A Comprehensive Tutorial and Survey of Applications of Deep Learning for Cyber Security

Vinayakumar R, Soman KP, Mamoun Alazab, *Senior Member, IEEE*, Sriram S, Simran K

***Abstract*—Deep learning (DL), a novel research direction in machine learning (ML) field, has achieved great success in many classical artificial intelligence (AI) tasks in comparison to classical ML algorithms (CMLAs). DL architectures are relatively recent and currently wisely used for diverse Cyber Security applications. This work aims to review the state-of-the-art DL architectures in Cyber Security applications by highlighting the contributions and challenges from various recent research papers. Initially, the concepts of most popular ML algorithms and DL architectures are discussed along with the mathematical representations. Following, we review the emerging researches of DL architectures for diverse anticipated applications of Cyber Security. This include Intrusion detection, Malware and Botnet detection, Spam and Phishing detection, Network traffic analysis, Binary analysis, Insider threat detection, CAPTCHA analysis, steganography. Additionally, the importance of DL architectures are discussed for cryptography, cloud security, biometric security, smart cities specific to Internet of things (IoT) and fog computing. We discuss the importance of big data, natural language processing, signal and image processing, blockchain technology, casual theory key concepts towards Cyber Security. Finally the paper concludes with a summary of the current state-of-the-art, a critical discussion of open challenges and directions for future research additionally we propose and discuss general DL based Cyber Security system.**

***Index Terms*—Cyber Security, Machine Learning, Neural Networks, Deep Learning.**

I. Introduction

RECENT days Internet and its technologies have become ubiquitous due to the rapid advancement in technologies particularly cloud computing, mobile computing, fog computing, IoT, etc. Internet has become an essential resource for people: in 2014, about 40% of the world's population uses the Internet and this figure increases up to 78% in the developed countries [1]. People very soon realized that the best way not only to promote sketchy products but also to steal account credentials, and spread computer viruses was to utilize the medium of wide distribution and free transmission created by researchers who had connected computers together by means of Internet to create a communications network that offers some value. All chunks of information concerned to Internet technologies and information stored in data bases, transmitted over the network can be protected by following the methods available in the field of Cyber Security.

Nowadays it seems that not a day goes by without a new story on the topic of Cyber Security, either a security incident on information leakage, or an abuse of an emerging technology such as autonomous car hacking, or the software we have been using for years is now deemed to be dangerous because of the newly found security vulnerabilities. Furthermore, as systems become increasingly complex and interconnected, it becomes harder and harder to ensure that there are no bugs or back doors that will give attackers a way in. According to the Symantec cyber crime report [2], the overall number of vulnerabilities has increased by 13% in 2018. Similarly, according to Cyber Security Ventures [3], zero-day exploits seen in the wild will grow from one per week (in 2015) to one per day by 2021. It is practically impossible for a human to keep pace with the sheer number of Cyber Security events (and related activities) on a daily basis on top of an already daunting threat landscape [3]. Furthermore, there is a crisis of skilled Cyber Security practitioners. According to study [4], the Cyber Security job market will grow by approximately 6 million USD globally by 2019, with potential shortages of trained professionals up to 25%. Automation of decisions and actions based on network and system generated alerts has the potential to help overcome the challenges related to security and privacy - both in a technological and business operations (e.g. labour shortages) dimension. The accurate detection of cyber-attacks, cyber-threat, intrusions and malicious software programs is an increasingly important problem in Cyber Security for ICT systems. Because the single incident related to cyber-attacks, cyber-threat, intrusions and malicious software programs can cause millions of dollars of damages. The increased vulnerability in data related to personal, corporate, and government information places a new urgency on Cyber Security.

Cyber Security has become an important area of research due to the explosive growth in the number of attacks to the computers and networks. This contains a set of concepts and procedures to protect ICT systems and networks, both hardware and software from malicious software programs and more importantly data from unauthorized access, theft, disclosure, as well as intentional or accidental harm [6]. Cyber Security evolves over time as the technology evolves to cope with the new types of patterns of malicious activity. To attack various threats, ID, social network security, malware analysis, advanced persistent threats, web application security, and applied cryptography, are few used tools in Cyber Security. However, even spam remains a major focus in an email system. This is also called as information technology security or elec-

This work was supported by the Department of Corporate and Information Services, Northern Territory Government of Australia, the Paramount Computer Systems, and Lakhshya Cyber Security Labs.

Vinayakumar R, Soman KP, Sriram S, and Simran K was with Center for Computational Engineering and Networking(CEN), Amrita School of Engineering, Coimbatore Amrita Vishwa Vidyapeetham, India e-mail: (vinayakumarr77@gmail.com).

Mamoun Alazab was with Charles Darwin University, Australia e-mail: (mamoun.alazab@cdu.edu.au).

tronic security. More formally Cyber Security is defined by [7], the preservation of confidentiality, integrity and availability of information in the Cyberspace. Cyber Security is a broad terms and it includes information security, network security, Internet security, Critical information infrastructure protection, cyber crime, cyber safety and ICT security. Information security deals with protection of electronic data from unauthorized access.

Network security includes integrity of networks and it supports the flow of information on networks within organizations and users. Internet security is an extension of network security which deals with the protection of Internet related services, ICT systems and networks. Critical information infrastructure protection deals with the confirmation of systems & networks protection and it also stays safe against information security risks, network security risks, Internet security risks, as well as Cyber Security risks. Cyber crime deals with criminal activities carried out in the cyber space. Cyber safety deals with the protection against various attacks in the cyberspace. There is no definition defined by ISO for ICT security, but it deals with the technical origins of computer security and the CIA principle.

Even though there is an increase in miscreants and adversaries in the field of Cyber Security over time, there has been no change in general threat categories. The main objective of security research is to prevent the attackers from achieving their goals and therefore, it is extremely important to have a good knowledge of various types of attacks. The existing Cyber Security systems have not adapted fast enough. In this dynamic technology environment, the only way to have a solid cyber defense is to have an adaptive Cyber Security framework that reacts and responds to changes proactively. Cyber Security world is rich with information, various logs to malware files, network traffic, just to name a few. The Cyber Security domain generates huge amount of data from network sensors, logs, and endpoint agents. Processing and extracting information from huge amount of data can provide information about malicious activities. The constant increase in the number of attacks to the ICT systems indicates that the classical Cyber Security tools and practices are not able to cope with the sophisticated threat landscape. Due to the large volume, large variety, large velocity and large veracity as the Big Data (BD) characteristics, BD causes many challenging issues on Data Mining (DM) and information processing. The data generated by various Internet technologies required to be processed to apply analysis. This analysis facilitates to extract the hidden features to differentiate between normal and malicious activities. The legacy Cyber Security solutions such as network-level and host-level firewalls, antivirus software, Intrustion Detection Systems (IDSs), and intrusion protection systems (IPSs) available in market are effective at detecting the known malicious activities. These solutions are based on rule based and rules have to be continually updated by Cyber Security domain experts. These systems completely fail to detect new types of malicious activities. Additionally, the legacy Cyber Security solutions are ineffective due to the reason that the amount of malicious activities landscape is rapidly changing and increasing. These huge volume data typically called as BD that creates additional challenges to Cyber Security. BD analytics has the capability to collect, store, process and visualize very large volume of data. Therefore, applying BD analytics to Cyber Security becomes critical and a new trend. The utilization of data from networks and computers facilitates the analysts to discover important patterns for legitimate and malicious activities.

As the technology evolves over time, the data generated by the end user system will follow different patterns. Moreover these technologies might suffer from vulnerabilities. The methods that we adopt should be capable to recognize the new kind of patterns that the system gets. Due to today's rapidly evolving technologies, security researchers are exploring the applicability of cognitive technologies typically called as cognitive security system which is basically ML to better anticipate and defend against cyber threats. AI as a Cyber Security tool is expected to capture a large market and it is clear that AI has the potential to impact the Cyber Security space [8]. Furthermore, there is sufficient market interest in both commercial (financial incentives) and academic research. It is understood that, there is a potential to mislead ML/DL deployment as discussed in existing literature [9], [10].

In recent years, ML and DL has become an essential tool for various applications in the field of NLP, computer vision and speech processing. The DL architectures have obtained better performance in compared to CMLAs and more importantly outperformed human performance in several computer vision and health related applications [11]. In recent days, the security researchers are employing DL methods to handle evolving malicious activities landscape. The application of ML and DL techniques are quickly moving past the domains of the scholarly world and theoretical fiction to enter the business standard. One of the major disadvantages of CMLAs is the reliance on the feature engineering methods. Feature engineering is a method dictated by domain experts to identify the important features or properties of each problem. DL scales well for very large amount of data samples compared to CMLAs since it can capture the important features from complex systems [9]. As the data of Cyber Security continuously grow day by day with the evolution of technology, the performance of DL based solutions also improve. CMLAs are shallow models where as DL architectures are deep in nature. Generally, DL architectures are composed of more than one hidden layer that helps to learn hierarchical representation. These architectures have obtained state-of-the-art performance is several long standing AI tasks. In recent days, variety of DL architecture designs have blossomed in the context of Cyber Security. On the other side, attackers are using the same tools to increase the sophistication of their attacks [11]. Thus, the DL based systems have to stay ahead of the bad actors.

A. Existing Surveys on Machine Learning and Deep Learning Applications in Cyber Security

Though there are many related surveys based on machine learning and deep learning applications in Cyber Security exist, to the best of our knowledge, a detailed survey on DL applications was not surveyed in such scope yet. In [626],

[627], [628], [629], [630], [631] describes classical machine learning frameworks for solving Cyber Security problems without including deep learning methodologies. Even though other authors have utilized deep learning frameworks for cyber security but for a very narrow set of applications of cyber security. [632] focusing on attacks related to intrusion detection whereas in [633], the work covers attacks related to not only intrusion detection but spam detection, and malware analysis also. [634] is a summary and review of the work related defending cyber-physical systems. Work related to Machine learning as well as deep learning methodologies for securing IoT technology was discussed in [635]. The speciality of this paper is that it covers a wide range of cyber attacks as well as the frameworks utilized for detecting them. The Deep learning frameworks used in this paper include CNNs, RNNs, and GANs. In [636] proposed a generalized deep learning framework for Cyber Security. The framework composed of sub modules for network traffic, android apps, PE files, emails and websites data analysis using deep learning. Additionally, the deep learning based works of each sub modules are summarized in detail. The article also provides a short tutorial on various deep learning architecture. In [637] surveyed the deep learning applications in detection of various attacks including malware, spam, insider threats, network intrusions, false data injection, and malicious domain names used by botnets. Additionally, the authors discussed the importance of benchmark datasets in Cyber Security.

B. Literature Search Methodology

Various journals and conference proceedings were covered by the literature search of this survey. Paper published in arXiv had been gone through because of nature of the work of this paper.

First, we reviewed survey-oriented journals related to ML and DL applications in Cyber Security like IEEE Communication Surveys and Tutorials, IEEEAccess other reputated journals and arXiv. Subsequently, we used IEEE Xplore, Google Scholar, and ACM Digital Library in order to search for the related papers utilizing the queries "Cyber Security with "Machine learning, "Cyber Security with "Neural Networks, "Cyber Security with "Deep Learning and "Deep Leraning". Later, we looked at the papers for publication citations or cited by already found works. We also looked at papers having the same author. The publications are presented in chronological order from 2000 to Mar, 2019. The papers published before 2000 are not incorporated into this work except when they are still highly relevant or have fundamental contribution. Despite the fact that the quantities of citations assessed by Scopus and Google Scholar used to identify the most influential research papers, we didn't take into account citation as the main important factor to choose the papers.

In this work, we systematically summarize all the published deep learning based Cyber Security applications related articles. The major contributions of the present survey of deep learning applications in Cyber Security research work are as follows:

1) Deep learning architectures that have been employed for various Cyber Security applications are reviewed and a walk-through of their evolution is provided. Additionally, this survey work summarize, compare and contrast the various deep learning architectures and forward a detailed understanding of the past, present and future deep learning applications in Cyber Security.
2) The classification of deep learning applications in Cyber Security based on the type of architecture and application, year, text representation, type of dataset and comparision with classical machine learning is presented.
3) Various issues and major challenges existing in Cyber Security applications involved in off-line and real-time deployment are discussed. Additionally, the importance of shared tasks in the field of Cyber Security is explained.
4) Importance of reinforcement learning, adversarial machine learning, and transfer learning applications in Cyber Security is examined.
5) Role and importance of big data in the field of Cyber Security are discussed.
6) Importance of Cyber Security in the field of smart cities, pervasive computing, biometric, IoT, fog computing, cloud technologies, and autonomous vehicle is discussed.
7) Significance of unsupervised learning for Cyber Security over semi-supervised and supervised learning is summarized.
8) Role of natural language processing, signal and image processing in Cyber Security applications is summarized.
9) Importance of explainable AI, visualization and hybrid framework in Cyber Security is summarized.
10) Many publically available datasets for various Cyber Security applications are reviewed and suggestions for future research directions are provided.

C. Paper Organization

This paper is divided into twenty five sections. Section II discusses various CMLAs and DL architectures. Section III discusses the application of NLP in Cyber Security. Section IV discusses the importance of signal and image processing techniques for Cyber Security. Section V discusses the importance of BD for Cyber Security. Section VI discusses the major issues exists in Cyber Security and the importance of shared tasks in Cyber Security. Most commonly used statistical measures in the context of DL based applications in Cyber Security are reported in Section VII. Section VIII discusses the importance of transfer learning approach in Cyber Security. Section IX contains adversarial DL in Cyber Security. Section X contains Reinforcement Learning (RL) in Cyber Security. Application of DL in Cyber Security in Section XI. Section XII contains the importance of DL in IoT security. Section XIII contains the importance DL with blockchain technology in Cyber Security. Section XIV contains the significance of DL in cryptography. Section XV discusses the importance of DL in cloud security. Section XVI discusses the importance of DL in biometric security. Section XVII contains information of DL based Cyber Security in fog computing. Section XVIII

contains the DL and Cyber Security in pervasive computing. Section XIX discusses the importance of unsupervised learning in Cyber Security. The major challenges involved in Cyber Security applications off-line and real-time deployment is discussed in Section XX. Section XXI contains the role of explainable AI in Cyber Security. The importance of casual theory with DL in Cyber Security is discussed in Section XXII. The detailed statistics of DL applications in Cyber Security is reported in Table XXIII. The suggested Cyber Security system is an organization is discussed in detail in Section XXIV. Finally the conclusion is placed in Section XXV.

The intended audience of this survey includes any Cyber Security researchers, i.e. who is interested in applying DL applications for Cyber Security particularly to understand why the CMLAs achieves less performance compared to DL techniques for Cyber Security applications. The sections II, and III are intentionally designed for novice Cyber Security researchers which provides theoretical and mathematical background of CMLAs, DL architectures and text representation of NLP. There is also a table included in each of the two sections (Tables II-III) that summarizes the CMLAs, DL architectures and various text representation methods in NLP. The importance of signal and image based data Cyber Security data representation method are discussed in Section IV. This type of representation method have been remained as one of the important research directions and anticipated to be more research works in this direction. An important section that need to understand is Section V which provides the terminologies of big data and big data frameworks. The Section V also contain a table that provides the detailed information of various deep learning libraries. Section VI lists the major challenges and issues in the existing study and contains two tables in which the first table provides the names of all the benchmark datasets and the second table lists out all the shared tasks. The Sections VI, and XX can provide important information for novices and advanced Cyber Security researchers. We also discussed the most commonly employed statistical measures in Section VII. Probably the most interesting part of the DL applications in Cyber Security survey can be found in Section XI. Section XI contains tables for each Cyber Security problems. The table includes the type of dataset and deep learning architecture, and information related to results comparison of classical machine learning algorithms and deep learning architectures. In XI section, there are several subsections in which intrusion detection subsection is organized based on the dataset used. The KDD-CUP, NSL-KDD and Mixed segments are further divided based on the presence of ML comparative study. The mixed segment contains various deep learning based intrusion detection solutions where the models are trained on multiple datasets. DL based DGA, URL, Email and Security log data analysis subsection in XI is organized as Email, DGA, URL and CAPTCHA segments. The DGA and URL segments are further divided based on DL architecture (Uni-directional RS, CNN, DNN, AE DBN, Hybrid CNN-RS, Bi-directional RS, Mixed). The mixed division contains DL studies where several multiple models are proposed. Deep Learning in Network Traffic Analysis subsection in XI is organized based on the presence of ML comparative study. Deep Learning in Windows Malware Analysis subsection in XI is organized based on the DL architectures. Deep Learning in Android Malware Detection subsection in XI is organized based on the analysis (static, dynamic, Both static and dynamic, Image processing). The static segment is further divided based on the presence of ML comparative study.

Other Sections IX, XIX, VIII, and X are very important and anticipated to be a significant research direction in future for DL Cyber Security research. The section IX is organized into four subsections based on the adversarial applications in DGA, Malware, IDS and attack and defence techniques. The section X is organized into two subsection based on RL based IDS and other RL applications in Cyber Security. The importance of Cyber Security in the emerging areas are discussed in Sections XIII, XIV, XV, XVI , XVII, XXII, and XVIII. Readers are advised to read Section XXIII that contains the detailed statistics of DL applications in Cyber Security. Various plots are used to showcase the statistics of different DL architectures, Cyber Security applications, text representation, published papers in chronological order from 2000 to Mar, 2019. Cyber Security practitioners are advised to read Section XXIV that contains a generalized framework for Cyber Security and deep learning practical implications and open problems in the Cyber Security field.

II. Background on Classical Machine learning Algorithms and Deep learning Architectures

Artificial intelligence (AI) means applying intelligence to vast amount of data with the aim to derive meaningful results. The term AI is a broad term that was coined by John McCarthy in 1955 and he defined AI as the ”science and engineering of making intelligent machines”. ML is a subset of AI which was introduced in 1950s. As the technology evolved, the amount of data increased in turn making the concept popular in 1990s. Mathematics and statistics are core important concepts in ML. These algorithms primarily discover patterns, correlations and anomalies in the data which varies widely in complexity. The outputs of ML algorithms are represented in terms of probabilities and confidence intervals. Manual analysis of these huge amount of data becomes a difficult task so ML algorithms are employed to automate the learning process. ML has pervaded Cyber Security in the last decade [11]. The algorithms in ML can be grouped into five different types, they are supervised, semi-supervised, unsupervised, reinforcement and active learning. Supervised learning algorithms are task-driven which relies on the labels of the sample input. For example in malware detection, we need to label whether the file is malware or legitimate. This kind of ML relies on preprocessing and feature engineering. Most commonly used CMLAs are Naive Bayes (NB), Logistic regression (LR), Decision Tree (DT), Ada Boost (AB), Random Forest (RF) and Support vector machine (SVM). The diagrammatic representation as well as mathematical details of these various and most commonly used CMLAs are reported in Table I. Unsupervised learning is a data driven approach. These models require only the data samples and implicitly learn the label based on the distribution of the data. Even

though the performance of unsupervised models is less when compared to supervised models, they are preferred in real-time Cyber Security applications as labeling data manually is a tedious task. Semi-supervised learning, as the name implies, combines both the supervised and unsupervised learning to get benefits from both the approaches. RL is an environment driven approach which works based on rewards. It is similar to learning system of a kid where the learning system is improved by a trial and error approach. Most of the DL based real-time systems in current days are based on RL. This is a suitable method for malware and botnet detection in the domain of Cyber Security. Active learning is a sub method of RL that contacts the user whenever a new data sample is seen. CMLAs composed of 3 main steps. They are given below.

1) Raw data collection
2) Feature extraction
3) Classification

Feature extraction is an important step in feature engineering which requires knowledge about the subject. The performance of the classifier implicitly relies on the feature extraction. NN is a machine learning technique which was introduced in 1950s. It is capable of automatic feature extraction and classification with out human intervention. The performance of the classical NN is considerable to a certain extent. However, feature engineering phase can be completely avoided by using advanced NN typically named as DL. The training and testing process involved in CMLAs and DL architectures are represented in Figures 1 and 2 respectively. This made the DL to achieve the best performance in long-standing AI applications related to various domains.

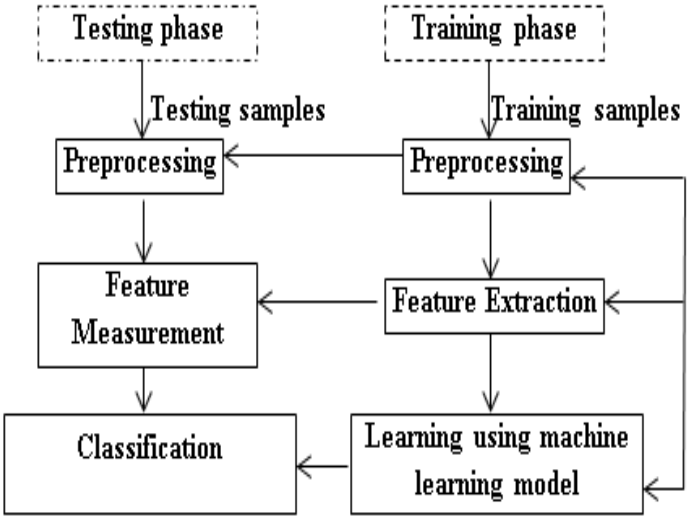


Fig. 1. Classification process involved in machine learning.

In recent days, DL has become a focal point for both the security researchers and people from security industries. Classifications of DL architectures are shown in Figure 3. ML, NNs and DL are all cognate words that influence any conversation about AI. There is an often confusion among all these fields. DL is a sub-field of ML that evolved from NNs. This imitates the way human brain works in the sense

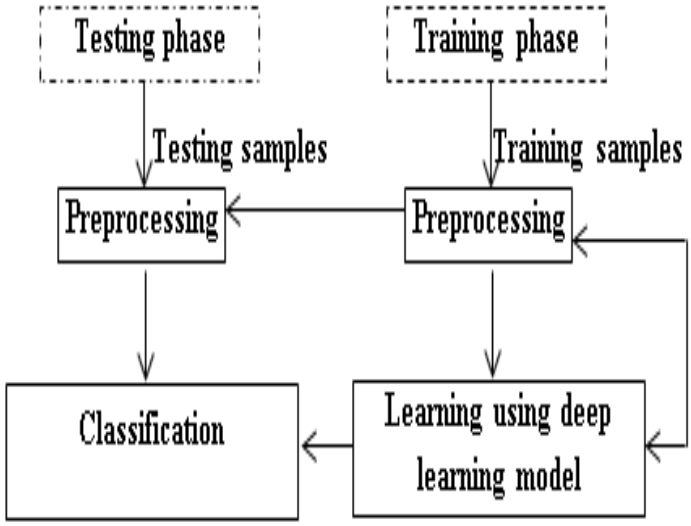


Fig. 2. Classification process involved in deep learning.

of processing data and creating patterns for use in decision making. NNs are the main important component in DL and generally DL means ”many NNs”. When the NNs are deep, the most common problems that arises are vanishing & exploding gradient issue and most importantly unavailability of high performance computing systems. In recent days, with the advancement in computing systems, introduction of new types of DL architectures and improvements in optimizers, activation functions, loss function, the vanishing and exploding gradient issues has been avoided. DL is now being employed in various problems existing in Cyber Security and it performs well in all use cases of Cyber Security in comparison to the classical ML. DL architecture can be classified into generative and discriminative and it is shown in Figure 3. Generative category composed of deep Boltzmann machine (DBM), Deep Autoencoder (DAE), deep belief network (DBN), recurrent structures and discriminative category composed of recurrent structures and convolutional neural network (CNN). Recurrent structures and CNN most commonly used DL architectures. Both DBN and DBM are based on the restricted Boltzmann machines (RBM). The GANs belong to both the generative and discriminative DL architecture category.

Artificial neural network (ANN) is a set of units (artificial neuron) connected together with edges, shown in Figure 5. Feed forward network (FFN) is a type of ANN in which edges connects a set of units in such a way that the connection is in single direction as well as there are no formation of cycle. Multi-layer perceptron (MLP) is a part of FFN which has three or more layers with a various number of units shown in Figure 4. The three layers in FFN are called as input, hidden and output layer.

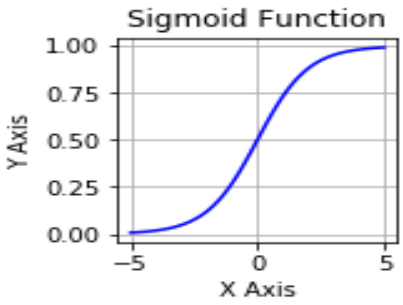
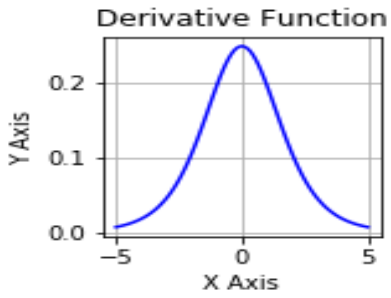
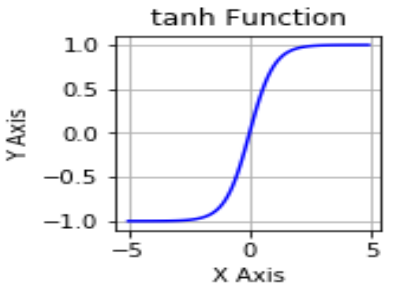
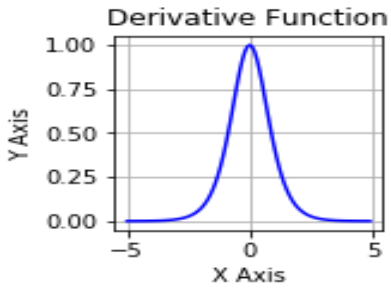
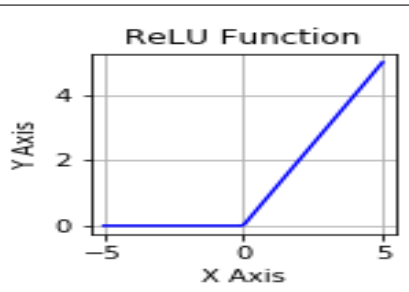
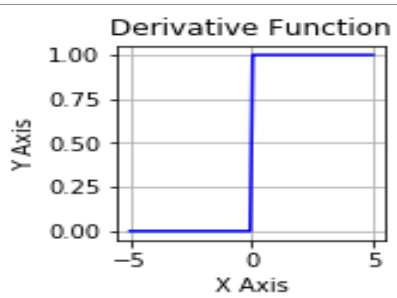
1) Input layer: It takes input which will be processed by the FFN.
2) Hidden layer: Learns features.
3) Output layer: It displays the result that is processed by the FFN.

The network uses BP approach to minimize the loss during

TABLE I
MOST COMMONLY USED CLASSICAL MACHINE LEARNING ALGORITHMS (CMLAS).

Algorithm	**Description**	**Mathematical Background**
Logistic regression (LR)	Logistic regression (LR) is primarily used when the dependent or output variable is nominal. LR uses the log-odds of the probability of an event which is a linear combination of independent or predictor variables. LR uses $Sigmoid$ activation function which results in either '0' or '1'.	$$\sigma(z) = \frac{1}{1+e^{-z}}$$ where z defines the input value.
Nave Bayes (NB)	Naive Bayes uses Bayes theorem in which features are independent in nature. It is simple to build because the algorithm doesn't involve any parameter estimation. Thus it can work on very large datasets.	$$P(c\|x) = \frac{P(x\|c)P(c)}{p(x)}$$ $$P(c\|x) = P(x_1\|c) \times P(x_2\|c) \times \cdots \times P(x_n\|c) \times P(c)$$ where c denotes class and x denotes input values, $P(c\|x)$ posterior probability of c given the data x, $P(x\|c)$ proability of input x given that the hypothesis was true, $P(c)$ prior probability of c, $P(x)$ is prior probability of x
Decision Tree (DT)	A Decision Tree (DT) is a tree like diagram. A tree is composed of nodes and edges. A node has condition on feature and edge contains the output. Each leaf node of the tree denotes a class and a path from parent to leaf node represent classification rules.	Tree splitting uses Gini Index, Chi-Square and Information Gain methods.
K-Nearest Neighbor (KNN)	K-nearest neighbor (KNN) is a non-parametric approach which stores all the possible cases and using similarity measure i.e. distance function classifies other new cases. It is computationally expensive and requires larger memory because it stores the entire training data.	Most commonly used distance function is Euclidean distance.
Ada Boost (AB)	Adaptive Boosting (AB) which aims to convert a set of weak classifiers into a strong one.	$$F(x) = sign(\sum_{m=1}^{M} \theta_m f_m(x))$$ where f_m stands for the Mth weak classifier and θ_m is the corresponding weight.
Random Forest (RF)	A DT with many leaf nodes can see overfit, to alleviate it random forest can be used. This makes a prediction by averaging the prediction of each component tree. As name suggests, random forest has many trees and those are random in nature. Random forest uses the bagging method which enhances the performance.	$$\overset{\Lambda}{f} = \frac{1}{B}\sum_{b=1}^{B} f_b(x^{'})$$ where B denotes bagging, f_b is a classification, $x^{'}$ denotes unseen training samples.
Support vector machine (SVM)	Support vector machine (SVM) can be linear and non-linear classifier. It finds a hyper plane and separates the training set with maximal margin. The points near to the separating hype plane are called support vectors and they determine the position of hyper plane. If the training dataset is not linearly separable then it can be mapped to high-dimensional space using kernels where it is assumed to be linearly separable.	SVM problem formulation is $$\min \|w\|^2 + C\sum_{i=1}^{n} \xi_n \ (\mathrm{L1-SVM})$$ $$\min \|w\|^2 + \frac{C}{2}\sum_{i=1}^{n} \xi_n \ (\mathrm{L2-SVM})$$ Subject to $$y_i(w \cdot \phi(x_i) + b \geq 1 - \zeta_i \, \zeta_1 \geq 0)$$ (x_i, y_i) input data samples, $\phi(x)$ is a transformation on the input data, ζ_i is a salck variable, C is a penality parameters Most commonly used kernel functions are linear, radial basis function (RBF), polynomial and hyperbolic tangent.

TABLE II
MOST COMMONLY USED ACTIVATION FUNCTIONS.

Activation Function	Representation	Derivative	Values In the range	Mathematical Background
$Sigmoid$			$(0, 1)$	$\sigma(z) = \frac{1}{1+e^{-z}}$
$tanh$			$(-1, 1)$	$\tanh(z) = \frac{e^{2z}-1}{e^{2z}+1}$
$ReLU$			$(0, \infty)$	$\mathrm{Re}LU(z) = \max(0.0, z)$

training. The classical NNs are shallow and in recent days the advancement of various concepts involved in NN enables it to train deeper network [11]. So, MLP with $ReLU$ activation function is typically called as DNNs [11]. The main difference between $Sigmoid$, $ReLU$ and $tanh$ activation function is reported in Table II.

DBN or deep networks is a generative engineering, appeared as a relative network of the classical ANN [12]. It contains an input layer, at least one hidden layers (with one layer of DBN is same as FFN) and a output layer. Both the input and hidden layer should have at least one neuron, scientifically termed as units. A output layer has one unit for every class which should be classified by the network. In addition, a network with in excess of one hidden layer may expend more opportunity for its assembly. To maintain a strategic distance from the arbitrary statement, [13] presented an unsupervised learning component, for example, restricted Boltzmann machine (RBM) to take in the minimized element vectors eagerly by passing an input vector through at least one RBM hidden layers amid the preparation stage. DBN's training phase has two steps namely pre-training and reconstruction. Given the training samples without class labels, the pre-training stage propagates the input stochastically across RBM layers. Associative memory is actually the top level of RBM layer. Each layer of RBM has learned some features which represents the data in previous layer. Conditional distribution is followed by each hidden layer unit to generate binary form feature vectors. These feature vectors of binary forms are propagated in reverse direction to reconstruct the training samples. This procedure is followed iteratively for all the training samples.

A set of NN architectures made to learn alias representations of input data via linear or nonlinear operations are called as Autoencoders (AE) which have identical input and output layer dimensions [14]. AE is the NN typically made for the purpose of dimensionality reduction [14]. Recent days, researchers utilize more than one hidden layer to learn discriminative and representative features of raw data. This type of network is called as DAE. Unlike the general NN architecture which are trained to learn predefined output variables, these are trained to learn from the input. As a result, the NN learns by itself to reconstruct the input data. The architecture of an AE is similar to the MLP; it has one input layer, one or more hidden layers and finally the output layer. If the AE is having multiple hidden layers, then the features extracted from one layer are further processed to different features and these should be capable of reconstructing the data. During data reconstruction process,

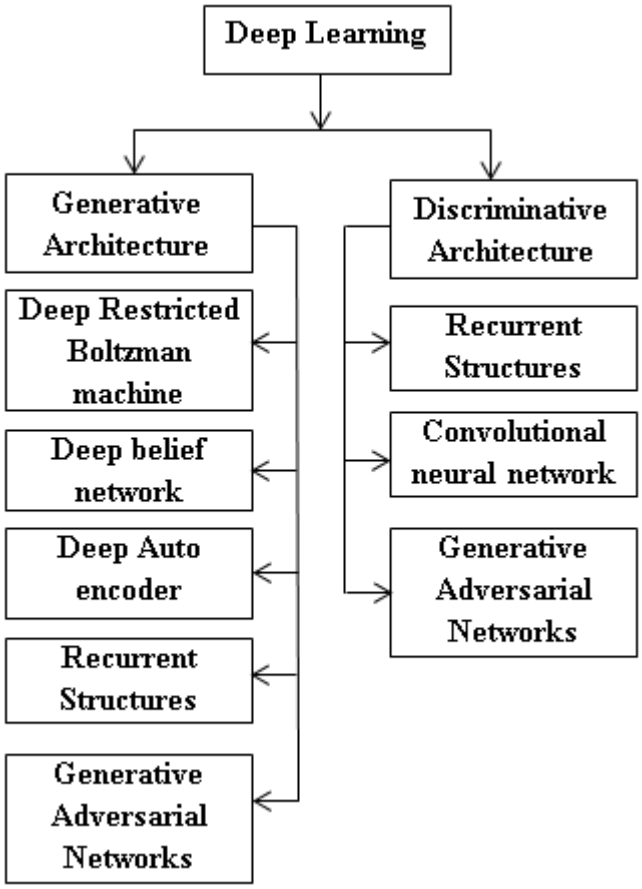


Fig. 3. Classification of Deep learning architecture.

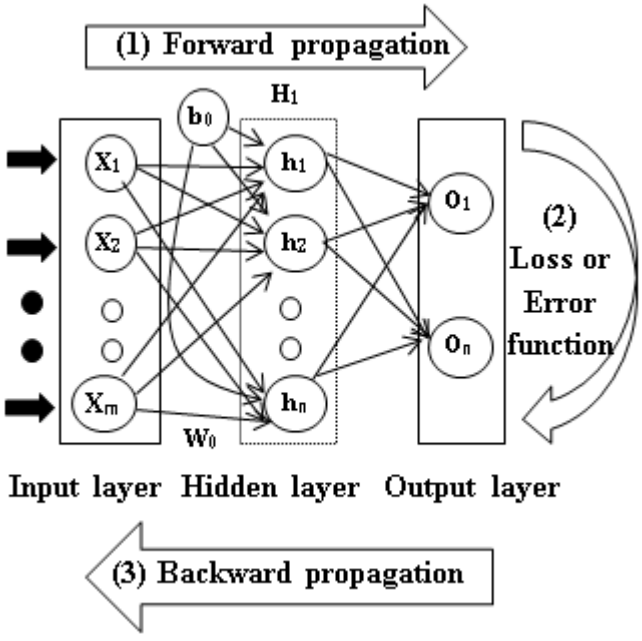


Fig. 4. Architecture of classical neural network.

AE aims to minimize the error. Therefore, the outputs of the intermediate layers are nothing but an encoded version of the input capable of reconstructing the input data under specific

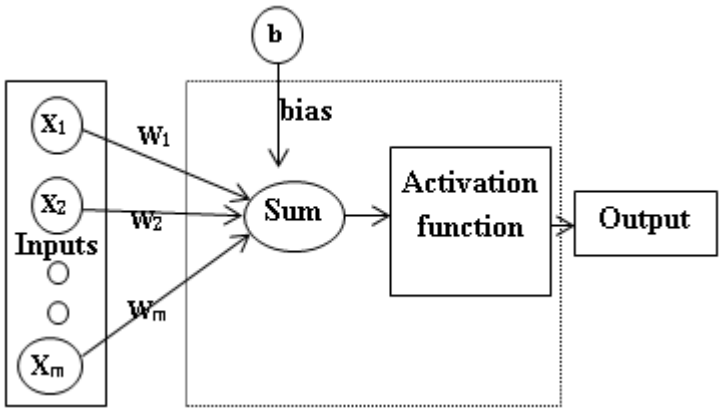


Fig. 5. A neuron in neural network, with several inputs and trainable weights and bias.

conditions. For example consider the architecture shown in Fig 4. Here, the input to the network is a N length vector X, which is reconstructed as $X1$. The idea is to select weights and biases to produce Y such that the error between X and $X1$ is as small as possible. Y is constructed from X via the transformation

$$Y_{1\times m} = f(X_{1\times n}W_{n\times m} + b_{1\times m})$$

where W and b represents the weights and biases corresponding to the first layer and f is some activation function. Similarly, in the next layer X is reconstructed from Y as $X1$ via the transformation

$$X1_{1\times n} = g(Y_{1\times m}W1_{m\times n} + b1_{1\times n})$$

where $W1$ and $b1$ are the corresponding weights and biases whereas g is the activation function. When Y and X are of the same dimensions, the obvious choice of W and $W1$ becomes the identity matrices with unity activation function. But when they are of different dimensions, the network is forced to learn newer representations of the input X via some transformations. It is not necessary that, AE implementations should be restricted to one layer. There can be multiple layers of different dimensions and in such cases, the features extracted from one layer are further processed into newer features which are still capable of reconstructing the data. Keeping the dimension of Y smaller than X results in learning some compressed encoding of the input and has found many applications in a variety of fields. This process is considered to be analogous to some feature extraction mechanism which can be used for applications such as classification. In usual feature based classification methods, some specific features are selected in prior and then calculated for all the data points so that they can be fed as input to the classification algorithm. In this approach, they go for unsupervised learning of features. In the case of multi-layered AE, different features are available from different layers which can be used as features for classification. Furthermore, outputs from deeper layers are features learned from features which can contain special information and cannot be directly extracted from the signal in usual transformations.

In usual classification methods, some particular features are selected utilizing the data points and then it is fed to classification algorithm as input whereas in AE, since it is following unsupervised approach, different features are extracted from different layers and is used as features to be passed into other DL layer such as CNN, RNN and hybrid networks such as CNN-RNN and CMLAs. Primarily AEs have 3 types of well-known variants; they are Sparse AE, Denoising AE and Contractive AE. To extract sparse features, spare AE can be utilized. In sparse AE, there are more hidden nodes than there are in the input and output layers, however, at a given time, only a portion of the hidden units are activated. This is accounted for by penalizing the activating additional nodes. DAE recovers the correct input from a corrupted version and thus it can increase the robustness of the model. Contractive AE increases the robustness of the model by adding an analytic contractive penalty to the reconstruction error function. Generally, the DAE architecture is more robust for noise and contractive AE is able to capture the local directions of variation dictated by the data.

