Nov 20, 2020].

[173] E. B. Karbab, M. Debbabi, A. Derhab, D. Mouheb, MalDozer: Automatic framework for android malware detection using deep learning, Digit. Investig. 24 (2018) S48–S59. doi:10.1016/j.diin.2018.01.007.
URL https://doi.org/10.1016/j.diin.2018.01.007

[174] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, J. Dean, Dis-

tributed representations of words and phrases and their compositionality, in: Advances in neural information processing systems, 2013, pp. 3111–3119.
[175] N. C. S. University. Android malware genome project [online] (Aug. 2012). [Accessed: Feb 20, 2021].
[176] D. Arp, M. Spreitzenbarth, M. Hübner, H. Gascon, K. Rieck, Drebin: Effective and explainable detection of android malware in your pocket, in: Proceedings 2014 Network and Distributed System Security Symposium, Vol. 14, Internet Society, 2014, pp. 23–26. doi:10.14722/ndss.2014.23247.
URL https://doi.org/10.14722/ndss.2014.23247
[177] G. LLC. Google play store. [online]. [Accessed: May 25, 2021].
[178] R. Feng, S. Chen, X. Xie, L. Ma, G. Meng, Y. Liu, S.-W. Lin, MobiDroid: A performance-sensitive malware detection system on mobile platform, in: 2019 24th International Conference on Engineering of Complex Computer Systems (ICECCS), IEEE, IEEE, 2019, pp. 61–70. doi:10.1109/iceccs.2019.00014.
URL https://doi.org/10.1109/iceccs.2019.00014
[179] Contagiodump. Contagio malware dump [online] (Jun. 2020). [Accessed: Feb 2, 2021].
[180] R. Feng, S. Chen, X. Xie, G. Meng, S.-W. Lin, Y. Liu, A performance-sensitive malware detection system using deep learning on mobile devices, IEEE Trans.Inform.Forensic Secur. 16 (2021) 1563–1578. doi:10.1109/tifs.2020.3025436.
URL https://doi.org/10.1109/tifs.2020.3025436
[181] I. U. Haq, T. A. Khan, A. Akhunzada, A dynamic robust dl-based model for android malware detection, IEEE Access 9 (2021) 74510–74521.
[182] F. Wei, Y. Li, S. Roy, X. Ou, W. Zhou, Deep ground truth analysis of current android malware, in: International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment, Springer, 2017, pp. 252–276.
[183] A. Azmoodeh, A. Dehghantanha, K.-K. R. Choo, Robust malware detection for internet of (Battlefield) things devices using deep eigenspace learning, IEEE Trans. Sustain. Comput. 4 (1) (2019) 88–95. doi:10.1109/tsusc.2018.2809665.
URL https://doi.org/10.1109/tsusc.2018.2809665
[184] J. Jeon, J. H. Park, Y.-S. Jeong, Dynamic analysis for IoT malware detection with convolution neural network model, IEEE Access 8 (2020) 96899–96911. doi:10.1109/access.2020.2995887.
URL https://doi.org/10.1109/access.2020.2995887
[185] M. Dib, S. Torabi, E. Bou-Harb, C. Assi, A multi-dimensional deep learning framework for iot malware classification and family attribution, IEEE Transactions on Network and Service Management.
[186] T. Kim, B. Kang, M. Rho, S. Sezer, E. G. Im, A multimodal deep learning method for android malware detection using various features, IEEE Trans.Inform.Forensic Secur. 14 (3) (2019) 773–788. doi:10.1109/tifs.2018.2866319.
URL https://doi.org/10.1109/tifs.2018.2866319
[187] T. P. Hut. Raspberry pi store. [online]. [Accessed: May 25, 2021].
[188] D. M. Research. Global botnet detection market – industry trends and forecast to 2027 [online]. [Accessed: Jun 20, 2021].
[189] J. Kim, A. Sim, J. Kim, K. Wu, Botnet detection using recurrent variational autoencoder, in: GLOBECOM 2020-2020 IEEE Global Communications Conference, IEEE, 2020, pp. 1–6.
[190] R. Vinayakumar, M. Alazab, S. Srinivasan, Q.-V. Pham, S. K. Padannayil, K. Simran, A visualized botnet detection system based deep learning for the internet of things networks of smart cities, IEEE Transactions on Industry Applications 56 (4) (2020) 4436–4456.
[191] R. R. Curtin, A. B. Gardner, S. Grzonkowski, A. Kleymenov, A. Mosquera, Detecting DGA domains with recurrent neural networks and side information, in: Proceedings of the 14th International Conference on Availability, Reliability and Security, ACM, ACM, 2019, p. 20. doi:10.1145/3339252.3339258.