CNN is a state-of-the-art method for many of the computer vision applications. CNN composed of convolutional, pooling and fully connected layers. A convolutional layer uses kernels or filters to move along the 1D or 2D or 3D or 4D data to extract optimal features, together called as feature maps. These feature maps are passed into the pooling layer. Both convolutional and pooling layers are translational invariants because they consider the neighboring data into account. Initially the feature maps are divided into partitions and various pooling functions are used to reduce the dimensionality of the feature maps. To simply put, the task performed by a pooling layer is a non-linear down sampling operation. The pooling operations are max, min, average, stochastic, spatial pyramid and def pooling. The max, min and average pooling take the maximum, minimum and average value from the partition. The stochastic pooling is similar to max pooling and at the same time it prevents the issue of overfitting. It checks for the activation within each pooling region according to a multinomial distribution and thereby replaces the conventional deterministic pooling operations with a stochastic procedure. Generally, CNN network can handle only the fixed length input image representations. To handle variable length input representations, spatial pyramid pooling can be used. Spatial pyramid pooling helps to handle input images of variable scales, sizes and aspect ratios. Def pooling handle deformation efficiently when compared to the max and average pooling. These different pooling layers can be combined to boost the performance of the CNN architecture. Most commonly used pooling operations are max, min and average pooling. Fully connected layer have connection to all other neurons of the previous layer and are used for classification purpose. Based on CNN, various benchmark architectures are proposed by well-known researchers and they are evaluated on ImageNet Large Scale Visual Recognition Challenge (ILSVRC). The various architectures based on CNN are LeNet, AlexNet, ZFNet, GoogleNet/Inception, VGGNet, SPPNet, ResNet, DenseNet, squeezenet, MobileNet and NASNet. All these architectures contain large number of parameters and typically these architectures are applied on large datasets. Primarily, obtaining large number of datasets for all the classes and as well as for all the tasks in real-time is very difficult. Data augmentation is the methodology used by researchers to increase the data samples without introducing extra labeling costs. Mostly, the operations used in data augmentation are divided into two forms. Translations and horizontal reflections are belongs to first form of data augmentation. Changes in the intensities of the RGB channels in training images belong to second form of data augmentation. Since these various CNN architectures are well-known and can be employed for parameter initialization instead of random parameter value in newer tasks. This is called as pretraining. This type of learning accelerates the learning process as well as improves the generalization ability.

Recurrent structures are mainly used in sequence and temporal data modeling tasks [27]. Recurrent neural network (RNN) is an advanced model of classical NN which has a self-recurrent connection in the hidden layer neurons that facilitates the network to remember the previous time step information. It suffers from vanishing and exploding gradient issue when dealing with long time-steps [26] during backpropogation through time (BPTT). To understand, let's define the recurrent relation for an RNN network with two time steps without non-linearity.

$$h_2 = (W^2)^T h_1$$

W revealed an Eigen decomposition of symmetric matrix, where the Eigen values λ and Eigen vectors V should be orthogonal to each other.

$$W = Q \wedge Q^T$$

This is substituted to as $h_2 = Q^T \wedge^T Q h_1$. Whenever, $|\lambda| < 1$ causes to vanishing gradient problem and if $|\lambda| > 1$ causes to exploding gradient problem. Fixing the λ to an appropriate norm solves the exploding problem but handling vanishing problem will be a challenge. Gradient clipping is one of the prominent strategies to avoid the exploding gradient issues [26]. Fortunately, $ReLU$ instead of $tanh$ or $Sigmoid$ and correct initialization of weight matrix and can alleviate the vanishing issue [15], [16], [17]. To alleviate, research on RNN progressed on the 3 significant directions. One is towards improving optimization methods in algorithms such as Hessian-free optimization methods belong to this category [18]. Second one is towards introducing complex components in recurrent hidden layer of network structure such as LSTM [19], [20], [21], a variant of LSTM network with reduced parameters set, GRU [22] and third one is towards the appropriate weight initializations with an identity matrix typically called as identity-recurrent neural network (IRNN) [23], [24]. Clockwork RNN is a variant of RNN which can handle vanishing and exploding gradient issue and it is computationally inexpensive compared to RNN [25]. To capture both spatial and temporal information, hybrid network i.e. CNN-RNN is used. The mathematical details and intuitive overview of all DL architectures are reported in Table III. In Table III, w denotes weights, b denotes bias, x denotes input, h defines hidden layer, A denotes activation function, i_t denotes

input gate, o_t denotes output gate, f_t denotes forget gate, c_t denotes memory cell, and u_t denotes update gate.

Batch normalization and dropout are two main important concept used in DL where batch normalization can prevent from vanishing and exploding gradient problem, increases the training speed, and dropout acts like a regularization parameter. A classical NN without dropout and with dropout is shown in Figure 6 and Figure 7 respectively. Dropout facilitates the network to reduce overfitting during training by randomly removing the neurons as well as its associated connections. Primarily, batch normalization is placed between the hidden layers in the DL architecture and it indicates the intermediate feature representation. It can be represented mathematically as:

Input: Let's consider hidden layer representation h over a mini-batch: $M = \{h_1, \cdots h_m\}$ and the objective is to learn hyper parameters, $\gamma and \beta$.

Output: $y_i = BN_{\gamma,\beta}(x_i)$

$$\mu_M = \frac{1}{m}\sum_{i=1}^{m} h_i \quad \text{mini} - \text{batch mean}$$

$$\sigma_M^2 = \frac{1}{m}\sum_{i=1}^{m} (x_i - \mu_M)^2 \quad \text{mini} - \text{batch variance}$$

$$h1_i = \frac{h_i - \mu_M}{\sqrt{\sigma_M^2 + \in}} \quad \text{normalize}$$

$$y_i = \gamma h_1 + \beta \equiv BN_{\gamma,\beta}(h_i) \quad \text{scale and shift}$$

DL is a right method for Cyber Security. This is mainly due to the reason that the amount of data in the field of Cyber Security is very large and typically called as BD. Additionally the availability of multi-core CPUs, GPUs and the concept of NNs evolved as to how best train NN which contains many hidden layers.

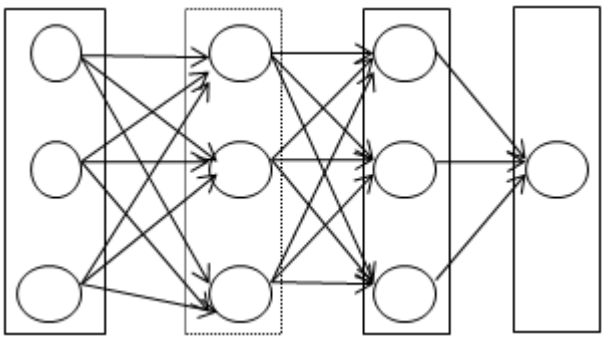


Fig. 6. Classical Neural Network.

III. Natural language processing for Cyber Security

Natural Language Processing (NLP) is the method of analyzing and extracting useful information from natural languages to make human-computer interaction simpler. The key to NLP success in Cyber Security is the availability of languages as data. The text data which Cyber Security domain come across are from various sources like emails, logs from various systems, the texts users share in online social networks. Exponential growth of text data has now become a common phrase these days that can have a direct impact on enhancing the detection rate of malicious activities. Leveraging NLP techniques have direct impact on providing situational awareness from various Cyber Security event logs. In recent years, the domain of Cyber Security has seen a tremendous growth in the amount of information available in the form of text as long as they continue to log each and every user activities from various systems.

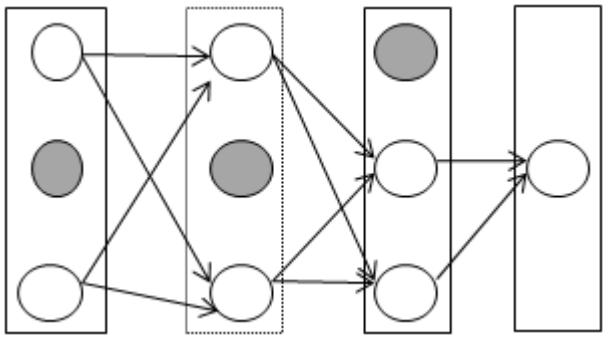
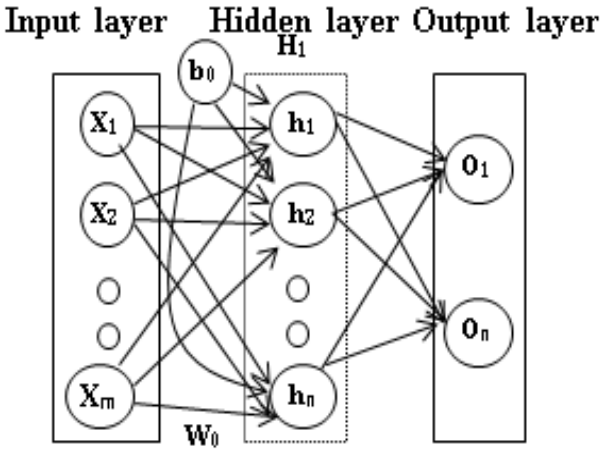
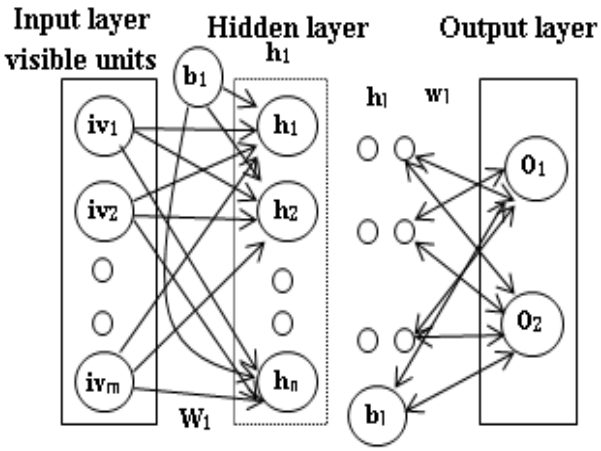
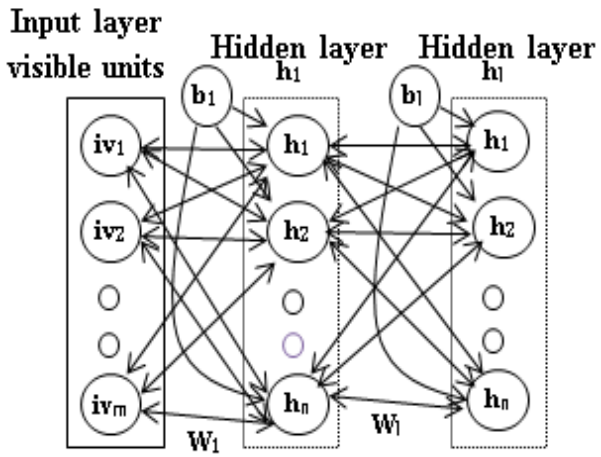
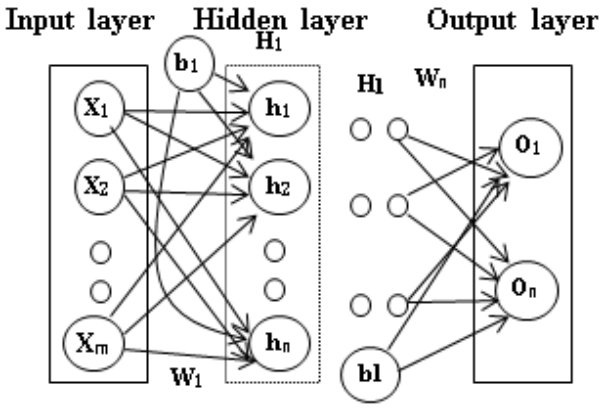
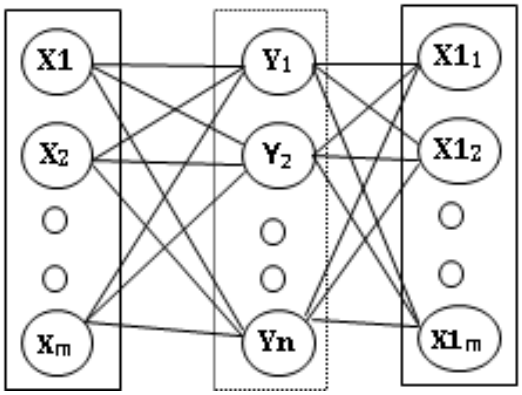
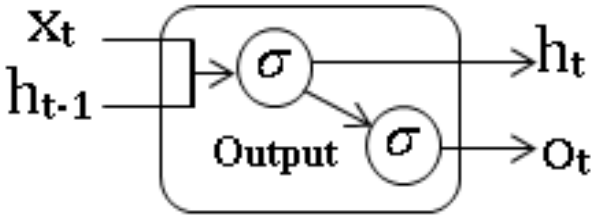
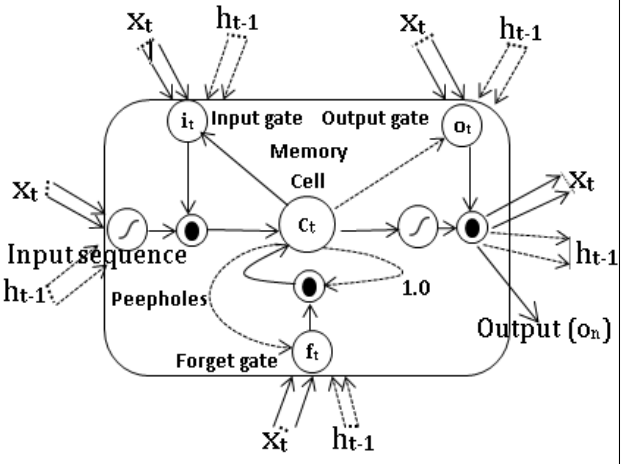


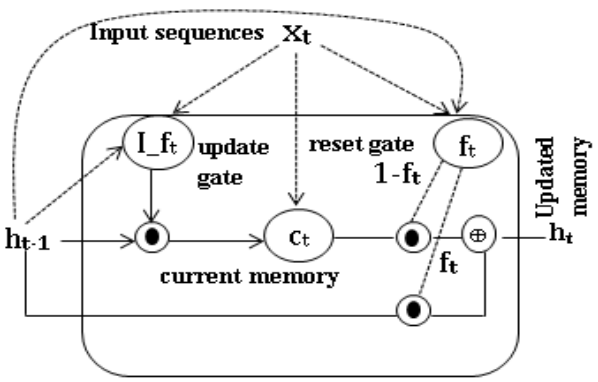
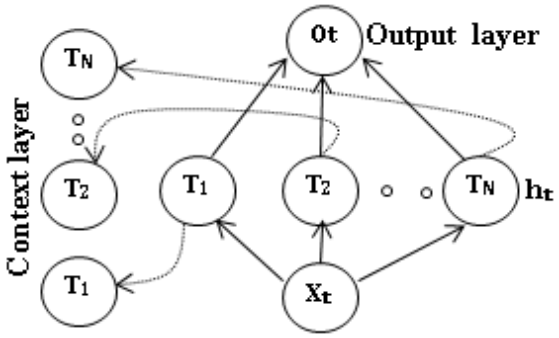
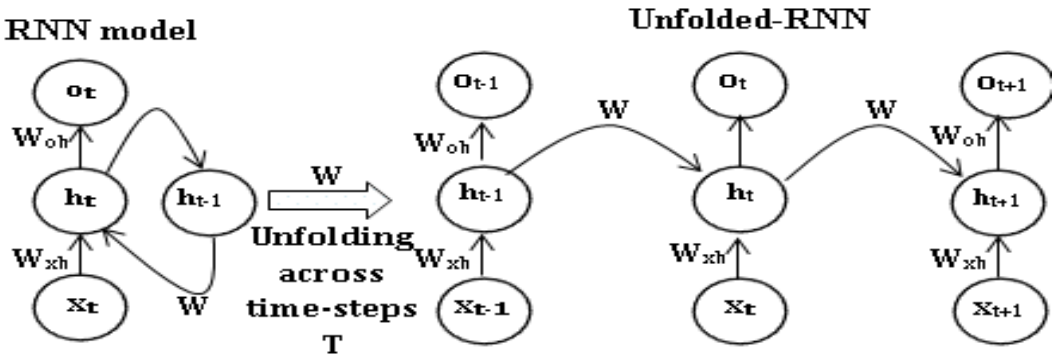
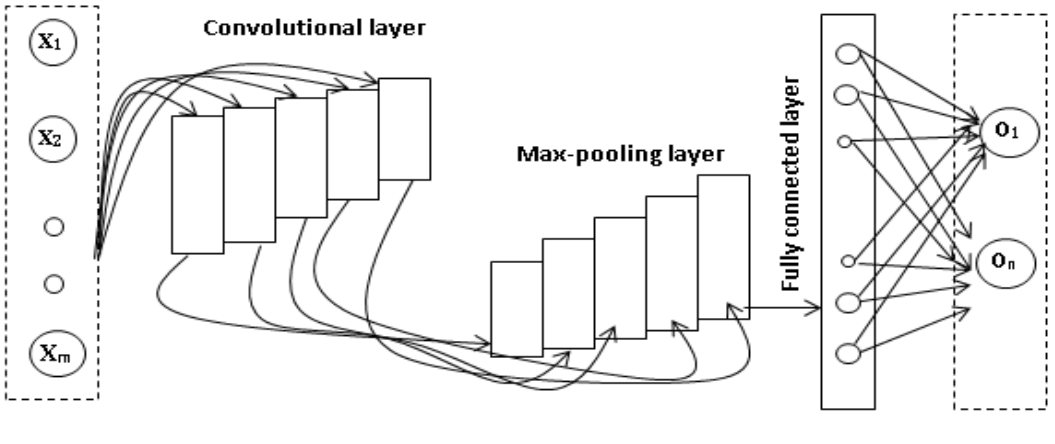
Fig. 7. Classical Neural Network with Dropout.

Text representation is a primary task to deal with the text [28]. There are various types of text representation existing to represent text in numeric form. They are vector space models, distributional representation and distributed representation. Representation of texts in word/character level is called as word/character level text encoding. Word/character level text encoding consists preprocessing followed by tokenization as the initial step. In preprocessing, the unnecessary information is discarded and words/characters are transformed into lowercase. Additionally, the unknown words/characters are assigned to the predefined key, 0. In tokenization, the texts are chopped into words/characters using word/character level tokenization respectively. Non-sequential and sequential inputs are the two main types of text representation. Bag of Words (BoW), Term document matrices (TDM) and Term frequency-Inverse document frequency matrices (TFIDF) belong to non-sequential representation. N-gram, Keras embedding, Word2vec, Neural-Bag-of-words and FastText belong to sequential representation. Sequential representations have the capability to extract the similarities in word meaning. In Cyber Security domain, capturing the sequential information is more important compared to the similarities in word meaning. This is due the fact that most of the data contain time and spatial information. The detailed description of various text representation methods is reported in Table IV.

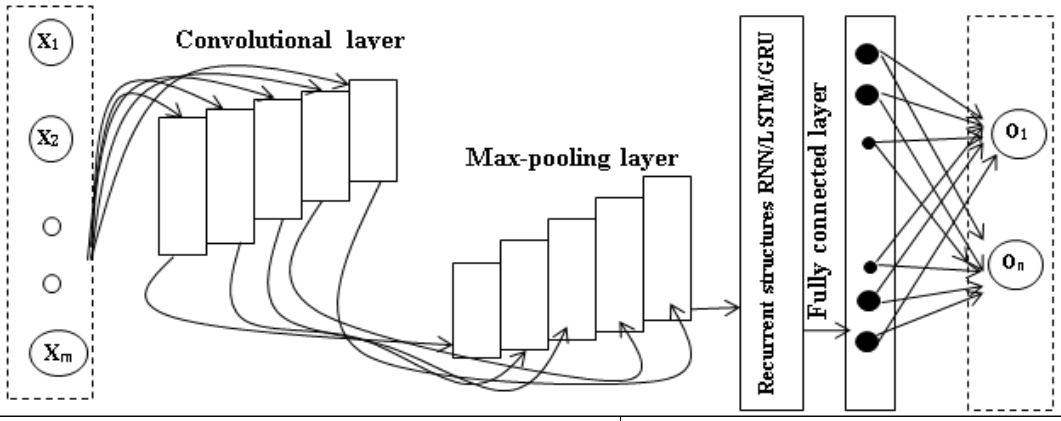
TABLE III: DEEP LEARNING ARCHITECTURES.

Model	Architecture	Mathematical Background
Shallow Artificial neural network (ANN)		$h_i(x) = A(w_i{}^T x + b_i)$ $h(x) = h_1(x)$
Deep Belief Network (DBN)		$R(iv, h) = -h^T iv - bi^T iv - bh^T h$ $P(iv_i = 1\|h) = \sigma(bi_i + \sum_k w_{ki} h_k)$ $P(h_k = 1\|iv) = \sigma(bh_k + \sum_i w_{ki} h_i)$ $\Delta w_{ki} = \eta((iv_i h_k)_{IS} - (iv_i h_k)_{ReS})$ $\Delta bi_i = \eta((iv_i)_{IS} - (iv_i)_{ReS})$ $\Delta bh_i = \eta((iv_i)_{IS} - (iv_i)_{ReS})$ η denotes the learning rate, IS and Res denotes input and reconstruct samples respectively.
Deep Restricted Boltzmann Machine (DBM)		

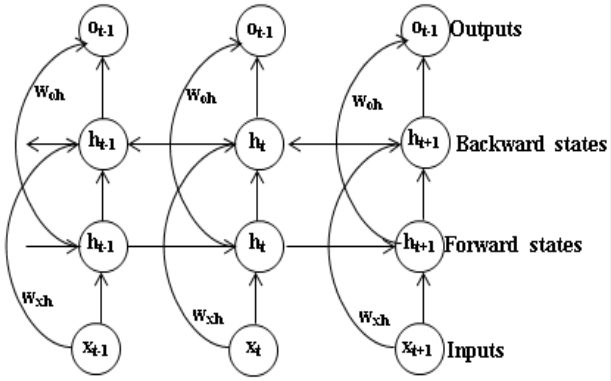
Deep Neural Network (DNN)		$h_i(x) = A({w_i}^T x + b_i)$ $h(x) = h_l(h_{l-1}(h_{l-2}(\cdots(h_1(x)))))$
Deep Autoencoder (DAE)		$Y_{1\times m} = A(X_{1\times n}W_{n\times m} + b_{1\times m})$ $X1_{1\times n} = A(Y_{1\times m}W1_{m\times n} + b1_{1\times n})$
Structure of neuron in Recurrent structures (RS)		
Recurrent Neural Network (RNN)		$h_t = A(w_{xh}x_t + w_{hh}h_{t-1} + b_h)$ $o_t = A(w_{ho}h_t + b_o)$
Long short-term memory (LSTM)		$i_t = A(w_{xi}x_t + w_{hi}h_{t-1} + w_{ci}c_{t-1} + b_i)$ $f_t = A(w_{xf}x_t + w_{hf}h_{t-1} + w_{cf}c_{t-1} + b_f)$ $c_t = f_t \odot c_{t-1} + i_t \odot \tanh(w_{xc}x_t + w_{hc}h_{t-1} + b_c)$ $o_t = A(w_{xo}x_t + w_{ho}h_{t-1} + w_{co}c_t + b_o)$ $h_t = o_t \odot \tanh(c_t)$

Gated recurrent unit (GRU)		$u_t = A(w_{xu}x_t + w_{hu}h_{t-1} + b_u)$ $f_t = A(w_{xf}x_t + w_{hf}h_{t-1} + b_f)$ $c_t = A(w_{xc}x_t + w_{hc}(f \odot h_{t-1}) + b_c)$ $h_t = A \odot h_{t-1} + (1-f) \odot c$
Clockwork recurrent neural network (CWRNN)		$h_t = A(w_{xh}x_t + w_{hh}h_{t-1} + b_h)$ $o_t = \sigma(w_{oh}.h_{t-1} + b_o)$
Training method employed for training DL architectures		
Backpropogation through time (BPTT)		
Convolutional neural network (CNN)		

CNN with recurrent structures (CNN-RS)



Bidirectional recurrent structures (BRS)



Forward pass:

$$\overrightarrow{h_t} = A(w_{x\vec{h}}x_t + w_{\overrightarrow{h}\overrightarrow{h}}\overrightarrow{h}_{t-1} + b_{\overrightarrow{h}})$$

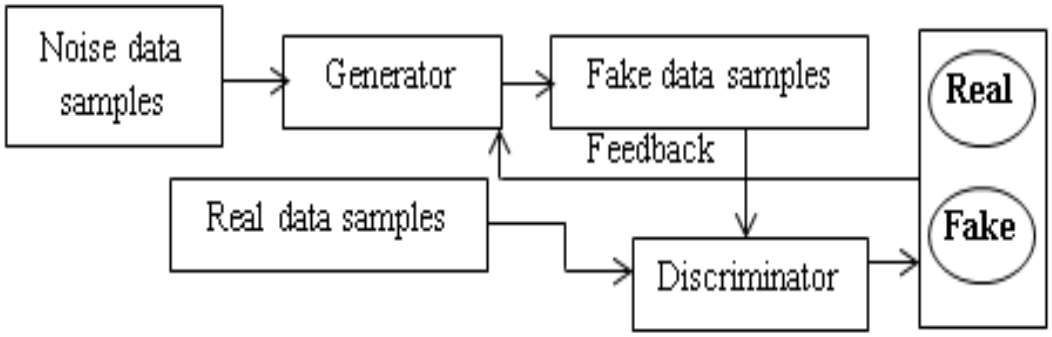
Backward pass:

$$\overleftarrow{h_t} = A(w_{x\overleftarrow{h}}x_t + w_{\overleftarrow{h}\overleftarrow{h}}\overleftarrow{h_{t+1}} + b_{\overleftarrow{h}})$$

Output layer:

$$o_t = (w_{\vec{h}o}\overrightarrow{h_t} + w_{\overleftarrow{h}o}\overleftarrow{h_t} + b_o)$$

Generative Adversarial Networks (GAN)



Reinforcement learning (RL)

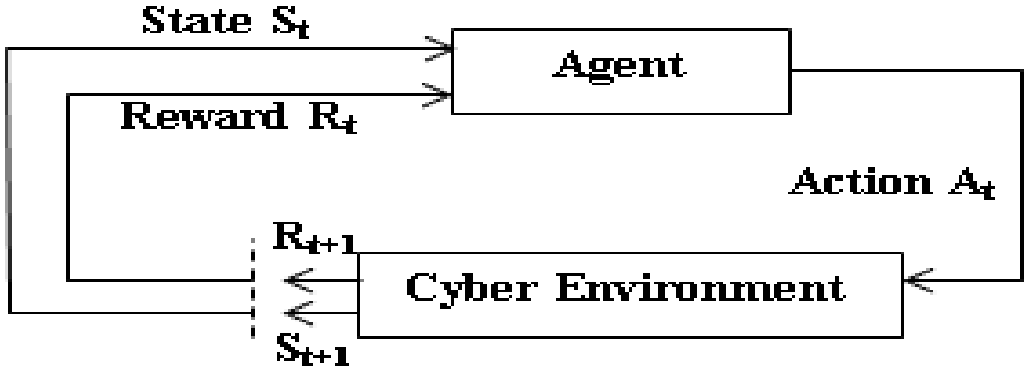


TABLE IV: TEXT REPRESENTATION METHODS IN NATURAL LANGUAGE PROCESSING (NLP).

Text representation	**Description**	**Mathematical equation**
Vector space models - Represents text as vector but fails to preserve the word order.		
Bag of Words (BoW)	Bag of words model represents each document as a bag of words.	$BoW(w,d) = number\,of\,times\,word\,w\,occurs\,in\,document\,d$
TDM and TF-IDF are weighting factor for BoW		
sTerm Document matrices (TDM)	In TDM, each cell actually holds the count of the occurrence of each word in the given document. In this matrix, the columns represent the vocabulary whereas the rows represent the documents.	$TF(w,d) = \frac{BoW(w,d)}{number\,of\,documents\,in\,which\,word\,w\,occurs}$
Term frequency-Inverse document frequency matrices (TF-IDF)	TF-IDF matrices are utilized when there is no need to consider the words frequently used in the final text analysis. In this matrix, the columns represent the vocabulary and the rows represent the documents. The speciality of this matrix is its ability to give importance to the unique words in the document. This is achieved by changing the count of the word with probability of word occurrence and dividing it with the total number of occurrences in all documents.	$TF{-}IDF(w,d) = \frac{BoW(w,d) * N}{number\,of\,documents\,in\,which\,word\,w\,occurs}$ where N denotes the total number of documents, $IDF(w,d) = \frac{N}{number\,of\,documents\,in\,which\,word\,w\,occurs}$ where $IDF(w,d)$ denotes the inverse document frequency.
Vector space model of Semantics or Distributional representation - Represent text as vector, preserves the word order and semantics of words to some extent but results in curse of dimensionality.		
TDM with non-negative matrix factorization and SVD	SVD is represented as $U_{m\times m}\sum_{m\times n}{V_{n\times n}}^T = SVD(D_{m\times n})$ where $D_{m\times n}$ can be output of TDM or TF-IDF, U is the column space of D, V is the row space of D, $\sum$ is the singular values.	$U_{m\times m}\sum_{m\times n}{V_{n\times n}}^T = SVD(TDM_{m\times n})$ $W_{m\times k}H^T_{k\times n} = NMF(TDM_{m\times n})$
TF-IDF with non-negative matrix factorization and SVD	NMF is represented as $W_{m\times k}H^T_{k\times n} = NMF(D_{m\times n})$ where W is the conceptual representation or features of D and H is the coefficients of D.	$U_{m\times m}\sum_{m\times n}{V_{n\times n}}^T = SVD(TF-IDF_{m\times n})$ $W_{m\times k}H^T_{k\times n} = NMF(TF-IDF_{m\times n})$

n-gram	Represent words in text as 1-gram, 2-gram or n-gram. This can preserve the word order in short context, it suffers from data sparsity and high dimensionality.	$$ngram_s = s - (NW - 1)$$ where s denotes sentence and NW is the number of words in sentence.
Distributed representation - Represents words as continuous vector and preserve the word order along with the context of word.		
Keras embedding	Keras embedding utilize the dictionary to map the words or characters represented by numeric indexes to their dense vector representations. The weights in Keras embedding are initialized with random weights mostly used Gaussian.	input-shape x weights-of-word-embedding = (nb-words, word-embedding-dim) input-shape = (nb-words, vocab-size), nb-words denotes the number of top words, vocab-size denotes the number of unique words, weights-of-word-embedding = (vocab-size, word-embedding-dimension), word-embedding-dimension denotes the size of word embedding vector.
Word embedding	Word embedding is a distributed representation method which represents the word in low dimensional space.	Continuous bag of words (CBoW), Skip-gram, global vectors (Glove), document to vector (doc2vec) and paragraph to vector (para2vec) are different types of word embedding.
Neural-Bag-of-words	It is a dense neural network which has the ability to map an input word sequence to output labels. Dense neural network receives a vector input, which is actually a composition function, in order to predict the probabilities for the output label.	$$z = \frac{1}{X}\sum_{w \in X} v_w$$ $$y1 = soft\max(w_l z + b)$$ where X denotes input text, w denotes word, and v_w denotes word vectors.
FastText	FastText, a library created by the Facebook research community, is utilized to catch the word representations and text classification. It works on n-grams of character level and n could range from 1 to the length of the word. It is better for morphological rich languages and it uses Skip-Gram model and a subword model. Subword model will see the internal structure of the words and learns an efficient vector representation for rare words. furthermore it can learn the vector representation for words which are not present in the dictionary. This works well when compared to other embedding on small datasets.	$$-\frac{1}{N}\sum_{n=1}^{N} y_n \log(f(BAx_n))$$ where N denotes number of document, x_n is the normalized bag of features of the nth document, y_n is the label, A and B are the weight matrices.

IV. Signal and Image processing approaches for Cyber Security

There is an alarming increase in the amount of malware that are generated daily. The present defense mechanisms for suspicious or malicious activity are based on scanning methods such as static analysis, dynamic analysis and heuristics based techniques, which are often slow to react for new attacks and threats. Static analysis is based on analyzing an executable file without executing it while dynamic analysis executes the malware binary file and studies its behavioral characteristics. The hybrid of static and dynamic analysis approach is used in commercial malware detection systems. To evade the static analysis, an adversary employed packers which uses compression, encryption or a combination of both to create a new packed executable that mimics the previous executable in function but reveals the actual code only upon execution runtime. The packed malware can be detected using dynamic analysis but is slow and time consuming. To alleviate the limitations of static and dynamic analysis, a series of studies based on orthogonal solutions from image and signal based malware analysis done by [220], [221], [222], [223], [225], [226], and [227]. These studies exploits the fact that most malware variants are similar in structure. To extract similarity features, the malwares are represented in the form of a grayscale image or a signal instead of viewing and editing malware binaries using Hex Editors. The range of this signal is [0, 255] (0: black, 255: white). In the case of an image, the width of the image is fixed and the height is allowed to vary depending on the file size. Various feature engineering mechanism available in signal and image processing domain were employed for malware classification and retrieval. Signal and image based malware analysis approach is fast, do not need disassembly, unpacking or execution. Recent days, a novel feature engineering methods such as spectral flatness, mel-frequency cepstrum coefficients (MFCC), chroma features and more are proposed by researchers to accurately extract important features from signal and images. These feature engineering mechanisms can be accomapnied with the deep learning to enhance the performance of malware ananlysis and retrieval. There are two major problems with the signal and image based malware detection. The first one is that the characterization of malware using signal and image based features does not give much information about the actual behavior of the malware. Secondly, since the approach relies on instance-based learning, its main limitation is that it can only detect or classify malware similar to what has already been observed. Thus, zero day attacks of new unseen malware cannot be prevented. However, this is a generic problem with any similarity based malware analysis framework.

V. Big data Analytics and tools to handle big data in the field of Cyber Security

The amount of data generated by Internet connected system is very large, fast and required to be processed in real-time. The real-time analysis of this BD is important for various Cyber Security applications with the aim to protect Internet connected devices from malicious activity. Another important factor is that the amount of data generated by the Internet technologies is unstructured and noisy. Moreover, the amount of data generated by the Internet devices is continuing to grow at an unprecedented rate. Advanced technologies in GPU, cluster computing frameworks enabled to process and handle very large amount of data in an efficient way [29]. The extreme scale of data can be considered in 4 angles of volume, variety, velocity and veracity in BD landscape. These terms are briefly defined as follows

1) **Volume:** The volume defines the amount, size, and scale of the data. The number of data samples vertically and the number of features horizontally indicates the size in terms of ML. It is also related to the type of data and according to [30], the smaller complex data samples can be considered equivalent to a larger quantity of simple data. The major challenges originated by volume while solving ML algorithms and DL architectures are discussed below;
   - Challenge in processing: The time taken is very high and the space needed is very huge for training the NN with large amount of data. The time complexity is related to the cube of number of samples whereas space complexity is proportional to the square of the number of samples. So the time and memory taken for computations will have an exponential increase with increase in the size of dataset. The solution to handle this challenge is to develop architectures capable of parallel processing of data [31] [32].
   - Curse of modularity: Many training/testing algorithms are designed assuming that the data is available in its entirety in memory. As the data volume is very high, it may not be possible for the entire data to be in memory or disk. Because of this, such algorithms cannot run successfully. This is known as the curse of modularity [33]. Distributed computing and parallelization can be resorted to tackle this challenge.
   - Curse of class imbalance: As data volume increases, we can no longer assume that there is uniform distribution of data among the available classes [34]. This variation in the probability of occurrence of classes leads to reduction in performance of ML algorithms. This issue is likely to happen in any sized datasets, but it has the potential to be a real challenge as the data size become enormous.
   - Curse of dimensionality: Dimensionality means the number of dimensions (which means the features/ attributes) present in the dataset [35]. BD involves a very high dimensional data space, handling of which is extremely challenging. As dimension increases, the time and space complexity of the ML algorithms increases considerably. The performance of training algorithm also falls considerably as the dimension of the data increases.
   - Feature engineering: In ML, proper selection of features is crucial using domain knowledge [36] [37]. As the dataset grows in dimension as well as

in sample size, it is extremely difficult to create relevant features. Feature selection is also very difficult in high dimensional data. Thus feature engineering, in total, is extremely complex and challenging task in BD analysis.
- Non-linearity: The presence of non-linearity in BD poses many challenges to the existing methodologies used to evaluate the dataset characteristics and algorithms performance. Most of these methods include the common assumption of linearity [38]. Overall, the presence of linearity and non-linearity pose challenges to the execution of ML algorithms in the context of BD.
- Variance and Bias: Generalization is a very important term in ML algorithms. Generalization error can be divided into two types such as variance and bias. Variance defines the consistency of a learner's ability to predict random things, whereas bias describes the ability of a learner to learn the wrong thing [39].

2) **Variety:** It defines the data type, variety and its sematic meaning [40]. The major challenges originated by variety while solving ML algorithms and DL architectures are discussed below;
   - Data Locality: Due to the large volume of data in the context of BD, the availability of the complete data in memory or a single disk file is not possible [41]. However, the datasets distributed over large number of different files residing in different physical locations.
   - Data heterogeneity: Primarily, the datasets are collected from different sources in the context of BD. These datasets are different data type, format, data model and semantics. Syntactic and semantic are two different types of heterogenity categories. Syntactic heterogeneity refer the data of different types, file formats, data encoding, and data model. Semantic heterogeneity refers to the different meanings and interpretation of data.
   - Dirty and Noisy data: BD typically acquired from different locations across different time range. The datasets are different format. These dataset may contain noisy [42] as well as dirty. Additionally, these datasets contain measurement errors, outliers, and missing values. [43] discussed the importance of preprocessing step for removal of noisy and dirty datasets.

3) **Velocity:** It mentions the rate at which the data generates and methods to handle them. Due to the deployment of IoT applications of smart cities and other various applications of smart cities, the velocity of BD has become an important factor to consider. The major challenges originated by velocity while solving ML algorithms and DL architectures are discussed below;
   - Data Availability: Primarily ML models rely on data to learn different patterns. These models assume that the complete dataset is available before starting the training phase. These models can't handle data streaming. As the time evolves the amount of data also increases. In order to cope with the new patterns, the ML models has to be retrained.
   - Real-time processing: The ML algorithms are not able to handle data in real-time stream processing. Real-time stream processing is very important in recent days due to the emergence of sensors, mobile devices, and IoT. To handle the real-time processing, various technologies are introduced. Mostly these technologies do not come with sophisticated ML algorithms. Integrating the streaming solutions with sophisticated ML algorithms is very much required.
   - Concept drift: The datasets in BD are non-stationery. This is primarily due the reason that the data arrival continuously in real-time systems and the distribution of current data flow may not be the same in future data. This is termed as concept drift. This can decrease the performance of ML model. It is very important factor in the field of Cyber Security because the trained models on the recent patterns of user behaviors and malware binaries are able to accurately predict malware.
   - Independent and Identically distributed random variables: Mostly, ML algorithms assume that the random variables are independent and identically distributed. However, this may not be the same case in real-time. Typical solution to handle this situation is that randomized the data samples before training the ML model.

4) **Veracity:** It deals with the completeness of the data. The major challenges originated by veracity while solving ML algorithms and DL architectures are discussed below;
   - Data provenance: Data provenance provides a historical record of the data and its origins. However in the case of BD, the provenance dataset itself becomes too large. These can add additional context to ML.
   - Data uncertainty: Generally, the data are continuously collected from different sources in a real-time environment. These different data can initiate uncertainty.
   - Dirty and Noisy data: The data collected from different sources are not accurate as well as are noisy. There may be possibility that the labels or contextual information of data being collected in real-time environment may not be accurate always.