URL https://doi.org/10.1145/3339252.3339258
[192] D. Tran, H. Mac, V. Tong, H. A. Tran, L. G. Nguyen, A LSTM based framework for handling multiclass imbalance in DGA botnet detection, Neurocomputing 275 (2018) 2401–2413. doi:10.1016/j.neucom.2017.11.018.
URL https://doi.org/10.1016/j.neucom.2017.11.018
[193] J. Woodbridge, H. S. Anderson, A. Ahuja, D. Grant, Predicting domain generation algorithms with long short-term memory networks, arXiv preprint arXiv:1611.00791.
[194] P. Lison, V. Mavroeidis, Neural reputation models learned from passive DNS data, in: 2017 IEEE International Conference on Big Data (Big Data), IEEE, IEEE, 2017, pp. 3662–3671. doi:10.1109/bigdata.2017.8258361.
URL https://doi.org/10.1109/bigdata.2017.8258361
[195] J. Spaulding, A. Mohaisen, Defending internet of things against malicious domain names using d-FENS, in: 2018 IEEE/ACM Symposium on Edge Computing (SEC), IEEE, IEEE, 2018, pp. 387–392. doi:10.1109/sec.2018.00051.
URL https://doi.org/10.1109/sec.2018.00051
[196] Cisco. Umbrella popularity list. [online]. [Accessed: May 25, 2021].
[197] T.-Y. Lin, P. Goyal, R. Girshick, K. He, P. Dollár, Focal loss for dense object detection, in: Proceedings of the IEEE international conference on computer vision, 2017, pp. 2980–2988.
[198] Y. Xu, X. Yan, Y. Wu, Y. Hu, W. Liang, J. Zhang, Hierarchical bidirectional rnn for safety-enhanced b5g heterogeneous networks, IEEE Transactions on Network Science and Engineering.
[199] Mnemonic. Passive dns [online]. [Accessed: Sep 2, 2021].
[200] J. Bromley, J. W. Bentz, L. Bottou, I. Guyon, Y. LeCun, C. Moore, E. Säckinger, R. Shah, Signature verification using a "siamese" time delay neural network, International Journal of Pattern Recognition and Artificial Intelligence 7 (04) (1993) 669–688.
[201] V. Ravi, M. Alazab, S. Srinivasan, A. Arunachalam, K. Soman, Adversarial defense: Dga-based botnets and dns homographs detection through integrated deep learning, IEEE Transactions on Engineering Management.
[202] J. Woodbridge, H. S. Anderson, A. Ahuja, D. Grant, Detecting homoglyph attacks with a siamese neural network, in: 2018 IEEE Security and Privacy Workshops (SPW), IEEE, 2018, pp. 22–28.
[203] P. Agten, W. Joosen, F. Piessens, N. Nikiforakis, Seven months' worth of mistakes: A longitudinal study of typosquatting abuse, in: Proceedings 2015 Network and Distributed System Security Symposium, Internet Society, Internet Society, 2015. doi:10.14722/ndss.2015.23058.
URL https://doi.org/10.14722/ndss.2015.23058
[204] O. R. R. C. Research. Global cyber physical systems market 2020 by company, regions, type and application, forecast to 2025. [online]. [Accessed: Sep 5, 2021].
[205] I. Stellios, P. Kotzanikolaou, M. Psarakis, C. Alcaraz, J. Lopez, A survey of IoT-enabled cyberattacks: Assessing attack paths to critical infrastructures and services, IEEE Commun. Surv. Tutorials 20 (4) (2018) 3453–3495. doi:10.1109/comst.2018.2855563.
URL https://doi.org/10.1109/comst.2018.2855563
[206] S. D. Anton, D. Fraunholz, C. Lipps, F. Pohl, M. Zimmermann, H. D. Schotten, Two decades of SCADA exploitation: A brief history, in: 2017 IEEE Conference on Application, Information and Network Security (AINS), IEEE, IEEE, 2017, pp. 98–104. doi:10.1109/ains.2017.8270432.
URL https://doi.org/10.1109/ains.2017.8270432
[207] R. Khatoun, S. Zeadally, Cybersecurity and privacy solutions in smart cities, IEEE Commun. Mag. 55 (3) (2017) 51–59. doi:10.1109/mcom.2017.1600297cm.
URL https://doi.org/10.1109/mcom.2017.1600297cm
[208] D. U. Case, Analysis of the cyber attack on the Ukrainian power grid, Electricity Information Sharing and Analysis Center (E-ISAC) 388.
[209] H. Boyes, Cybersecurity and cyber-resilient supply chains, Technology Innovation Management Review 5 (4) (2015) 28–