Infrastructure is the primary component in BD technology. BD infrastructure provides an environment and method to store, process and analyze data using ML techniques. Generally, BD technologies are divided into 2 categories namely batch processing and stream processing. Batch processing basically analysis the data at rest whereas stream processing analysis the data in motion. The most famous technology of batch processing is Hadoop. The Hadoop framework consists of Hadoop Distributed File System and MapReduce pro-

TABLE V
COMPARISON OF POPULAR DEEP LEARNING FRAMEWORKS.

Deep learning Frameworks	**Creator**	**License**	**Core Languages**	**Interface Support**
Caffe [232]	Berkeley Center	FreeBSD	C++	Python, & MATLAB
Torch [233]	Ronan Collobert et al.	BSD	C, & Lua	C/C++, Lua, & Python
Theano [234]	University of Montreal	BSD	Python	Python
Deeplearning4j [235]	Skymind	Apache 2.0	Java	Java, Scala, & Python
MXNet [236]	Apache Software Foundation	Apache 2.0	C++	C++, Python, R, Scala, Perl, Julia, & etc.
TensorFlow [237]	Google Brain Team	Apache 2.0	C++, & Python	Python, C/C++, Java, & Go
Neon [238]	Intel	Apache 2.0	Python	Python
H2O [239]	H2O.ai	Apache 2.0	Java	R, Python, Scala, & Rest API
Chainer [240]	Japanese venture company Preferred Networks in partnership with IBM, Intel, Microsoft, & Nvidia	MIT	Python	Python
CNTK [241]	Microsoft Research	MIT	C++	Python, C++, & BrainScript
Keras (higher level library for TensorFlow, CNTK & Theano) [242]	Franois Chollet	MIT	Python	Python

gramming model. Hadoop Distributed File System is utilized by the developers to store large files whereas MapReduce programming model is tuned to work on large-scale data processing problems which can be parallelized and distributed. Various tools are there which have the ability to help analysts create and work with complex queries and run ML frameworks on top of Hadoop. These tools include Hive (an SQL-friendly query language), Pig (a platform and a scripting language for complex queries), and Mahout and RHadoop (DM and ML algorithms for Hadoop). New framework were designed to improve the performance of DM and ML algorithms such as Spark 4. To improve the performance of advanced data analytics algorithms, these frameworks repeatedly reuse the working data set. Various databases were specifically designed for efficient storage and query of BD such as, CouchDB, Cassandra, HBase, Greenplum Database, Vertica, and MongoDB. Stream processing unlike batch processing does not have a single dominant technology like Hadoop, it is still a growing field of development and research. Complex Event Processing (Luckham 2002), is one of the models for stream processing. In this model, high level events are produced by aggregating and combining notification of events which are considered from the information flow. Storm, InfoSphere Streams, and Jubatus are few other implementations of stream technologies.

The conservation of privacy solely depends on technological drawbacks on the ability to extract, analyze and correlate potentially sensitive datasets. BD analytics development is providing tools not only to extract data but also to utilize this datat to make violations of privacy easier. Therefore, with development of these BD tools, safeguards creation has become very important to prevent abuse these BD tools [57].

The significance of dimensionality reduction in Cyber Security: Autoencoder is a generative model which learns the latent representation of different feature sets. In an unsupervised way, it learns very important features and is considered as a suitable method for network traffic analysis. This is due to the reason that ICT systems generates a very large amount of data in fraction of time and within this time the prepossessing of data should be performed without any lose of information. AE can also be used as dimensionality reduction techniques. SVD and PCA are the most commonly used classical method for dimensionality reduction.

Main factors limiting the growth of AI and DL is the burden of handling massive datasets and the heavy computational requirements associated with their processing. Large datasets were earlier handled by clusters of computers using CPU's to perform these tasks. Recently, the power of GPU's has been harnessed better and has proven to be faster and better for large scale data processing. These developments had made large scale data handling and processing quite trivial as compared to the earlier burdens and hence a number of platforms tailored for developing ML and DL applications have come up recently. Such popular platforms for implementation of DL architectures include TensorFlow, Theano, Torch, Caffe, DeepDist etc. Detailed information of popular DL framework is given in Table V. Most of the well-known platforms have used C++ back end supporting high level ML computations with little overhead and a simple to use Python front end.

TABLE VI
DETAILS OF BENCHMARK DATASETS.

Task	Most Commonly used benchmark Dataset
Intrusion detection	KDDCup-99 [179], NSL-KDD [180], UNSW-NB15 [181], ADFA-LD [182], UNM [183], ISCX-IDS-2012 [184], CICIDS2017 [185], Kyoto [186], UNIBS [187], CAIDA [188], LBNL [189], CIC DoS [190], CSE-CIC-IDS2018 [191], AWID [192], & WSN-DS [193]
Botnet, & DGA Analysis	UNB Botnet [194], DGArchive [195], & AmritaDGA [196]
URL Analysis	ISCX-URL-2016 [197], & Sophos URL [198]
Spam, & phishing Email detection	CSDMC [199], Enron [200], TREC [201], & SpamAssassin [202]
Malware Detection	EMBER [203], Microsoft malware classification challenge [204], Microsoft Malware Prediction [205], & Malrec [206]
Binary Analysis	ByteWeight [207]
Image spam detection	Image spam hunter [208]
Android	Android Adware & General Malware Dataset [209], CICAndMal2017 [210], Drebin [211], & Kharon [212]
Traffic Analysis	ISCXVPN2016 [213], & ISCXTor2016 [214]
Security in IoT	N-BaIoT [215], & Bot-IoT [216]
Side channel attacks	DPA contest [217], & ASCAD [218]
Insider threat detection	CERT [219]

Moreover, Keras has become most commonly used higher level API which has support for TensorFlow, CNTK and Theano.

Most commonly used tool for CMLAs implementation is scikit-learn [44]. It is an open-source ML library which was developed by David Cournapeau in 2007 as Google summer of code project. The entire library was written in Python and includes python numerical and scientific libraries like NumPy and SciPy. Additionally, it includes other libraries like pandas, matplotlib, seaborn, etc. It gives various tools for ML like classification, clustering, regression algorithms. It also contains several feature engineering techniques and additionally gives the learning tutorials for every concept.

Of all these choices available, the TensorFlow platform has been chosen for all the NN implementations in recent days. It is an open source library for numeric computation using data flow graphs. TensorFlow is the second generation of ML platforms developed by the Google Brain team after DistBelief. As the name suggests, TensorFlow represents a problem with a data flow model acting on N dimensional arrays (tensors). The key advantage of the framework is its flexibility; the model can be mapped onto a range of hardware platforms ranging from a mobile device to massive GPU clusters. Further, the team has provided great documentation and support for easy development so that existing problems could be easily mapped onto this scenario and tested. The framework comes with an efficient C++ back end supporting high level ML computations with little overhead and a simple to use Python front end. However, [45] study the vulnerabilities of common DL frameworks like Caffe, TensorFlow and Torch. Unlike small code size of DL models these are complicated and have heavy dependencies on varied open source packages. It considers the risks by examining their impact on common DL applications or control-flow hijack attacks which result in compromise of systems or evasion of recognition. It draws attention on software implementation and need for improvement of security of DL frameworks.

VI. MAJOR ISSUES IN THE EXISTING CYBER SECURITY SOLUTIONS AND IMPORTANCE OF SHARED TASKS IN CYBER SECURITY

In spite of the fact that vast distributed ML and DL based Cyber Security identification arrangements exists, undertaking organizations are still battling with a contradictory situation between the determinations of ML and DL algorithms and benchmark datasets. Recently, the challenges and issues involved in employing the data science techniques for Cyber Security applications were discussed in detail by [47]. The details of benchmark dataset are reported in Table VI. Finding a satisfactory dataset for Cyber Security use cases is often troublesome. on top of that, some of the datasets have their own particular issues. Most regular issues are, (1) the vast majority of the datasets are outdated (2) they are not genuine agent datasets (3) Most of the security researchers follow different splitting methodology to divide data into train, valid and test categories (4) Most of the present datasets in the field of Cyber Security are not broadly accessible to the research community due to security and privacy reasons. This leads to experimental results that are not reproducible. Due to these issues, the use cases of Cyber Security doesn't have a standard approach and most enterprises avoid using ML and DL solutions for improving their Cyber Security applications [46].

Most recent way to enhance the performance of a system is by organizing the shared tasks as a part of conference and workshop. Shared tasks are competitions to which researchers or teams of researchers submit systems that address specific, predefined challenges. Initial phase of the shared task is to distribute the train dataset among the participants. Evaluation of trained models is performed utilizing the test dataset. Finally,

the results are made publically available and give an option for publication. Shared tasks are most familiar in the field of NLP, computer vision and speech recognition. Recently, CDMC[1], IWSPA-AP[2], DMD 2018[3], and AICS 2019[4] are 4 shared tasks in Cyber Security is organized and the details of the shared tasks in Cyber Security are reported in Table VIII.

VII. Statistical Measures

To evaluate the performances of the DL models, various statistical measures are used. The most commonly used statistical measures in the field of Cyber Security are discussed below.

Confusion matrix: The confusion matrix is a matrix representation which shows the classification results in detail, whether they are correctly or incorrectly classified and different classes are distinguished. Each row of the matrix represents the instances in a predicted class while each column represents the instances in an actual class (or vice versa). The confusion matrix for binary class classification is shown in Table VII. The dimension of confusion matrix for classes is . Let P and N be the number of positive and negative samples in the test set respectively.

True Positive (TP)**:** Positive samples correctly classified by the DL model.

False Negative (FN)**:** Positive samples that are misclassified by the DL model.

False Positive (FP)**:** Negative samples that are misclassified by the DL model.

True Negative (TN)**:** Negative samples that are correctly classified by the DL model.

Using confusion matrix, the following metrics can be estimated

Accuracy: is the measure that gives total number of correct predictions made out of the all the predictions made by the model. It is a good measure to use when target classes are nearly balanced in the data.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

Precision: is the measure for ability of the model to give quality positive predictions. It can be interpreted as the probability that positive prediction made by the model is positive.

$$\text{Precision} = \frac{TP}{TP + FP}$$

Recall or True Positive Rate (TPR) **or Sensitivity:** measure is the ratio of the true positive (TP) divided by the total of true positive (TP) and false negative (FN) predictions.

$$\text{Recall} = \frac{TP}{TP + FN}$$

F1-measure or F1-score: measure is given by the harmonic mean between the precision and recall. It can be a better measure when target classes are unevenly distributed.

$$\text{F1} - \text{score} = \frac{2 * \text{Re}call * \text{Pr}ecision}{\text{Re}call + \text{Pr}ecision}$$

False Positive Rate (FPR)**:** measure is the ratio between the false positive values and the total false positive and true negative values.

$$\text{FPR} = \frac{FP}{FP + TN}$$

True Negative Rate (TNR)**:** measure is the ratio between the true negative values and the total true negative and false negative values.

$$\text{TNR} = \frac{TN}{TN + FN}$$

False Negative Rate (FNR)**:** measure is the ratio between the false negative values and the total true positive and false negative values.

$$\text{FNR} = \frac{FN}{TP + FN}$$

The values of Accuracy, Precision, Recall, F1-score, FPR, TNR and FNR range from 0 to 1 and larger Accuracy, Precision, Recall, F1-score, FPR, TNR and FNR represent better performance. All these measures are corelated for example, the desire to increase the TPR may result in the undesired increase of the FPR. Thus during design phase, an optimal detection accuracy is usually assessed based on a discrimination threshold that reects the dependency of TPR on FPR, which is called the Receiver Operating Characteristics (ROC) curve. It is obtained by plotting the TPR on X axis and FPR on Y axis. For comaprision purpose, the area under the curve is estimated. Generally, AUC values range from 0.5 to 1.0, and larger AUCs represent better performance.

VIII. Importance of Transfer learning in Cyber Security Applications

Transfer learning is a method of making use of already existing model of a particular task on another related task. This method is very popular in DL particularly in various problems related to natural language processing and computer vision. This is achieved by replacing the output layer for classification by new output layer. This type of learning method is very useful when the particular task contains very less amount of data. It saves time and there is a possibility to get better performance. For example, there are different types of logs can be collected from an end user hosts. As we know that the logs provide important information that can be used to identify the reason behind each and every activity. Suppose the aims of the model is to detect the malicious activity, the logs from system, proxy, DNS and other can be used. Initially, the DL model trained on these datasets can be used to detect malicious activities. However, later suppose we need to develop a DL model to detect botnets using DNS query information. In this case instead of training from scratch, the performance of the DL model can be enhanced for botnet detection by using the already trained model. This can further save time and is computationally inexpensive. This is illustrated in Figure 8. In [515], a new clustering-enhanced hierarchical TL technique

[1]http://www.csmining.org/cdmc2018/
[2]https://dasavisha.github.io/IWSPA-sharedtask/
[3]https://nlp.amrita.edu/DMD2018/
[4]http://www-personal.umich.edu/ arunesh/AICS2019/index.html

TABLE VII
CONFUSION MATRIX.

	Predicted as Positive	**Predicted as Negative**
Positive Label	True Positive (TP)	False Negative (FN)
Negative Label	False Positive (FP)	True Negative (TN)

TABLE VIII
DETAILS OF SHARED TASKS ORGANIZED IN CYBER SECURITY.

Shared Task	**Description**
CDMC	It is a multidisciplinary competition conducted once in a year as part of AICS & ICONIP. They allow top performed system to submit their method in the form of working note.
IWSPA-AP	Phishing email detection organized as part of ACM CODASPY 2018. They have allowed all the participants to submit their method in the form of working note & published in CEUR workshop proceedings.
DMD 2018	Detecting malicious domain names organized as part of ICACCI'18 & SSCC'18. They allowed all the participants to submit their method in the form of working note & published in Springer CCIS workshop proceedings.
AICS 2019	This task is organized as part of AAAI-19 conference. The main objective of this task is to build robust malware classification to adversarial evasion attacks. They have allowed participants to write the system description paper & submit into arxiv.

which finds the relation between known and new attack is presented. The proposed TL technique is evaluated using combination of different traditional ML classifiers such as DT, random forest, KNN, SVM and naive bayes with various existing TL approaches.

IX. ADVERSARIAL DEEP LEARNING FOR CYBER SECURITY

As ML is being applied for deployment in various critical systems, it is extremely imperative to consider the reliability of such algorithms. The threats that are presented on ML framework by adversarial agents are genuine. Hackers can carry out their malicious activities by exploiting the vulnerabilities of ML framework using adversarial samples just like how they access web servers by exploiting firewalls vulnerabilities. So it is very important to consider the shortcomings of ML framework and understand how much influenced they can be under stress before putting such solution in the line of fire. ML vulnerabilities studied in adversarial environments is known as adversarial ML. As a large portion of ML frameworks behave as black boxes in critical systems, adversarial ML faces a lot of difficulties. It is exceptionally troublesome for experts and clients to comprehend the model results on the grounds as there is no straightforwardness in what is going on inside a classifier and a indicators. As there is no much explanation about the decision made by the framework, the users can't recognize whether any malicious activities are influencing the framework. As long as there is no assurance about the robustness of these frameworks, the resistance for adoption and acceptance of ML frameworks used for secure and critical system will be there. A short review on adversarial DL for Cyber Security applications is reported in Table IX.

A. Domain Generation Algorithms

In [377], a DL based architecture (GAN) that can generate adversarial domain names is presented. The authors have demonstrated that these adversarial examples which are generated for DL architecture can also fool, a totally different classifier like random forest. They have also enhanced the performance of random forest classifier by training the model using these adversarial samples.In [402], the authors have proposed an oversampling method based on GAN to produce adversarial URLs. The URLs from GAN and other sources are used to train a phishing URL detector which outperforms other oversampling techniques. In [408], a GAN based text-captcha solver is proposed. The authors generate synthetic text-based captchas using GAN and applies TL for the captcha solver training. The proposed method performs better when compared to other state-of-the-art models and requires very less human intervention. In [410], the authors have proposed a text based captcha solver based on GAN and CNN. The proposed framework requires only small sample of data with no complicated preprocessing and it achieves better results when compared to normal CNN based baseline model. In [418], a black box adversarial attack approach for evading DGA detection classifier is proposed. The proposed adversarial approach adds a small perturbation to the character-level representation of the DGA domains without any prior knowledge of the classifier model. The proposed approach is tested using DMD-2018 dataset and it successfully degrades the F1-score to 0.495 from 0.977.

B. Malware or Malicious Software

In [378], the authors have presented a novel method called Malware Recomposition Variation which generates adversarial malware examples based on semantic analysis of various existing malwares to evade malware detector. They have also evaluated three defenses techniques to enhance the robustness of the detector against the proposed approach. In [380], Grosse et al. have employed existing adversarial example

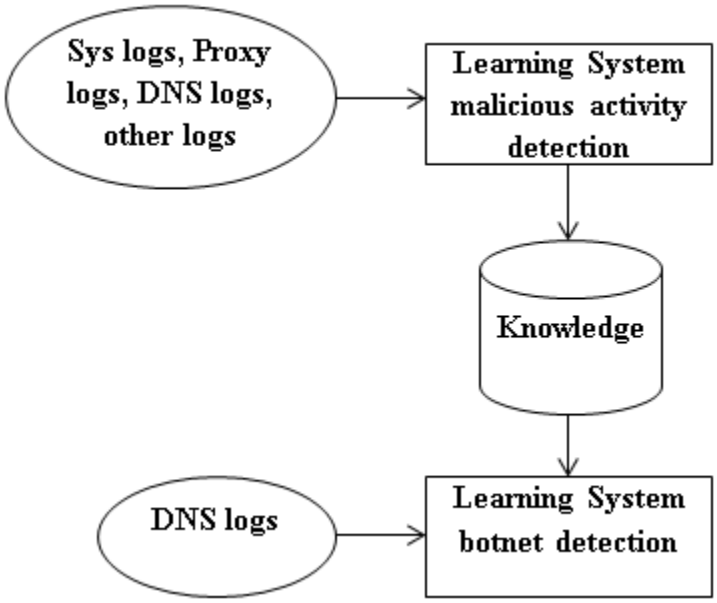


Fig. 8. Transfer Learning.

generation algorithms to generate malwares without losing its intrusive functionality. They have evaluated the robustness of a malware detector based on NN against adversarial attacks. They have reduced the accuracy of the model considerably using adversarial examples, and they have also investigated some defense techniques for adversarial attacks. In [381], a malware detection framework based on transferred GAN (tGAN) is proposed which is specifically designed for zero-data attack detection. The proposed model has good learning stability and it achieves 96.39% accuracy which is better when compared to other related DL and traditional ML detection classifier. In [382], a GAN based black box attack malware detection model is proposed. The proposed method generate adversarial examples to trick the neural network based malware detector model with almost 100% success rate. In [383], the authors proposes a novel approach to construct DL model that are robust against adversarial attack. The proposed approach nullifies random features in the data in order to boost the robustness. They have shown significant improvement of robustness of the malware detection model against adversarial malware binaries with a slight decrease in accuracy. In [385], the authors have presented a novel framework based on saddle-point optimization for training malware detector that are robust to adversarial examples. They have also presented four techniques to generate adversarial examples without losing its malware functionalities. In [386], the authors have proposed a gradient based attack method for generating adversarial malware examples to evade DL based malware detector which learns from raw bytes. The adversarial examples are generated by modifying the few bytes of malware samples without losing its intrusive functionality. It can be observed that the proposed method can evade the MalConv network architecture with 60% probability in best case. In [393], Rigaki et al. presents a GAN based adaptive malware which avoids detection by mimicking benign network traffic based on the parameter given by GAN. The proposed malware system sends its blocking status as feedback to enhance the performance of the GAN model. In [396], a GAN based malware detection framework is proposed which can detect zero-data attacks. The proposed variant of GAN has better learning stability than other GANs and it achieves 95.74% accuracy which is better when compared to other state-of-the-art detection models. In [398], the efficacy of existing adversarial attack techniques against DL based malware detection model is studied. The authors have also proposed a novel adversarial attack which alters the bytes of the binary to generate adversarial examples. In [399], the authors have proposed a framework for improving the robustness of DL based malware detection model against adversarial attacks based on the six principles that they have compiles. In [400], authors propose robust malware detection module using GAN. For interpretation, various visualization methods were proposed by them. These visualization helped to visualize the malware behaviours. In [401], a new adversarial attack against CNN based malware detection model is proposed. The proposed approach generates adversarial malware binaries by injecting few bytes into the original malware file and evades MalConv model with high probability. In [411], the authors studies the robustness of API call based malware detection models against adversarial examples. They have proposed a black box attack which generate adversarial API call and static features sequences to trick the classifiers like RNN, DNN, and other ML classifiers. In [412], a framework based on hash function for improving the robustness of DNN against adversarial malware binaries is proposed. The proposed method uses locality preserving hash functions and SdA to avoid the effects of adversarial attack. In [413], a novel approach to detect obfuscate malware using GAN is proposed. The proposed method extracts feature from VAE and trains the GAN generator to knowledge space and achieves accuracy of 96.97% and performs better when compared to

TABLE IX
A SHORT REVIEW ON ADVERSARIAL DEEP LEARNING FOR CYBER SECURITY APPLICATIONS.

Reference	Dataset
[377]	Alexa, & Private
[378]	Genome, Contagio, VirusShare, & Drebin
[379]	KDDCup-99, DARPA98 & NSL-KDD
[380]	Drebin
[381]	Microsoft malware classication challenge (BIG 2015)
[382]	Private
[383]	Malicious Behavior Windows Audit Logs
[384]	Private
[385]	VirusShare
[386]	VirusShare, Citadel, & APT1
[387]	Spambase
[388]	Wild dataset, & Microsoft malware classication challenge (BIG 2015)
[389]	GTSRB
[390]	Enron Spam Dataset
[391]	Mozilla Common Voice Dataset
[392]	VirusShare
[393]	Private
[394]	NSL-KDD
[395]	ISCX BOTNET
[396]	Microsoft malware classication challenge (BIG 2015)
[397]	NSL-KDD
[398]	Private
[399]	AICS'2019 challenge dataset
[400]	Private
[401]	Microsoft malware classication challenge (BIG 2015),& benign samples are collected privately
[402]	Private
[403]	NSL-KDD
[404]	Car-Hacking Dataset from HCRL
[405]	ADFA-LD
[406]	Private
[408]	NSL-KDD
[409]	Publically available for further research
[410]	Private
[411]	Private
[412]	Private
[413]	Microsoft malware classication challenge (BIG 2015)
[414]	CIDDS-001
[415]	KDDCup-99
[416]	Ember
[417]	KDDCup-99
[418]	AmritaDGA
[419]	CycleGAN, & StarGAN
[407]	NSL-KDD

other conventional ML models. In [416], the authors have proposed a new adversarial attack against MalConv which evades detection with high certainty. They have found few vulnerabilities in MalConv by analyzing its learned weights and results. The proposed attack exploits these vulnerabilities to generate adversarial malware examples by modifying few bytes in file header.

C. Intrusion Detection Systems (IDSs)

In [379], the authors have evaluated the robustness of various Network Intrusion detection classifiers like DT, Random forest, Linear SVM, Voting ensemble against adversarial examples. It can be observed that the accuracy has been degraded from 73% to 45% in best case. In [394], a new multi discriminator GAN architecture to enhance the performance of anomaly detection system is proposed. The proposed approach generates adversarial examples that is used to improve the detection model. In [397], the authors presented a deep AEs based adaptive ID System and a novel framework to tests its robustness against adversarial examples. It can be observed that the proposed approach achieves 15% more accuracy when compared to PCA based detection system. In [403], a blackbox attack based on GAN against ID System is proposed. The proposed framework generates adversarial network traffic which are malicious to evade detection with high success rate. In [404], The authors have proposed a GAN based ID System model for in-vehicle networks. Since the amount of known attacks are very less in vehicle networks, they have used GAN to generate fake data and used it to train the model. It can be observed in the given experiment results that the proposed model performs well with high detection rate for even unknown attacks. In [405], a Host-based ID System (HIDS) based on GAN is proposed. The GAN produces adversarial anomalies which is used to train the Artificial NN model along with the original dataset. The proposed method improves the detection accuracy of unseen anomalies from 17.07% to 80.49%. In [406], an SDN-based port scan detecting framework based on DL architectures such as CNN, MLP, and LSTM are studied and their robustness against four adversarial attack algorithms are tested as well. It can be observed that among 4 adversarial attack methods, JSMA performed better and reduced the accuracy comparatively well. In [407], the authors have studied the effect of three block box adversarial attack against DNN based Network ID Systems (NIDS). It can be observed from the results that the accuracy of DNN model degraded notably and among the three attack method, ZOO achieves best results. In [415], the authors have proposed a network ID system which uses two data augmentation modules to address the data insufficiency challenge. The proposed method employs probabilistic generative model and DL based adversarial sample generation model to generate synthesized and adversarial data respectively. These generated data is then used to enhance the performance of the detection system. In [417], the author proposed a method based on GAN to trick the ID system and evade detection. The proposed method uses GAN to generate adversarial DDoS attack traffic and tricks the CNN based detection model. It can be observed that the detection accuracy drops significantly from 97.3% to 47.6%.

D. Other Adversarial based Attack and Defence techniques in Cyber Security

In [384], the authors studies the efficacy of two existing defences against adversarial attack and proposes a weight decay defence. They have analyzed distillation and ensemble defences and found that ensemble technique significantly improves the robustness of the model against adversarial attacks. They have shown that by adding few more hidden layers, the robustness can be improved further. In [387], an outlier based defense technique against poisoning attacks against linear classifiers is proposed. It can be observed that the proposed approach is effective but fails to detect attacks that are less aggressive like label flipping. In [388], the authors employs generalized distillation learning approach to train the DL based detection model using privileged features which are available at the time of training. They have shown that the proposed method leads to better accuracy when compared to systems with no privileged information. In [389], a novel real-world attack against computer vision based modules of autonomous vehicles. The proposed approach utilizes adversarial examples concept to trick the system to misclassify advertisements and innocuous signs as adversary's desired traffic sign with high certainty. It can be observed that the proposed method misclassifies with a success rate of 95% in the physical environment. In [390], a novel adversarial attack against DL classifier in a black box environment is proposed. The proposed approach modifies the data with small text perturbation to produce adversarial examples. They have also developed a new score metrics which rates the effectiveness of each word in contribution to misclassification. It can be observed that the proposed method degrades the accuracy from 87% to 26% on IMDB and from 99% to 40%on Enron datasets. In [391], the authors demonstrates an optimization-based adversarial attack on DL based automatic speech recognition system. The proposed approach is capable of generating adversarial example from any input audio waveform by adding small perturbation and it can trick the model to give out a specific target transcription as output with 100% success. In [392], a DL based deceptive review detection model is proposed. It uses GAN with two discriminators to detect fake reviews. The proposed approach achieves accuracy of 89.1% and performs better when compared to the baseline models. In [395], the authors presents a GAN based framework which generates adversarial examples which are used during the training phase. The proposed approach enhance the performance of the original detection model in terms of precision, accuracy, and other performance metrics and also reduces the specificity of the original detection model. [409] proposed a GAN based method for bypassing Perceptual ad-blocking. In [414], the authors have proposed a novel DL based network traffic generation method. It generates flow-based network traffic by using GANs. They have proposed three preprocessing methods and a novel evaluation technique in order to convert the flow-base data into continuous values which are fed into GAN and to evaluate the quality of the generated traffic data. In [419], a novel approach for the detection of fake images generated by GAN is proposed. The proposed approach computes the co-occurrence matrices on the RGB channels of the images and uses those matrices to train the CNN model to detect fake GAN images. The proposed framework is test using two GAN datasets such as cycleGAN and StarGAN and the model achieves 99% of accuracy for both datasets.

X. Reinforcement learning for Cyber Security

This is a revolutionary technique that is inspired by the psychological concept of Pavlov's classical conditioning technique and the mathematical concept of Markov decision process [638]. The abilities of these concepts are exploited in order to make an algorithm learn to make decisions in a certain scenario by making it experience the same scenario again and again. There are three vital elements that construct this algorithm namely, observation, reward and action. Each time, the algorithm is allowed to take a decision and observes the changes in the scenario which in turn receives a corresponding negative or positive reward. Generally, the motive is to gain the maximum reward. Therefore the algorithm, aided by the reward uses the Markov decision process to either receive the maximum reward or reach a particular goal. This process is known as classical reinforcement learning which are only suitable for smaller problems. In case of larger problems, Deep Reinforcement Learning approaches are used. This approaches utilize NNs or approximation methods for finding the optimal value or solution for the problem. Deep reinforcement leaning based solutions for the applications of Cyber Security are still in this beginning stage. This methodology can be suitable for Cyber Security applications like botnet detection, malware detection etc. A short review on RL for Cyber Security is reported in Table X.

A. Reinforcement learning based Intrusion Detection

In [323], an adaptive ID system based on ML is proposed. The proposed framework uses multi-class SVM with PCA for feature reduction and RL approach for prediction. The model is trained on benchmark datasets such as the KDDCup-99 data and the system call data and it achieves promising results. In [324], a distributed ID system based on multi-agent RL is proposed. The proposed approach uses RL sensor agents to learn to differentiate normal and abnormal network states and the decision agents to learn the semantics of the actions sent by the sensor agents. The authors have given a detailed analysis using realistic traffic from network simulation. In [325], a decentralized DDoS detection and response system is proposed. The framework uses multi RL agent router throttling to respond to DDoS attacks. The proposed approach is more secure, autonomous, scalable and it achieves better performance when compared to other related approaches. In [326], the authors have proposed a new distributed scalable framework for ID and response system based on RL. The proposed decentralized approach is designed especially for DDoS detection and response. The approach is to install RL agents on a set of routers which learns from the traffic to respond to DDoS attacks. In [328], the authors have proposed a RL based detection system for flooding based DoS and DDoS attacks. The proposed approach uses RL agents to

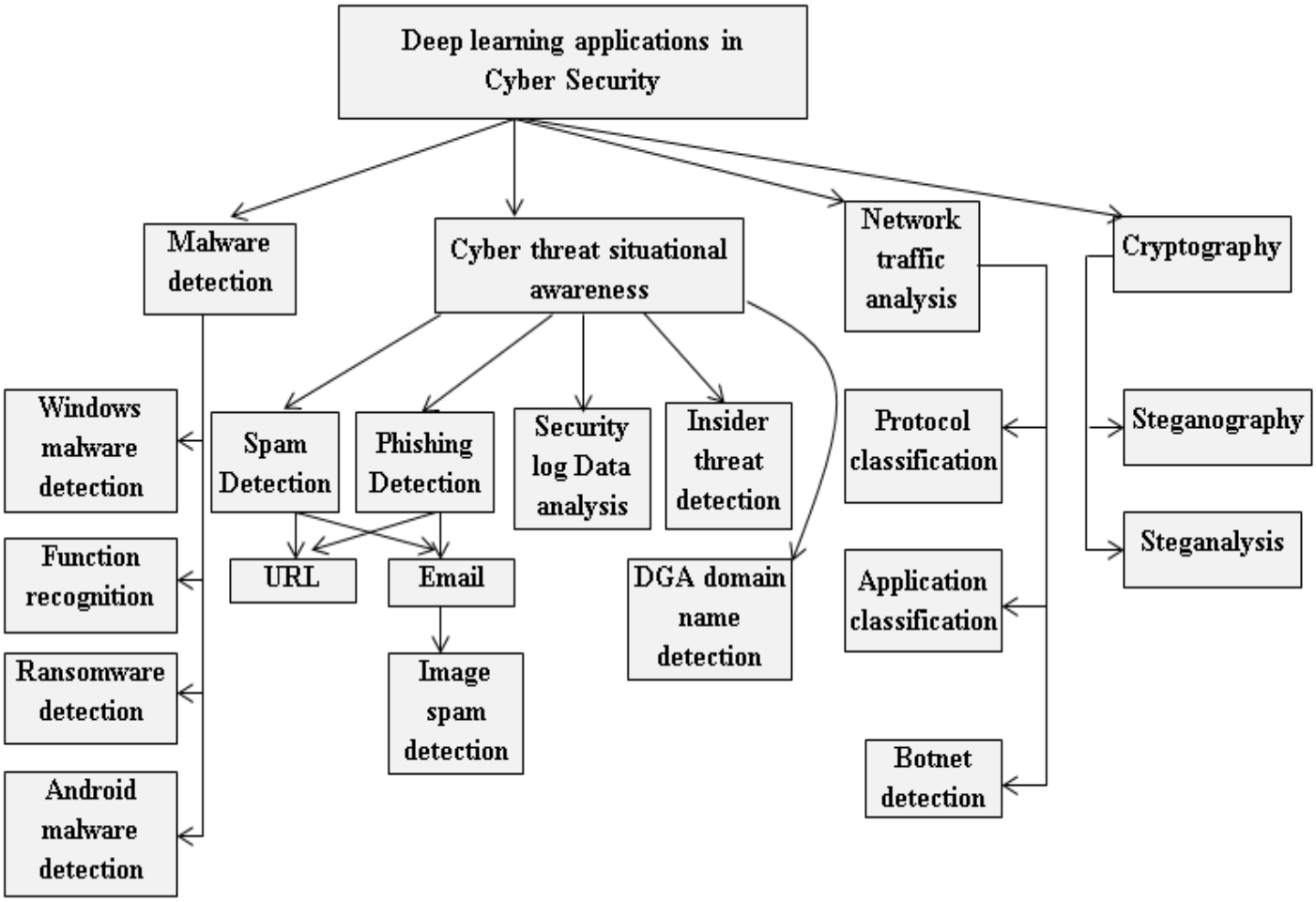


Fig. 9. Deep learning methods for Cyber Security Applications.

analyse the data flow information between hosts in order to differentiate normal and malicious traffic. In [331], a deep RL based real-time DDoS attack mitigation framework for SDN environment is proposed. The RL agents learns the optimal policies for various attack scenario in order to mitigate the attack. The proposed framework can responds well for various attacks such as TCP SYN, UDP, and ICMP flooding and it outperforms the other baseline models.

B. Other various applications in Cyber Security

In [327], the authors have proposed a RL based framework to attack anti-malware engine and evade detection. The RL agent in the proposed model learns by performing set of operations against the anti-malware engine which will most likely leads to evasion and gives insights about the loopholes of the malware detector. In [329], an RL based black box attack to evade static PE malware detection engines is proposed. The RL agent with a set of malicious functionalities performs a series of attack on the anti-malware engine and it learns the sequences of operations which are more likely to evade detection. In [330], the authors have proposed a RL based fuzzing approach which finds security issues using modified inpurs. They have used markov decision processes to formalize fuzzing as RL problem and applied deep Q-learning algorithms in order to optimize the rewards. They have evaluated the proposed model and found that it performs better than other baseline fuzzing approaches. In [332], an evolving NN and RL based online phishing email detection framework is proposed. The proposed model is capable adapting itself and it can detect zero-day phishing attacks. The experimental analysis shows that the proposed approach achieves 98.63% accuracy and a very low FPR of 1.81%. In [333], a RL based malware generation framework is proposed which is used to improve the performance of the ML classifier. The proposed framework generates malware samples which can evade detection and trains the ML model with then generated samples to improve its accuracy. The experimental analysis shows that the accuracy improved to 93.5% from 15.75%. In [334], a new deep RL based malware execution control framework is proposed. The deep RL based framework learns to halt the execution of malware files at best time and prevents the attackers from evading the detection. The experimental analysis shows that the proposed model halts the execution of 91.3% of the files automatically and it improves the TPR by 61.5% when compared to the baseline approach. Deep RL algorithm was utilized as malware execution control model in [302].

TABLE X
A SHORT REVIEW ON REINFORCEMENT LEARNING FOR CYBER SECURITY APPLICATIONS.

Reference	Dataset
[323]	KDDCup-99
[324]	Private
[325]	Private
[326]	Private
[327]	Private
[328]	Private
[329]	VirusShare, & VirusTotal
[330]	Private
[331]	Private
[332]	PhishingCorpus, SpamAssassin, & PhishTank
[333]	Ember
[334]	Private
[302]	Private

XI. APPLICATIONS OF DEEP LEARNING IN CYBER SECURITY

In recent days, various applications utilized the frameworks of DL for various use cases of Cyber Security. Additionally, the application of NLP, signal and image processing and BD analytics leveraged along with DL. In this section, we discuss the number of DL based Cyber Security applications. DL for Cyber Security applications is in the infancy stage due to lack of labeled data, data quality, high complexity, and dynamic environment. Various deep learning architectures that can be employed to Cyber Security are shown in Figure 9. Each of these main Cyber Security applications contain sub applications. The hierarchies of Cyber Security applications is shown in Figure 9. In the last years, Security researchers employed various deep learning methods for Cyber Security applications. These published researches are surveyed, summarized and identified the advantages, limitations and future scope of the work.

A. Deep Learning in Intrusion Detection

Today's most of the communications rely on ICTs and Internet based services. Growing reliance on ICTs, Internet and applications makes systems, networks and its services more vulnerable to attacks against critical infrastructures. The quick paced innovative progressions in this cutting edge time has encourged organization aroung the glode to adopt the integration of information and communication technology (ICT). Thus making a domain where each activity is directed through that framework making the association defenseless if the security of the ICT framework is endangered. Consequently, this for a multilayered discovery and assurance conspires that can deal with really novel assaults on the framework and in addition ready to independently adjust to the new information.

Everything in this day-to-day work is becoming connected with each other. One of the most important among them is autonomous driving. In the foreseeable future, they can see the streets flooded with autonomous driving cars due to the initiation of companies like Tesla, Waymo and several other major companies and startups. Since autonomous cars have more similarities with a modern smartphone than a traditional combustion engine car, it raises the question of cyber safety, security robustness and hack ability of the system that runs these autonomous cars. Cyber-attacks on cyber-physical systems like Controller Area Network (CAN) has been shown to be potentially vulnerable. A short review on DL applications in ID is reported in Table XI.

1) KDDCup-99 Dataset: Deep learning with ML comparative study: In [423], DBN based ID system is proposed. A greedy multi-layered DBN is capable of extracting features from large unlabeled dataset. The proposed model performs better than SVM and ANN based detection models and it achieves an accuracy of 93.49%. In [432], an LSTM-RNN based ID system is proposed. The proposed model learns from network traffic data to differentiate between normal and malicious traffic. The model achieves an accuracy of 96.93% and performs better when compared to other classifiers such as SVM, KNN and GRNN. In [424], LSTM is used for detection network intrusion and its performance is evaluated for different feature sets. The model outperforms other approaches used in KDD Cup '99 challenge with an accuracy of 93.82%. In [441], various recurrent structures are are employed for ID. These architectures are evaluated on three different datasets. In [443], the authors have studied the performance of DNN for ID system. The model is trained on normalized dataset and its performance is evaluated. The proposed model outperforms other models like SVM, DBN and etc. and it achieves an accuracy of 95.57%. In [463], a multi-layered echo-state machine is employed to model the ID based on network traffic data. The proposed model is compared with other traditional ML classifiers such as SVM, Nave Bayes, Random Forest, KNN, DT and MLP. In [474], the performance of DNN is studied for network IDS. various DNN models with different number of layers are evaluated and compared with other traditional ML classifiers. It is found that DNN with 3 layers achieved the best results. In [519], a DL based semi-unsupervised technique is proposed for network anomaly detection framework. The proposed framework reduces the dimensionality of the feature by using the novel technique and learns to classify normal and anomalous behaviours with good accuracy.