34. doi:10.22215/timreview888.
URL https://doi.org/10.22215/timreview888

[210] M. Macas, W. Chunming, Enhanced cyber-physical security through deep learning techniques, in: Proc. CPS Summer School Ph. D. Workshop, 2019, pp. 72–83.

[211] J. Goh, S. Adepu, M. Tan, Z. S. Lee, Anomaly detection in cyber physical systems using recurrent neural networks, in: 2017 IEEE 18th International Symposium on High Assurance Systems Engineering (HASE), IEEE, IEEE, 2017, pp. 140–145. doi:10.1109/hase.2017.36.
URL https://doi.org/10.1109/hase.2017.36

[212] J. Inoue, Y. Yamagata, Y. Chen, C. M. Poskitt, J. Sun, Anomaly detection for a water treatment system using unsupervised machine learning, in: 2017 IEEE International Conference on Data Mining Workshops (ICDMW), IEEE, IEEE, 2017, pp. 1058–1065. doi:10.1109/icdmw.2017.149.
URL https://doi.org/10.1109/icdmw.2017.149

[213] M. Kravchik, A. Shabtai, Detecting cyber attacks in industrial control systems using convolutional neural networks, in: Proceedings of the 2018 Workshop on Cyber-Physical Systems Security and PrivaCy, ACM, 2018, pp. 72–83. doi:10.1145/3264888.3264896.
URL https://doi.org/10.1145/3264888.3264896

[214] M. Kravchik, A. Shabtai, Efficient cyber attack detection in industrial control systems using lightweight neural networks and PCA, IEEE Trans. Dependable and Secure Comput. (2021) 1–1doi:10.1109/tdsc.2021.3050101.
URL https://doi.org/10.1109/tdsc.2021.3050101

[215] X. Xie, B. Wang, T. Wan, W. Tang, Multivariate abnormal detection for industrial control systems using 1D CNN and GRU, IEEE Access 8 (2020) 88348–88359. doi:10.1109/access.2020.2993335.
URL https://doi.org/10.1109/access.2020.2993335

[216] K.-D. Lu, G.-Q. Zeng, X. Luo, J. Weng, W. Luo, Y. Wu, Evolutionary deep belief network for cyber-attack detection in industrial automation and control system, IEEE Transactions on Industrial Informatics.

[217] S. Boettcher, A. Percus, Nature's way of optimizing, Artificial intelligence 119 (1-2) (2000) 275–286.

[218] T. Morris, W. Gao, Industrial control system traffic data sets for intrusion detection research, in: International Conference on Critical Infrastructure Protection, Springer, 2014, pp. 65–78.

[219] S. Huda, J. Yearwood, M. M. Hassan, A. Almogren, Securing the operations in scada-iot platform based industrial control system using ensemble of deep belief networks, Applied soft computing 71 (2018) 66–77.

[220] B. Hussain, Q. Du, B. Sun, Z. Han, Deep learning-based ddos-attack detection for cyber–physical system over 5g network, IEEE Transactions on Industrial Informatics 17 (2) (2021) 860–870. doi:10.1109/TII.2020.2974520.

[221] G. Barlacchi, M. De Nadai, R. Larcher, A. Casella, C. Chitic, G. Torrisi, F. Antonelli, A. Vespignani, A. Pentland, B. Lepri, A multi-source dataset of urban life in the city of milan and the province of trentino, Scientific data 2 (1) (2015) 1–15.

[222] Y. He, G. J. Mendis, J. Wei, Real-time detection of false data injection attacks in smart grid: A deep learning-based intelligent mechanism, IEEE Trans. Smart Grid 8 (5) (2017) 2505–2516. doi:10.1109/tsg.2017.2703842.
URL https://doi.org/10.1109/tsg.2017.2703842

[223] X. Niu, J. Li, J. Sun, K. Tomsovic, Dynamic detection of false data injection attack in smart grid using deep learning, in: 2019 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT), IEEE, IEEE, 2019, pp. 1–6. doi:10.1109/isgt.2019.8791598.
URL https://doi.org/10.1109/isgt.2019.8791598

[224] J. Wang, D. Shi, Y. Li, J. Chen, H. Ding, X. Duan, Distributed framework for detecting PMU data manipulation attacks with deep autoencoders, IEEE Trans. Smart Grid 10 (4) (2019) 4401–4410. doi:10.1109/tsg.2018.2859339.
URL https://doi.org/10.1109/tsg.2018.2859339

[225] Y. Wang, D. Chen, C. Zhang, X. Chen, B. Huang, X. Cheng, Wide and recurrent neural networks for detection of false data injection in smart grids, in: International Conference on Wireless Algorithms, Systems, and Applications, Springer, 2019, pp. 335–345.

[226] Y. Zhang, J. Wang, B. Chen, Detecting false data injection attacks in smart grids: A semi-supervised deep learning approach, IEEE Transactions on Smart Grid 12 (1) (2020) 623–634.

[227] K. P. Schneider, B. Mather, B. Pal, C.-W. Ten, G. J. Shirek, H. Zhu, J. C. Fuller, J. L. R. Pereira, L. F. Ochoa, L. R. de Araujo, et al., Analytic considerations and design basis for the ieee distribution test feeders, IEEE Transactions on power systems 33 (3) (2017) 3181–3188.

[228] I. Siniosoglou, P. Radoglou-Grammatikis, G. Efstathopoulos, P. Fouliras, P. Sarigiannidis, A unified deep learning anomaly detection and classification approach for smart grid environments, IEEE Transactions on Network and Service Management.

[229] P. R. Grammatikis, P. Sarigiannidis, E. Iturbe, E. Rios, A. Sarigiannidis, O. Nikolis, D. Ioannidis, V. Machamint, M. Tzifas, A. Giannakoulias, et al., Secure and private smart grid: The spear architecture, in: 2020 6th IEEE Conference on Network Softwarization (NetSoft), IEEE, 2020, pp. 450–456.

[230] F. van Wyk, Y. Wang, A. Khojandi, N. Masoud, Real-time sensor anomaly detection and identification in automated vehicles, IEEE Trans. Intell. Transport. Syst. 21 (3) (2020) 1264–1276. doi:10.1109/tits.2019.2906038.
URL https://doi.org/10.1109/tits.2019.2906038

[231] D. Bezzina, J. Sayer, Safety pilot model deployment: Test conductor team report, Report No. DOT HS 812 (171) (2014) 18.

[232] D. A. Hahn, A. Munir, V. Behzadan, Security and privacy issues in intelligent transportation systems: Classification and challenges, IEEE Intell. Transp. Syst 1.