Deep learning without ML comparative study: In [421], the authors have proposed a RNN based outlier detection system. The performance of RNN is compared with other DM and statistical methods such as Hadi94, Donoho-Stahel and MML. In [422], the effectiveness of LSTM based approach is studied for ID. The LSTM model is evaluated in terms of confusion matrix, accuracy and AUC. In [425], a hybrid AEs and DBN based malicious code detection framework is proposed. The AE model facilitates dimensionality reduction to extract the useful features from the data and DBN model is used for classification. The proposed architecture performs better than simple DBN model. In [427], a DBN based online

anomaly based IDS is proposed. The DBN model is used to extract the useful features from the dataset and a LR classifier is used as detector. The proposed model achieves an accuracy of 97.9% and it performs better than other related models. In [433], the authors have evaluated the effectiveness of CNN, MLP and CNN based hybrid architectures such as CNN-LSTM, CNN-RNN and CNN-GRU for ID system. It can be observed that the CNN based architecture performs better than other hybrid models. In [438], the effectiveness of LSTM based ID classifiers with 6 different gradient descent optimizers are studied.It is found that the model with $adam$ optimizer performs better with 98.95% of accuracy when compared to other optimizers such as $rmsprop$, $adagrad$, $adadelta$, $adamax$, and $adam$. In [439], a LSTM-RNN based anomaly detection system is proposed. The model is trained on time series data and it detects anomalies collectively by observing the prediction error from a number of time steps. The application of AE is discussed for ID by [445]. In [447], SAE and stacked RBM models are studied for anomaly based network IDS. The experimental analysis shows that the SAE model performed better than RBM and the SAE consumed more time to train when compared to stacked RBM. In [456], LSTM and RNN based ID system is proposed which models the network behaviour in order to differentiate between normal and malicious traffic data. Both LSTM and RNN model performs well with an accuracy of 83% and 82% respectively in best case. In [457], effectiveness of three DL approaches for NIDS is studied. The study uses DNN, AEs and LSTM models to construct a ID classifier. It can be observed that the AEs outperforms the other models with an accuracy of 98.9%. In [469], two DL approach for detection network intrusion is proposed. Both LSTM and CNN-LSTM outperforms simple RNN based IDS with an accuracy of 89.23% and 94.12% respectively in best case. In [482], a greedy DAE based network anomaly detection system is proposed. The proposed model extracts useful features from imbalanced data and learns to differentiate normal and malicious behaviour. The proposed model achieves an accuracy of 94.71%. In [502], the effectiveness of various LSTM variants such as Dynamic-RNN for GRU and LSTM, GRU and Bi-directional LSTM are studied for IDS. PCA and RF are used as detection classfiers in which RF outperformed PCA model. They found that simple LSTM performed better than all other model in terms of accuracy where as BLSTM trained in just 190 second with accuray less than 90%. In [510], the authors have proposed a DNN based IDS. They have shown that the model achieves an accuracy of 99.9% which is better when compared to other traditional ML models.

2) NSL-KDD Dataset: Deep Learning without ML Comparative study: In [426], the authors have studied the effectiveness of DNN model based IDS. The proposed system uses fine-tuned AE model to extract useful features from preprocessed dataset and DNN model is used to differentiate between normal and malicious network traffic. In [428], a DL based flexible network ID system is proposed where self taught learning (STL) is used for learning the difference between normal and malicious network traffic. The proposed STL model achieves more than 98% accuracy in all the classification. In [440], an LSTM based DDoS attack detection system using TensorFlow framework is proposed. The model parameters are fine tuned and its performance is evaluated in both CPU and GPU environment. /the model achieves accuracy of 99.968% in best case. In [464], MLP based ID system with online and office feedback is proposed. The feedback with most relevant features are provided to the user to improve the user trust. Robustness of AE is discussed for ID [466]. In [468], a LSTM-RNN approach is proposed for NIDS. The performance of the model is compared with simple RNN based detection system. It can be observed that the LSTM-RNN outperforms the simple RNN with 75% accuracy in best case. In [477], a CNN based real-time ID system is proposed for very large network. The raw network traffic data is converted to image and give as input to CNN. The proposed model is compared with RNN and other traditional ML classifiers. It is found that the CNN achieves better false alarm rate with very few computation while compared to RNN based detector. In [480], the authors have proposed a NIDS based on stacked non-symmetric DAEs (S-NDAE). The proposed technique facilitates dimensionality reduction of non-symmetric data and it uses random forest as classifier. In [483], the authors have studied the effectiveness of CNN architectures of different depth for NIDS. They have used 3 CNN models of varying depth called shallow, moderate and deep CNN. They have found out that the deep model does not improve the performance and proposed model sometimes performs better than VAE based model. In [484], studied a detailed analysis of application of ML and DL models for ID. The study also discussed the importance feature learning in ID. In [486], the authors have used MLP and AE models for IDS. The network traffic data is preprocess where the outliers are eliminated and statistical analysis is used to obtain useful features from the data. They have found that AE model achieves 87% of overall accuracy and it outperforms MLP and other related models. In [490], proposed cooperative DBN based method for ID in cloud environment. In [493], proposed modified DBN based method for attacks and other security related event detection for heavy-duty robots. In [504], a DL based distributed attack detection system is proposed. The proposed DL model achieves 99% of accuracy in best case. In the experimental analysis, the proposed model is compared with other centralized and traditional ML approaches. In [521], the authors have proposed a ID and classification system using AEs. The AEs extracts the features from normalized dataset and learns to differentiate normal and malicious data. The proposed model achieves an accuracy of 99.3% and it outperforms other DL models like CNN, CNN-LSTM, DBN, CNN, S-NDAE and etc.

Deep Learning with ML Comparative study: In [430], a flow based anomaly detection system for SDN scenario is proposed. The proposed model learns to detect anomalies from the 6 extracted features. The experiment analysis compares the performance of the proposed model with other traditional ML approaches. In [434], two CNN based ImageNet architectures are studied for ID system and its performance is compared with other traditional ML approaches. It is found that the CNN model performed better than traditional ML classifiers. In [436], the authors have proposed a malware and network

ID system based on AEs. The proposed model automatically learns the semantic similarities among the features and performs better than other classifiers like DT, gaussian naive bayes tree and etc. In [442], the authors have proposed a LSTM based ID system which uses PCA for feature extraction. PCA is reduces the dimensions of the feature while maintaining its variance. The proposed model achieves 98.85% accuracy and it outperforms other related models. In [444], The performance of RNN is studied for ID system. RNN model is used for both binary and multi-class classification with various learning rate and number of neurons. The experimental analysis found that 80 hidden neurons and learning rate of 0.1 achieves the best result in binary classification whereas in multi-class classification, 80 hidden neurons and learning rate of 0.5 achieves the best results. In [446], two CNN based IDS with a new feature representation approach is proposed. The symbolic and continuous features from raw network traffic packets are extracted and converted into a image which is taken as input by CNN. The experimental analysis compares the proposed models with traditional ML classifier like SVM, random forest and etc. IoT based network ID method proposed by [449] using NN and implemented in FPGA. In [452], two DL based ID system is presented. The feedforward neural network and CNN models are compared with traditional ML classifiers such as DT, random forest, SVM, Naive Bayes. It can be observed that the proposed models outperforms the traditional classifiers. In [467], a STL framework based on AEs for ID is proposed. The proposed approach learns efficiently from the features and reduces the dimensionality of the features to aid SVM based detector. The model is compared with other ML based classifiers like DT, random forest, naive bayes and etc. In [471], a CNN based character level ID system is proposed. The system preprocess the data by considering the network traffic records as sequences of character. The proposed system performs better that traditional ML classifiers with an accuracy of 85.07%. In [472], the effectiveness of CNN, LSTM and AE models are studied for anomaly based ID system. Experimental analysis shows that LSTM and CNN performs better than AEs and other traditional ML classifiers. In [485], the authors have proposed a LSTM based ID system which uses semantic features of the data to classify. They have used a new feature extraction technique which extracts semantic features from various network traffic data. The proposed LSTM model is compared with other traditional ML classifiers. In [488], a GRU based IDS is proposed where a novel technique called local adaptive SMOTE is used to deal with imbalanced network traffic data. The GRU model extracts the temporal features from the data and learns to classify it. The proposed model is compared with other state-of-art approaches. In [489], a SVM and AE based IDS is proposed where AE is used for dimensionality reduction and to extract the useful features automatically and SVM is used as detection classifier. It can be observed that model using $ReLU$ and cross entropy gave the best result. In [503], a Gaussian-Bernoulli RBM is proposed for detection of DDoS attack. The RBM model has 7 hidden layers and the hyperparameters are optimized. The proposed model is trained using normalized dataset and it performs better than other models like Bernoulli-Bernoulli RBM and DBN with 73.2% of accuracy in best case. In [509], a stacked sparse AE based IDS framework is proposed where high dimensional sparse features are extracted automatically to classify normal and maliciou traffic. The experimental analysis found that the features extracted in this study accelerates the detection process and they are far more discriminative for intrusion behaviors compared low dimensional features. The DL architectures performed well in compared to all other architectures [608]. Various DNNs performances are evaulated for network intrusion detection by [625]

3) Private Dataset: In [429], the authors have proposed a DL based DDoS detection framework in SDN environment. They have used SAEs to extract features from large network traffic and to learn to differentiate normal and DDoS attack traffic. The proposed system achieves an accuracy of 99.82% in binary classification and 95.65% in multi-class classification. In [437], the LSTM based centralised host ID system is proposed. Three LSTM models are trained on efficiently proposed system call sequences from the host system. All three models outperforms the other ML classifiers such as random forest, SVM and LR with 0.924 precision in best case. In [458], various ML and DL approaches are used for DDoS detection in consumer IoT devices. KNN, SVM, DT, Random forest and DNN models are trained on network traffic data. The experimental analysis compares the performance of these models in context of stateless and stateful features and all models achieves good accuracies ranging from 0.91 to 0.99 approximately. In [481], a new lightweight network IDS based on ensemble of AEs is proposed. The proposed system performs online anomaly detection by extracting the features from network traffic. The system is lightweight and its performance is evaluated on devices like IoT network, camera surveillance network and Raspberry Pi router network. In [491], DL models are studied for real-time financial fraud detection framework. SAE and restricted boltzmann machines (RBM) are trained using a dataset extracted from one month user transaction logs of a private money service company. It can be observed that the RBM model achieves 92% accuracy and it performs better than SAE model whose accuracy is 81%. In [492], the authors have proposed a low speed port scan detection system based on CNN model. The system filters the normal packets and group the remaining suspicious packets using its source and destination IP. The proposed CNN model extracts the interval and sequential features from the input and learns to detect port scan. The experimental analysis shows that the proposed model achieves precision of 97.4%. In [501], a centralized localization attack detection framework in wireless sensor network based on DL model is proposed. The SdAs are used to learn from the topological and positional features of the data and its performance is evaluated. The proposed system achieves an accuracy of 94.39%. [514] proposed a NN architecture for ID. In [517], the authors have proposed a CNN based IDS where the controller area network (CAN) data is preprocessed and mapped to 2D images and fed into CNN. The CNN models extracts the useful features and learns to detect malicious ECU attacks with an accuracy of more than 90% using limited data.

4) UNSW-NB15 Dataset: In [451], a CNN based ID system (IDS) which can be employed in the router is evaluated. The layout of features are rearranged by the genetic algorithm in order to enhance the performance of the IDS. The genetic algorithm enhances the detection capacity to 0.77 from 0.71. In [461], an Bi-directional LSTM based approach is applied for detecting intrusion in Iot network. The model learns to differentiate between normal and malicious traffic. The proposed model performs well and it achieves an accuracy of 95%. In [475], a novel encoding approach is proposed which enhance the performance of CNN for network anomaly detection. The CNN model is compared with random forest based detetor. The experimental analysis shows that the proposed encoding approach consistently gives better results when compared to gray-scale encoding. [513] proposed hybrid DAE and MLP for ID. The DAE was used for feature extraction and MLP for classification.

5) Kyoto Dataset: In [454], a hybrid GRU and SVM based ID system which uses network traffic data is proposed. The proposed system uses SVM instead of softmax in the final output layer in order to enhance the performance of the detection system. The model achieves an accuracy of 81.54 while the accuracy of softmax approach is 63.07.

6) ISCX-IDS-2012 Dataset: In [455], the authors have proposed a anomaly detection framework based on LSTM-RNNs models. The proposed models uses flow sequences to model the network behaviour. In [498], a IDS based on CNN and random forest algorithm is proposed. Word embedding is applied to the payloads obtained from raw network traffic is fed into CNN which extracts payload features. The statistical features extracted from the network traffic and payload features from the CNN is used to train the random forest classifier. The proposed model achieves accuracy of 99.13% and FAR of 1.18% which is better when compared to other SVM, NN, CNN and Random forest models. In [499], a CNN and LSTM based network IDS is proposed where word embedding is used for dimensionality reduction. The CNN model learns from payload text features and its output given to LSTM model with header feature in order to learn the temporal features. The proposed model achieves 99.97% accuracy and a very impressive FAR of 0.02% which are better than other state-of-the-art approaches. In [500], a hybrid SAE and SVM based IDS is proposed. The SAE model is used for dimensionality reduction and for extraction of 10 latent features and then the SVM model is used as classifier. The proposed model whose accuracy is 92% performs better than PCA-GMM and RBM approach in terms of accuracy and execution speed. In [505], a distributed IDS based on random forest (RF) and DL is proposed. The research uses a imbalanced network traffic dataset comprising of multiple attack types and it resolves the class imbalance problem by using oversampling technique. The proposed approach uses specific model for each attack type for first phase and uses distributed RF and DL models in the second phase.

7) CICIDS2017 Dataset: In [462], the authors have proposed two port scan attempt detection framework. They have used DL and SVM models which are trained on normalized dataset. The performance of both the models are evaluated and it can observed that DL model acheveils 97.80% of accuracy while SVM achieves an accuracy of 69.79%. In [479], an anomaly detection model based on LSTM network is proposed. The model uses multiple flows to extract the temporal features. The attention approach is used by the model to focus on useful features. The performance of the proposed model is evaluated and it is compared with various ML and DL models. The proposed model achieves an accuracy of 91%.

8) AWID Dataset: In [470], the authors have proposed a DL based solution for WiFi NIDS. The SAEs and DNN are used to classify normal and attack network traffic. The proposed model classifies the traffic into 4 classes such as normal, impersonation attack, flooding attack and injection attack. The model achieves an accuracy of 98.4%, 98.3%, 73.1% and 99.9%, for 4 different classes respectively.

9) HTTP DATASET CSIC 2010 Dataset: In [478], a character level CNN based web application firewall is proposed which learns to differentiate between normal and malicious http requests. The model is trained using unicode encoded raw http requests and its performance is evaluated. The proposed model achieves an accuracy of 98.8% and its average processing time is 2.35ms.

10) CIDDS-001 Dataset: In [507], the effectiveness of LSTM for flow-based network IDS is studied. LSTM models of different combination of hyperparameters are tested using a flow-based network traffic dataset and its performance is evaluated. The experimental analysis compares the performance of proposed LSTM models with other traditional ML approaches. In [508], the effectiveness of various machine and DL models are studied for anomaly-based IDS. The study used various technique to fix the imbalanced dataset and trains DNN, VAE, random forest, voting, and stacking ML models.The experimental analysis founds that the proposed DNN model with down-sampling and class balancer achieves 99.99% accuracy and RF model works effectively even when large amount of the data is missing.

11) CTU-13 Dataset: In [511], a two level DL based adaptive anomaly detection system for 5G networks is proposed. The proposed model uses flow-based features from the network traffic data and trains DBN or SAE model in the first level. LSTM is used in the second. In [512], the authors have proposed a DL model for network anomaly detection which provides a MEC-oriented solution for 5G networks. The proposed approach extracts the flow based features automatically and learns to detect anomalous traffic in real-time.

12) Mixed Dataset: Deep Learning with ML Comparative study: In [431], a LSTM based host ID system is proposed. The LSTM network models the system calls to learn the semantic features and a new ensemble method is used to detect the anomalies. The proposed system is trained on 3 different datasets and it achieves high accuracy and low false alarm rate when compared to other related approaches. In [453], a novel CNN-LSTM based ID system is proposed. The CNN model learns the low level spatial features while the LSTM model learns the high level temporal features from the network traffic data. The proposed model is trained on two different dataset and it outperforms other related approaches. In [465],

a DL based solution for detection cyber attack in mobile cloud network is proposed. The proposed model is trained using NSL-KDD, KDDCup-99 and UNSW-NB15 datasets and it achieves an accuracy of 97.11% in best case. Its performance is compared with other traditional machine algorithms like SVM, DT, MLP, random forest and etc. In [476], a CNN-LSTM based ID system is studied. The CNN model learns the spatial features while the LSTM model learns the temporal features from flow features extracted from raw network traffic data. Optimal parameters for the model is found by using tree structured Parzen estimator. The study further investigate the impact of flow size and flow status interval on the performance of the detector. [487] proposed the LSTM architectures for cyber attack detection in fog-of-things environment. The architecture is scalable and also it can work in a distributed way to detect cyber-attacks. In [494], a distributed DBN and ensemble SVM based malicious behaviour detection framework in large scale network is proposed. The proposed distributed DBN is used for non-linear dimensionality reduction and apache spark based ensemble SVM model is used as detection classifier. The model is trained on 4 different data and it is compared with other related models. In [496], a DAEs and DNN based anomaly based IDS is proposed which is specifically designed for IICS. The proposed DAE model learns the normal behaviour of network and produces the optimal parameters which are used to effectively tune the parameters of DNN based classifier. In [497], the authors have proposed an online DBN based anomaly IDS which uses a new activation function. The proposed new fast adaptive linear activation function enhances the convergence speed of the proposed model and reduces the training time by 80% when compared to other activation functions such as $ReLU$, $tanh$ and $Sigmoid$. It also increases the accuracy of the model to 98.59% which is better than other state-of-the-art models. In [506], a DNN based scalable routing attack detection framework is proposed where the attack dataset is extracted from the Cooja IoT simulator. The proposed framework preprocess the extracted simulator data and selects the useful features which is fed into DNN model as input. In [516], a stack AE based network IDS is proposed where the model is capable of learning the important features from a large quantity of unlabeled data to classify them. The proposed model is trained using UNSW-NB15 and KDDCup-99 dataset and it achieves an accuracy of 89.134% and 99.996% for both datasets respectively which is better when compared to the other existing approaches. In [522], a DBN based IDS is proposed where a novel clustering algorithm is used to split the training data into several subsets. These subset of data are trained on several sub-DBN classifiers which reduces the feature dimension and classifies the data. The proposed model is trained on NSL-KDD and UNSW-NB15 datasets. Various DL architectures and tensor decomposition methods were evaluated for ID [604]. [614] proposed a DNN based architecture for host and network level ID. Various DL architectures such as LSTM, GRU, RNN, and IRNN were evaluated for on NSL-KDD dataset. Detailed experimental analysis were also shown on minimal feature sets and compared with the classical MLA. The DBN architecture was modeled for intrusion detection by [596]. The performance of DBN was evaluated on both the KDDCup-99 and NSL data sets. The results of DBN were compared with the MLP, LR, NB, KNN, DT, SB, RF, SVM and ELM. The DBN architecture outperformed the other models in all the experiments.

Deep Learning without ML Comparative study: In [435], a hybrid CNN and AEs based network ID system is proposed. The proposed system combines the advantages of both architecture to automatically extract and learn from raw network traffic and it achieves high accuracy. This proposed model has the potential to be used in large scale and real world scenarios. In [448], a SdA based network ID framework where the model is trained using session based features extracted from raw network traffic packets. The proposed model is evaluated using CTU-13 and ISCX-IDS-2012 dataset and its performance is evaluated. In [450], a DL based online anomaly-based IDS for FPGA hardware is proposed where the number of computation are reduced by utilizing dynamic fixed point arithmetic. The proposed system uses DBN model and it is evaluated on 2 different dataset. The experimental analysis shows that the proposed model achieves an accuracy of 94.6% on the NSL-KDD dataset and 95.1% on the HTTP DATASET CSIC 2010 dataset with the detection speed of just .008ms. In [459], a CNN based authentication system using mouse behaviour is proposed. The 2D-CNN model is trained on 2 publicly available datasets called Balabit and TWOS and its performance is compared with 1D-CNN and SVM models. The proposed model outperforms the other models and it achieves an average AUC of 0.96 in best case. In [460], the effectiveness of CNN based NIDS is studied and compared with other approaches like SAE and DBN models. The CNN model is trained using UNSW-NB15 and NSL-KDD datasets and its performance is evaluated. It can be observed that the CNN outperforms other models in normal class classification with an accuracy of almost 99%. In [495], a DBN based online anomaly detection system is proposed where FPGA is used for better power efficiency than CPU and GPU and for enhancing the inference speed to .008ms. The proposed model is trained on different datasets and it converges faster than other state-of-the-art DL models. it achieves an accuracy of 97.% in best case. In [518], the effectiveness of several DL models are studied for DDoS attack detection system where four DL models such as MLP, CNN, LSTM, CNN-LSTM are trained using CICIDS2017 datasets. The proposed model CNN-LSTM model performed better than other models and it achieves an accuracy of 97.16%. In [520], a CNN and RNN based network attack detection models are proposed. The network traffic payloads are preprossed into byte and character streams which are taken as input by CNN and RNN respectively. Both the models are trained using KDDCup-99 dataset and its performance are evaluated. It can be observed that the processing speed of CNN and RNN are 4.2 ms and 2.8 ms for a single sample. In [523], a DL based adaptive and scalable misuse IDS is proposed. The MAPE-K reference model and STL model are used to learn from reconstructed data and to create a self adaptive misuse IDS. The proposed model is trained on NSL-KDD and KDDCUUP 99 datasets and it is compared with a static IDS.

TABLE XI: A SHORT REVIEW ON DEEP LEARNING APPLICATIONS IN INTRUSION DETECTION.

Reference	**Architecture**	**Dataset**	**Compared CML**
[421]	RS	KDDCup-99	No
[422]	RS	KDDCup-99	No
[423]	DBN	KDDCup-99	Yes
[424]	RS	KDDCup-99	Yes
[425]	AE, DBN, & RBM	KDDCup-99	No
[426]	DNN	NSL-KDD	No
[427]	DBN	KDDCup-99	No
[428]	Sparse Autoencoder	NSL-KDD	No
[429]	Autoencoder	Private	No
[430]	DNN	NSL-KDD	Yes
[431]	RS	ADFA-LD, KDDCup-99, & UNM-lpr	Yes
[432]	RS	KDDCup-99	Yes
[433]	CNN, CNN-RS	KDDCup-99	No
[434]	CNN	NSL-KDD	Yes
[435]	Dilated CNN	CTU-UNB, & Contagio-CTU-UNB	No
[436]	Autoencoder	NSL-KDD	Yes
[437]	RS	Private	Yes
[438]	RS	KDDCup-99	No
[439]	RS	KDDCup-99	No
[440]	RS	NSL-KDD	No
[441]	RS	KDDCup-99	Yes
[442]	RS	NSL-KDD	Yes
[443]	DNN	KDDCup-99	Yes
[444]	RS	NSL-KDD	Yes
[445]	AutoEncoder	KDDCup-99	No
[446]	CNN	NSL-KDD	Yes
[447]	SAE	KDDCup-99	No
[448]	Autoencoder, DBN, & CNN	ISCX-IDS-2012, & CTU-13	No
[449]	DNN	NSL-KDD	Yes
[450]	DBN	NSL-KDD, & HTTP DATASET CSIC 2010	No
[451]	CNN	UNSW-NB15	No
[452]	FFN, & CNN	NSL-KDD	Yes
[453]	CNN, & RS	DARPA1998, & ISCX-IDS-2012	Yes
[454]	RS, & SVM	Kyoto	No
[455]	RS	ISCX-IDS-2012	No
[456]	RS	KDDCup-99	No
[457]	DNN, & RS	KDDCup-99	No
[458]	RS	Private	Yes
[459]	CNN	Balabit, & TWOS	No
[460]	DBN, SAE, & CNN	NSL-KDD, & UNSW-NB15	No
[461]	Bidirectional RS	UNSW-NB15	No
[462]	DNN	CICIDS2017	Yes
[463]	RS	KDDCup-99	Yes
[464]	DNN	NSL-KDD	No
[465]	RBM	KDDCup-99, NSL-KDD, & UNSW-NB15	Yes
[466]	Autoencoder	NSL-KDD	No
[467]	Autoencoder	NSL-KDD	Yes
[468]	RS	NSL-KDD	No

[469]	CNN, & RS	KDDCup-99	No
[470]	SAE, & DNN	AWID	No
[471]	CNN	NSL-KDD	Yes
[472]	DNN	NSL-KDD	Yes
[473]	CNN, & RS	DARPA1998, & ISCX-IDS-2012	Yes
[474]	DNN	KDDCup-99	Yes
[475]	CNN	UNSW-NB15	No
[476]	CNN-RS	ISCX-IDS-2012, & CICIDS2017	Yes
[477]	CNN	NSL-KDD	No
[478]	CNN	HTTP DATASET CSIC 2010	Yes
[479]	RS	CICIDS2017	Yes
[480]	Autoencoder, & DNN	NSL-KDD	No
[481]	Autoencoder	Private	No
[482]	Autoencoder	KDDCup-99	No
[483]	CNN	NSL-KDD	No
[484]	Autoencoder, & RS	NSL-KDD	No
[485]	RS	NSL-KDD	Yes
[486]	Autoencoder	NSL-KDD	No
[487]	RS	ISCX-IDS-2012, & AWID	Yes
[488]	RS	NSL-KDD	Yes
[489]	Autoencoder	NSL-KDD	Yes
[596]	DBN	KDDCup 99, & NSL-KDD	Yes
[490]	DBN	NSL-KDD	No
[491]	SAE	Private	Yes
[492]	CNN	Private	No
[493]	DBN	NSL-KDD	No
[494]	DBN	KDDCup-99, NSL-KDD, UNSW-NB15, & CICIDS 2017	Yes
[495]	DBN	HTTP DATASET CSIC 2010, KDDCup-99, & NSL-KDD	No
[496]	Autoencoder	NSL-KDD, & UNSW-NB15	Yes
[497]	DBN, & Autoencoder	HTTP DATASET CSIC 2010, KDDCUP'99, NSL-KDD, & Kyoto	Yes
[498]	CNN	ISCX-IDS-2012	Yes
[499]	DNN	ISCX-IDS-2012	No
[500]	SAE	ISCX-IDS-2012	No
[501]	SdA,	Private	No
[502]	RS	KDDCup-99	No
[503]	RBM	NSL-KDD	Yes
[504]	DNN	NSL-KDD	No
[505]	DNN	ISCX-IDS-2012	No
[506]	DNN	Real-time traffic, KDDCup-99, & UNSW-NB15	Yes
[507]	RS	CIDDS-001	Yes
[508]	DNN	CIDDS-001	Yes
[509]	Sparse Autoencoder	NSL-KDD	Yes
[510]	DNN	KDDCup-99	Yes
[511]	DBN, SAE, & RS	CTU-13	No
[512]	DNN, & RS	CTU-13	No
[513]	DAE	UNSW-NB15	Yes
[514]	NN	Private	Yes
[516]	Autoencoder, & DNN	KDDCup-99, & UNSW-NB15	Yes
[517]	CNN	Private	No
[518]	CNN, & RS	CICIDS2017	No

[520]	CNN, & RS	CNTC-2017 Webshell, HTTP DATASET CSIC 2010, & DARPA1998-all-attacks	No
[521]	Autoencoder	NSL-KDD	No
[522]	DBN	NSL-KDD, & UNSW-NB15	Yes
[523]	Autoencoder	KDDCup-99, & NSL-KDD	No
[604]	DNN	KDDCup 99, NSL-KDD, UNSW-NB15, WSN-DS, & CICIDS 2017	Yes
[614]	DNN, RS, & CNN	KDDCup 99, & Kyoto	Yes
[608]	RS	NSL-KDD	Yes
[625]	RS	NSL-KDD	Yes

TABLE XII: A short review on on deep learning based DGA, malicious URL, phishing URL, spam email, phishing email and image spam detection.

References	**Architecture**	**Task**	**Data set**	**Text representation**	**Compared CML**
[537]	DBN	Email	LingSpam, Enron, & SpamAssassin	Term frequency	Yes
[538]	RS	Domain name	Publically available sources	Keras embedding	Yes
[539]	CNN	URL	Private	One-hot	Yes
[540]	CNN	URL	Private	Word2vec, & TF-IDF	Yes
[541]	RS	Domain name	DGArchive, & publically available sources	One-hot	No
[542]	AlexNet, VGG16, VGG19, SqueezeNet, Inception-BN-21k, Inception-BN-1k, Inception V4, & ResidualNet152	Domain name	Publically available sources, & samples collected from real-time system	Characters are converted into images	No
[543]	AlexNet, VGGNet, & GoogleNet	CAPTCHA	Private	-	No
[544]	DNN, & CNN	CAPTCHA	Publically available sources	-	No
[545]	CNN	URL	Publically available sources	One-hot	Yes
[546]	CNN	Domain name, & URL	Publically available sources	One-hot	No
[547]	CNN, & RS	Domain name	DGArchive, publically available sources, & real-time data set	Keras embedding	Yes
[548]	CNN, & RS	Domain name	Publically available sources	Keras embedding	Yes
[549]	CNN	CAPTCHA	Publically available sources	-	Yes

[550]	RS	Domain name	Private	One-hot	No
[551]	CNN, & RS	Domain name	Publically available sources	Keras embedding	No
[593]	RS, CNN, & CNN-RS	URL	Publically avilable sources	Keras embedding	Yes
[601]	CNN, & CNN-RS	URL	Publically avilable sources	Keras embedding	Yes
[552]	VAE	YouTube video address	Private	Sent2vec embedding	Yes
[553]	DNN	Web page	Private	bag-of-words	Yes
[554]	CNN, & RS	Email	IWSPA-2018	Word2vec, & Neural Bag-of-ngrams	No
[555]	FastText	Email	IWSPA-2018	FastText	No
[556]	Various deep learning architectures	Email	IWSPA-2018	Various text representations	Yes
[557]	CNN	URL	Private	Keras embedding, & Bag-of-words	Yes
[558]	DBN	URL	Publically available sources	Manual feature engineering	Yes
[559]	RS	URL	Publically available sources	One-hot	Yes
[560]	RS	URL	Publically available sources	One-hot	Yes
[561]	RS	URL	Publically available sources	One-hot	Yes
[562]	CNN	Image spam	Image Spam Hunter	-	No
[563]	RS, CNN & CNN-RS	Domain name	Publically available sources	Keras embedding	Yes
[564]	RS	Domain name	Publically available sources	Keras embedding	Yes
[565]	CNN, & RS	Domain name	Publically available sources, & real-time data set	Keras embedding	Yes
[566]	CNN, & RS	Domain name	Publically available sources	Keras embedding	Yes
[567]	CNN, & RS	Domain name	Publically available sources	Keras embedding, & one-hot encoding	Yes
[568]	RS	Domain name	DGArchive, & Publically available sources	TF-IDF & Keras embedding	Yes
[569]	CNN, & RS	Domain name	Publically available sources	Keras embedding	Yes
[570]	RS	Domain name	Publically available sources	Keras embedding	No
[571]	CNN, & RS	Domain name	Publically available sources	Keras embedding	No
[572]	CNN, & RS CNN-RS	Domain name	Publically available sources	Keras embedding	Yes
[573]	DNN	Domain name & URL	Publically available sources	n-gram & feature engineering	Yes
[574]	BLSTM	Network flow	Private	Keras embedding	No
[575]	CNN	Network flow	CTU-13 Dataset & real-time data set	Traffic to image representation	Yes
[576]	BLSTM	Network flow	Private	Keras embedding	No

[577]	DNN	Network flow	HogZilla dataset, CTU-13 dataset, & ISCX-IDS-2012 dataset	-	No
[578]	DNN	URL	Private	Bag-of-words	No
[579]	DNN	URL	Publically available sources	Manual feature engineering	No
[580]	RS	Domain name	Publically available sources	Keras embedding	No
[581]	DNN (Invincea, Endgme, NYU, CMU, & MIT)	Domain name	AmritaDGA	ASCII representation	Yes
[582]	CNN, CNN-NB, & CNN-XGB	Domain name	AmritaDGA	Keras embedding	Yes
[583]	RS & Bidirectional RS	Domain name	AmritaDGA	Keras embedding	No
[584]	Bidirectional RS	Domain name	AmritaDGA	Keras embedding	No
[585]	DNN	Domain name	AmritaDGA	n-gram	No
[591]	CNN	Email	TREC 2007, & SpamAssassin	Keras embedding	Yes
[592]	CNN	Email	IWSPA-2018	Keras embedding	No
[586]	CNN, & RS	Domain name	Publically available sources	Keras embedding	Yes
[587]	RS	Domain name	Publically available sources	Keras embedding	Yes
[588]	DNN, & RS	Domain name	Publically available sources	Keras embedding	Yes
[589]	CNN, & RS	Domain name & URL	DGArchive, & publically available sources	Keras embedding	Yes
[590]	CNN, & RS	URL	Private	Keras embedding	Yes
[603]	CNN, RS & CNN-RS	Domain name	AmritaDGA	Keras embedding	No
[605]	Endgame, NYU, MIT, CMU, Invincea, DNN, & CNN-RS	Domain name	Publically avilable sources, & real-time dataset	Keras embedding	Yes
[606]	RS, CNN-RS, & Bidirectional RS	Domain name	AmritaDGA	Keras embedding	Yes
[612]	RS, CNN, & CNN-RS	Domain name	AmritaDGA	Keras embedding	No
[609]	AE & CNN	Domain name	AmritaDGA	Keras embedding	No
[607]	CNN, RS, & CNN-RS	Email & URL	Publically avilable sources	Keras embedding	Yes
[610]	DNN, RS, CNN, & CNN-RS	URL	Publically avilable sources	Keras embedding	Yes
[604]	RS & CNN-RS	URL	Publically avilable sources	Keras embedding	Yes
[622]	RS	Domain name	AmritaDGA	Keras embedding	No

B. Deep Learning in Developing Cyber Threat Situational Awareness using DGA, URL, Email and Security log Data analysis

The traditional antimalware systems may not be able to identify malicious activities quickly. This is primarily due to the reason that these methods consume a lot of time for reverse engineering and there is a chance that a significant amount of damage might have happened. However, it may be detected faster through cyber threat situational awareness data analysis. Situational awareness data are DNS, email, URL and social media data. Timely collection of data from various sources and analysis enable to detect malware quickly. The concepts behinds these various data sources are discussed below.

Domain Name System (DNS): is one of the main Internet protocols. Individuals all around the globe usually access Internet through a browser works by rendering the web pages and portals. First the domain name of the web page is typed by the users in the browser's address bar. Then, Internet assists the users in information exchange. DNS servers can be distinguished into two broad categories: Recursive servers and Non-recursive/Iterative servers. Non-recursive DNS servers basically work as the Start of Authority (SOA), replying to the queries which are inside their governed/local domain only without worying about the queries of other DNS servers regardless if they can cater to the requested answer or not. On the other hand, Recursive DNS servers reply to the queries of not only local domain but also all types of domains by sending the queries to other servers and then sending back the response to the user. Some of the most serious attacks on the Recursive DNS servers are root name server performance degradation, DNS cache poisoning, Distributed Denial of Service (DDoS) attacks, unauthorized use of resources. As DNS protocol was not basically created with security issues in mind and has vulnerabilities, the large expanse of event data produced by these systems can be used to create situational awareness about Cyber threat. Earlier day's adversary embeds malware with fixed domain name and IP address. This can be detected by using blacklisting methods. In order to bypass the blacklisting method, adversary uses the concept of fluxing. There are two types of fluxing; they are domain and IP fluxing. Fluxing means an adversary constantly changes the IP address and domain name. Most commonly used method for domain fluxing is domain generation algorithms (DGAs).

Domain generation algorithms (DGAs): The Domain generation algorithm (DGA) facilitates the generation of large set of domain names using a seed value which is known to the attacker. The attacker uses the known seed value to generate same set of domains and register one of the many generated domains and deploy the C&C server. The DGAs are broadly classified into 2 types. One is binary-based DGAs which are embedded in the malware binary and triggered after the installation of the malware. The second type is script-based DGAs which are embedded in the Javascript and triggered when the user opens a malicious website. The flow diagram of domain flux attack is shown in Figure 10. As shown in Figure 10, an infected system attempts to access many domains in an attempt to contact the command and control server. It contacts three domains, abc.com, xyz.com and secure123.com. Both abc.com and xyz.com are not registered and an infected system receives an NXDOMAIN response from DNS server. The third domain is an active and registered domain. Hence the DNS server uses this domain to call C&C server.

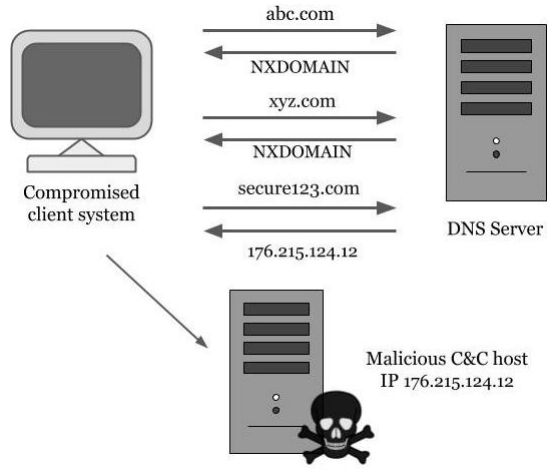


Fig. 10. Active domain name discovery process employed by recent malwares.

Botnets are networks formed by devices that are compromised by malware. It can be controlled remotely by the bot master using the command and control (C&C) channel [243]. A compromised device in a network is called bot and the bot master uses these bots to conduct various illegal activities like DDoS [243], phishing, identity theft, malware distribution, etc. Even though the structure and size varies, they have same stages of lifecycle [243]. The C&C server is used by the botmaster to issue commands to the botnets based on which the botnets perform their assigned tasks and sends back the results. Based on the command and control communication channel, the botnet are grouped into Internet Relay Chat (IRC) botnet, Hyper Text Transfer Protocol (HTTP) botnet, Peer to Peer (P2P) botnet, and Hybrid botnet where IRC botnet uses centralized architecture, P2P uses distributed architecture. Hybrid architecture is a hybrid of centralized and distributed architecture and detection of botnet which uses hybrid architecture is often difficult compared to centralized and distributed architecture.

Spam and Phishing Email Detection: The most popular form of spam being email spam commonly referred to as junk mail'. The spammers or cybercriminals send us these spam emails in mass amount, either to make money from the small percentage of recipients that actually respond to such emails or to carry out phishing scams to obtain passwords, credit card numbers, bank account details and more or maybe to simply infect the recipient's computer with malicious code. Spam emails are usually used for commercial purposes. Phishing is another online scam where cybercriminals send emails asking for sensitive information. These mails are made in such as way that they appear to be from a legitimate company. In recent days, this phishing email has become one of the major issue of Internet not only resulting in annoying individual users but also creating great financial losses for organizations. These mails usually consist of links which will direct to a

website appearing like the company's website to fill in the information but the information provided by you us misused by the criminals since that link will directly take you to the fake website. Phishing mail are basically a form of spam email but is more manipulative and causes more harm since it tries to extract the confidential information from the user and carry out fraudulent activities. This particular type of spam employs two techniques: deceptive phishing and malware-based phishing. The first category uses social engineering scheme, which generates a spam mail which fake the legitimate company or a bank such that the victim is redirected to a fake website to trick the victim to obtain financial data [228].