[233] M. Hanselmann, T. Strauss, K. Dormann, H. Ulmer, Canet: An unsupervised intrusion detection system for high dimensional can bus data, IEEE Access 8 (2020) 58194–58205.

[234] D.-A. Clevert, T. Unterthiner, S. Hochreiter, Fast and accurate deep network learning by exponential linear units (elus), arXiv preprint arXiv:1511.07289.

[235] C. Yue, L. Wang, D. Wang, R. Duo, X. Nie, An ensemble intrusion detection method for train ethernet consist network based on cnn and rnn, IEEE Access 9 (2021) 59527–59539.

[236] M. Hanselmann, T. Strauss, K. Dormann, H. Ulmer. Syncan dataset [online]. [Accessed: Jun 20, 2021].

[237] D. E. Sorkin. Spam statistics and facts. [online]. [Accessed: Jun 19, 2021].

[238] B. Feng, Q. Fu, M. Dong, D. Guo, Q. Li, Multistage and elastic spam detection in mobile social networks through deep learning, IEEE Network 32 (4) (2018) 15–21. doi:10.1109/mnet.2018.1700406.
URL https://doi.org/10.1109/mnet.2018.1700406

[239] Y. Gao, M. Gong, Y. Xie, A. Qin, An attention-based unsupervised adversarial model for movie review spam detection, IEEE Trans. Multimedia 23 (2021) 784–796. doi:10.1109/tmm.2020.2990085.
URL https://doi.org/10.1109/tmm.2020.2990085

[240] S. Madisetty, M. S. Desarkar, A neural network-based ensemble approach for spam detection in twitter, IEEE Trans. Comput. Soc. Syst. 5 (4) (2018) 973–984. doi:10.1109/tcss.2018.2878852.
URL https://doi.org/10.1109/tcss.2018.2878852

[241] A. Makkar, N. Kumar, An efficient deep learning-based scheme for web spam detection in IoT environment, Future Gener. Comp. Sy. 108 (2020) 467–487. doi:10.1016/j.future.2020.03.004.
URL https://doi.org/10.1016/j.future.2020.03.004

[242] P. K. Roy, J. P. Singh, S. Banerjee, Deep learning to filter SMS spam, Future Gener. Comp. Sy. 102 (2020) 524–533. doi:10.1016/j.future.2019.09.001.
URL https://doi.org/10.1016/j.future.2019.09.001

[243] S. Seth, S. Biswas, Multimodal spam classification using deep learning techniques, in: 2017 13th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS), IEEE, IEEE, 2017, pp. 346–349. doi:10.1109/sitis.2017.91.
URL https://doi.org/10.1109/sitis.2017.91

[244] T. Wu, S. Liu, J. Zhang, Y. Xiang, Twitter spam detection based on deep learning, in: Proceedings of the Australasian Computer Science Week Multiconference, ACM, 2017, pp. 1–8. doi:10.1145/3014812.3014815.
URL https://doi.org/10.1145/3014812.3014815

[245] C. Yang, R. Harkreader, G. Gu, Empirical evaluation and new design for fighting evolving twitter spammers, IEEE Trans.Inform.Forensic Secur. 8 (8) (2013) 1280–1293. doi:10.1109/tifs.2013.2267732.
URL https://doi.org/10.1109/tifs.2013.2267732

[246] A. Makkar, U. Ghosh, P. K. Sharma, Artificial intelligence and edge computing-enabled web spam detection for next generation iot applications, IEEE Sensors Journal.

[247] S. Sedhai, A. Sun, HSpam14, in: Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, 2015, pp. 223–232. doi:10.1145/2766462.2767701.
URL https://doi.org/10.1145/2766462.2767701

[248] B. Wang, A. Zubiaga, M. Liakata, R. Procter, Making the most of tweet-inherent features for social spam detection on twitter, arXiv preprint arXiv:1503.07405.

[249] G. Lingam, R. R. Rout, D. V. L. N. Somayajulu, S. K. Ghosh, Particle swarm optimization on deep reinforcement learning for detecting social spam bots and spam-influential users in twitter network, IEEE Systems Journal 15 (2) (2021) 2281–2292. doi:10.1109/JSYST.2020.3034416.

[250] S. Cresci, R. Di Pietro, M. Petrocchi, A. Spognardi, M. Tesconi, The paradigm-shift of social spambots: Evidence, theories, and tools for the arms race, in: Proceedings of the 26th international conference on world wide web companion, 2017, pp. 963–972.

[251] T. Xu, G. Goossen, H. K. Cevahir, S. Khodeir, Y. Jin, F. Li, S. Shan, S. Patel, D. Freeman, P. Pearce, Deep entity classification: Abusive account detection for online social networks, in: 30th USENIX Security Symposium (USENIX Security 21), USENIX Association, 2021, pp. 4097–4114.
URL https://www.usenix.org/conference/usenixsecurity21/presentation/xu-teng

[252] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, arXiv preprint arXiv:1706.03762.

[253] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al., Language models are few-shot learners, arXiv preprint arXiv:2005.14165.

[254] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, arXiv preprint arXiv:1810.04805.