Uniform resource locator (URL): A universal address of documents and other resources on the World Wide Web is called as URL, which plays the important role of locating documents and other web resources that are available online, and find a method for accessing it via web browser. It has a linear structure which is shown in figure 11and it generally consists of some of the following:

- Scheme name: This finds the protocol that must be used to access the required resource on the web. The most commonly used protocol are ftp, http, https, and mailto.
- Host name: The hostname distinguishes the host where the asset is found. A hostname is a space name allocated to a host PC. This is typically a blend of the host's neighbourhood name with its parent space's name. For instance, www.google.com comprises of host's machine name www and the area name google.com.
- Port Number: Servers regularly convey in excess of one kind of administration, so you should likewise tell the server what benefit is being asked. These solicitations are made by port number. Understood port numbers for an administration are ordinarily discarded from the URL. For instance, web benefit HTTP is normally conveyed on port 80.
- Path: This distinguishes the particular asset inside the host that the client needs to get to. For instance,/html/html-url.php

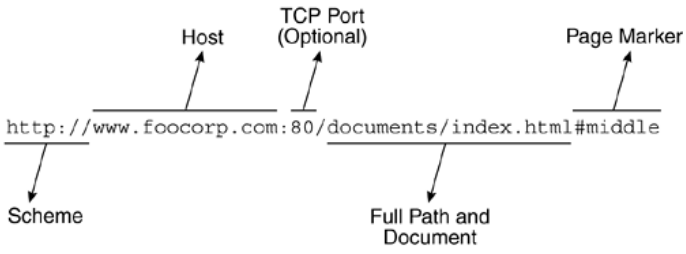


Fig. 11. Structure of URL.

Internet Protocol (IP) is a system convention that makes it feasible for a host to communicate with another host on the Internet, paying little mind to the fundamental systems administration equipment. The key rule behind IP is that every host that connects to the Internet is assigned a unique logical address which is used for identification of host and location addressing.

As Internet grows, the URL has become one of the most commonly used tool to host malicious contents by an adversary and the problem of protection against those malicious contents has come to the forefront in recent years. Many open-source and commercial products are there for the same. The traditional methods of blacklisting and filtering are simple but not scalable, though some advanced methods using fuzzy matching techniques exist. Other approaches try to use ML techniques by extracting features from URL strings.

Completely Automated Public Turing Test to Tell Computers and Humans Apart (CAPTCHA): It is a technique used to identify whether the user is a human or a bot. The user has to pass the CAPTCHA test which is a challenge thrown to him to prove that the user is human. It consists of an image with distorted and obscured letters in the foreground, noisy and graphical background. It is an active security technique to prevent the application from DDoS, attacks. It is simple, easy to implement and have many significant advantages and generally it is the first line of defense. And it can be mainly designed in a way that it is easy for a human to identify the letters and numbers present in it but very difficult for optical character recognition (OCR) software and any other automated text recognition software and it very difficult to solve buy present advanced AI. The reason is it uses three techniques segmentation, invariant recognition and parsing which can be easily done by a human user but the difficult and challenging task for a computer program the human brain performs the contextual dynamic thinking by mapping the contours of the individual segmentation. Segmentation is the separation of the letters and numbers individually text and by reducing the space and baffling each other. Invariant recognition deals with the human eye can identify infinite variations in the shapes of the letters but a computer has to train explicitly to handle all the spatial variations of the letters. Parsing is the ability to identify the letters holistically based on the context is also important when solving a CAPTCHA. Complete context has to be taken into account and any letter should not be identified as the wrong letter. The types of CAPTCHA's are given below

- Text-based
- Image based
- Video based
- Audio based
- Puzzle based

A short review on DL applications in DGA, URL, Email and security log data analysis is reported in Table XII.

1) Email: In [537], the authors have proposed a greedy layered DBN based approach for spam detection. They have trained the proposed model on three different datasets and they have found that the proposed model outperforms the SVM based approach for all three datasets. In [554], various DL architectures like CNN, RNN, LSTM, and MLP are studied for phishing email detection. The experiment uses world embedding and neural bag of n-grams with DL models to obtain semantic and syntactic email similarity. The experiment results shows that word embedding with LSTM approach performs well with accuracy of 99.1 and 97.1 in two different task in anti-phishing shared task corpus at IWSPA-AP 20181. In [555], a DL based anti-phishing system is proposed where a distributed representation method is used to differentiate phishing and legitimate emails. The proposed approach uses

word embedding and neural bag-of-ngrams in order to extract semantic and syntactic features from the email data and it achieves f1 score of 99% in the best case. In [556], A summary of anti-phishing shared task at ACM IWSPA 2018. There were two subtask is given. First one is to identity phishing emails from a collection of emails and the second task is to separate phishing emails from the collection given header and body information. Two new metrics such as balanced and normalized balanced detection rate are used for performance evaluation of all the teams. DL models and logistic regression performed well in first sub task but when header information is given in second sub task, Multinomial NB outperformed the other models. In [562], a DL based image spam detection system is proposed where CNN is trained using dataset containing 1738 images. The proposed model automatically extracts the useful features and performs binary classification. The results show that the proposed method outperforms other ML techniques with an accuracy of 91.7%. In [591], two CNN models are studied for email spam classification. Both character level embedding and word embedding based models performs better than baseline SVM based classifier. It can be observed that character level linear SVM based approach got almost same accuracy when compared with both CNN models. In [592], a CNN based phishing e-mail detection system is proposed. Dataset containing email with a without header are taken and keras word embedding is applied. The proposed CNN model learns to classify legitimate and phishing mails and achieves good accuracy even though the used dataset is highly imbalanced.

2) DGA: Recurrent Structures: In [538], a LSTM based real-time DGA domain detection model is proposed. The model is evaluated on open datasets and it better than other state-of-the-art approaches. It achieves 90% detection rate with AUC of 0.9993 for binary classification. In [541], a RNN based DGA domains detection framework is proposed which is based on domain names as input with no external resources or human intervention. The proposed framework achieves detection accuracy of 93% and F1 score of 93% with very low false positive rate. In [550], a DL based approach to address the problem of time series deinterleaving. The proposed approach generates synthetic dataset and evaluates various inference strategies for the problem using AHMM and LSTM. The experimental results show that the LSTM method outperforms the model based on AHMM. In [564], the effectiveness of various DL based approaches are studied for scalable DGA detection which uses DNS logs. DL models such as RNN and LSTM extracts useful features from DNS log data and its classification performance is evaluated. The results show that DL models perform well when compared to ML models and LSTM achieves the highest detection accuracy. In [568], a RNN based robust DGA detection system which takes advantage of additional WHOIS information if available to enhance the performance of the system and to classify much more difficult DGA families. A novel measure called smashword score is also proposed which ranks DGAs based on how close the domains resemble English words. In [570], a DL based DGA detection engine is proposed where 1D-CNN model is to extract the important features from a large dataset containing URLs from 51 DGA malware families. The results show that the proposed approach achieves an accuracy of 97% with 0.7% FAR. In [580], a LSTM based anti-DGA system is proposed where the model uses attention mechanism in order to give more focus on important substrings in the domain names which enhances the overall performance of the system. The results shows that the proposed model achieves a FAR of 1.29%. In [587], two state-of-the-art DGA detection approaches are compared and dangers of manual feature engineering are discussed. Random forest based FANCI approach is compared with LSTM based approach and it is found that the DL model achieves an accuracy of 98.7% while RF method achieves an accuracy of 93.8%. This study also created a new DGA based on feature set used in FANCI which reduced the accuracy of RF and LSTM based model to 59.9% and 85.5%. In [622] proposed LSTM architecture for DGA detection and as well as classification. This has performed well in both the tasks.In [609], The authors extracts the hidden layer features and fed it as input to many classical ML models for further learning. This type of learning approach can be called as transfer learning.

Convolutional Neural Network: In [542], the authors have presented various DGA domain detection approaches based on pre-trained ImageNet models like VGG, Res Net, Inception, Squeeze Net, and Alex Net. The models are evaluated using real-world malware dataset containing 34,000,000 unique malware samples and the best model achieves a true positive rate 99.86. In [575], a CNN based botnet detection framework for IoT and wearable devices is proposed where the model is trained using CTU-13 Dataset. The network traffic data is converted into image format and fed into CNN model. The experiment analysis shows that the proposed system using transfer learning enhances the accuracy up to 99.98% while the SVM and logistic regression based approaches achieves an accuracy of 83.15% and 78.56% respectively. In [582], the authors have presented a transfer learning technique by combining CNN with NB classifier for DGA analysis. They have used CNN-XGB ensemble for multi-class classification. The performance of CNN-NB ensemble is then compared with the performance of other classifiers such as NB, IRNN, Random forest, CNN and Bigram-LR.

Mixed: In [547], a character level DL approach for DGA detection and filtering framework is proposed where the model is trained using a large set of real network traffic data. The DNS resolved domains are test against classifier to predict if they are benign or malicious. The framework is capable of filtering and blocking potentially malicious domains from communicating to C&C server. The proposed system achieves a false positive rate of 0.01%. In [548], various manual and automatic feature extraction based supervised learning algorithms such as SVM, decision tree, ELM, HMM, LSTM, recurrent SVM, CNN-LSTM and bi-directional LSTM are studied for DGA detection. All the models are trained using 1 million alexa benign domains and OSINT DGA feed for evaluation of the performance. The result shows that the recurrent SVM and bi-directional LSTM method achieves high detection accuracy for both binary and multi-class classification. In [551], the authors have studied five different DL architectures

based on RNN, CNN or a combination of both in order to differentiate between malicious and benign domain names. It can be observed that there is very little difference in the performance of all the 5 DL architecture but they all easily outperforms random forest model which is based on human engineered features. In [565], a DL based scalable DGA detection framework which works at ISP level is proposed. The framework uses event data from the DNS to detect DGA and provides situational awareness. In [566], a novel CNN-LSTM based approach is proposed for DGA and malicious URL detection which incorporates NLP technique for enhancing the performance. The proposed system is compared with traditional ML classifiers which uses bi-gram feature representation and character-level CNN model. The results show that the proposed system achieves an accuracy of 99% for malicious URL detection and 98.3% for DGA detection. In [567], the authors have studies the effectiveness of five character-level RNN and CNN based approaches for DGA detection. They have found that there is only little differences among the performance of the proposed models while they all comfortably outperform random forest based approach. All the proposed models achieves 97-98% of detection accuracy with very low FAR of 0.001. [569] proposes a cost-sensitive method to handle multiclass imbalanced DGA classification. A detailed analysis of various DL architectures performances were evaluated for DGA classification with the data set split including time and seed [571]. In [586], the authors have proposed a scalable, distributed and unified framework which provides the situational awareness and decision capabilities required to take active measures to handle complex threats. The proposed framework groups several Cyber Security sub systems as a single unified security solution. The experimental analysis compares the proposed DL based system with traditional ML based approaches. In [588], a machine and DL based DGA detection models are proposed where the models are trained using real-time traffic data collected for a year. A two-level model and DNN model is proposed. The two-level model classifies the DGAs and clusters the domains into groups based on different DGAs. The experimental analysis shows that the optimized DNN model performs better with an accuracy of 97.79% while the other model obtains an accuracy of 97.79%.

Hybrid Recurrent Structures: DL architecture based on CNN and CNN-LSTM was proposed for malicious URL detection by [601]. This was compared with the classical method LR with bigram text representation method. The DL architectures outperformed the classical method in all different types of experiments. In [563], various character and bigram level DL based approaches are studied for DGA detection. RNN, IRNN, LSTM, CNN and CNN-LSTM are trained on domains from OpenDNS, Alexa, and a corpus of domains from by 17 DGA malware families. In the experimental analysis, the proposed models are evaluated using OSNIT dataset and it is found that the LSTM and CNN-LSTM achieves the highest detection rate of 99.45% and 98.79% respectively. In [572], two RNN and CNN model based DGA botnet detection system is proposed where system detects the malicious domains using DNS traffic data and gives information about the infected host and C&C domain. The results shows that CNN-LSTM achieves highest accuracy of 98.7%. Various well-known character based short text classification were modeled for DGA detection and classification by [605]. The experiments of these models were conducted on various datasets. The performances obtained by all these architectures are closer. In [612] proposed a cost-sensitive DL architecture to handle imbalanced problem in DGA multiclass classification.

Deep Neural Network: [573] proposes a DL based method for DGA analysis. To find out the generalized DL architectures, the experiments are conducted on various other Cyber Security applications. In [577], a multi-layered neural network based botnet detection system in SDN environment is proposed where the model is trained using HogZilla dataset. The proposed approach performs feature selection and filtering on the dataset to make it realistic for SDN scenario. The experimental analysis shows that the proposed approach achieves 96% accuracy. In [581], the authors have studied the performance of featureless DNN classifiers and Random forest classifier for detection and categorization of domain names generated by DGAs. In [585], the authors propose a DNN model for detection and categorization of DGA domains. The DNN model takes 3-gram representation of domain names as input and has 5 hidden layers.

Bi-directional Recurrent Structures: In [574], a bi-directional LSTM based botnet detection engine is proposed where word embedding is used for conversion of network traffic packets into tokenized integer values. The proposed model is compared with simple LSTM approach and the results shows that both achieves good accuracy and performed well for mirai, udp, and dns attacks. In [576], a bi-directional LSTM based botnet detection system is proposed where the model is trained using a generated labeled dataset which contains botnet activities and DDoS attacks. The detection model extracts the useful features using word embedding and learns to detect model and inform the user in case if there is an infection. The experimental analysis shows that the proposed model works well for mirai, udp, and dns attacks but less favorable for ack attack. In [583], the authors have studied the performance of two DL architectures such as LSTM and Bidirectional LSTM for binary and multi-class classification of domain names. It is observed from their result that their binary classification model performed better that multi-class classification model. Similar to [583], [584] proposes a system to detect DGA domains using Bidirectional LSTM and character embedding. In [603] proposed a data set called as AmritaDGA for DGA analysis. Various DL architectures were proposed for DGA detection and classification. This was called as AmritaDeepDGA. Both AmritaDGA and AmritaDeepDGA have been made publically available for further research. Various DL architectures such as RNN, LSTM, GRU, CNN-LSTM, BRNN and BLSTM were employed for DGA detection and classification using AmritaDGA data set [606].

3) URL: Recurrent Structures: In [559], a LSTM based scalable phishing URL detection approach is proposed and compared with random forest based approach. The proposed method extracts the useful features from a large dataset of URLs automatically whereas the RF approach uses features

extracted from lexical and statistical analysis. The experimental analysis shows that the LSTM and RF approach achieves an accuracy of 98.7% and 93.5% respectively. In [560], the authors have proposed a LSTM based phishing attack technique which is capable of evading the detection systems. The proposed model learns the intrinsic patterns and generates synthetic malicious URLS that can bypass the detection framework with high success rate. In [561], a GRU based malicious URL detection technique is proposed and compared with random forest based approach. The proposed method automatically extracts the useful features from a large dataset of containing 240000 URLs of six types whereas the RF approach uses features extracted from lexical and statistical analysis. The experimental analysis found that the GRU achieves 98.5% accuracy and consistently outperformed the RF model which is trained using well-selected features.

Convolutional Neural Network: In [539], CNN architecture is used to learn to extract features from a short character string input which could be malicious URLs, registry keys, file paths, named mutexes, and named pipes. The proposed model outperforms other related models and achieves very low false positive rate. In [540], the authors have proposed a CNN based approach for the prediction of malicious URLs. The model is evacuated using 75,643 malicious URLs and 344,821 benign URLs and it achieves more than 96% of accuracy. The proposed model performs better than other models based on SVM and linear regression. In [545], the authors have proposed an event de-noising CNN based malicious URL redirection sequence detection system which uses proxy logs to extract the sequences. They have compared the proposed method with simple malicious URL detection and CNN based approaches and found that the proposed approach performs well with very low FAR. In [546], a character-level CNN based malicious URL and DNS detection framework is proposed where NLP methods are used to map the URL and DNS strings into vector. The CNN model extracts features automatically and learns to classify. The proposed model is trained using real-world dataset and it outperforms other baseline approaches in terms of scalability and efficiency. In [557], a character and word level CNN based malicious URL detection framework is proposed where the model learns to detect malicious URL by extracting semantic features from the URL data. The proposed method is test using large-scale dataset and it performs better than the other related approaches.

Mixed: In [590], a character-level convolutional GRU model based malicious URL detection system is proposed where feature representation technique of URLs is based on malicious keywords. The GRU model is used for feature extraction and 407, 212 URLs are used for training process. The proposed approach achieves an accuracy of 99.6%.

Hybrid Recurrent Structures: The performance evaluations of various DL architectures such as LSTM, RNN, I-RNN, GRU, CNN, CNN-LSTM are evaluated for malicious URL detection [593]. This was compared with various classical ML classifiers such as RF, DT, MT, AB, and NB. To convert characters in malicious URL, Keras embedding was employed and additionally the bi-gram text representation was used with RF for performance comparison. Overall, the DL architectures performed well in compared to other methods. A unified DL architecture was proposed for email and URL data analysis by [607]. The importance random split and time split method for dividing the data into train, valid and test datasets briefly discussed by [610]. They have done various experiments on both random and time splitting in malicious URL detection. Various DL architectures were evaluated for malicious URL detection [604].

Deep Neural Network: In [553], a DL based malicious URL detection framework based on features extracted from static HTML files. The proposed framework uses regular expressions to extract features and uses spatial information to yield high accuracy of 97.5% with very low false positive rate. In [578], the authors have proposed a deep neural network based framework for classifying normal and malicious URL where byte value are extracted from URL to construct a URL vector. The proposed model is trained using real-life datasets obtained from phishtank.com and from a private research organization and it achieves an accuracy of 94.18%. In [579], two DL based phishing URL detection system is proposed where ANN, DNN and few ML models are trained using 73575 URLs to differentiate normal and phishing URL. The experimental analysis shows that the ANN and DNN approaches performed better than ML classifiers with an accuracy of 92% and 96% respectively.

Autoencoder and DBN: [552] proposes a VAE based method was proposed for clickbait problem in Youtube videos. In [558], the authors have proposed a DL based malicious URL detection system which uses greedy multi-layered DBN for extracting the useful features aromatically and DNN for classification. They have trained the proposed model using 27,700 URLs and they have found that it achieves better results with very low false positive rate.

4) CAPTCHA: In [543], a text-based CAPTCHA technique with a modal completion is proposed and its robustness against DL CAPTCHA solver is analyzed. The proposed approach enhances the CAPTCHA using after effects and it uses CNN based AlexNet. The experimental analysis found the DL based solver takes more time solve as it is hard to emulate the a modal completion. In [544], a CNN based CAPTCHA solving technique is proposed where character localization and recognition methods are used to solve text based CAPTCHAs. The proposed method is capable of breaking 11 CAPTCHA schemes with accuracy more than 50%. The experimental analysis compares MLP, SLP and CNN based solvers and found that CNN based approach performs well. In [549], a DL based CAPTCHA breaker is proposed where CNN architecture is used to solve letter-based CAPTCHAs. While the proposed method works well for classification of single letter CAPTCHA, it does not generalize for multiple letter ones.

C. Deep Learning in Network Traffic Analysis

The scale and the thickness of system movement are developing step by step. The sorts of convention are more. Distinguishing each stream of information is an important issue both in big business system and web. Port based, signature

based and factual highlights based distinguishing pieces of proof are the standard methodologies. One of the soonest strategies is working based on uncommon or predefined ports. For instance, standard HTTP port is 80. The default port of SSL is 443. Be that as it may, as an ever increasing number of new conventions don't take after manage of port registration; the mistake rate is becoming higher. Signature based activity distinguishing proof has been utilized after the year 2002. A signature is a portion of payload information that is static and recognizable for applications, which can be portrayed as a grouping of strings or hex qualities. The mistake rate is hypothetically lower than 10%. Signature-based technique is basic and its effectiveness generally high, so most frameworks of convention ID receive it. Be that as it may, when a convention determination changes or another convention produces, individuals must begin once again to find profitable signatures. It will be extremely tedious and work escalated. As of late, an approach of programmed grouping based on factual highlights and ML is exceptionally mainstream. This approach relies upon the highlights of activity transmission, for example, the time interim between bundles, parcel estimate, rehashing example, etc. At that point the highlights are sustained into different customary ML algorithms. A short review on DL applications in network traffic analysis is reported in Table XIII.

1) Deep Learning solutions with ML Comparative study: DL based approach, Deep Packet, was proposed in [526] for classifying encrypted traffic. As this model is based on DL it also has the capability of feature extraction. This model have two functionalities namely, traffic characterization and application identification. Traffic characterization is that this classifier can classify encrypted traffic into major classes like P2P, FTP etc. Application identification is a process through which applications like Skype and BitTorrent from end user point can be identified. A DL based multitask architecture for forecasting mobile Internet traffic was proposed in [529]. Various experiments with architectures of RNN, 3D CNN and combination of RNN and CNN were performed in this study. The experiments showed that geographical and temporal traffic features were extracted by CNN-RNN architecture.Software defined networking architecture along with DL technology was utilized to classify network applications in [531]. SDN controller of the proposed model takes advantage of the powerful computing capability and logical centralized control to collect and process the massive traffic network data easily. The hybrid DL network which is a combination of softmax regression layer and SAE is trained using the data from SDN controller. This model is effective as it achieves high accuracy for classification compared SVM classifier. Different DL techniques to classify mobile network encrypted traffic were compared in [532]. These DL techniques were first reproduced then were dissected and finally were set into a framework so hat comparison can be performed. Three different datasets collected from real-time activities of human users were utilized for this work. Various recurrent structures such as RNN and LSTM were evaluated for identifying SSH traffic and as well as application classification in both SSH and Non-SSH traffic [594]. For comparison purpose, the other classifiers such as RF, AB, DT, KNN, NB, and SVM with linear and Non-linear were used. To find out the optimal method, these methods are evaluated on 4 different types of publically available network traffic data sets. The DL deep architectures outperformed other classical classifiers. However, the proposed method relies on feature engineering. This can be avoided by passing the entire payload and input to the DL architectures. Various DL architectures such as RNN, LSTM, GRU, IRNN, CNN and CNN-LSTM were evaluated for SSH traffic identification and as well as SSH application classification [597]. These architectures are evaluated on publically available data sets. The results are compared with the shallow models, ELM and MLP. The deep model performed well compared to shallow models in all the experiments.

2) Deep Learning solutions without ML Comparative study: A DL framework was proposed in [524] to identify network traffic. DL framework has the capability to automatically learn features. This works also discusses about the applications of DL & ANN frameworks to identify network traffic. Experiments were performed utilizing real-time data and outcomes show that this proposed network works well on application like protocol classification, unknown protocol identification, and anomalous protocol detection. In [525], four different ML and DL algorithms which predict the network traffic were compared. This work shows that MLP and RNN performed better than SAE (complex models). DL based approach for Internet communication traffic classification was proposed in [527]. This work also investigated the viabilities of utilizing DL based models for classifying network traffic to detecting malicious traffic as well as manage network applications. The outcome of the experiments done in this work shows that utilizing the initial 50 bytes of traffic flow, the classifier will have high accuracy. DL based model for classifying IoT network traffic was proposed in [528]. This model is a combination of RNN and CNN. Various experiments were performed utilizing different architectures which integrates of RNN and CNN. The experiments performed in this work shows that this methodologies performance is better than ML algorithms for network traffic classification. Various RNN architectures were utilized for network traffic prediction in [530]. The networks utilized were LSTM, GRU, and IRNN. Real-time data from GANT backbone networks was utilized in the experiments conducted in this work. The experiments showed that LSTM performs better with respect to other RNN architectures. In [620] proposed a DL architecture for protocol and application classification. The architecture uses AE for reducing the dimensionality of byte information and CNN for classification. In [533] proposed AE based method for traffic analysis which uses log information to learn the characteristics between the legitimate and anomalous activity. Byte segment neural network (BSNN) and RNN was utilized to classify network traffic in [534]. BSNN basically breaks a data gram into bytes. RNN based encoders utilizes these bytes segments. The final vector representation of datagram, which is given to a softmax function for prediction, is a combination of information extracted from all the encoders. Real-world data was utilized for the experiments conducted in this work. LSTM network was utilized to detect unauthorized and unmanaged

TABLE XIII
A SHORT REVIEW ON DEEP LEARNING APPLICATIONS IN NETWORK TRAFFIC ANALYSIS.

Reference	Architecture	Dataset	Compared CML
[524]	Autoencoder	Private	No
[525]	Autoencoder, & SAE	Time Series Data Library from the DataMarket	No
[526]	SAE, & CNN	VPN-nonVPN (ISCXVPN2016)	Yes
[527]	DNN	UNSW-NB15	No
[528]	CNN	Private	No
[529]	CNN, & RS	TIM Big Data Challenge 2015	Yes
[530]	RS	GEANT	No
[620]	Autoencoder, &CNN	Private	No
[531]	CNN, & RS	Moore	Yes
[532]	DNN	Private	Yes
[533]	Autoencoder	CTU-13	No
[534]	RS	Private	No
[535]	RS	Private	No
[594]	RS	DARPA 1999, AMP, MAWI, & NIMS	Yes
[597]	RS, CNN, & CNN-RS	AMP, MAWI, & NIMS	Yes
[608]	Autoencoder, CNN, RS & CNN-RS	Private, & UNSW-NB15	No

devices in [535]. The devices which are unworthy will be flagged with unusual names using lexical content of networked devices. The proposed network utilizes the names of the devices to learn which devices to flag. In [608] proposed DL architectures for application network traffic classification, malicious traffic classification and malicious traffic detection. The architecture composed of AE and CNN, RNN, LSTM and CNN-LSTM where AE was used for reducing the network traffic features and other DL architectures for classification.

D. Deep Learning in Windows Malware Analysis

Malware represents software programs which are specifically designed for malicious tasks. There are various types of malware exists which includes viruses, Trojans, worms, backdoors, rootkits, spyware, ransomware and panic software, etc. Malicious software programs are similar to software program but intentionally designed to target ICT systems and networks. Ir remains a serious problem for both government and private entities. It continues to open windows to crime, espionages and other illegal activities. Therefore, it becomes vital to push forward computer security in order to eradicate this gateway. Malware discovery can be significantly ordered into static and dynamic analysis. Static analysis comprises of analyzing the executable record without review the genuine guidelines. Dynamic analysis is performed by watching the conduct of the malware while it is really running on a host framework. These two techniques are used to generate signatures. Signature-based detection completely fails to detect variants of existing or new malware itself. To alleviate, the ML and DL algorithms are used. Feature engineering in malware has been one of the difficult tasks due to the rapidly changing behavioral characteristics and the numbers of malwares are very large. To handle this, application of DL architectures are used in recent days. Ransomware is a subset of malware in which the information on a casualty's PC is bolted, normally by encryption and installment is requested before the recovered information is decoded and gets to came back to the casualty [623]. The rationale in ransomware assaults is almost constantly money related, and dissimilar to different kinds of assaults, the casualty is normally advised that an adventure has happened and is given guidelines on the best way to recoup from the assault. The installment is often requested in virtual money, for example, bitcoin with the goal that the cyber criminal's identity isn't known.A short review on DL applications in windows malware analysis is reported in Table XIV.

1) Deep Neural Network (DNN): For reducing the dimensionality of the input data for a DL algorithm, random projection was utilized in [251]. In this work over 2.6 million labeled input data was utilized to train extremely large NNs which in turn allows to train complex supervised classification. 2D binary features were utilized by a DL framework to detect malware in [252]. 400,000 software binaries were used in this work and 0.1% FPR was achieved. To perform feature extraction, this DL framework uses modest computation and accomplishes good accuracy inside a decent time period. Low false rate is due to the fact that this framework relays on syntactic features and not on semantics features. A multi-task DL framework for classifying binary malware was proposed in [257]. The experiments in this work shows that there is a gain of 0.5 for detection compared to DL based detection. 4.5 million files were used for training the DL architecture whereas 2 million files were utilized for testing the model. Furthermore, this work demonstrated the error rate can be reduced significantly by dropout for both deep and shallow neural frameworks and number of epochs to train the model were also reduced by using rectified linear activation function. DL framework was proposed in [261] to detect ransomwares. This network is a combination of ML algorithm and deep packet inspection. The model has the capability of detecting ransomwares with high speed and good accuracy. Raff et al utilized raw bytes of exe files to detect malware by using

ML algorithm [263]. These work deals with a problem of sequential classification as the bytes are treated as units in a sequence with approx two million time steps. Despite the difficulties of learning from a sequence problem, this framework has the capacity to accomplish consistent generalization in both the test datasets. They also recognized some unique ML challenges as well as talked about certain procedures which can be utilized to address the problem of classification of data with amazingly long sequences. Detection and classification of ransomware utilizing deep and shallow networks with the help of API calls was proposed in [270]. Binary and multiclass classification was performed in this work and the results showed that deep networks are performed better than shallow networks. Ransomware were distinguished from benign as well as from its families. MLP were utilized in this work and had the highest accuracy of 1.0. DNN based approach for malware detection using static analysis is proposed in [276]. Score based gist descriptors were utilized for implementing, testing and analyzing malwares in [279]. Robustness of this proposed method along with feature reduction for determining minimum number of gist features required were also analysed in this work. The effectiveness of the proposed approach with respect to DL techniques is also evaluated. Based on malwares opcodes, Word2vec is utilized to detect malware in [281]. For classifying the malware, Gradient Boosting algorithm whereas for validating the model, k-fold cross-validation is utilized in this work. 96% accuracy was achieved in this work with using limited data samples. Efficacy of deep and shallow networks for statically detecting malware is evaluated in [284]. EMBER malware dataset, which is a labeled benchmark dataset, is utilized in this work. Numbers of experiments are performed to choose the parameters based on performance comparison of network topologies and network parameters. The outcome of these experiments shows those deep networks are better in performance when compared with shallow networks. Various DNN frameworks were combined to detect malware in [298]. A very large real-time dataset was created and was utilized to train and test this model in this work. The proposed method has 96.24 percentage of accuracy. DNN and random forest for classification of malwares is compared in [299]. Four different feature sets are used to compare these models. The experiments show that random forest performance better than DNN in all the feature sets. BD is utilized in [301] to train DL model to efficiently detect malwares. The robustness of the proposed architecture was evaluated utilizing a complex dataset. This model has the capability of real-time monitoring, analyzing and detecting malwares. The model has achieved 97% accuracy whereas the ROC is at 0.99. TL approach was applied to classify malwares in [303]. The DNN trained with computer vision data was used to classify static malwares. Three experiments were performed in this work and the results showed that the proposed approach has better accuracy, TPR, FPR, and F1 score that other ML approaches. Biggest advantage is that this model accelerates the training time of the network and at the same times have high performance. In [598] proposed DNN architecture for malware classification. This has performed well in compared to other shallow models such as LR, NB, KNN, DT, AB, and SVM. The experiments are done on EMBER benchmark data set. However, the main limitation of this method is that the proposed DNN architecture relies on the feature engineering.

2) Convolutional Neural Network (CNN): Two scalable approaches were presented in [259] for multi-class classification problem of malwares utilizing convolution neural networks. Microsoft for the BD Innovators Gathering Cup was utilized for this study. First approach converts the malware binary to gray-scale images and then performs classification. In second approach, the architecture of English language classification were utilized to classify malwares. Two different models were trained in the second approach. One model has been pretrained on word embedding whereas the other one is not pretrained. A convolutional FFN which employs hierarchical feature extraction mechanism to detect and classify malwares [264]. This approach is based on data from static analysis and uses meta data of portable executable files. It basically classifies malware into 13 different predefined classes by separating malicious exe files from benign programs. Not only classification was done in this work but they also differentiated predefined classes and benign exe files. The experiments proved that this proposed method is better that normal ML mechanisms such as SVM and FFNs. Malware programs were represented as an image for malware detection in [267]. The images were fed into CNN for classifying the malware.Training of the DL algorithm was done with different kernel and data size whereas the evaluation of this this work was done using ROC AUC. The classification of malware images achieved an higher accuracy and AUC equal to 0.9973. CNN for classifying different types of malwares was proposed in [273]. The malware programs were converted into grayscale images which were given to the network. This proposed network achieved an accuracy of 98 percent. To mitigate the issue of imbalance in classifying malware images utilizing CNN, a simple and effective method was proposed in [274]. In this methodology, the last layer of the CNN employs a weighted softmax loss. This makes the error of classification for different classes to have different weights which makes the classifier to treat the error differently. This proposed method is fearile on any existing working CNN models. DL framework to classify malware is proposed in [275]. Initially, malware was converted into images. This framework uses deep CNN for malware classification and has achieved 91.7 percentage in this work. In [620] proposed AE and CNN based hybrid architecture for malware classification. The AE was employed for feature reduction and CNN for classification. In [277], convolutional deep neural network was utilized to detect malwares using program binaries as input. This proposed work learns only from raw sequence of bytes and labels. This architecture does not need any domain specific features engineering. 20 million dataset of portable exe file was utilized in this work to train the model. It achieves almost same performance as a network which will take hand crafted features as input. SimHash along with CNN was utilized for malware classification in[278]. Conversion of disassembled malware code into a gray scale images is done by SimHash for visualization whereas CNN uses these images to identify the malware family. For improving the

performance methods such as major block selection, multi-hash, and bilinear interpolation were utilized. 10805 malware dataset samples were used in this work and achieved an average accuracy of 98.862%. To recognize a new sample only 1.41 s is needed by this model. DDoS malware detection using a light weight CNN for IoT environment in [283]. Binary malware are converted to one channel gray scale images to be fed into the convolution network which classifies the malware. The accuracy for classification of DDoS malware and benign was 94.0% whereas for the classification of two main malware families and benign is 81.8%. For classifying malware, a DL approach was proposed in [287]. Malimg and Microsoft malware, two benchmark datasets for classifying malware, were utilized in this work. The dataset was first converted to grayscale images to be fed into CNN. 98.52% accuracy was achieved for Malimg dataset whereas for Microsoft dataset the accuracy was 99.97%. GoogleNet and ResNet models were analysed for malware detection in [290]. Two datasets were utilized, one dataset from Microsoft and other dataset which contains 3000 benign files. The dataset is first converted into opcode from exe files and later to images using opcode. GoogleNet has an accuracy of 74.5 percent whereas ResNet got an precision of 88.36 percentage. In [296], combination of CNN and attention mechanism is used to classify malwares. Initially, binary data is converted into images and then attention mechanism uses these images to calculate and show the regions having higher importance. The experiment shows that this model has more accuracy than normal convolutional network. Hierarchical DL model is proposed in [297] to detect and classify targeted malwares. This proposed framework has the accuracy of 97 percentage and a false negative rate of 2.8 percentage. Convolution neural network effectiveness for detecting malware in cloud platform is discussed in [300]. First, a two dimensional CNN is trained utilizing metadata. To decrease the mislabeling of the data and to increase the accuracy of the CNN, a three dimensional CNN is utilized in this work. Various randomly selected malwares were run on virtual machines to collect data to fed into the network. The two dimensional CNN has the accuracy of 79 percent whereas the three dimensional CNN has 90 percent. DL based malware classification and detection was proposed in [305]. The new technique was utilized to convert malware into gray scale image. Convolution neural network was utilized in this work to detect various malwares. Bat algorithm for data equilibrium was utilized in this work to reduce the data imbalance issue. The efficient and effectiveness of the proposed approach was shown in the experiments conducted in this work.

3) Recurrent Structures (RS): An hybrid model which has the ability to learn the language of malware to classify malware was proposed in [253]. The projection stage of this model is a combination of Echo state networks and RNN which are used as feature extractors in this work. Unsupervised data is used to train the model and the feature detected by the projection stage is used by the classifier to detect the malware. Few different experiments were performed with the projection stage which included Half-Frame models and Max Pooling. 98.3% improvement was achieved by TPR and FPR of 0.1% in the final hybrid model which was selected. Effectiveness of DL framework to dynamically detect malware was analyzed in [262]. This analyzing was based on the behavior of the network to collect malware communications exhaustively as well as efficiently. Common latent function and change in the purpose of the communication was the two behaviors that were focused in this work. RNN was also applied to this proposed model. Keeping collection coverage of URLs was 97.9 percentage, analysis time of 67.1 percentage was reduced by the experiments conducted in this work. In [265], the profits of utilizing semi-supervised learning were shown. LSTM and GRU are used by multiple classifiers which classifies malwares. File representation from neural features was constructed utilizing attention mechanism and temporal maximum pooling. Furthermore based on character level CNN they also proposed a single stage malware classifier. Long-Short Term Memory (LSTM) network was utilized to detecting ransomware in [271]. This proposed network uses binary sequences of API calls made by a process. For extracting API calls, an automatic methodology was presented in this work. 96.67% accuracy was achieved for classifying the ransomware behavior by this proposed architecture. Mimicry resilient program behavior model was built in [282] utilizing LSTM and branch modeling. Program behavior model against mimicry attacks were harden using branch sequencing in this work whereas during run time, for extracting branch information hardware features were utilized. LSTM was handling large scale branch sequencing. CNN combined with GRU was utilized for malware classification in [291]. This network has the ability to classify nine different malware families and has 92.6 percentage accuracy. In [536], a RNN and CNN based malware detection framework is proposed where the RNN model learns from features extracted from static and dynamic analysis of PE files. The output of RNN is converted into image format and fed into CNN model in order to classify the PE file as either normal or malicious. The proposed model performs better than other traditional ML and DL approaches with an accuracy of 97.3%.

4) Autoencoder (AE) and Deep belief network (DBN): In [254], malware signature generation and classification is performed using DBN. For compact representation of the malware behavior, deep stack of DAEs were utilized in this model of DBN. Using signatures generated, an accuracy of 98.6% was achieved for classification in this work. SAEs models resting on Windows API were utilized by DL framework to detect malwares in [255]. This proposed architecture first employs unsupervised feature learning and then performs fine tuning with the help of supervised parameters. Large dataset from Comodo Cloud Security Center was utilized for experimentation in this work. The experiments done in this work shows that compared with shallow learning algorithms, this method has better overall performance in detection malware. Malware is represented as an opcode sequence and fed to deep belief nets to detect malware classification in [258]. This framework performance is compared with three other algorithms which are DT, SVM and KNN. This accuracy achieved by this model is equal to the accuracy achieved by the best of the other compared algorithms. When unlabeled data is fed into the model the accuracy is improved. In [266], DBNs for detecting

malware utilizing unlabeled data was proposed. Malware was represented as a opcode sequence in this work. This work shows that there is a increase in performance when used unlabeled data as it produces better classification outcomes. DBNs were treated as AEs to decrease the feature vector dimensions. In [269], feature learning model based on AEs was proposed. Latent representation of different feature sets was learned by AEs model. A fixed 10 size raw features vectors are removed from exe files and fed as input to AEs to extract semantic similarity to produce a code vector. This approach can minimise the memory requirement as it reduces the dimensionality of the features. Malware classification and network based anomaly ID tasks were performed in this work. In [289], a GAN based obfuscated malware detection system is proposed where VAE is used to extract features from the latent space projection of the data. The GAN takes extracted features as input and generates malware samples with features from a specific Gaussian distribution. The generated malware samples are used to enhance the knowledge space of the detection model. The experimental analysis shows that the proposed approach achieves an accuracy of 96.97%. In [295], DBN is utilized to increase the online accuracy of detecting ransomware. Random bit-stream in memory are stored by this method to produce cross-correlations for computing stochastically in FPGA. The network has a precision rate of 91 percentage.