[255] J. Cao, C. Lai, A bilingual multi-type spam detection model based on m-bert, in: GLOBECOM 2020-2020 IEEE Global Communications Conference, IEEE, 2020, pp. 1–6.

[256] Y. Kou, C.-T. Lu, S. Sirwongwattana, Y.-P. Huang, Survey of fraud detection techniques, in: IEEE International Conference on Networking, Sensing and Control, 2004, Vol. 2, IEEE, 2004, pp. 749–754.

[257] M. Intelligence. Fraud detection and prevention market - growth, trends, covid-19 impact, and forecasts (2021 - 2026) [online]. [Accessed: Jun 20, 2021].

[258] S. Pandit, J. Liu, R. Perdisci, M. Ahamad, Applying deep learning to combat mass robocalls, in: 2021 IEEE Security and Privacy Workshops (SPW), 2021, pp. 63–70. doi:10.1109/SPW53761.2021.00018.

[259] S. Xu, S. Lai, Y. Li, A deep learning based framework for cloud masquerade attack detection, in: 2018 IEEE 37th International Performance Computing and Communications Conference (IPCCC), IEEE, IEEE, 2018, pp. 1–2. doi:10.1109/pccc.2018.8711277.
URL https://doi.org/10.1109/pccc.2018.8711277

[260] C.-Y. Yu, C. K. Chang, W. Zhang, An edge computing based situation enabled crowdsourcing blacklisting system for efficient identification of scammer phone numbers, in: 2020 International Conference on Computational Science and Computational Intelligence (CSCI), IEEE, 2020, pp. 776–781.

[261] S. Rezaei, X. Liu, Deep learning for encrypted traffic classification: An overview, IEEE Commun. Mag. 57 (5) (2019) 76–81. doi:10.1109/mcom.2019.1800819.
URL https://doi.org/10.1109/mcom.2019.1800819

[262] K. Abe, S. Goto, Fingerprinting attack on tor anonymity using deep learning, Proceedings of the Asia-Pacific Advanced Network 42 (2016) 15–20.

[263] V. Rimmer, D. Preuveneers, M. Juárez, T. van Goethem, W. Joosen, Automated website fingerprinting through deep learning, in: 25th Annual Network and Distributed System Security Symposium, NDSS 2018, San Diego, California, USA, February 18-21, 2018, The Internet Society, 2018. doi:10.14722/ndss.2018.23105.
URL https://doi.org/10.14722/ndss.2018.23105

[264] P. Sirinam, M. Imani, M. Juarez, M. Wright, Deep fingerprinting: Undermining website fingerprinting defenses with deep learning, in: Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, 2018, pp. 1928–1943.

[265] G. Aceto, D. Ciuonzo, A. Montieri, A. Pescapé, Mobile encrypted traffic classification using deep learning, in: 2018 Network traffic measurement and analysis conference (TMA), IEEE, 2018, pp. 1–8.

[266] T. Shapira, Y. Shavitt, Flowpic: Encrypted internet traffic classification is as easy as image recognition, in: IEEE INFOCOM 2019-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), IEEE, 2019, pp. 680–687.

[267] W. Wang, M. Zhu, X. Zeng, X. Ye, Y. Sheng, Malware traffic classification using convolutional neural network for representation learning, in: 2017 International Conference on Information Networking (ICOIN), IEEE, 2017, pp. 712–717.

[268] S. Rezaei, X. Liu, Multitask learning for network traffic classification, in: 2020 29th International Conference on Computer Communications and Networks (ICCCN), IEEE, 2020, pp. 1–9.

[269] P. Wang, Z. Wang, F. Ye, X. Chen, Bytesgan: A semi-supervised generative adversarial network for encrypted traffic classification of sdn edge gateway in green communication network, arXiv preprint arXiv:2103.05250.

[270] H. Wu, L. Wang, G. Cheng, X. Hu, Mobile application encryption traffic classification based on tls flow sequence network, in: 2021 IEEE International Conference on Communications Workshops (ICC Workshops), IEEE, 2021, pp. 1–6.

[271] T. Wang. Website fingerprinting [online]. [Accessed: May 25, 2021].

[272] V. Rimmer, D. Preuveneers, M. Juarez, T. Van Goethem, W. Joosen. Dataset-website fingerprinting. [online]. [Accessed: Jun 20, 2021].

[273] A. Radford, L. Metz, S. Chintala, Unsupervised representation learning with deep convolutional generative adversarial networks, arXiv preprint arXiv:1511.06434.

[274] J. Su, V. Danilo Vasconcellos, S. Prasad, S. Daniele, Y. Feng, K. Sakurai, Lightweight classification of IoT malware based on image recognition, in: 2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC), Vol. 2, IEEE, IEEE, 2018, pp. 664–669. doi:10.1109/compsac.2018.10315.
URL https://doi.org/10.1109/compsac.2018.10315

[275] D. Sahoo, Q. Pham, J. Lu, S. C. H. Hoi, Online deep learning: Learning deep neural networks on the fly (2017). arXiv:1711.03705.

[276] M. McCloskey, N. J. Cohen, Catastrophic interference in connectionist networks: The sequential learning problem, in: Psychology of Learning and Motivation, Elsevier, 1989, pp. 109–165. doi:10.1016/s0079-7421(08)60536-8.