5) Mixed Deep learning (DL) architectures: Two stage DNNs malware detection based on process behavior to check if a terminal is infected or not was proposed in [256]. API call sequences were recorded to observe the process behavior. LSTM with API call based on language model was utilized to construct the features from this API call sequences. Initially a RNN was utilized for extraction of features from the behavior of the process. These extracted feature images were fed into a CNN for classification. In [260], hybrid NN for classification of malware was proposed. Two convolution layers combined with one recurrent layers was utilized in this hybrid network. This method has the capability of extracting hierarchical features which has both full sequential modeling and convolution of n-gram features. The performance of this architecture outperforms many methods such as SVMs, hidden markov models and so on. DL based intelligent anti malware system was proposed in [268]. Malimg dataset which is a collection of malware images was used in this work. A L2SVM was used along with DL model for multiclass classification. CNN-SVM, GRU-SVM (Agarap, 2017), and MLP-SVM were used in this work for classification and in which GRU-SVM model achieved the highest accuracy of around 84.92%. Binary files were utilized for malware classification as well as a data signature and flow features with DL approaches were applied to classify network protocol in [272]. In this work their own dataset was utilized for traffic identification whereas for classification of malware they utilized Microsoft Kaggle dataset. Two convolution with two dense layers were used for malware classification as they give maximum accuracy for this particular dataset. MalNet, a automatic feature learning model was proposed in [280] for detecting malware from raw data. Malware files are converted to grayscale images and decompilation tool is utilized to extract opcode sequences. This proposed work uses CNN and LSTM to learn from the constructed data. 40000 samples were used to train this model in this work.The validation accuracy achieved by this model for detecting malware is 99.88%. 9 malware families for malware family classification were also performed in this work. In [286], DL approach was proposed to classify malwares using malware images. The proposed method is combination of RNNs and CNNs. RNN was utilized to reduce the training dependences with respect to categorical labels. This predictive code was combined with the original code to generate image features by minhash. CNN took this newly combined data to classify the malware. Various ML algorithms and DL framework to detect and analyse malware was used in [288]. Opcode frequency was utilized as a feature vector in this work. Given the opcode frequency, random forest has the best performance compared to any other ML or DL architecture. Heterogeneous DL architecture was proposed in [292] to classify malware. This model consists of AE, multilayer boltzmann and layers of associate memory. Exe files are used to extract windows API calls. Two phases are utilized in this framework, pre training and fine tuning. Various methods based on AE and DNN were proposed for malware detection in [293]. The performance f these architectures were evaluated on Malicia dataset and the proposed method performed better than the feature engineering based method. An anti malware engine generates a very long API call sequences which is a problem for detecting malware. The problem is solved in [294] using neural malware detection models. In this paper, experiments were conducted using different end to end models. These models are combinations of CNN and LSTM networks.

6) Hybrid Recurrent Structures (RSs): In [285], DL framework was utilized for classifying malware. This framework is a combination of CNN and BLSTM network. It can identify complex features and patterns as it is based on data driven approach. One of the big advantage of this framework is that it does not require any expert domain knowledge. Utilizing raw binary files, this framework with high accuracy can classify a malware into nine different types of malwares and takes only 0.0.2 to do this. In [304], a neural sequential classification malware model was proposed. API calls are processed by this proposed network where the input includes two additional parameters. Anti malware engines produced data was collected and the evaluation of this model was done using this data. This proposed model's low FPR is better than any other neural classification model. [611] have done detailed analysis of DL architectures for malware classification. [613] proposes a hybrid of static and dynamic analysis framework for malware detection. Additionally, the image processing based malware classification was done to classify the malware to their corresponding malware family. [621] proposes a cost-sensitive DL approach for handling multiclass imbalance in malware classification. This has performed well compared to the existing cost-insensitive deep earning architectures.

TABLE XIV: A SHORT REVIEW ON DEEP LEARNING APPLICATIONS FOR MALWARE ANALYSIS.

Reference	Architecture	Dataset	Compared CML
[251]	DNN	Private	Yes
[252]	DNN	Private	Yes
[253]	RS	Private	Yes
[254]	DBN	Private	No
[255]	Autoencoder	Private	Yes
[256]	CNN, & RS	Private	
[257]	DNN	Private	No
[258]	DBN	Private	Yes
[259]	CNN	Microsoft malware classication challenge (BIG 2015)	No
[260]	CNN, & RS	VirusShare, Maltrieve, & Private	Yes
[261]	DNN	Private	No
[262]	RS	Private	Yes
[263]	DNN	Private	No
[264]	CNN	VirusShare, Maltrieve, & Private	Yes
[265]	RS	Private	No
[266]	DBN	Microsoft malware classication challenge (BIG 2015), VX Heavens, & Offensive computing malware	Yes
[267]	CNN	VirusShare	No
[268]	CNN, & RS	Malimg	No
[269]	Autoencoder	Microsoft malware classication challenge (BIG 2015), & NSL-KDD	Yes
[270]	DNN	Private	Yes
[271]	RS	Private	No
[272]	CNN, & RS	Microsoft malware classication challenge (BIG 2015)	No
[273]	CNN	Malimg	No
[274]	CNN	Malimg	No
[275]	CNN	Microsoft malware classication challenge (BIG 2015)	No
[620]	AE-CNN	Microsoft malware classication challenge (BIG 2015)	No
[276]	DNN	Private	Yes
[277]	CNN	Avast Repository	No
[278]	CNN	Microsoft malware classication challenge (BIG 2015)	Yes
[279]	DNN	malimg, Malicia	Yes
[280]	CNN, & RS	Microsoft malware classication challenge (BIG 2015)	Yes
[281]	DNN	Microsoft malware classication challenge (BIG 2015)	No
[282]	RS	Private	No
[283]	CNN	Private	No
[284]	DNN	Ember	Yes
[285]	CNN, & CNN-RS	Microsoft malware classication challenge (BIG 2015)	No
[420]	CNN	Microsoft malware classication challenge (BIG 2015)	Yes

[286]	CNN, & RS	Microsoft malware classication challenge (BIG 2015)	No
[287]	CNN	Microsoft malware classication challenge (BIG 2015), & Malimg	Yes
[288]	Autoencoder, & DNN	Malicia	Yes
[289]	Autoencoder	Microsoft malware classication challenge (BIG 2015)	No
[290]	CNN	Microsoft malware classication challenge (BIG 2015), & privately collected samples	No
[291]	RS	Microsoft malware classication challenge (BIG 2015)	No
[292]	DBN, Autoencoder, Stacked Autoencoder,	Private	Yes
[293]	Autoencoder, & DNN	Malicia project	No
[294]	CNN, & RS	Private	No
[295]	DBN	Private	No
[296]	CNN	Private	Yes
[297]	CNN	Private	No
[298]	DNN	Private	No
[299]	DNN	Private	Yes
[300]	CNN	Private	No
[301]	DNN	MalwareTrainingSets	Yes
[303]	DNN	Malimg	Yes
[304]	LSTM, & CNN-RS	Private	No
[305]	CNN	Malimg	Yes
[536]	RS	Private	Yes
[598]	DNN	EMBER	Yes
[611]	CNN, & CNN-RS	Malimg	Yes
[613]	CNN, & CNN-RS	Malimg	Yes
[621]	CNN-RS	Malimg	Yes

TABLE XV: A short review on Deep learning applications in Android Malware detection.

Reference	**Analysis**	**Features**	**Architecture**	**Dataset**	**Compared CML**
[335]	Dynamic / Static analysis	Sensitive API calls	DBN	Publically available sources	Yes
[336]	Static, & Dynamic Analysis	192 binary features	DBN	Publically available sources	Yes
[337]	Dynamic Analysis	API calls	CNN	Publically available sources	No
[338]	Static, & Dynamic Analysis	10 static, & dynamic feature sets	Autoencoder	Publically available sources	No
[339]	Static, & Dynamic Analysis	API calls	DBN & CNN	Publically available sources	Yes
[340]	Static analysis	Requested permissions, used permissions, sensitive API calls, & Actions-app components	DBN	Drebin, & apps collected from publically available sources	No
[341]	Static analysis	API call blocks	DBN	Publically available sources	Yes

[342]	Dynamic / Static analysis	Required permissions, sensitive API calls, & apps actions collected dynamically	DBN	Publically available sources	Yes
[343]	Dynamic analysis	System calls	SAE	Publically available sources	Yes
[344]	Static analysis	More than 1,000 features	DNN	Drebin, & collected from publically available sources	Yes
[346]	Static analysis	Android permissions	RS	CDMC 2016	No
[347]	Static analysis	Risky Permissions, & dangerous API calls	DBN	Private	Yes
[348]	Static analysis	Requested permission	CNN-AlexNet	Drebin, & apps collected from publically available sources	No
[349]	Static data flow analysis	323 features	DBN	Publically available sources	Yes
[350]	API calls	1,058 API calls	DBN, & SAE	Publically available sources	Yes
[351]	Opcode Sequence	Learn to detect sequences of opcode that indicates malware	CNN	Publically available sources	No
[352]	Static analysis	API call sequences	CNN	Publically available sources	Yes
[353]	Transfer classes. dex into RGB color images	Extract features from the transferred images	CNN	Collected by the researchers	No
[354]	Dynamic analysis	System calls	CNN	Drebin, & collected from publically available sources	No
[355]	Dynamic analysis	System call Sequences	CNN	Publically available sources	No
[356]	Static analysis	Opcode sequences	RS	Drebin, & collected from publically available sources	Yes
[357]	Static analysis	Byte sequences	CNN	Publically available sources	Yes
[358]	Static, & Dynamic Analysis	Features from Static & Dynamic Analysis	DBN	Publically available sources	Yes
[359]	Static analysis	Dangerous API calls, & risky permissions	DBN	Drebin, & apps collected from publically available sources	No
[360]	Static analysis	API calls, Permissions, & Intent filters	CNN	Drebin, & apps collected from publically available sources	Yes
[361]	Static analysis	API calls	DBN	Drebin, & apps collected from publically available sources	No
[362]	Static analysis	Permissions requested, permissions filtered intents restricted API calls, hardware features, code related features, & suspicious API calls	CNN	Publically available sources	Yes

[363]	Static analysis	API sequence calls	CNN	Drebin, & apps collected from publically available sources	No
[364]	Static analysis	Semantic structure of Android bytecode	CNN-RS	Publically available sources	No
[365]	Static analysis	Permissions API Calls	DNN	Drebin, & apps collected from publically available sources	No
[366]	Static analysis	Code Analysis	CNN	Drebin, & apps collected from publically available sources	No
[367]	Dynamic / Static analysis	Permissions Intents, app components network activities, & Linux system call	DNN	Drebin, & apps Publically available sources	No
[368]	Dynamic / Static analysis	Permissions events generated by Monkey Tool	RS	Publically available sources	Yes
[369]	Static code analysis	Token & semantic features from smali	DNN	Private	Yes
[370]	Static analysis	Features from permission & API calls	DNN	Drebin, & collected from publically available sources	No
[371]	Static analysis	Opcode sequences	CNN	Drebin, & collected from publically available sources	Yes
[372]	Static Analysis	Features from control flow graph, & data flow graph	CNN	Drebin, & collected from publically available sources	No
[373]	Static / Dynamic analysis	API sequences to image	CNN	Publically available sources	No
[374]	Static, & Dynamic Analysis	String feature, Method opcode feature, Method API feature, Shared library function, opcode feature, Permission feature, Component feature, & Environmental feature	Autoencoder	Publically available sources	Yes
[375]	Static, & Dynamic Analysis	Features from Static, & Dynamic Analysis	Autoencoder	Publically available sources	No
[376]	Dynamic Analysis	API calls graph	CNN	AMD, AndroZoo, Drebin Malware Collection, & ISCX Android Botnet	Yes
[600]	Static analysis	Permission & API calls	DNN	CDMC2017	Yes
[602]	Static analysis	Permission & API calls	RS	CDMC2017	Yes
[608]	Static Analysis	Opcode	RS, CNN, CNN-RS,	Drebin	No

E. Deep Learning in Android Malware Detection

In recent times, Android OS has earned attention from different organization ranging from academia to industry. An android OS (OS) is an open source, Linux based OS which is most commonly used OS for mobile and handheld devices. Due to its importance in several applications, Android OS has become a target for attackers to conduct criminal and illegal activities. As the attacks to Android OS continue to grow, various methods have been introduced to fight against attacks. The developers use software development kit (SDK) to build and publish their applications. It is basically designed for mobile devices, smartphones, tablets and additionally it can support other platforms like TVs, cars, embedded and wearable devices. Due to this portability nature, many companies have been involved in development of apps for android platform that apparently runs on all the devices. Applications are hosted in official app store called Google Play. Android is being served as most popular OS, third-party distribution centers, a rich SDK and uses Java programming language. In recent days, Android devices have been largely used by peoples. These devices stores lots of sensitive information like financial information like bank details, user credentials, personal information like photos, and videos. This has become more interesting to the attackers. Malicious authors develop malware to steal the private data or delete or alter the existing data and monitor the user activities with the aim to get benefits. Additionally, Android applications are hosted in various third party stores which allow the user to repackage Android applications with adding a piece of malicious code. Generally, Android OS automatically assigns a unique Linux user ID during the installation phase to know each app runs its own instance of virtual machine. This facilitates to the creation of a sandbox which isolates the apps from each other's. It provides authorization mechanism through the use of Android permissions. Android architecture composed of different sections, as shown in Figure 12. They are:

- System applications: The architecture of Android contains system applications at the top which offers the basic functionality such as email management, calendar etc.
- Application framework: This provides a set of reusable modular components for the development of new applications in Java language.
- Libraries and Android runtime: A developer can also access native libraries in C/C++ via Android native development kit (NDK). To maintain low consumption of resources, Android Runtime runs multiple instances of virtual machines. Each virtual machine runs the DEX bytecode which is written in Java.
- Linux Kernel: It is composed of hardware resources which are utilized by the upper level, Application framework. It is a fundamental unit for hardware of the device and provides the basic functionality such as memory, process and inter-process communication (IPC) management.

Generally, Android features is collected via rooted [229], [230] or unrooted [231] devices that can be passed as an input to ML models to learn the characteristics to distinguish between the benign and malicious apps. Android uses Google play as an official market store that hosts the apps and there are more than hundred third-party app stores that also host the Android applications. It has an in-built security mechanism, called as bouncer. It frequently scans the Google play store for malicious app and assigns a signature. Android permission systems is an another built-in security mechanism that are meant to control the app permissions. During the app installation, Android permission systems seek explicit permission from the users to access any sub-systems. However, most of the users follow the blind approach to provide grant permissions during the installation procedure of apps as the impact is less known to the end user and the process is tedious and doesn't have many options to provide selective permissions. Since the intention of the app is difficult to identify, the damage could be serious including, compromise of user data, identity theft and taking over the control of the entire device. Secondly, the set of permissions might be same for both the benign and malicious apps. Hence, Android permission system can be considered as an initial shelter for risk assessment rather than malware detection. Attacks on Android OS will continue to grow as the technology evolves. There is a sudden surge in Android malware and this sheer number of new malware instances requires newer approaches as writing a signature for each malware is a daunting task. Signature based and heuristics based methods are belongs to rule based system. Rule based system relies on signature database. This database has to be continuously updated by domain experts whenever a new malware occurs. This can be an effective solution to detect already existing malware but completely fails in detecting the variants of new malware and completely new malware itself. While signature-based or heuristics-based detection is important, using self-learning systems for the analysis and detection of growing Android malware are increasingly being studied. Self-learning system composed of DM, ML and DL algorithms. These techniques provide new sensing capabilities for Android malware detection which work at scale. Moreover, these approaches have the capability to detect the variants of already existing malware or entirely new malware itself. There are two fundamental taxonomies of techniques followed by researchers for collecting features from Android OS. They are static and dynamic analysis [624]. Static analysis collects a set of features from apps by unpacking or disassembling them without the runtime execution and by contrast, dynamic analysis examines the run-time execution behavior of apps such as system calls, network connections, memory utilization, power consumption, user interactions, etc. The hybrid analysis is a two-step process where initially static analysis is performed before the dynamic one which results in less computational cost, low resource utilization, light-weight and less time-consuming in nature. Hybrid analysis approach is increasingly being used by anti-virus providers for the smartphones as it provides higher detection rates. A short review on DL applications in Android malware analysis is reported in Table XV.

1) Dynamic Analysis: : In [337], the authors have proposed a CNN based android malware detection system where API calls sequences are used to train the detection model. The

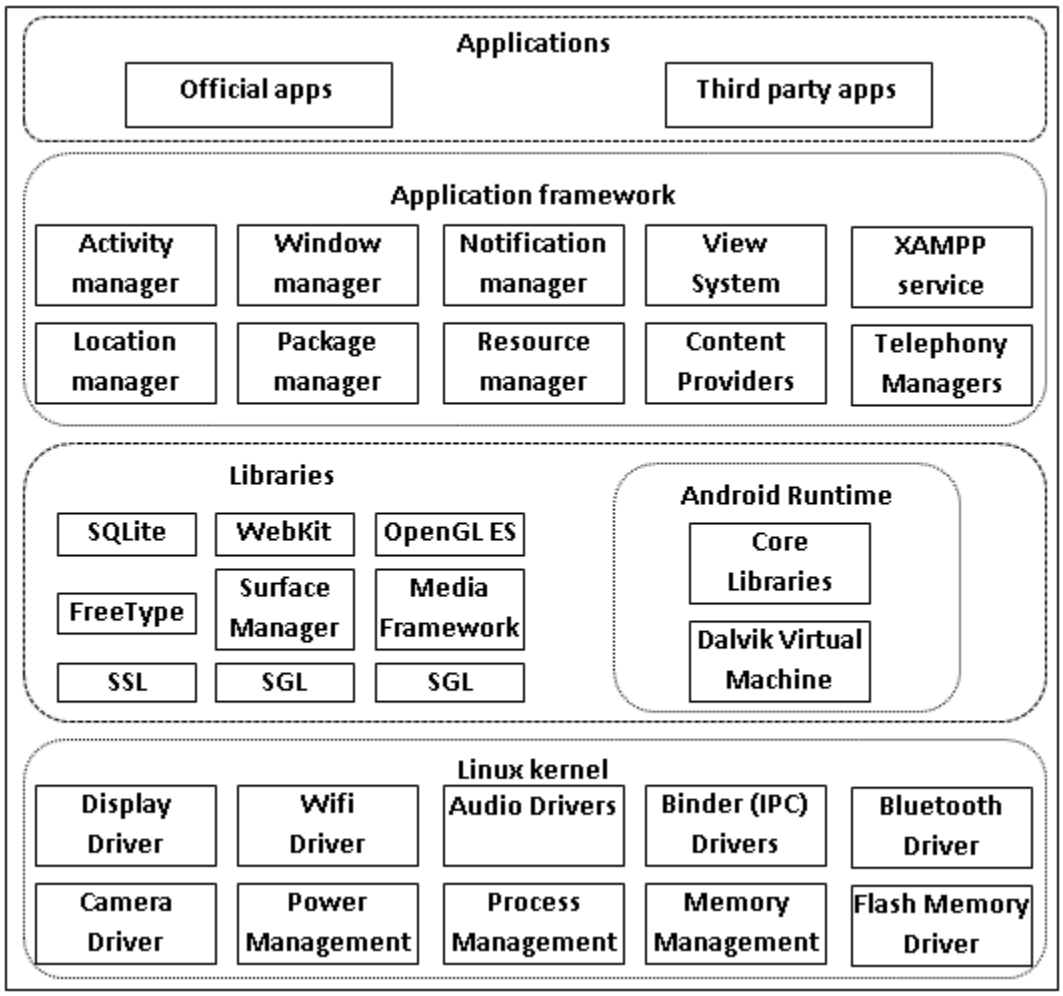


Fig. 12. Structures of Android Operating System (OS).

proposed model performs better than traditional ML models like SVM and naive bayes with almost 100% accuracy. In [343], a DL framework for android malware detection is proposed which learns system call graphs. The framework performs a new dynamic analysis which executes code routines of the app and constructs a graph based on the system calls. The experimental analysis shows that the proposed system outperforms other traditional detectors when tested on real-world malware samples. In [354], the authors have presented a CNN-based android malware detection system which uses dynamic analysis to extract system calls to build the model. The model is evaluated using real-world dataset containing 7100 apps and it achieves an accuracy of 85% to 95%. In [355], a DL model to detect malwares in android systems is presented. The proposed model takes system call sequences as textual input and differentiates between benign and malicious texts. The model is evaluated using 14231 apps and it achieves an accuracy of 93.16%. In [376], a new DL based android malware detection approach is proposed where API calls are represented as graph which shows the execution paths of malware samples. A CNN based architecture is used for extracting the local features and it classifies with 98.86% accuracy. The experimental analysis shows the performance of the proposed architecture with various graph embedding techniques.

2) Static and Dynamic Analysis: : In [335], a DBN based malware detection framework for android is proposed which performs static and dynamic analysis to obtains more than 200 feature. The proposed model achieves an accuracy of 96% and it outperforms other models such as Naive bayes, DT, SVM, MLP, and LR. In [336], a DL based android malware detection framework is proposed. The proposed framework extracts features from static and dynamic analysis of apps and performs better than traditional ML classifiers with a detection accuracy of 96.76%. In [338], the authors have proposed a new hybrid android malware classifier which is partly based on DL approach. The proposed system trains the DNN with extracted static and dynamic features and combines the original features with the features extracted from the DNN. The hierarchical multiple kernel learning is applied on combined feature set which improves the detection accuracy to 94.7%. In [339], a hybrid parallelized graph-based ML approach for android malware detection is proposed. The original features are combined with features from DNN which is trained on static and dynamic features. The graph-based kernels are applied on this combined features and at at last hierarchical Multiple Kernel

Learning is applied. In [342], the authors have proposed an online android malware detection framework based on DBN. The framework extracts 192 features from static and dynamic analysis and build the detection model. The proposed model is evaluated using 21760 applications and it achieves an accuracy of 96.76%. In [358], a DBN based android malware detection framework is proposed. The framework extracts features from static and dynamic analysis and system calls. The proposed model performs better when compared to SVM, naive bayes and random forest based models and achieves an accuracy of 99.1%. In [367], a DL based android malware detection application for on-device usage is proposed. The application extracts features by performing static and dynamic analysis and builds the DL model. The proposed model is evaluated using 4208 apps and it achieves an accuracy of 95%. In [368], The effectiveness of RNN and LSTM models for detection of malware in android systems is studied. The models are evaluated using well-known datasets and it is found that the LSTM performs better than RNN. The LSTM model achieves detection accuracy 93.9% and 97.5% in dynamic and static analysis respectively. In [373], a DL based malware identification framework for android is proposed. The framework extracts sequence of API calls and protection levels as featured and builds the CNN based detection model. In [374], the authors have proposed a multi-model DL based malware detector for android systems. The proposed model is trained using 7 different features obtained by analyzing different files from the APK file. Detained experimental analysis is given where the proposed system is compared with other DL model. In [375], a DL based android malware detection framework using AE is presented. The model is trained on features extracted from APK file and its performance is evaluated using real-world dataset containing various benign and malicious sample apps. The proposed framework works better than conventional systems with a high accuracy of 96.81%.

3) Image Processing: : In [353], a CNN based android malware detection framework is proposed. The proposed system converts the bytecode of classes.dex into images which is taken as input by CNN detection model.

4) Static: : Comparison with Machine learning algorithms: In [341] , the authors have proposed a novel feature representation and a DBN based malware detector for android systems. The detection framework uses block of API calls as features instead of simple API calls which makes it harder for the attacker to evade detection. The proposed model is tested on real-world samples and it performs better than other conventional approaches. In [344], the authors have proposed an evolving DNN based android malware detection system. The proposed system uses a genetic algorithm to modify the parameters and configurations of the DL model with the goal of increasing the accuracy and minimizing the complexity. This system performs better when compared with SVM based models and it achieves an accuracy of 91%. In [347], a DBN based android malware detection framework is presented. The framework utilizes risky API calls and permissions as features to build the detection model. Detailed experimental analysis is given where It is found that the proposed model performs better when compared to traditional SVM based approaches. In [349], a DBN based android malware detection system is presented. The system identifies 323 features by extracting data flows from 1000s of malicious and benign apps. The proposed model outperforms various traditional ML models and achieves F1 score of 95.05%. In [350], a novel feature representation and two DL based malware detectors for android systems is proposed. The detection framework uses block of API calls as features instead of simple API calls which makes it harder for the attacker to evade detection. DBN and AE detection models are tested on real-world samples and it is found tha DBN model performed better than AE model. In [352], the authors have studied the effectiveness of DL models for android malware detection. They have used CNN and LSTM models for the detection based on API call sequences and both models have performed better than n-gram based detection models. It can be observed that CNN model surprisingly performed better than LSTM model. In [356], a LSTM-based hierarchical denoise network is proposed for android malware detection where opcode sequences are computed by first level LSTM and malware detection takes place in second level LSTM. The experimental analysis shows that proposed method can capture long feature sequences and achieves better detection accuracy than malware detector based on n-gram. In [357], a CNN based generalized malware detection system is proposed. The proposed method extracts important sequences of bytes by evaluating attention map which could be useful for the malware analysts. The results shows that The model achieves better detection accuracy than conventional methods. In [360], a malware detection framework for android systems based on CNN is proposed. The framework extracts 5 different feature sets from static analysis and builds the detection model. The proposed model achieves an accuracy of 97.4% and outperforms other traditional ML based detection models like KNN and Linear SVM. In [362], a hybrid large-scale android malware detection system is proposed based on CNN and DAE. The proposed framework uses DAE as a pre-training method for CNN based detection model. The proposed hybrid model is tested with 13000 malicious and 10000 benign applications. It can be observed that the proposed model reduces the training period by 83% when compared to simple CNN model. In [369], a DNN based android malware detector is proposed which uses a novel technique to obtain token and semantic features of smali files. The model is trained using smali files from 50 apks and it outperforms traditional ML models with an AUC of 85.98% and 70% in both WPDP and CPDP mode respectively. In [371], a CNN based android malware detection framework is presented. The model uses opcode sequences from decompiled apk files to learn to differentiate between malicious and benign apps. The proposed approach achieves a detection accuracy of 99% with very low false positives. In [372], the authors have proposed a android malware detector based on CNN which uses control and data flow graphs as features to build detection model. It perform better than Drebin and most of anti-virus tools gathered in VirusTotal. DNN architecture was proposed for Cyber Security applications [600]. The performance of this architecture was evaluated on Incident detection, Android malware classification, and fraud detection. Various experiments were done

on these various Cyber Security applications and results are compared with the XGBoost. In all the experiments DNN performed well in compared to the classical method. RNN architecture was proposed for Cyber Security applications [602]. The performance of this architecture was evaluated on Incident detection, Android malware classification, and fraud detection. Various experiments were done on these various Cyber Security applications and results are compared with the SVM. In all the experiments DNN performed well in compared to the classical method.

Comparison with Machine learning algorithms: In [340], the authors have proposed a DBN based android malware detector which can be adapted to large scale real-world detection. The framework extracts 32,247 features from static analysis of the apps and build the DL model which works better than other conventional approaches with an accuracy of 99.4%. In [346], the author have proposed a LSTM based android malware detection framework which detects malicious apps based on android permissions. The LSTM is trained on permission sequence with bag-of-words embedding and optimal parameters and it achieves an accuracy of 89.7% on the real-world Android malware dataset, provided by CDMC2016. In [348], a CNN-based android malware detector is proposed. The framework extracts features from static analysis of android app permissions to build the CNN model. The model is test on real world android malware dataset with 2000 malicious and 500 benign apps and it performed better than other related approaches with an accuracy of 93%. In [351], the authors have proposed a CNN based android malware detection framework. The framework extracts features from static analysis of raw opcode sequences. The proposed approach performs better and computationally efficient when compared to n-gram based detection model. In [359], the authors have proposed a DBN-based android malware detection system which extracts features from risky API calls and permissions to build the detection model. While DREBIN attain accuracy of 90% with 545000 features, the proposed model achieves the same accuracy with just 237 features. In [361], a DBN and Restricted Boltzmann Machine based malware detector for android systems is proposed. The framework extracts features from API calls and grayscale texture image obtained from malicious or benign code. The proposed approach achieves accuracy of 94% with low false positives. The proposed approach achieves an accuracy of 93.64%. In [363], a DL based automatic malware detection framework for android systems is proposed which can be deployed on servers, mobile and IoT devices. The proposed framework extracts raw sequences of API calls from app and builds the model which detect malware and identify its family. The experiment analysis shows that the proposed approach achieves F1 score of 96%-99%. In [364], the authors have proposed an android malware detector with multi-detection layer based on MLP and LSTM. The first layer uses MLP which is learns from xml files and the second layer uses LSTM which learns from bytecode semantics. The proposed approach outperforms other state-of-the-art detection systems and achieves accuracy of 97.74%. In [365], a DL based android malware detection and categorization engine is proposed. The proposed model is tested using real-world malware dataset containing 131611 applications and it achieves an accuracy of 97%. In [366], the authors have proposed a lightweight android malware detector based on 1-D convolution which analyses last 1KB of raw APK file to differentiate benign and malicious apps. The proposed system is tested using different datasets containing 7000 applications and it achieves an accuracy of 96- 97%. In [370], the authors presents a DL based android malware detector which extracts a large set of features from static analysis to build the detection model. The proposed model classifies the malware based on its family with more than 90% accuracy in just 0.5 seconds which shows its potential to be used in a real device. In [608] proposed DL based method for static Android malware detection. DL architectures implicitly learn the optimal feature representation from raw opcode sequences extracted from disassembled Android programs. The various numbers of experiments were run for various DL architectures with Keras Embedding as the opcode representation method. RNN, CNN, LSTM, CNN-RNN and CNN-LSTM architectures were employed and CNN-LSTM performed well in compared to other DL architectures.

F. Deep Learning in Side Channel Attacks Detection

Kocher introduced Side-Channel attacks in 1996 [250]. These attacks are basically a type of physical attacks which are utilized by the hackers to break into cryptographic devices. Side channel information leaked by the hardware like timing information, electromagnetic radiation, power consumption statistics of encryption devices can be utilized to launch a side channel attack. These attacks are very fast and can be implemented for only few hundred dollars so they pose a great threat to security. The devices used to be attacked can be any device from small embedded devices such as RFID to laptops. In recent days, state of art DL frameworks have been applied to this domain of side channel attacks detection.

A short review on DL applications in side channel attacks detection is reported in Table XVI. In [306], the authors studied the application of DL for the analysis of side-channel attacks. They have discussed about several parameterization options, benchmark models and choice of hyper parameters for DL models. The results shows that VGG-16 model outperforms many baseline models. In [307], presented an overview of DL for side channel attacks detection. They also showed experiments related to CNN based side channel attacks detection. In [308], the authors have proposed two approaches to enhance the effectiveness of side-channel attacks based on DL methods. First approach is to decrease the training and attack traces to retrieve the key by using new spread layer in NNs. The second approach is to efficiently correct the model predictions based on confusion matrix. In [309], the authors studies several ML and DL models for side-channel attacks. They have found that CNN performs better when the noise level is low and number of features are high. They have also shown that random forest and XGBoost performs better than CNN with low computational cost in other scenarios. In [310], a novel DL based method for non-profiled side-channel attacks is proposed. The proposed method uses DL metrics and

combined key guess to retrieve insights about secret key. They have enhanced the performance by using data augmentation techniques with MLP and CNN models and they have shown that the proposed approach outperforms classic Non-Profiled attacks. In [311], The performance of CNN is studied on four side channel data is studied. The proposed CNN model is compared with CMLAs. It can be observed that in best case, CNN model can achieve accuracy of 99.3% and for DPA contest v2 dataset, CNN is outperformed by ML models like SVM. In [312], the authors have proposed a novel side-channel attack on CNN based model. The proposed attack recovers the input image using the measured power traces. They have proposed background detection and power template method to recover image. The experimental analysis on MNIST datasets shows that the proposed approach achieves high accuracy. In [313], A DL based side-channel attack is used to retrieve the secret key of AES cryptographic circuit. The relationship between EM noise and power noise is modeled using DNN by analysing the captured EM emission and power dissipation and the secret key is retrieved. The proposed work can effective get the key by only analysing 32,500 number of plaintexts. In [314], studies the robustness of CNN on various side channel information leaks.

TABLE XVI
A SHORT REVIEW ON DEEP LEARNING APPLICATIONS IN SIDE CHANNEL ATTACKS DETECTION.

Reference	Method	Dataset	Compared CML
[306]	Self-Normalizing Neural Networks, & CNN	ASCAD Database	No
[307]	CNN	DPA Contest v4	No
[308]	DNN, & its variants	ASCAD Database	No
[309]	CNN	DPA Contest v4	Yes
[310]	CNN	ASCAD Database	No
[311]	CNN	DPA Contest v2, & DPA Contest v4	No
[312]	CNN	MNIST	No
[313]	DNN	Private	No
[314]	CNN	Private	No

G. Deep Learning in Function recognition

Function recognition is an important step which is very useful in malware detection. For malware analysis, one of the most important tools is bytecode or binary data analysis. For binary data analysis, the biggest challenge is function recognition. The applications of DL architectures are have been studied for function recognition in x86 and x64.

In [83], a DL mechanism was proposed to recognize functions in a binary file. Great results were found in various applications such as language modeling, speech recognition etc. The comparison between ML and DL algorithms was also performed. In [84], EKLAVYA, a RNN was introduced to address the problem of function type signature recovery. This recovery was done from a disassembled binary code. This system was able to learn not only about the idioms that match the given domain knowledge but also about the calling conventions. In [85], Gemini was introduced by the authors which are a DNN based approach for generating embedding for binary function. Additionally, they also demonstrated that Gemini was able to identify vulnerable firmware images more significantly than Genius. In [86] presented VulDeePecker which is a DL based vulnerability detection system. Experiments showed that this approach has less false negative rate than many other vulnerability detection system. In [87] proposed architecture MobileFindr which is dynamic strategy for function similarity mapping system. It was developed for on-mobile device applications. The outcome showed that the proposed system identifies the fine-grained function similarities successfully. The author in [88] proposed and implemented a prototype namely Diff to solve cross-version BCSD problem. This system is basically a DNN augmented solution which applies three semantic features namely, inter-module, inter-function and intra-function features. DeepMem is a graph based object detection methodology proposed in [89]. The author has basically visualized memory as an intermediate graph representation. Later this graph was utilized to detect different types of object. In [90], the author proposed a word embedding method for extracting features from two binary file with the aim to find the similarity between the files utilizing only their compiled form. They employed state of art DL method along with graph embedding. The experiments showed that there is an increase of two percent in performance with respect solution based on ML approaches. SAFE was introduced in [91] for embedding of functions. This architecture is based on self-attentive NN and does not need CFG due to which there is an improvement in speed. In [92], the author proposed a methodology to find software weakness using DL approaches. Initially, Instruction2vec was also introduced in this work which basically vectorizes the assembly code effectively. The DL approach takes this vector to learn about the assembly code function and classify whether the function has a software weakness or not. The outcomes of the proposed model with word2vec were compared. SySeVR which stand for Syntax-based, Semantics-based and Vector Representations is another DL architecture used for detecting software vulnerabilities [93]. The specialty of this algorithm is that it can detect 15 different vulnerabilities among which seven are unknown. A similarity comparison tool INNEREYE-BB was introduced in [94] which is a based on a NN and word embedding. Instructions are represented using word embedding and the encoding of instruction embedding and dependencies is done using LSTM. This proposed work is able to solve the problem of cross-architecture code containment problem. DL approach was applied to binary code visualization to solve the problem of binary code similarities was proposed in [95]. Basically binary code was represented as an images and then DL algorithm for image classification were used in this work. Decompilation is a method of recovering the structure of source code from binary machine code which was implemented using RNN architecture [96]. The model was trained and evaluated on binary machine code which was compiled from C language source code. This proposed methodology is not language specific and does not require domain knowledge of the language. Automatically learning code similarities from

diverse representations of codes like ASTs, CFGs etc using DL approach for SE tasks is proposed by [97]. This model is reusable which eliminates large time sink of a DL approach. Patch detection in binary analysis is proposed by [98]. It is created using DNN and the model automatically identifies whether the function is patched or not.

H. Deep Learning for Steganalysis and Steganography

Steganography technique is an art of sending messages while hiding the existence of the communication. Utilizing this technique, secret messages concealed inside some ordinary information can be sent without anyone's knowledge; making the secret message invisible. The main property of a good steganographic system is by utilizing the right key only, finding the message should be possible. The process of detecting the presence of communication which has the concealed messages is known as steganalysis. Steganalysis has been remained as a significant area of research in Cyber Security in the last years to recognize covert attacks in public network. A brief introduction to steganography and steganalysis is provided in [99]. In [100], experiments were performed utilizing ANN to show ML based steganography. Steganography and steganalysis techniques can be applied on different kinds of data, for example, on texts, images and videos.

Deep Steganography was proposed by Shumeet Baluja which utilized DNN to work as a pair to not only hide but also uncover the concealed messages [101]. The Study is basically related to hiding a fixed color image inside an another image by using DL methodology and the experiments were performed using Imagenet database. A two stage process for hiding information was proposed in [102]. Generally the secret message of information is embedded inside a complex object. With the aim to identify the complex object this paper utilized the application of DL architecture after information hiding is studied using multiple steganographic algorithms. The capability of DNN for data hiding is studied and compared with the classical data hiding techniques. The proposed DNN models are efficient and more robust [103]. Texture synthesizing is a well-known method in computer vision and used image concealing in steganography and watermark [104]. The classical method exists for Texture synthesizing are based on features and mathematical functions. This is avoided by introducing the GANs and detailed experimental analysis shown for Texture synthesizing. CNN based image steganography is studied for cover image by [105]. With the aim to enhance the robustness of the CNN model in image steganography [106], GANs are employed and additionally to improve the invisibility by hiding the secret image only in the Y channel of the cover image. Payload capacity is very important factor in the domain of image steganography due to the reason that suppose if more information is hidden in the cover image then there may be possibility that cover image is altered largely and risk of detection is higher. With the aim to increase the payload capacity without changing the appearance of cover image in a larger term, CNN based method is employed [107]. CNN based image steganography is mapped into video steganography by [108]. DNN based secret information removal is studied by [109]. Unsupervised GANs was introduced to avoid lot of expert knowledge and complex artificial rules in steganography [110]. The method generates the stego image from the secret message without the cover image. The detailed experimental analysis was shown with a case study. U-Net DL architecture was employed for reversible image steganography [111]. The DL architecture for image steganography studied by [112], particularly the DNN was used in decoding of secret message approximately followed by domain adaptation method based on GANs for transferring image into high quality RGB image.

The modified method for cohort intelligence was proposed and applied on JPEG image steganography [113]. GANs based method was proposed for hiding the binary data inside an image. The proposed method provides how much information can be hidden successfully [114]. The experimental analysis based on DL based steganalysis was studied for to defeat LSB-based steganography [115]. To avoid embedding information in steganography deep convolutional generative adversarial networks (DCGANs) was employed [116]. Using DCGAN, secure steganography method was proposed. This method automatically generates container for images and stays more secure against steganalysis method compared to container derived from the original images [117]. Application of GANs was used for steganography. A detailed case study on GANs based steganography discussed and confirmed that the proposed method stays more robust in an adversarial environment compared to the classical steganography methods [118]. A detailed study on DL based steganography and steganalysis was studied by [119]. A detailed study on DL based steganography was studied by [120]. A detailed study on the effectiveness of deep residual network was shown for steganalysis by [121]. Residual CNN based approach was studied for steganalysis by [122]. The existing CNN based model for steganography relies on handcraft or heuristics to identify the value for its various parameters [123]. To avoid this, an architecture based on deep residual was introduced and a detailed experimental analysis of the proposed method was shown [124]. Deep residual multi-scale convolutional network was proposed for steganalysis which outperformed the existing methods based on CNN and also the classical steganography methods. TL method was employed for image steganalysis [125]. The detailed experimental analysis was shown using the deep residual NN. CNN based framework was proposed for JPEG steganography [126].

Text steganography method using LSTM encoder and decoder models was proposed by [127]. The detailed experimental analysis was shown for Chinese quatrains generation. To hide information in VoIP streams, Quantization index modulation (QIM) is the most commonly employed method. In [128]., the author proposed a RNN based method for QIM steganalysis. [129] proposed RNN based linguistic steganography. CNN based text steganalysis was proposed which can work in semantic analysis phase. Primarily CNN was used to extract high level semantic features of texts and estimates the difference of with and without embedding secret information. In [130], the method was proposed for text steganalysis which

is more fast and efficient. This method finds the correlations between words of steganographic texts and then these words are mapped to a semantic space. The correlations between the words are extracted using a hidden layer and these features are passed into output layer for classification. In [131], the author proposed DNN based method for steganography in speech signals. LSTM based stegnographic text generation proposed by [132]. In [133], the author proposed a method for audio steganography using CNN. [134] proposed a method based on Deep Residual Network for steganalysis which uses the spectrogram as the main feature to detect steganography schemes in different embedding domains for AAC and MP3. In [135], a CNN based steganalysis method was proposed for MP3 Steganography in the Entropy Code Domain. Recently, deep RNN models were employed for DNA steganography by [136].

I. Deep Learning in Insider threat detection

There have been several types of research on the field of detection of external malware entering the system. Till day this has been handled by adding IDS layer at the choke points of the network. But there is another variety of problem where there is a possibility of theft within the system. To prevent this so-called insider threat, researchers have been developing insider threat detection systems based on various ML architectures. The ultimate aim of this system is to identify hostile activities from behavior data. As the network threat evolution in recent years, identification of internal threat has become more difficult. Like in IDS, classical insider threat detection systems have been functioning on the acquired knowledge of past attacks which has been deemed to be inefficient.

A short review on DL applications in insider threat detection is reported in Table XVII. In [315], the authors proposes an online unsupervised DL approach that uses the system logs in real-time which can extract the anomaly scores into individual user behavior to efficiently identify the threat. They have used DNN and LSTM models which outperformed the existing anomaly detection baselines that are based on Isolation Forest, PCA and SVMs. In [316], The authors have presented a novel insider threat detection system based on LSTM and CNN. The LSTM extracts the behavior features from user actions which is given to CNN for classification. It can be observed that the proposed model achieves AUC of 0.9449 in best case. In [317], Aaron et al. have proposed a flexible unsupervised technique for the detection of anomalous activities using bidirectional LSTM which is trained on computer security log data. The proposed method is performs significantly better when compared to standard PCA and isolation forest based detection models and it achieves an AUC of 0.98. In [318], the authors proposes an insider threat detection framework based on LSTM-RNN and PCA. The proposed system classifies the attribute features and evaluates the behaviour abnormality. It can be observed that the proposed framework performs better when compared to insider threat detection models based on SVM, PCA, and Isolation Forest. In [319], a LSTM based anomalous user behaviour detection framework is proposed. The proposed approach learns the behaviour pattern by analysing the user logs and detects anomalies. In [320], the authors have proposed a novel insider threat detector based on adaptive optimization DBN. The proposed DL model with multiple hidden layers learns the behaviour pattern by analysing user logs and can achieve detection accuracy of 97.872% with significant advantages. In [321], a LSTM based insider threat system is proposed. The proposed method uses system log to train the model to dierentiate anomalous behaviour from normal user behaviour. It can be observed that the proposed method perform better that the existing log based anomaly detection techniques. In [322], Teng et al. have proposed a CNN based user authentication technique which uses the dynamic behaviour of the mouse. The proposed model learns from the dynamic behaviour pictures and can authenticate user continuously for every 7 seconds. It has FPR of only 2.94%.

TABLE XVII
A SHORT REVIEW ON DEEP LEARNING APPLICATIONS IN INSIDER THREAT DETECTION.

Reference	Method	Dataset	Compared CML
[315]	DNN, & RS	CERT Insider Threat	Yes
[316]	CNN, & RS	CERT Insider Threat	No
[317]	RS	Los Alamos National Laboratory (LANL) Cyber Security	Yes
[318]	RS	CERT Insider Threat	No
[319]	RS	CERT Insider Threat	No
[320]	DBN	CERT Insider Threat	No
[321]	RS	CERT Insider Threat	No
[322]	CNN	Balabit Mouse Dynamics Challenge	Yes

J. Deep Learning and Cyber Security in autonomous vehicle technology

Everything in this day-to-day life is becoming connected with each other. One of the most important among them is autonomous driving. In the foreseeable future, they can see the streets flooded with autonomous driving cars due to the initiation of companies like Tesla, Waymo and several other major companies and startups. Since autonomous cars have more similarities with a modern smartphone than a traditional combustion engine car, it raises the question of cyber safety, security robustness and hackability of the system that runs these autonomous cars. Cyber attacks on cyber-physical systems like CAN (automotive Controller Area Network) has been shown to be potentially vulnerable. The possible attacks, vulnerabilities and exploitatios for autonomous vehicles was briefly outlined by [599]. [616] has made an analysis of this CAN data broadcasts and have tested multiple statistical methods to detect the anomalies in the CAN traffic in time windows which will yield a valuable collection of data. The built-in DNNs of a typical modern-day autonomous vehicle system often may demonstrate potentially fatal incorrect errors. To solve this, DeepTest [617] systematically explores various parts of the logic of DNN. It acts as a tool for

automated testing of DNN-driven autonomous cars which acts as an important step towards building robust systems based on DNNs. While [617] is using a module that surveys the DNN network, [618] has taken the route of using adversarial deep reinforcement learning algorithm in order to make the system more robust in the training stage itself unless DeepTest corrects itself in real-time. It is true that the security of the system that is functioning in real-time in autonomous cars are important but on the other hand enhances the in-vehicular networks are also vital for the overall safety of autonomous driving suites. [619] implemented an intrusion detection system (IDS) using deep neural networks for the same. A short review on DL and Cyber Security in Autonomous Vehicles is reported in Table XVIII.

TABLE XVIII
A SHORT REVIEW ON DL AND CYBER SECURITY IN AUTONOMOUS VEHICLES.

Comments	Architecture	Data set
[616]	Hybrid	Udacity Self-Driving Challenge Dataset
[617]	Reinforcement Learning	Private
[618]	DNN	Private

K. Social media data for Cyber Security

Social media is a concept that came into limelight in the recent years. Peoples got introduced to social media platforms like Google Talk, Orkut, Facebook, Twitter and WhatsApp etc. which gained popularity at an exponential rate. Along with the several advantages of social media, the biggest of which is real-time communication, regardless of the distance between those communicating, came the negative aspects like cyber stalking, cyber bullying, hacking, anti-social elements spreading their propaganda and ideologies, and even the spread of fake news. Ongoing investigation on the US election 2016 is one of the greatest instances of their time. It isn't just professed to have been fixed by the Russians but social media has likewise assumed a huge role in it. Summing up all these negative aspects of social media, anonymity has turned into a common aspect which has the capacity to serve as a common helping hand in accomplishing these insidious goals. Anonymity is a treacherous enemy to the decorum of any online community. Aside for North Koreans, every individual has access to Internet and are utilizing their freedom to hide behind fake profiles on social media platforms. Due to this fake profiling, the field of cyber forensics faces the challenge of Author Profiling.

In recent days, the data from social media resources such as Facebook, Twitter, etc. can be effectively used to create cyber threat situational awareness platform. This can enhance the malware detection rate. A characteristic of DDoS defense is that rescue time is limited since the attack's launch. In [244] aims at the prediction of the likelihood of DDoS attacks by monitoring relevant text streams in social media, so that the level of security can be dynamic aimed at cost effectiveness. [245] aims at creating a novel application of NLP models to detect denial of service attacks using only social media as a source. It evaluates two learning algorithms for this task, both of which outperform the previous state-of-the-art techniques - a FFN and a partially labelled LDA model. [246] studied the clusters of Twitter users tweeting about Ransomware and Virus and other malware since 2010. Investigating the quality of the information on Twitter about malware, the paper concludes that a great quality and there is a great possibility to use this information as the automatic classification of new attacks. There is very less number of works existing in this direction because still in its infancy stage. An improved method based on LSTM and CNN approach for DDoS attacks prediction in social media proposed by [247]. To detect DDoS attack in Social media [248] adopted the DL approach. In [249] ransomware detection and classification was done on Twitter posts using DNN with embedding method. The proposed method continuously monitors the online posts in Twitter and provides early warning about ransomware spreads. In [615] utilized the Twitter resource to detect and classify the ransomware events. The proposed architecture acts as cyber threat situational awareness which uses DNN architectures to detect and classify the tweet into corresponding ransomware family. Recurrent structures such as RNN and LSTM were employed for encrypted text classification [595]. To convert the data in encrypted form to numeric representation, Keras embedding was used. The performances of both the architectures are evaluated on both character and word level text representation.

XII. DEEP LEARNING IN IOT APPLICATIONS OF SMART CITIES

Governments of different countries all around the world want to make their urban area more livable, sustainable and productable. To achieve this, they promote smart city projects which use Internet enabled devices. These devices are used in lot of cities major applications like power grid, water treatment etc and due to which the market of Internet-of-Things (IoT) is rapidly growing. The IoT has become increasingly popular and innovative. IoT networks of various cyber physical devices with some storage capacity and processing power which collaborate, associate, exchange information and generate lot of data typically know as BD. The principle objective of IoT is to make secure, reliable, and fully automated smart environments e.g., buildings, smart homes, smart vehicles, smart grids, smart cities, smart healthcare, smart agriculture, and so on. Be that as it may, there are numerous technological challenges in deploying IoT. This incorporates connectivity and networking, timeliness, energy and power consumption dependability, privacy and security, compatibility and longevity, network/protocol standards, and so on with respect to resource-constrained embedded sensors and devices. As these devices have very low security, they pose a great threat to smart cities. It is very easy for attackers to gain access to these IoT devices. Different types of attack like Denial-of-service (DDoS), Distributed Denial-of-Service (DDoS), DDoS as a service, botnet etc. are already implemented using IoT devices by hackers to commit crimes [63]. Botnet attacks basically take the control of IoT devices to make these devices

weaponized. Later these compromised IoT devices are used to launch attacks such as DDoS attack. These attacks can be used to not just invade corporate networks, but also to endanger lives or cause widespread panic across new smart city.

Security policies such as network and host level systems, firewalls, and heuristics based approaches can be employed to secure the IoT environment from malicious activities. Along with these methods application of ML and DL have been employed in academia for IoT security. DL approaches have been applied for IoT botnet detection [64], [65]. Meidan, Yair, et al. proposed an empirically evaluated network-based DL approach for detecting attacks launched from compromised IoT devices [64]. Anomaly detection is performed using these DL methods. Training a DAE with IoT's typical behavior for each IoT device was done to gain proficiency. This was done utilizing behavioral snapshots of the IoT traffic. The DAE endeavors to compress snapshots. The IoT device is said to be compromised when an AE fails to recreate a snapshot. For training and validating the credibility of the system, a new well-structured dataset, Bot-IoT was proposed by [65]. This dataset is a combination of various types of attacks with legitimate and simulated IoT traffic. For addressing the existing dataset downsides, a testbed environment was also introduced. This dataset was made available for further research.

XIII. Deep Learning with Blockchain technology for Cyber Security

DL technology have state of art algorithms which have been employed in various applications in wide range of areas such as Natural language processing, speech recognition etc., and had achieved high accuracy. However, when it comes to Cyber Security area, DL algorithms applied to applications is still in its beginning stages. The major reason to this is FPR. On the off chance that there is even a little blunder as a result of false positive in security territory, it will be an incredible misfortune to the company that is the reason behind organization taking a step back to implement DL algorithms to security area. Not just this, the framework ought to be in adversarial environment which implies that any hacker should not have the capacity to circumvent the security.

Privacy preserved DL is a strategy in which neither the model nor the training data ought to be exposed to the outside world. DeepChain which is a robust and fair decentralized platform for secure collaboration of deep training was proposed by [68]. Three important goals namely, auditability, confidentiality and fairness were achieved in this work. A prototype of this proposed work was made and evaluation on feasibility was done utilizing four different aspects in particular, throughput, ciphertext size, training time and training accuracy. If the distributed computer nodes are compromised, they will be exposed the algorithm and data to various Cyber Security threats. Alternating Direction Method of Multipliers (ADMM) is a tool for distributed optimization problem can be used for detection of the attacks on these models [70]. [71] discussed the detailed analysis on potential attack vectors for generalized distributed optimization problems, with a focus on the ADMM.

Blockchain technology can be defined as a realistic solution for centralization as it enables decentralized and secure public ledger on multiple computers. The reason behind the popularity of this technology is that it prevents any identity theft, data breach, criminal attacks and so forth. It has turned into an amazing advancement that is bringing extraordinary changing in different businesses, for example, healthcare, manufacturing, financial service and son on. It basically ensures that data will be secure and remains private. A comprehensive survey on using blockchain technology for various security related services was done by [66]. This paper gave insights on the present security service, featured methods that provide these services and their challenges. Furthermore, they discussed how blockchain is able to resolve these challenges and compared various blockchain based methodologies providing security services. Additionally, discussion on the current challenges that are restringing blockchain technology in security services was also done. In real-time, to train a model of DL, we have big datasets which are actually available across different data servers. The distributed solution offered by the blockchain technology to maintain tamper-resistant system. This system has the ability to provide various solutions to tackle security problems [69], [67]. Blockchain technology can utilize its decentralized and coordinated platform as computing power for this BD. It will likewise make the DL decisions or outcomes to be more trustworthy, transparent and explainable. Additionally it provides secure data sharing environment. [72] proposed a decentralized blockchain-based architecture for DL applications. Recent days to enhance the performance of the ID model, security researchers use collaborative ID networks. The main significant issues with such IDS model is data and trust management and this can degrade the effectiveness of IDS model. To avoid this, [73] showed application of blockchain technology in IDS.

XIV. Deep learning for Cryptography

Data security has become highly important as more and more electronic communication devices are using Internet for communication. Cryptography is a methodology though which data can be kept secret while communication. It basically converts the data into unreadable format that is not understandable by anyone except for the authorized individuals. This converted unreadable data is safe enough to be transmitted to the right user though Internet and can be decrypted to the original form. Modern cryptography major aspects are data integrity, authentication, data confidentiality, and non-repudiation. It can be broadly classified into two types: symmetric-key (single key) and asymmetric-key (public and private key). In recent year, states of art DL algorithms have been applied for cryptography [137]. The data encrypted using DL cannot be broken down without the right key. Even though DNNs have high computation cost, they are very useful in applications of smart cities. With the help of Quantum computing in future, this computation cost will be very minimal and breaking the encryption would become possible. NNs are trained for encryption and decryption adversarially [138]. This was shown with a case study. They also reported that the NNs

are good at cryptanalysis and can be further considered for steganography, pseudorandom-number generation, or integrity checks. In [139] discussed the theoretical aspects of applying NNs to encrypted data. [140] discussed the applicability of NNs in cryptography and steganography. [141] showed various experiments how and in which conditions an artificial agent can learn a secure encryption method. The detailed experimental analysis on RNN based methods for cipher design was done by [142]. In [143] showed detailed experimental analysis of NNs on the design of S-boxes used in ciphers. Cryptographic Primitive classification based on CNN was done by [144]. CNN based method was proposed which has the capability to run over encrypted data [145]. In [146], a method was proposed which uses NNs for encrypted data typically called as cryptonets. This allows the user to send the encrypted data to the cloud. In cloud, the DL models predict without decrypting the data and against the results are passed into the user. At last the user decrypts the data using a cipher. In this study various experiments were done for cryptonets to enhance the performance [147].

XV. Deep learning for Cloud Security

Internet is being reformed by cloud computing. Cloud computing is a convenient, on-request network accessibility to huge amount of shared, configurable computer system resources. The computing resources that are provided are data storage, servers, services, applications and so on. Cloud computing is picking up popularity as it not only saves cost but also there is no requirement for any immediate active administration by the user. The three important aspects of cloud computing are Software as a Service (SaaS), Platform as a Service (PaaS), and Infrastructure as a Service (IaaS). To deploy applications or use services, there is no need for user to have any knowledge, control, and ownership of the computer system infrastructures. The user can simply rent and use the hardware or software by paying some minimal amount for what they used, from anywhere, at any time. The servers should to be in secure environment which means they should not be accessed by unauthorized users. Utilization of firewalls and ID can secure the servers but in recent year's lot of methods have come into picture which can bypass the firewall very easily. This can be avoided using host level as well as network level ID mechanisms.

In recent years, ML and DL techniques have been used in cloud computing. In [148], ML feasibility for cloud computing is studied. The real hinders for the acceptance of cloud computing are issues related to trust, privacy, and security. Google which provides cloud platform puts security as the top concern. They use ML and AI not only to block wrong messages but also show warning for suspicious messages.

Cloud computing is seeing the era of BD. BD is typically a very large amount of data. This data can be stored in cloud and cloud servers can use various DL architectures for training this BD where as the standard approaches presently require centralized data training methodology. Google has already built secure and robust cloud infrastructures for processing this data. This is basically making ML as-a-service (MLaaS) and DL as-a-service (DLaaS) possible. DLaaS provides users to design, develop and train DL applications faster. DL as a service is provided by various cloud platform such as Microsoft Azure ML [149], Google Cloud ML Platform [150], Amazon ML [151], IBM Watson Analytics [152] and so on. IBM's DLaaS software architecture details are provided in [153]. Virtual machines are one of the important components in infrastructure as a service (IaaS). Various experiments based on heuristics were done for VMs network traffic anomalous behavior detection [154]. They also suggested that the application of DL can be used for increasing the accuracy of the model. In [155] proposed LSTM based method for anomaly detection in a cloud environment. DL based ID for vehicles was discussed in cloud environment [156]. RNN based ID was proposed for cloud environment in the context of BD approach [157]. In a DL approach was proposed for cyberattack detection in mobile cloud computing environment [158]. A method based on CNN was proposed for malware detection in the cloud infrastructure [159].

XVI. Deep learning for Biometric security

Biometrics security deals with identifying a person by their unique characteristics. Physiological characteristics and behaviors characteristics can be used for this purpose. Physiological characteristics include face, fingerprints, palmprints, iris etc., whereas behaviors characteristics involve voice, signature, gait, keystroke and so on. By using these measures it is extremely difficult to break into any system making it very useful to be used for security purpose like access to confidential data. From data, DL algorithms can learn hierarchical features making them very popular with multiple fields such as speech, natural language processing, computer vision etc. Surveys on DL for Biometrics have been done in [160].

XVII. Deep learning based Cyber Security in fog computing

Fog computing, an extension of cloud computing technology, is the new emerging technology of Cyber Security. In fog computing the information exchanged between the end users is through the fog layers which processes the input information by changing structure, size and validity of the data. The data collected at this level will be very large. These layers have fog nodes which are from various providers which makes is difficult to trust them. This fact leads to new privacy and security problems in fog computing. In [169], utilizing a set of application scenarios, the authors have highlighted the privacy challenges and data security issues in the fog layers. This upcoming technology is very much needed for application of smart city, e-Health, smart homes, mobile applications, and so on. In [166] gives a great introduction on analysis of BD for security in this new emerging technology. [167] provides a great detailed view on fog computing, DL, and BD.

XVIII. Deep learning and Cyber Security in Pervasive computing

Pervasive computing is a new simple technology which is trying to make embedding systems to be available everywhere at any time. It is also called as ubiquitous computing. It

is trying to increase embedding computational capability in everyday things so that they can communicate and perform more efficiently. Pervasive security deals with security in this new emerging technology. A brief discussion on the current stage of pervasive security and open problems in this area is done in [170]. [171], [174] gives a complete study on security in this new technology. Experiments using falsification and singleton invariant were done to identify if the light in refrigerator is off when the door is closed were done in [172] to see the security in pervasive computing. In [173], experiments were performed to prove that monitoring, evidence gathering and reconciliation are a better way for security. To address the challenges faced by pervasive security, [175] proposed a method which combines decentralized trust and reputation management systems, network-level observations and Semantic Web languages declarative policies. DL framework have been applied in this domain to process and predict the data coming from multiple sensors and how these devices can be used to implement these architectures. MicroDeep is basically a CNN over a distributed sensor network which was proposed in [176]. They demonstrated two experiments which show that MicroDeep performs better than ordinally CNN. In [177] shows how DL frameworks can be squeezed into these small embedding devices. DAEs were utilized to detect suspicious network traffic of compromised IoT devices so that IoT botnet attacks can be identified [178].

XIX. Unsupervised machine learning leads to build better Cyber Security system for an organization

Unsupervised learning is a ML approach of inferring a function to describe the hidden structure. Recently there has been a lot of interest in unsupervised learning methods to understand and learn the representation of words, popular methods like word2vec embedding model, which learns the syntactic and semantic representation of a word. Unlike the openly available datasets, most of the real-life datasets are often unlabeled or poorly labeled. As a result, supervised learning and CMLAs can't be trusted. Therefore a new wing of ML has been born called unsupervised ML to evade this disadvantage. One of the classical examples of unsupervised learning is the ANN. [49] has demonstrated a novel approach for IDS using unsupervised learning proving the advantage of it in the field of Cyber Security where the data is almost always unlabeled. The use of unsupervised learning approach is not just in IDS. Like the approach in [50], the ability of it can be exploited in signature extraction which is the key part in forensic log analysis. They have proposed a method based on a neural language model that has promisingly outperformed the current signature extraction techniques. [51] has developed an enterprise-grade framework that uses a divide and conquer strategy combing the analytics of behavior and modeling of time series. This approach has achieved an area under the curve receiver operating characteristics curve of 0.943. In [52], the author proposed a method based on a sparse variant of DBN that holds promise for modeling of higher order features. [53] proposes a real-time collective anomaly detection architecture based on NN learning that is built on time series version of KDDCup-99 dataset.

Tensors are multi-dimensional arrays that contain numerical values and hence generalize matrices of more than one dimension. Tensor decomposition is a method of representing a complex tensor in the form of one or more simpler tensors for easier manipulation and understanding. [54] has developed a joint probabilistic tensor factorization method to derive the latent tensor subspace, which extracts common behaviors that vary in time across the views. By doing so, they have achieved significant results in temporal multi-view inconsistency detection for network traffic analysis. It is often difficult to find dense blocks when the tensor is of complex and high order. The current decomposition techniques that are used for finding dense blocks are not satisfactory with respect to accuracy, speed and flexibility. Therefore, [55] has developed a solution called M-ZOOM that gives promising results in terms of scalability, accuracy, flexibility and effectiveness with an AUC score of 0.98. Due to the fact that it is often time-consuming to do complex, multi-dimensional tensor decomposition which cannot be afforded in real-time Internet anomaly detection with high accuracy. Therefore, [56] has proposed TensorDet which can solve the problem directly and efficiently. It exploits the factorization structures with novel methodologies like sequential tensor truncation and two-phase anomaly detection.

XX. Cyber Security applications in off-line and real-time deployment

The application of DL architectures towards Cyber Security is in the initial stage and there are many challenges involved in both off-line and real-time Cyber Security applications. There are many important factors that should be considered during designing ML based system. They are;

1) **Interpretation and understanding of Deep learning architectures:** Interpreting and understanding what the trained ML model has learnt is an important factor of a robust validation procedure. Interpretability is an essential factor in applications related to Cyber Security where the reliance of the model on the correct features must be guaranteed. Generally, the simple models are easier to interpret than the complex models. Simpler models are linear models whereas complex models are non-linear models. Interpretation which means the human has to be able to understand what the predictions are, for example texts, images etc. whereas non-interpretable are hidden layer features, vectors spaces produced by text representations, word embedding. Heat map is one of the most commonly used approaches to understand the classification decision. The pixel of heat map image provides the contribution towards the classification.
   There is no clear mathematical proof as well as theory to DL architectures interpretation and transparency. Thus it is very difficult to arrive at a specific reason to identify why DL architecture model misclassify a data sample. Identifying which DL architecture is more suitable, identifying optimal parameters for network structure and

network parameters is one of the daunting tasks. Additionally, more practical knowledge is required to identify sensible values for parameters such as learning rate, regularizer, etc. Currently these are determined on an ad-hoc basis. [58] proposed a method to identify the optimal number of feature maps. This method worked well for extremely small receptive fields and later [59] proposed a visualization approach that facilitated for intermediate feature visualization. Following, [60] proposed a visualization method for hidden layer feature visualization. These visualization methods facilitated to design better DL architecture. Later many methods are proposed for gradient visualization, these are explained in detail by [61]. Recently, the detailed survey on visualization and visual analytics in DL is done by [62].

2) **Unavailability of well-known labeled benchmark datasets:** Dataset is one important component in ML. Due to privacy and security reasons, the labeled datasets are not publically available for research purpose. Labeling data samples by using manual approach is one of the daunting tasks. Mostly, to label the data samples heuristics method is followed. Most commonly used solution to label data sample is based on vendor provided blacklist and whitelist. Basically there are 3 different types of datasets are used in ML. They are called as train, valid and test datasets and these datasets are disjoint to each other. It means when we are collecting a data sample to develop network traffic analysis system, we have to collect these datasets from different networks which include different users as well as different applications access. These 3 datasets should also include time information. It means the train data should be from $t-1$, test dataset should be from $t+2$ and valid dataset from $t+1$. Anomaly detection is more popular in many domains and less preferred in the area of Cyber Security. This is due to the reason that achieving low false positive is one of the biggest tasks in Cyber Security anomaly detection. There are chances where a single misclassification can cause millions of dollars damage to the company. The semi-supervised and mostly unsupervised learning methodology is the prefer method in the domain of Cyber Security. The main important factors to be considered during dataset collection are

   a) Different qualities of measurement
   b) Different subjects
   c) Evolution of technology over time
   d) Different ways of labeling examples
   e) Different level of concentration
   f) Different environments
   g) Different protocols
   h) Time of the day

3) **Attacker-Defender Approach and Concept of Drift:** Cyber Security is an evolving area, to adapt to new types of patterns used by adversary; the ML based system has to be continuously trained. Since the datasets generated by various ICT systems is huge, feature engineering is a difficult task, thus in this case application of DL architectures can be used. This helps to lean the different types of new patterns used by an adversary simply following pertaining method.
4) **Imbalanced data samples:** Data imbalance is one of the most common problems in Cyber Security. Most of the time the samples of malware are rare and particularly almost all the data's are imbalanced in multiclass classification in the field of Cyber Security.
5) **Domain adaptation:** Domain adaptation is a method to measure the difference between train and test datasets. Both of these datasets distribution should be completely different. The domain of Cyber Security contains many forms of datasets; this includes network traffic, spam, phishing, etc. These are highly correlated and can help to detect malware effectively. A major challenge is to adopt an effective defense method from one domain to another.
6) **Important factors to be considered in deployment of ML models in real-time systems:** Though as these ML algorithms and DL architectures have the capability to discriminate the new types of malicious patterns, there are still is in early stage in adopting in enterprise security systems. Recently, a new research direction typically called as explainable AI can give better reasons for incorrect decision. The incorrect decision in Cyber Security system can cause dollars of damage. For example If a legitimate application is flagged as malicious and the application is not acceptable by any of them in a working hours in an enterprise system, then it is going to cause a lot of damages. The explainable AI can better understand the complex problems. Interpretability is crucial for CMLAs and DL architecture because a single wrong decision can be extremely costly. Generally, DNNs learn hierarchical feature representations. Each layer has multiple neurons with similar structure but with different weight parameters. In the presence of the data heterogeneity in Cyber Security systems, it can be tricky to ensure that the classifier uses the right features. Interpretable ML model can be used to validate a trained model, or to learn something from the models. Variation in the prediction can be learned by using sensitivity analysis. It also discusses the importance of interpretable DNNs modeling explaining the predictions.

XXI. Role of Explainable Artificial Intelligence in Cyber Security

In the previous decade, AI systems have begun to perform tasks that previously needed a human's cognitive skills. These systems can perform specific tasks quicker and better than a human, due to which humans have started to rely on these systems to be in charge for making major decision in real-time. Nonetheless, the fundamental principle that these framework are utilizing to settle on these choices is often hidden such as for what reason did the AI do what it did, how the choice was made, etc. For example, movie suggestions are made by film streaming services however most of the times the clarification

of how the framework chose these particular motion pictures isn't given. Situations like these makes the machine generated decision to be less confident, less trustworthy and so forth. Trust is one of the major factors when safety comes into picture and this trust on decision made by machine over time can be improved by Explainable AI [80]. Explainable AI is a possible and desirable concept where humans can access the decision making procedure of the AI [81]. In other words it will give reasons and explanations for everything that is happening by the algorithms such as the reason behind the outcome. The biggest challenge here is to create machines which humans can trust. A detailed survey and study an explainable AI is done by [82].

DL is a state of art technique where models are composed of multiple layers. Which actually resemble the human brain. In [74], interpreting the concepts learned by the model and understanding the model decision of DNNs is well presented. DL is still in beginning stage in Cyber Security area especially in the case real-time deployment application.

In recent days, Explainable AI has been applied in Cyber Security applications. In [75], an adversarial approach with Explainable AI was utilized to explain the reason for the incorrect classification given by the ID system. Initially, the minimum modifications needed in order to correctly classify the misclassified values were discovered utilizing the methodology. Later these recently discovered modifications were utilized to visualize the related features which were responsible for the incorrect classification. Linear and MLP models were utilized for the experimentation and intuitive plots were used for displaying the clarifications by the author in this paper. The outcomes demonstrated that conflicting characteristics between classes was the reason behind the incorrect classification. In [76], a survey to improve the generalization capability of DL Cyber-Physical Systems was done using Regularization techniques. The DL architectures used in this work are CNNs, LSTMs, RBMs, AE, DBNs and DFFNN whereas regularization techniques exploded in this work are weight decay, dropout and sparse regularization. In [77] gives an overview of the security and resilience related to a critical infrastructures. An introduction to organizational units dealing with Critical Infrastructure Security and Resilience (CISR) as well as different topics related to vision and future of CISR are discussed in this paper. LEMNA is a method which treats a DL model as a blackbox to derive explanations for each and every classification outcome [78]. This method was dedicated to security application. Various interpretation and visualization methods are mapped to 3 different tasks and detailed analysis is shown [79].

XXII. Casual theories with deep learning for Cyber Security

State of art DL algorithms have been applied to various application of Cyber Security and have got great accuracy but still are not widely used commercial uses. This is due to fact that there is no trust in the outcomes predicted by these models and how it is working. In order to establish trust, it is very important to formalize a framework to understand the working of these models. Causal inferences are one of the methods which are used for this purpose as it has the ability to answer such questions. Causal model includes a statistical model and additional structure which has the capability to answer the questions related to distribution and interventions changes. What if type of questions which involve making some changes to the existing framework can be answered using this type of model. This causal theory to understand DL models have been applied to different application of Cyber Security. In [161] proposed a framework to understand a DL architecture using causal inferences. They not only built a structural causal model but also showed the effectiveness of this model. In [162], Granger Causality was used to confirm the TCP flooding attacks. This work was based on analyzing causal data in network traffic to find the presence of TCP-SYN flooding DDoS attacks. Causality countermeasures were utilized for detection of attacks in [163]. PRIOTRACKER was proposed in [164] which can utilize for tracking processes. It basically prioritizes the investigation of abnormal causal dependencies. [165] demonstrated a developmental learning algorithm for understanding a set of causal models which describes Cyber Security.

XXIII. A brief statistics of Deep learning applications in Cyber Security

Recent years applying novel DL methods and as well as evaluating the performance evaluation of various existing DL architectures to find out optimal one has been remained as a significant direction of research for security researchers. The surveyed DL based Cyber Security applications papers are shown in Figures 13, 14, 15, 16, 17, and 18. Figure 13 represents the statistics of the DL architectures for various Cyber Security applications. Since there are many DL architectures, we have grouped similar architectures to a group and the details are given below.

1) DBN, RBM (all other variants of DBN and RBM related architectures.)
2) DNN - a network with many fully connected layers with $ReLU$ (includes Invincea, Endgme, NYU, CMU, MIT.)
3) Autoencoder (stacked autoencoder, denoising autoencoder, variational autoencoder, contractive autoencoder and other variants of Autoencoder.)
4) CNN (this includes all cnn architectures like (AlexNet,VGG16,VGG19,SqueezeNet,Inception-BN-21k,Inception-BN-1k,Inception V4, ResidualNet152.)
5) Recurrent structures - RNN, LSTM, GRU, and IRNN.
6) Hybrid of CNN and Recurrent structures - CNN-RNN, CNN-LSTM, CNN-GRU, CNN-IRNN
7) Bidirectional recurrent structures - Bidirectional recurrent neural network, Bidirectional long short-term memory, Bidirectional gated recurrent unit and other variants of Bidirectional recurrent structures.
8) Reinforcement learning (includes variants of Reinforcement learning).
9) Adversarial machine learning (includes various Cyber Security applications based on Adversarial DL.)

Figure 13 shows that the DL architectures based on recurrent structures, CNN and DNN are largely used. More

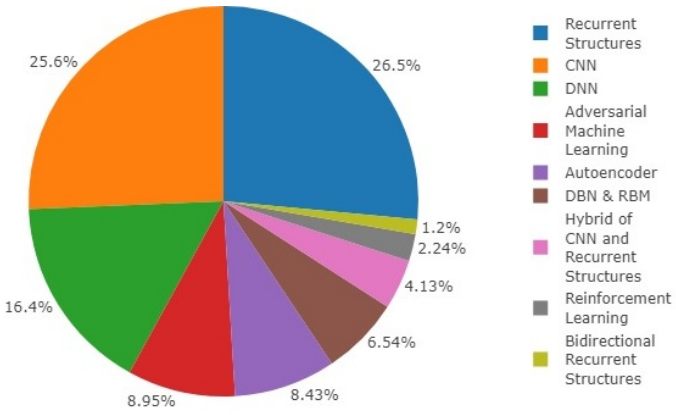


Fig. 13. Statistics of the Deep learning Architectures in Cyber Security.

importantly, DL architectures based on recurrent structures is largely used compared to CNN and DNN. This may be due to the reason that most of the Cyber Security applications datasets involves the sequence and time series information. DL architectures related to recurrent structures are good at learning sequence and time related features compared to other DL architectures.

Figure 14 represents the statistics of various Cyber Security applications using DL. We have considered the following Cyber Security applications;

1) Windows malware detection
2) Android malware detection
3) Intrusion detection
4) Network traffic analysis
5) DGA, Email, URL and Security log Data analysis
6) Side channel attack detection
7) Insider threat detection
8) Function recognition
9) Steganalysis and Steganography
10) Insider threat detection
11) Attacks detection in autonomous vehicles
12) Security events detection in social media
13) Cryptography applications

Application of DL architectures are largely used in intrusion detection, Windows malware detection, Android malware detection and DGA, Email, URL and Security log data analysis. Most importantly, DL architectures are largely used for intrusion detection. This has been remained as a significant direction of research for the past several years.

There are many research works exists based on DL based Cyber Security applications. However, the main significant issue is that most of the published researches have not compared the performance of DL architecture with the existing classical ML algorithms. This is very much required because for few Cyber Security applications the classical ML algorithms are more sufficient than the DL architectures. To identify this, we have shown the statistics of the various DL based Cyber Security applications in Figure 15. We have considered the following Cyber Security applications;

1) Intrusion detection
2) DGA, Email, URL and Security log Data analysis
3) Network traffic analysis
4) Windows malware detection
5) Android malware detection
6) Side channel attacks detection
7) Insider threat detection

Figure 15 indicates that only few published research works based on DL architectures have compared the results with classical Ml algorithms. Most importantly, the DL architectures outperformed the classical ML algorithms in all the research works. As a result, this facilities to understand DL architectures are more efficient and robust than the classical ML algorithms.

NLP is an important domain which has many important application in Cyber Security. It deals with conversion of text to numerical representation. There are many text representation exists and the performance implicitly depends on the text representation. In the last years, various DL based published papers have used various text representations. We have shown the statistics of various text representation methods of published works specific to DGA, Email, URL and Security log Data analysis is shown in Figure 16. Most of the research works have utilized keras embedding. This is primarily due to the reason that it helps to preserve the sequence information of words or characters in the texts. There are very less number of work exists based on the word embedding models because in most of the Cyber Security text data doesn't involve semantic and contextual representation. We have considered the following text representations;

1) Bag of words (Term frequency)
2) n-grams
3) One hot
4) ASCII representations
5) Keras embedding
6) Word2vec (Sent2vec)
7) FastText
8) Characters converted into image
9) Manual feature engineering

Figure 16 shows that the Keras embedding text representation is largely used in published research papers based on DL. This may be due to the reason that sequential features are more important in Cyber Security text data and Keras embedding has the capability to capture sequential information while the other text representations such as Bag of Words, term frequency, and one hot encoding are not capable of capturing sequential features. Word embedding can also learn sequential features in the text but it is computationally expensive compared to Keras embedding.

The DL based Cyber Security applications published research works are estimated for the years, 2000 to Mar, 2019 and published articles for 2000-2012, 2013, 2014, 2015, 2016, 2017, 2018, and 2019 (till March) years statistics is represented in Figure 17. It shows that the DL applications towards Cyber Security has evolved along with the time and the number of DL applications in Cyber Security gradually increased over the years.

The datasets are important and plays an important role in development of DL based Cyber Security applications. We

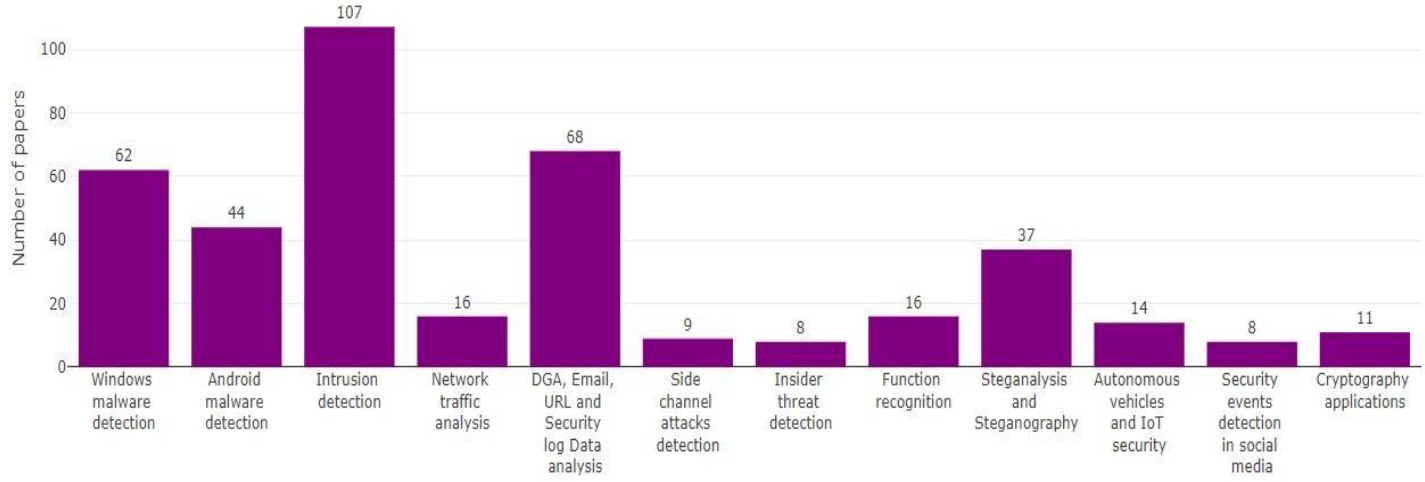


Fig. 14. Statistics of the Deep learning based Cyber Security Applications.