URL https://doi.org/10.1016/s0079-7421(08)60536-8
[277] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, D. Hassabis, C. Clopath, D. Kumaran, R. Hadsell, Overcoming catastrophic forgetting in neural networks, Proceedings of the National Academy of Sciences 114 (13) (2017) 3521–3526. doi:10.1073/pnas.1611835114.
URL https://doi.org/10.1073/pnas.1611835114
[278] R. Kemker, M. McClure, A. Abitino, T. Hayes, C. Kanan, Measuring catastrophic forgetting in neural networks, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 32, 2018.
[279] P. Tune, D. Veitch, Sampling vs sketching: An information theoretic comparison, in: 2011 Proceedings IEEE INFOCOM, IEEE, 2011. doi:10.1109/infcom.2011.5935020.
URL https://doi.org/10.1109/infcom.2011.5935020
[280] Z. Liu, A. Manousis, G. Vorsanger, V. Sekar, V. Braverman, One sketch to rule them all, in: Proceedings of the 2016 ACM SIGCOMM Conference, ACM, 2016. doi:10.1145/2934872.2934906.
URL https://doi.org/10.1145/2934872.2934906
[281] T. Yang, J. Jiang, P. Liu, Q. Huang, J. Gong, Y. Zhou, R. Miao, X. Li, S. Uhlig, Elastic sketch, in: Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication, ACM, 2018. doi:10.1145/3230543.3230544.
URL https://doi.org/10.1145/3230543.3230544
[282] J.-Y. Li, T. Chow, Y.-L. Yu, The estimation theory and optimization algorithm for the number of hidden units in the higher-order feedforward neural network, in: Proceedings of ICNN'95 - International Conference on Neural Networks, IEEE. doi:10.1109/icnn.1995.487330.
URL https://doi.org/10.1109/icnn.1995.487330
[283] D. Menotti, G. Chiachia, A. Pinto, W. Robson Schwartz, H. Pedrini, A. Xavier Falcao, A. Rocha, Deep representations for iris, face, and fingerprint spoofing detection, IEEE Trans.Inform.Forensic Secur. 10 (4) (2015) 864–879. doi:10.1109/tifs.2015.2398817.
URL https://doi.org/10.1109/tifs.2015.2398817
[284] A. Panesar, Evaluating Machine Learning Models, Apress, 2020, Ch. Evaluating Machine Learning Models, pp. 189–205. doi:10.1007/978-1-4842-6537-6_7.
URL https://doi.org/10.1007/978-1-4842-6537-6_7
[285] X. He, K. Zhao, X. Chu, AutoML: A survey of the state-of-the-art, Knowledge-Based Systems 212 (2021) 106622. doi:10.1016/j.knosys.2020.106622.
URL https://doi.org/10.1016/j.knosys.2020.106622
[286] H. Hindy, D. Brosset, E. Bayne, A. K. Seeam, C. Tachtatzis, R. Atkinson, X. Bellekens, A taxonomy of network threats and the effect of current datasets on intrusion detection systems, IEEE Access 8 (2020) 104650–104675. doi:10.1109/access.2020.3000179.
URL https://doi.org/10.1109/access.2020.3000179
[287] A. Gharib, I. Sharafaldin, A. H. Lashkari, A. A. Ghorbani, An evaluation framework for intrusion detection dataset, in: 2016 International Conference on Information Science and Security (ICISS), IEEE, IEEE, 2016, pp. 1–6. doi:10.1109/icissec.2016.7885840.
URL https://doi.org/10.1109/icissec.2016.7885840
[288] I. Sharafaldin, A. Habibi Lashkari, A. A. Ghorbani, Toward generating a new intrusion detection dataset and intrusion traffic characterization, in: Proceedings of the 4th International Conference on Information Systems Security and Privacy, SCITEPRESS - Science and Technology Publications, 2018, pp. 108–116. doi:10.5220/0006639801080116.
URL https://doi.org/10.5220/0006639801080116
[289] C. Chio, D. Freeman, Machine Learning and Security: Protecting Systems with Data and Algorithms, " O'Reilly Media, Inc.", 2018.
[290] R. F. Fouladi, T. Seifpoor, E. Anarim, Frequency characteristics of DoS and DDoS attacks, in: 2013 21st Signal Processing and Communications Applications Conference (SIU), IEEE, IEEE, 2013, pp. 1–4. doi:10.1109/siu.2013.6531200.
URL https://doi.org/10.1109/siu.2013.6531200
[291] P. Ruvolo, E. Eaton, ELLA: An efficient lifelong learning algorithm, in: S. Dasgupta, D. McAllester (Eds.), Proceedings of the 30th International Conference on Machine Learning, Vol. 28 of Proceedings of Machine Learning Research, PMLR, Atlanta, Georgia, USA, 2013, pp. 507–515.
URL https://proceedings.mlr.press/v28/ruvolo13.html
[292] P. Ren, Y. Xiao, X. Chang, P.-Y. Huang, Z. Li, X. Chen, X. Wang, A survey of deep active learning (2020). arXiv:2009.00236.
[293] D. Shu, N. O. Leslie, C. A. Kamhoua, C. S. Tucker, Generative adversarial attacks against intrusion detection systems using active learning, in: Proceedings of the 2nd ACM Workshop on Wireless Security and Machine Learning, 2020, pp. 1–6.
[294] A. Shahraki, M. Abbasi, A. Taherkordi, A. D. Jurcut, Active learning for network traffic classification: A technical study (2021). arXiv:2106.06933.
[295] F.-L. Fan, J. Xiong, M. Li, G. Wang, On interpretability of artificial neural networks: A survey, IEEE Transactions on Radiation and Plasma Medical Sciences (2021) 1–1doi:10.1109/trpms.2021.3066428.
URL https://doi.org/10.1109/trpms.2021.3066428
[296] J. R. Geis, A. P. Brady, C. C. Wu, J. Spencer, E. Ranschaert, J. L. Jaremko, S. G. Langer, A. B. Kitts, J. Birch, W. F. Shields, R. van den Hoven van Genderen, E. Kotter, J. W. Gichoya, T. S. Cook, M. B. Morgan, A. Tang, N. M. Safdar, M. Kohli, Ethics of artificial intelligence in radiology: Summary of the joint european and north american multisociety statement, Canadian Association of Radiologists Journal 70 (4) (2019) 329–334. doi:10.1016/j.carj.2019.08.010.
URL https://doi.org/10.1016/j.carj.2019.08.010
[297] M. Wang, K. Zheng, Y. Yang, X. Wang, An explainable machine learning framework for intrusion detection systems, IEEE Access 8 (2020) 73127–73141. doi:10.1109/access.2020.2988359.
URL https://doi.org/10.1109/access.2020.2988359
[298] S. M. Lundberg, S.-I. Lee, A unified approach to interpreting model predictions, in: Proceedings of the 31st International Conference on Neural Information Processing Systems, NIPS'17, Curran Associates Inc., Red Hook, NY, USA, 2017, p. 4768–4777.
[299] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, M. A. Riedmiller, Deterministic policy gradient algorithms, in: Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014, Vol. 32 of JMLR Workshop and Conference Proceedings, JMLR.org, 2014, pp. 387–395.
URL http://proceedings.mlr.press/v32/silver14.html
[300] I. Grondman, L. Busoniu, G. A. Lopes, R. Babuska, A survey of actor-critic reinforcement learning: Standard and natural policy gradients, IEEE Trans. Syst., Man, Cybern. C 42 (6) (2012) 1291–1307. doi:10.1109/tsmcc.2012.2218595.
URL https://doi.org/10.1109/tsmcc.2012.2218595
[301] T. Jung, S. Kim, K. Kim, DeepVision: Deepfakes detection using human eye blinking pattern, IEEE Access 8 (2020) 83144–83154. doi:10.1109/access.2020.2988660.
URL https://doi.org/10.1109/access.2020.2988660
[302] T. T. Nguyen, C. M. Nguyen, D. T. Nguyen, D. T. Nguyen, S. Nahavandi, Deep learning for deepfakes creation and detection, arXiv preprint arXiv:1909.11573 1.
[303] S. Rezaei, X. Liu, Security of deep learning methodologies: Challenges and opportunities, arXiv preprint arXiv:1912.03735.
[304] T. T. Nguyen, V. J. Reddi, Deep reinforcement learning for cyber security, arXiv preprint arXiv:1906.05799.
[305] Y. Liu, S. Ma, Y. Aafer, W. Lee, J. Zhai, W. Wang, X. Zhang, Trojaning attack on neural networks, in: 25th Annual Network and Distributed System Security Symposium, NDSS 2018, San Diego, California, USA, February 18-21, 2018, The Internet Society, 2018.

URL http://wp.internetsociety.org/ndss/wp-content/uploads/sites/25/2018/02/ndss2018_03A-5_Liu_paper.pdf
[306] X. Chen, C. Liu, B. Li, K. Lu, D. Song, Targeted backdoor attacks on deep learning systems using data poisoning, arXiv preprint arXiv:1712.05526.
[307] T. Gu, K. Liu, B. Dolan-Gavitt, S. Garg, BadNets: Evaluating backdooring attacks on deep neural networks, IEEE Access 7 (2019) 47230–47244. doi:10.1109/access.2019.2909068.
URL https://doi.org/10.1109/access.2019.2909068
[308] C.-Y. Hsu, P.-Y. Chen, S. Lu, S. Liu, C.-M. Yu, Adversarial examples for unsupervised machine learning modelsarXiv:2103.01895.
[309] Gdpr.eu. General data protection regulation (GDPR) compliance guidelines. [online] (2020). [Accessed: Sep 5, 2021].
[310] B. McMahan, E. Moore, D. Ramage, S. Hampson, B. A. y. Arcas, Communication-Efficient Learning of Deep Networks from Decentralized Data, Vol. 54 of Proceedings of Machine Learning Research, PMLR, 2017, pp. 1273–1282.
URL https://proceedings.mlr.press/v54/mcmahan17a.html
[311] V. Smith, C.-K. Chiang, M. Sanjabi, A. Talwalkar, Federated multi-task learning, in: Proceedings of the 31st International Conference on Neural Information Processing Systems, NIPS'17, Curran Associates Inc., Red Hook, NY, USA, 2017, p. 4427–4437.
[312] L. T. Phong, Y. Aono, T. Hayashi, L. Wang, S. Moriai, Privacy-preserving deep learning via additively homomorphic encryption, IEEE Transactions on Information Forensics and Security 13 (5) (2018) 1333–1345. doi:10.1109/tifs.2017.2787987.
URL https://doi.org/10.1109/tifs.2017.2787987
[313] D. Gao, Y. Liu, A. Huang, C. Ju, H. Yu, Q. Yang, Privacy-preserving heterogeneous federated transfer learning, in: 2019 IEEE International Conference on Big Data (Big Data), IEEE, 2019. doi:10.1109/bigdata47090.2019.9005992.
URL https://doi.org/10.1109/bigdata47090.2019.9005992
[314] H. Yang, H. He, W. Zhang, X. Cao, FedSteg: A federated transfer learning framework for secure image steganalysis, IEEE Transactions on Network Science and Engineering 8 (2) (2021) 1084–1094. doi:10.1109/tnse.2020.2996612.
URL https://doi.org/10.1109/tnse.2020.2996612
[315] X. Yin, Y. Zhu, J. Hu, A comprehensive survey of privacy-preserving federated learning, ACM Computing Surveys 54 (6) (2021) 1–36. doi:10.1145/3460427.
URL https://doi.org/10.1145/3460427
[316] O. Alkadi, N. Moustafa, B. Turnbull, K.-K. R. Choo, A deep blockchain framework-enabled collaborative intrusion detection for protecting IoT and cloud networks, IEEE Internet of Things Journal 8 (12) (2021) 9463–9472. doi:10.1109/jiot.2020.2996590.
URL https://doi.org/10.1109/jiot.2020.2996590
[317] X. Liu, H. Li, G. Xu, S. Liu, Z. Liu, R. Lu, PADL: Privacy-aware and asynchronous deep learning for IoT applications, IEEE Internet of Things Journal 7 (8) (2020) 6955–6969. doi:10.1109/jiot.2020.2981379.
URL https://doi.org/10.1109/jiot.2020.2981379
[318] R. Gilad-Bachrach, N. Dowlin, K. Laine, K. Lauter, M. Naehrig, J. Wernsing, Cryptonets: Applying neural networks to encrypted data with high throughput and accuracy, in: M. F. Balcan, K. Q. Weinberger (Eds.), Proceedings of The 33rd International Conference on Machine Learning, Vol. 48 of Proceedings of Machine Learning Research, PMLR, New York, New York, USA, 2016, pp. 201–210.
URL https://proceedings.mlr.press/v48/gilad-bachrach16.html
[319] K. Nandakumar, N. Ratha, S. Pankanti, S. Halevi, Towards deep neural network training on encrypted data, in: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), IEEE, 2019. doi:10.1109/cvprw.2019.00011.
URL https://doi.org/10.1109/cvprw.2019.00011
[320] Z. Wu, S. Shen, X. Lian, X. Su, E. Chen, A dummy-based user privacy protection approach for text information retrieval, Knowledge-Based Systems 195 (2020) 105679. doi:10.1016/j.knosys.2020.105679.
URL https://doi.org/10.1016/j.knosys.2020.105679
[321] Z. Wu, S. Shen, H. Zhou, H. Li, C. Lu, D. Zou, An effective approach for the protection of user commodity viewing privacy in e-commerce website, Knowledge-Based Systems 220 (2021) 106952. doi:10.1016/j.knosys.2021.106952.
URL https://doi.org/10.1016/j.knosys.2021.106952
[322] Z. Wu, G. Li, Q. Liu, G. Xu, E. Chen, Covering the sensitive subjects to protect personal privacy in personalized recommendation, IEEE Transactions on Services Computing 11 (3) (2018) 493–506. doi:10.1109/tsc.2016.2575825.
URL https://doi.org/10.1109/tsc.2016.2575825
[323] Z. Wu, G. Xu, C. Lu, E. Chen, F. Jiang, G. Li, An effective approach for the protection of privacy text data in the CloudDB, World Wide Web 21 (4) (2017) 915–938. doi:10.1007/s11280-017-0491-8.
URL https://doi.org/10.1007/s11280-017-0491-8
[324] Z. Wu, R. Wang, Q. Li, X. Lian, G. Xu, E. Chen, X. Liu, A location privacy-preserving system based on query range cover-up or location-based services, IEEE Transactions on Vehicular Technology 69 (5) (2020) 5244–5254. doi:10.1109/tvt.2020.2981633.
URL https://doi.org/10.1109/tvt.2020.2981633
[325] Z. Wu, G. Li, S. Shen, X. Lian, E. Chen, G. Xu, Constructing dummy query sequences to protect location privacy and query privacy in location-based services, World Wide Web 24 (1) (2020) 25–49. doi:10.1007/s11280-020-00830-x.
URL https://doi.org/10.1007/s11280-020-00830-x
[326] A. Shafique, A. Mehmood, M. Elhadef, Survey of security protocols and vulnerabilities in unmanned aerial vehicles, IEEE Access 9 (2021) 46927–46948. doi:10.1109/access.2021.3066778.
URL https://doi.org/10.1109/access.2021.3066778
[327] J. Ratcliffe, F. Soave, N. Bryan-Kinns, L. Tokarchuk, I. Farkhatdinov, Extended reality (XR) remote research: a survey of drawbacks and opportunities, in: Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems, ACM, 2021. doi:10.1145/3411764.3445170.
URL https://doi.org/10.1145/3411764.3445170