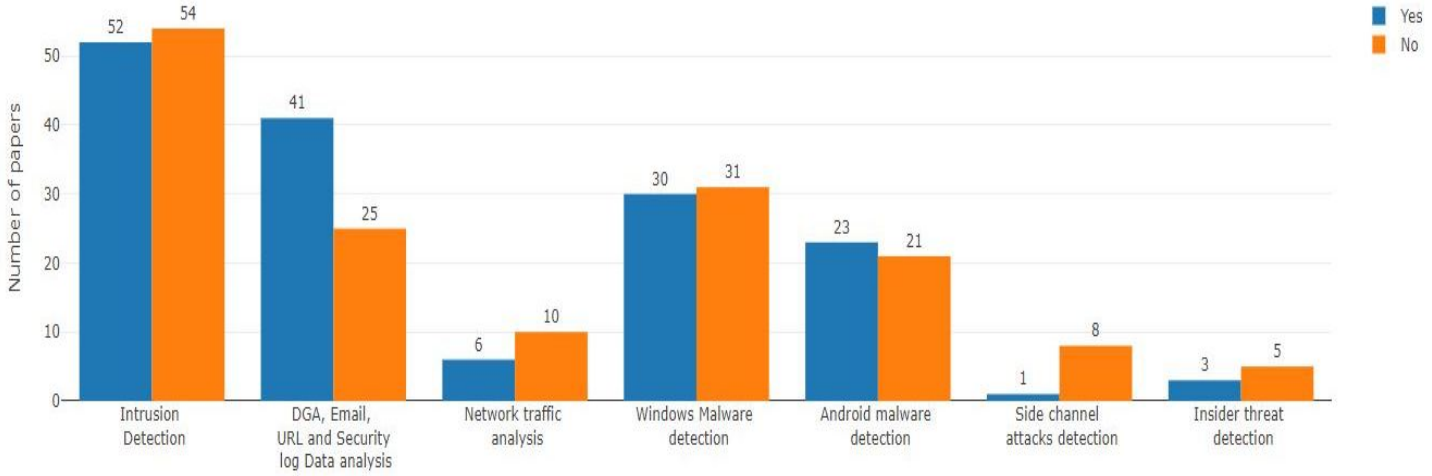


Fig. 15. Statistics of Deep learning approach for Cyber Security Applications based on comparison with classical machine learning algorithms.

have discussed the major issues exists in the various available datasets in detail in the above section. The datasets used in the existing deep learning based Cyber Security application published works are categorized based on the following;

1) Benchmark: These datasets are available publically for research purpose. These datasets can be used for performance evaluation of existing and as well as newly introduced algorithms.
2) Collected from publicly available sources: These datasets are collected from various publically available sources. In most of the cases these datasets are not publically avilable for research purpose.
3) Private: The datasets doesn't belongs to benchmark, collected from publically available sources and real-time category are considered as private datasets. These datasets are not publically available for research purpose.
4) Real-time: The datasets are collected from real-time environment are considered as real-time datasets.

The detailed statistics of the DL application in Cyber Security based on dataset type is shown in Figure 18. While

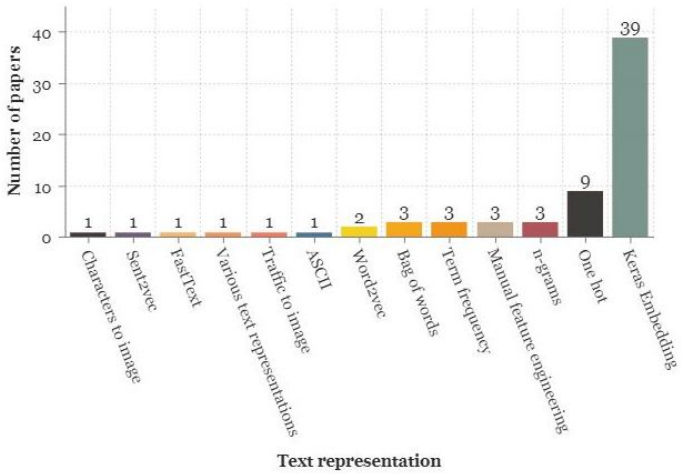


Fig. 16. Statistics of Deep learning approach for DGA, Email, URL and Security log Data analysis based on comparison with NLP text representation.

most of the research works have used the benchmark datasets,

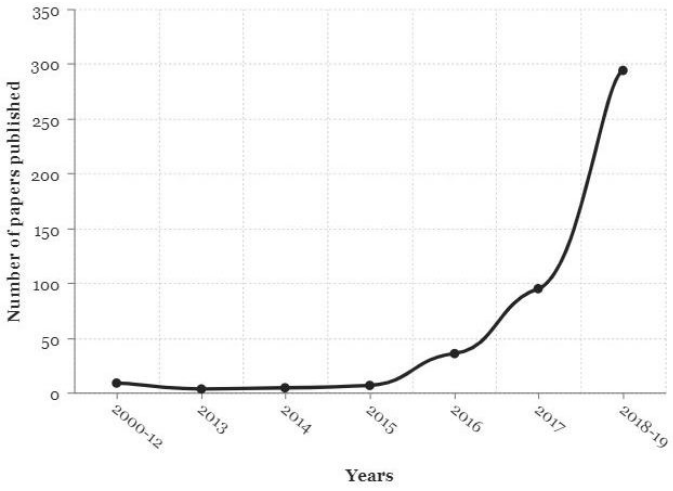


Fig. 17. Dynamics of the Deep learning based Cyber Security Applications per year.

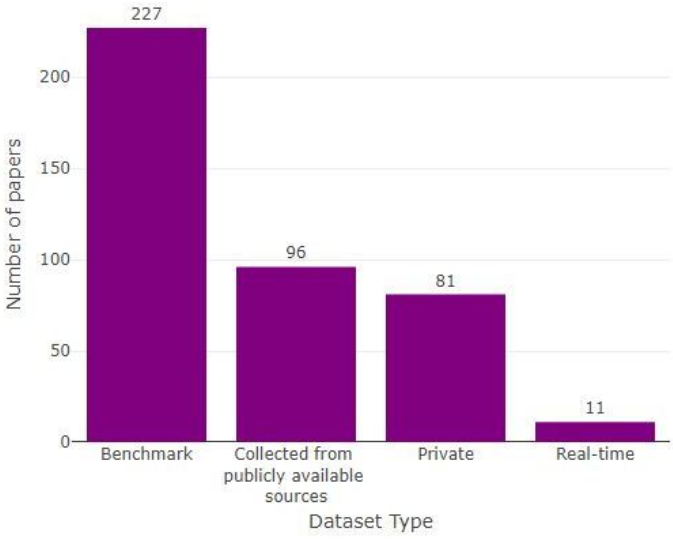


Fig. 18. Statistics of Deep learning based Cyber Security Applications based on whether the datasets are private, collected from publicly available sources and benchmark.

few of them uses dataset which are belongs to the following dataset categories;

- Collected from publicly available sources
- Private
- Real-time

Most importantly, the results obtained in research works based on the benchmark datasets can be reproduced, compared and enhanced in future. Thus, the need for benchmark datasets is critical for advancing DL applications in Cyber Security.

XXIV. Suggested Hybrid systems for organizational Cyber Security

Cyber criminals are in constant improvement of their tools and knowledge. A huge amount of data is collected from different sensors about user behaviors from different sources over the Internet and these data can be processed using DL methods to monitor and trace them in real-time. The massive amount of data needs new tools and technologies to preprocess and apply DL methods on them. To achieve this, highly scalable distributed computing platform can be used. The proposed tool can collect data in a distributed manner and use distributed algorithms to analyze data with the aim to detect and classify security events. This facilitates the user to know whether the administrator has to take an action. DL architectures are used to not only process and find patterns but also to interpret data with the aim to find the degree of risk each threat. The tool monitors the network and the devices which are connected to see for variations from normal events tell you whether those attacks actually are hitting or already compromised their systems. The proposed model has the capability to detect an attack is being detected. It is designed to provide organizations with the situational awareness needed to deal with their most pressing issues.

Based on the knowledge obtained from the surveyed papers, we propose a general DL framework for Cyber Security applications. This framework has multiple layer of security with the aim to detect malicious activities more accurately. The proposed framework is considered to be as generic as possible and it is a hybrid of many Cyber Security modules with the aim to meet today's Cyber Security challenges. Primarily the proposed DL framework contains Data collection, Data preprocessing, DL based classification modules. In the Data collection phase, the data samples are collected passively from various sensors and stored in NoSQL data base. These raw data samples were further passed into preprocessing module which extracts important information using distributed log parser. Finally, the information will be passed into DL based classification module to detect and as well as classify the malicious activities. This proposed framework is more general to handle various Cyber Security challenges in the modern society. This framework contains the following sub modules;

- Cyber threat situational awareness sub module based on DNS data analysis using deep learning.
- A sub module to analyze the global BGP updates for Cyber Security threat detection using deep learning.
- Deep learning based sub module for Spam and Phishing detection using URL, Email and social media data analysis.
- Deep learning based hybrid intrusion detection sub module which can detect attacks at network and host level.
- Deep learning approach sub module for network traffic analysis.
- A sub module for identification of detailed information on the structure and behavior of the malware using malware binary analysis using deep learning.
- Deep learning based sub module for ransomware identification.
- Deep learning and Visualization sub module for Botnet Detection in the Internet of Things of Smart Cities.
- Deep learning based sub module for Android malware analysis.
- Deep learning based anomaly detection sub module using operational logs in cloud applications.
- Malware spread modeling sub module using Deep learn-

ing and scientific computing models.
- A Casual approach with deep learning sub module for the network anomaly detection.
- Malware visualization sub module using image processing and deep learning.
- Privacy preserved deep learning sub module using block chain technologies.
- Deep learning based Cyber Security sub module for Cloud environment.
- Deep learning based Cyber Security sub module for fog computing.
- Deep learning based Cyber Security sub module for Cryptography.
- GAN based sub module for enhancement of robustness of deep learning models in an adversarial environment.
- Reinforcement learning based sub module for enhancing the system performance through pretraining.

An enterprise Cyber Security system composed of above mentioned sub modules correlatively can detect attacks more accurately and quickly can block the malicious activities communication point between a bot and target host.

XXV. Conclusion

Over the last decade, Cyber Security has become an important area of research due to the explosive growth in the number of attacks to the computers and networks. Mostly, the existing commercial Cyber Security products in markets are based on blacklisting and heuristics methods which completely fail to detect new types of attacks to the computers and networks. Later, machine learning algorithms are employed. This has been remained as an active area of research from the last 10 years. However, the main significant limitation in machine learning is that, the performance implicitly relies on feature engineering. To avoid feature engineering, deep learning methods are employed in the last two years. Primarily deep learning is a machine learning model which helps to learn the hierarchical and abstract feature representation implicitly. This has outperformed classical machine learning algorithms in many Cyber Security applications. Recent days, employing deep learning applications in Cyber Security is an active area of research.

This survey paper presented a short tutorial-style description of classical machine learning algorithms and deep learning architectures. Further, the importance of natural language processing (NLP), signal and image processing, and big data analytics in Cyber Security applications is discussed. These techniques helps to handle very large amount of datasets and to extract important hidden features more accurately. Next, we discussed the information of various deep learning software libraries, major issues of existing Cyber Security solutions, and importance of shared tasks in Cyber Security. Along with the deep learning architecture, the importance of, explainable AI, transfer learning, reinforcement learning and adversarial machine learning applications in Cyber Security is discussed. Next, the literature review of deep learning applications in Cyber Security was carried out. The major objective and limitations were reported while summarizing the literature review of deep learning applications in Cyber Security. Next, the importance of Cyber Security in emerging areas such as Smart cities, IoT, fog computing, cloud technologies, biometrics, pervasive computing is discussed. Applicability of emerging research areas such as blockchain and casuality in deep learning based solutions for the Cyber Security domain is examined. Finally, the surveyed deep learning based Cyber Security applications are summarized according to the type of deep learning architecture, datasets used, year, type of Cyber Security application. The statistics indicates that the deep learning architectures related to recurrent structures is the largely used method. This is mainly due the fact that most of Cyber Security datasets are time series and sequence in nature. Since most of the datasets in Cyber Security are sequence in nature, Keras embedding is the largely employed method to convert Cyber Security texts into numerical representations. Most importantly few published deep learning based Cyber Security applications have not utilized benchmark datasets and also the results were not compared with the classical machine learning methods. This factors do not allow for fair comparison and results reproducibility in future. Finally, we proposed deep learning based hybrid framework which contains different layers of security to learn the characteristics of malware and legitimate activities and evolves in real-time to detect and prevent from advanced attacks. As such, this detailed survey on deep learning applications in Cyber Security provides an overall summary of the work which can motivate researchers to advance the state of deep learning for Cyber Security applications.

Attackers are continuously advancing their methods to develop potential and new kind of malware which can bypass and remain hidden from the existing anti-malware system. DL techniques have the ability to capture clear signal from large volume of security related data to distinguish the legitimate and malicious activities. The various DL architectures were introduced for each Cyber Security applications. However, the robusteness of these methods were not discussed in an adversarial environment. Recent days, attackers follow adversarial machine learning to bypass the deep learning based models. Thus, studying the robustness of the deep learning models in an adversarial environment has been considered as significant direction towards future work.

Due to limited availability of benchmark datasets, few of the published research studies have utilized the private datasets and the modified version of benchmark datasets. Mostly, the private and modified version of benchmark datasets are not publically available for further research. Thus most of their solutions are not directly comparable. This can lessens the research towards choosing best DL architecture. When choosing an effective architecture several checklists have to be considered. They are accuracy, space and time complexity, ability to detect the new malware, easy integration and deployment in real-time system, and robustness in an adversarial environment. However, only few research studies mentions all the checklists in their research study. Another important factor to be considered is that most of the existing DL based Cyber Security applications learning approach is supervised. Supervised learning requires the labelled datasets

and it is fairly expensive to collect them. Thus DL based Cyber Security applications learning methodology is anticipated to be at least semi-supervised and at most unsupervised. To adapt to new attacks, DL architectures require retraining. Thus, Cyber Security domain implicitly relies on incremental learning and continual lifelong learning. The incremental learning and continual lifelong learning with deep learning applications in Cyber Security is an another significant directions towards research in enhancement of security solutions.

Research on deep learning applications in Cyber Security is still in its infancy stage. various good research studies are required to identify an optimal deep learning architecture. Shared tasks are one of the prominent way to push deep learning applications in Cyber Security research forward and organisation of shared tasks is anticipated to be more in the near future. To conclude, deep learning is an important method for all the Cyber Security applications due to the reason that the rise of big data. The data distribution in big data is highly non-linear, noisy and dirty. Classical machine learning algorithms might not suffice to deal with big data and the performance can be very lesser in all various Cyber Security applications. Finally, we report that the deep learning applications can be used on the Cyber Security problems which require to learning complex non-linear hypotheses with large number of features and high-order polynomial terms and in domains with big data.

NOMENCLATURE

AB	Ada Boost
ADMM	Alternating direction method of multipliers (ADMM)
AE	Autoencoder
AI	Artificial Intelligence
ANN	Artificial Neural Network
BD	Big data
BLSTM	Bidirectional Long Short-term Memory
BoW	Bag of Words
BPTT	Backpropagation Through Time
BP	Backpropagation
BRNN	Bidirectional Recurrent Neural Network
BRS	Bidirectional Recurrent Structures
CNN	Convolutional Neural Network
CWRNN	Clock Work Recurrent
DAE	Denoising Autoencoder
DBN	Deep belief network
DDoS	Distributed Denial of Service
DGA	Domain Generation Algorithm
DL	Deep Learning
DM	Data Mining
DNN	Deep Neural Network
DNS	Domain Name System
DoS	Denial of Service
DT	Decision Tree
Email	Electronic mail
FFN	Feed Forward Network
GAN	Generative Adversarial Network
GPU	Graphics Processing Unit
GRU	Gated Recurrent Unit
HIDS	Host Intrusion Detection
ICT	Information and Communication technology
ID	Intrusion Detection
IoT	Internet of Things
IRNN	Identity Recurrent Unit
ISP	Internet service provider
KNN	K Nearest Neighbour
LR	Logistic Regression
LSTM	Long Short-term Memory
MLP	Multi-Layer Perceptron
ML	Machine learning
MT	Maximum Entropy
NB	Navie Bayes
NIDS	Network Intrusion Detection
NN	Neural Network
OS	Operating System
PCA	Principal component analysis
RBN	Restricted Boltzmann Machine
ReLU	Rectified Linear Unit
RL	Reinforcement Learning
RNN	Recurrent Neural Network
SAE	Stacked Autoencoder
SdA	Stacked denoising Autoencoder
SGD	Stochastic Gradient Descent
SL	Supervised Learning
SVD	Singular value decomposition
SVM	Support Vector Machine
TDM	Term document matrices
TFIDF	Term frequency-Inverse document frequency matrices
TL	Transfer learning
UL	Unsupervised Learning
URL	Uniform Resource Locator
VAE	Variational Autoencoder

ACKNOWLEDGMENT

The authors would like to thank NVIDIA India, for the GPU hardware support to research grant. They would also like to thank Computational Engineering and Networking (CEN) department for encouraging the research.

REFERENCES

[1] International Telecommunication Union (2014) the world in 2014: ICT Facts and figures. Technical report
[2] 2018 Internet Security Threat Report, https://www.symantec.com/content/dam/symantec/docs/reports/istr-23-executive-summary-en.pdf
[3] S. Morgan, "2017 Cybercrime report, Cybercrime damages will cost the world us$6 trillion by 2021, Cyber Security Ventures, Herjavec Group, Online Report, 2017.
[4] J. Trull, "Top 5 best practices to automate security operations, Microsoft Secure, Enterprise Cyber Security Group, Online Blog, August 2017.
[5] S. Ciccone, "Cyber Security: More threats, but also more opportunities, Paloalot Networks, Online, June 2016.
[6] Craigen, D., N. Diakun-Thibault, and R. Purse, Defining Cyber Security. Technology Innovation Management Review, 2014. 4(10).
[7] International Organization for Standardization. (2012). ISO/IEC 27032: 2012: Information Technology-Security Techniques-Guidelines for Cyber Security.
[8] M. Armstrong, "The future of a.i.. Statista Infographics, Statista, Online Report, November 2016.

[9] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks, in 2017 IEEE Symposium on Security and Privacy (SP). IEEE, 2017, pp. 39-57.

[10] N. Carlini, P. Mishra, T. Vaidya, Y. Zhang, M. Sherr, C. Shields, D. Wagner, and W. Zhou, "Hidden voice commands, in 25th USENIX Security Symposium (USENIX Security 16), Austin, TX, 2016

[11] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. nature, 521(7553), 436.

[12] Hinton, G. E. (2009). Deep belief networks. Scholarpedia, 4(5), 5947.

[13] Hinton, G. E., Osindero, S., & Teh, Y. W. (2006). A fast learning algorithm for deep belief nets. Neural computation, 18(7), 1527-1554.

[14] Hinton, G. E., & Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. science, 313(5786), 504-507.

[15] Glorot, X., Bordes, A., & Bengio, Y. (2011, June). Deep sparse rectifier neural networks. In Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics (pp. 315-323).

[16] Maas, A. L., Hannun, A. Y., & Ng, A. Y. (2013, June). Rectifier nonlinearities improve neural network acoustic models. In Proc. ICML (Vol. 30, No. 1).

[17] Nair, V., & Hinton, G. E. (2010). Rectified linear units improve restricted boltzmann machines. In Proceedings of the 27th international conference on machine learning (ICML-10) (pp. 807-814).

[18] Martens, J., & Sutskever, I. (2011). Learning recurrent neural networks with hessian-free optimization. In Proceedings of the 28th International Conference on Machine Learning (ICML-11) (pp. 1033-1040).

[19] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural computation, 9(8), 1735-1780.

[20] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: Continual prediction with LSTM, Neural Computation, vol. 12,no. 10, pp. 2451-2471, 2000.

[21] F. A. Gers, N. N. Schraudolph, and J. Schmidhuber, "Learning precise timing with LSTM recurrent networks, Journal of Machine Learning Research, vol. 3, pp. 115-143, Mar. 2003.

[22] Cho, Kyunghyun, van Merrienboer, Bart, Gulcehre, Caglar,Bougares, Fethi, Schwenk, Holger, and Bengio, Yoshua.Learning Phrase Representations using RNN EncoderDecoder for Statistical Machine Translation. arXiv preprint arXiv:1406.1078, 2014. URL http://arxiv.org/abs/1406.1078.

[23] Le, Q. V., Jaitly, N., & Hinton, G. E. (2015). A simple way to initialize recurrent networks of rectified linear units. arXiv preprint arXiv:1504.00941.

[24] Talathi, S. S., & Vartak, A. (2015). Improving performance of recurrent neural network with relu nonlinearity. arXiv preprint arXiv:1511.03771.

[25] Koutnik, J., Greff, K., Gomez, F., & Schmidhuber, J. (2014). A clockwork rnn. arXiv preprint arXiv:1402.3511.

[26] Bengio, Y., Simard, P., and Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. IEEE Transactions on Neural Networks, 5(2), 157-166.

[27] Jeffrey L. Elman. Finding structure in time. Cognitive science, 14(2):179-211, 1990.

[28] Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., & Kuksa, P. (2011). Natural language processing (almost) from scratch. Journal of Machine Learning Research, 12(Aug), 2493-2537.

[29] Vinayakumar, R., Poornachandran, P., & Soman, K. P. (2018). Scalable Framework for Cyber Threat Situational Awareness Based on Domain Name Systems Data Analysis. In Big Data in Engineering Applications (pp. 113-142). Springer, Singapore.

[30] Gandomi, A., & Haider, M. (2015). Beyond the hype: Big data concepts, methods, and analytics. International Journal of Information Management, 35(2), 137-144.

[31] M. Zaharia et al., Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing," in Proc. 9th USENIX Conf. Netw.Syst. Design Implement. (NSDI), 2012, p. 2.

[32] M. Zaharia, M. Chowdhury, M. J. Franklin, S. Shenker, and I. Stoica,Spark: Cluster computing with working sets," in Proc. 2nd USENIX-Conf. Hot Topics Cloud Comput., 2010, p. 10.

[33] C. Parker, Unexpected challenges in large scale machine learning," in Proc. 1st Int. Workshop Big Data, Streams Heterogeneous Source Mining Algorithms, Syst., Programm. Models Appl. (BigMine), 2012, pp. 1-6.

[34] M. Ghanavati, R. K. Wong, F. Chen, Y. Wang, and C.-S. Perng, An effective integrated method for learning big imbalanced data," in Proc. IEEE Int. Congr. Big Data, Jun. 2014, pp. 691-698.

[35] P. Domingos, A few useful things to know about machine learning," Commun. ACM, vol. 55, no. 10, pp. 78-87, 2012

[36] M. M. Najafabadi, F. Villanustre, T. M. Khoshgoftaar, N. Seliya, R. Wald, and E. Muharemagic, Deep Learning Applications and Challenges in Big Data Analytics," J. Big Data, vol. 2, no. 1, p. 1, Feb. 2015

[37] J. Fan, F. Han, and H. Liu, Challenges of big data analysis," Nat. Sci. Rev., vol. 1, no. 2, pp. 293-314, 2014.

[38] B. Ratner, Statistical and Machine-Learning Data Mining: Techniques for Better Predictive Modeling and Analysis of Big Data. Boca Raton, FL: CRC Press, 2011

[39] P. Domingos, A few useful things to know about machine learning," Commun. ACM, vol. 55, no. 10, pp. 78-87, 2012

[40] H. V. Jagadish et al., Big data and its technical challenges," Commun. ACM, vol. 57, no. 7, pp. 86-94, 2014

[41] J. Leskovec, A. Rajaraman, and J. D. Ullman, Mining of Massive Datasets, vol. 13. Cambridge, U.K.: Cambridge Univ. Press, 2014

[42] J. Fan, F. Han, and H. Liu, Challenges of big data analysis," Nat. Sci. Rev., vol. 1, no. 2, pp. 293-314, 2014

[43] M. Swan, The quantified self: Fundamental disruption in big data science and biological discovery," Big Data, vol. 1, no. 2, pp. 85-99, 2013.

[44] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Vanderplas, J. (2011). Scikit-learn: Machine learning in Python. Journal of machine learning research, 12(Oct), 2825-2830.

[45] Xiao, Q., Li, K., Zhang, D., & Xu, W. (2017). Security Risks in Deep Learning Implementations. arXiv preprint arXiv:1711.11008.

[46] Sommer, R., & Paxson, V. (2010, May). Outside the closed world: On using machine learning for network intrusion detection. In Security and Privacy (SP), 2010 IEEE Symposium on (pp. 305-316). IEEE.

[47] Verma, R. (2018, March). Security Analytics: Adapting Data Science for Security Challenges. In Proceedings of the Fourth ACM International Workshop on Security and Privacy Analytics (pp. 40-41). ACM.

[48] Camp, J. (2009). Data for Cyber Security Research: Process and "whish list". Retrieved July 15, 2013, from http://www.gtisc.gatech.edu/files_nsf10/data-wishlist.pdf.

[49] Alom, M. Z., & Taha, T. M. (2017, June). Network intrusion detection for Cyber Security using unsupervised deep learning approaches. In Aerospace and Electronics Conference (NAECON), 2017 IEEE National (pp. 63-69). IEEE.

[50] Thaler, S., Menkovski, V., & Petkovic, M. (2017, September). Unsupervised signature extraction from forensic logs. In Joint European Conference on Machine Learning and Knowledge Discovery in Databases (pp. 305-316). Springer, Cham.

[51] Arnaldo, I., Cuesta-Infante, A., Arun, A., Lam, M., Bassias, C., & Veeramachaneni, K. (2017, June). Learning Representations for Log Data in Cyber Security. In International Conference on Cyber Security Cryptography and Machine Learning (pp. 250-268). Springer, Cham.

[52] Lee, H., Ekanadham, C., & Ng, A. Y. (2008). Sparse deep belief net model for visual area V2. In Advances in neural information processing systems (pp. 873-880).

[53] Bontemps, L., McDermott, J., & Le-Khac, N. A. (2016, November). Collective anomaly detection based on long short-term memory recurrent neural networks. In International Conference on Future Data and Security Engineering (pp. 141-152). Springer, Cham.

[54] Xiao, H., Gao, J., Turaga, D. S., Vu, L. H., & Biem, A. (2015, May). Temporal multi-view inconsistency detection for network traffic analysis. In Proceedings of the 24th International Conference on World Wide Web (pp. 455-465). ACM.

[55] Shin, K., Hooi, B., & Faloutsos, C. (2016, September). M-zoom: Fast dense-block detection in tensors with quality guarantees. In Joint European Conference on Machine Learning and Knowledge Discovery in Databases (pp. 264-280). Springer, Cham.

[56] Xie, K., Li, X., Wang, X., Xie, G., Wen, J., Cao, J., & Zhang, D. (2017). Fast tensor factorization for accurate Internet anomaly detection. IEEE/ACM transactions on networking, 25(6), 3794-3807.

[57] Bryant, R., R. Katz & E. Lazowska. (2008). Big-Data Computing: Creating revolutionary breakthroughs in commerce, science and society. Washington, DC: Computing Community Consortium

[58] J.L. Chu, A. Krzyak, Analysis of feature maps selection in supervised learning using convolutional neural networks. Advances in Artificial Intelligence, Springer International Publishing, 2014, pp. 59-70.

[59] M.D. Zeiler, R. Fergus, Visualizing and understanding convolutional neural networks, in: Proceedings of the ECCV, 2014.

[60] W. Yu, K. Yang, Y. Bai, et al., Visualizing and comparing convolutional neural networks, arXiv preprint, arXiv: 1412.6631, 2014.

[61] Montavon, G., Samek, W., & Mller, K. R. (2018). Methods for interpreting and understanding deep neural networks. Digital Signal Processing, 73, 1-15.

[62] Hohman, F. M., Kahng, M., Pienta, R., & Chau, D. H. (2018). Visual analytics in deep learning: An interrogative survey for the next frontiers. IEEE transactions on visualization and computer graphics.

[63] Angrishi, K. "Turning Internet of Things (IoT) into Internet of Vulnerabilities (IoV): IoT Botnets, 2017."

[64] Meidan, Yair, et al. "N-BaIoTNetwork-Based Detection of IoT Botnet Attacks Using Deep Autoencoders." IEEE Pervasive Computing 17.3 (2018): 12-22.

[65] Koroniotis, Nickolaos, et al. "Towards the Development of Realistic Botnet Dataset in the Internet of Things for Network Forensic Analytics: Bot-IoT Dataset." arXiv preprint arXiv:1811.00701 (2018).

[66] Salman, Tara, et al. "Security services using blockchains: A state of the art survey." IEEE Communications Surveys & Tutorials (2018).

[67] Dinh, Thang N., and My T. Thai. "Ai and blockchain: A disruptive integration." Computer 51.9 (2018): 48-53.

[68] Weng, J., et al. "Deepchain: Auditable and privacy-preserving deep learning with blockchain-based incentive." Cryptology ePrint Archive, Report 2018/679 (2018).

[69] Salah, K., Rehman, M. H. U., Nizamuddin, N., & Al-Fuqaha, A. (2019). Blockchain for AI: Review and Open Research Challenges. IEEE Access, 7, 10127-10149.

[70] Zhang, C., Ahmad, M., & Wang, Y. (2019). Admm based privacy-preserving decentralized optimization. IEEE Transactions on Information Forensics and Security, 14(3), 565-580.

[71] Munsing, E., & Moura, S. (2018). Cyber Security in Distributed and Fully-Decentralized Optimization: Distortions, Noise Injection, and ADMM. arXiv preprint arXiv:1805.11194.

[72] Mendis, G. J., Sabounchi, M., Wei, J., & Roche, R. (2018). Blockchain as a Service: An Autonomous, Privacy Preserving, Decentralized Architecture for Deep Learning. arXiv preprint arXiv:1807.02515.

[73] Meng, W., Tischhauser, E. W., Wang, Q., Wang, Y., & Han, J. (2018). When intrusion detection meets blockchain technology: a review. Ieee Access, 6, 10179-10188.

[74] Montavon, Grgoire, Wojciech Samek, and Klaus-Robert Mller. "Methods for interpreting and understanding deep neural networks." Digital Signal Processing 73 (2018): 1-15.

[75] Marino, Daniel L., Chathurika S. Wickramasinghe, and Milos Manic. "An adversarial approach for explainable ai in intrusion detection systems." IECON 2018-44th Annual Conference of the IEEE Industrial Electronics Society. IEEE, 2018.

[76] Wickramasinghe, Chathurika S., et al. "Generalization of Deep Learning for Cyber-Physical System Security: A Survey." IECON 2018-44th Annual Conference of the IEEE Industrial Electronics Society. IEEE, 2018.

[77] Rieger, Craig, and Milos Manic. "On Critical Infrastructures, Their Security and Resilience-Trends and Vision." arXiv preprint arXiv:1812.02710 (2018).

[78] Guo, Wenbo, et al. "Lemna: Explaining deep learning based security applications." Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security. ACM, 2018.

[79] Samek, W., Wiegand, T., & Mller, K. R. (2017). Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models. arXiv preprint arXiv:1708.08296.

[80] Gunning, D. (2017). Explainable artificial intelligence (xai). Defense Advanced Research Projects Agency (DARPA), nd Web.

[81] Montavon, G., Samek, W., & Mller, K. R. (2018). Methods for interpreting and understanding deep neural networks. Digital Signal Processing, 73, 1-15.

[82] Adadi, A., & Berrada, M. (2018). Peeking inside the black-box: A survey on Explainable Artificial Intelligence (XAI). IEEE Access, 6, 52138-52160.

[83] Shin, E. C. R., Song, D., & Moazzezi, R. (2015). Recognizing functions in binaries with neural networks. In 24th USENIX Security Symposium (USENIX Security 15) (pp. 611-626).

[84] Chua, Z. L., Shen, S., Saxena, P., & Liang, Z. (2017). Neural nets can learn function type signatures from binaries. In 26th USENIX Security Symposium (USENIX Security 17) (pp. 99-116).

[85] Xu, X., Liu, C., Feng, Q., Yin, H., Song, L., & Song, D. (2017, October). Neural network-based graph embedding for cross-platform binary code similarity detection. In Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security (pp. 363-376). ACM.

[86] Li, Z., Zou, D., Xu, S., Ou, X., Jin, H., Wang, S., ... & Zhong, Y. (2018). VulDeePecker: A deep learning-based system for vulnerability detection. arXiv preprint arXiv:1801.01681.

[87] Liao, Y., Cai, R., Zhu, G., Yin, Y., & Li, K. (2018, September). Mobilefindr: Function similarity identification for reversing mobile binaries. In European Symposium on Research in Computer Security (pp. 66-83). Springer, Cham.

[88] Liu, B., Huo, W., Zhang, C., Li, W., Li, F., Piao, A., & Zou, W. (2018, September). Diff: cross-version binary code similarity detection with DNN. In Proceedings of the 33rd ACM/IEEE International Conference on Automated Software Engineering(pp. 667-678). ACM.

[89] Song, W., Yin, H., Liu, C., & Song, D. (2018, October). DeepMem: Learning Graph Neural Network Models for Fast and Robust Memory Forensic Analysis. In Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security (pp. 606-618). ACM.

[90] Baldoni, R., Di Luna, G. A., Massarelli, L., Petroni, F., & Querzoni, L. (2018). Unsupervised Features Extraction for Binary Similarity Using Graph Embedding Neural Networks. arXiv preprint arXiv:1810.09683.

[91] Massarelli, L., Di Luna, G. A., Petroni, F., Querzoni, L., & Baldoni, R. (2018). SAFE: Self-Attentive Function Embeddings for Binary Similarity. arXiv preprint arXiv:1811.05296.

[92] Lee, Y. J., Choi, S. H., Kim, C., Lim, S. H., & Park, K. W. (2017, December). Learning binary code with deep learning to detect software weakness. In KSII The 9th International Conference on Internet (ICONI) 2017 Symposium.

[93] Li, Z., Zou, D., Xu, S., Jin, H., Zhu, Y., Chen, Z., ... & Wang, J. (2018). SySeVR: A Framework for Using Deep Learning to Detect Software Vulnerabilities. arXiv preprint arXiv:1807.06756.

[94] Zuo, F., Li, X., Zhang, Z., Young, P., Luo, L., & Zeng, Q. (2018). Neural machine translation inspired binary code similarity comparison beyond function pairs. arXiv preprint arXiv:1808.04706.

[95] Marastoni, N., Giacobazzi, R., & Dalla Preda, M. (2018, September). A deep learning approach to program similarity. In Proceedings of the 1st International Workshop on Machine Learning and Software Engineering in Symbiosis (pp. 26-35). ACM.

[96] Katz, D. S., Ruchti, J., & Schulte, E. (2018, March). Using recurrent neural networks for decompilation. In 2018 IEEE 25th International Conference on Software Analysis, Evolution and Reengineering (SANER) (pp. 346-356). IEEE.

[97] Tufano, M., Watson, C., Bavota, G., Di Penta, M., White, M., & Poshyvanyk, D. (2018, May). Deep learning similarities from different representations of source code. In 2018 IEEE/ACM 15th International Conference on Mining Software Repositories (MSR) (pp. 542-553). IEEE.

[98] Feng, Q., Zhou, R., Zhao, Y., Ma, J., Wang, Y., Yu, N., ... & Ning, P. (2019, January). Learning Binary Representation for Automatic Patch Detection. In 2019 16th IEEE Annual Consumer Communications & Networking Conference (CCNC)(pp. 1-6). IEEE.

[99] Krenn, R. (2004). Steganography and steganalysis. Retrieved September, 8, 2007.

[100] Wu, H. Z., Wang, H. X., & Shi, Y. Q. (2016). Can machine learn steganography?-implementing LSB substitution and matrix coding steganography with feed-forward neural networks. arXiv preprint arXiv:1606.05294.

[101] Baluja, S. (2017). Hiding images in plain sight: Deep steganography. In Advances in Neural Information Processing Systems (pp. 2069-2079).

[102] Meng, R., Rice, S. G., Wang, J., & Sun, X. (2018). A fusion steganographic algorithm based on faster R-CNN. Computers, Materials & Continua, 55(1), 1-16.

[103] Zhu, J., Kaplan, R., Johnson, J., & Fei-Fei, L. (2018). Hidden: Hiding data with deep networks. In Proceedings of the European Conference on Computer Vision (ECCV) (pp. 657-672).

[104] Li, C., Jiang, Y., & Cheslyar, M. (2018). Embedding image through generated intermediate medium using deep convolutional generative adversarial network. Comput., Mater. Continua, 56(2), 313-324.

[105] Wu, P., Yang, Y., & Li, X. (2018). StegNet: Mega image steganography capacity with deep convolutional network. Future Internet, 10(6), 54.

[106] Zhang, R., Dong, S., & Liu, J. (2018). Invisible steganography via generative adversarial networks. Multimedia Tools and Applications, 1-17.

[107] Wu, P., Yang, Y., & Li, X. (2018, September). Image-into-Image Steganography Using Deep Convolutional Network. In Pacific Rim Conference on Multimedia (pp. 792-802). Springer, Cham.

[108] Weng, X., Li, Y., Chi, L., & Mu, Y. (2018). Convolutional Video Steganography with Temporal Residual Modeling. arXiv preprint arXiv:1806.02941.

[109] Jung, D., Bae, H., Choi, H. S., & Yoon, S. (2019). PixelSteganalysis: Destroying Hidden Information with a Low Degree of Visual Degradation. arXiv preprint arXiv:1902.11113.

[110] Wang, Z., Gao, N., Wang, X., Qu, X., & Li, L. (2018, December). SSteGAN: Self-learning Steganography Based on Generative Adversarial Networks. In International Conference on Neural Information Processing (pp. 253-264). Springer, Cham.

[111] Duan, X., Jia, K., Li, B., Guo, D., Zhang, E., & Qin, C. (2019). Reversible Image Steganography Scheme Based on a U-Net Structure. IEEE Access, 7, 9314-9323.

[112] Babaheidarian, P., & Wallace, M. (2019). Decode and Transfer: A New Steganalysis Technique via Conditional Generative Adversarial Networks. arXiv preprint arXiv:1901.09746.

[113] Sarmah, D. K., & Kulkarni, A. J. (2019). Improved Cohort IntelligenceA high capacity, swift and secure approach on JPEG image steganography. Journal of Information Security and Applications, 45, 90-106.

[114] Zhang, K. A., Cuesta-Infante, A., & Veeramachaneni, K. (2019). SteganoGAN: Pushing the Limits of Image Steganography. arXiv preprint arXiv:1901.03892.

[115] Kim, D. H., & Lee, H. Y. (2017, November). Deep Learning-Based Steganalysis Against Spatial Domain Steganography. In 2017 European Conference on Electrical Engineering and Computer Science (EECS) (pp. 1-4). IEEE.

[116] Hu, D., Wang, L., Jiang, W., Zheng, S., & Li, B. (2018). A novel image steganography method via deep convolutional generative adversarial networks. IEEE Access, 6, 38303-38314.

[117] Volkhonskiy, D., Borisenko, B., & Burnaev, E. (2016). Generative adversarial networks for image steganography.

[118] Hayes, J., & Danezis, G. (2017). ste-gan-ography: Generating steganographic images via adversarial training. arXiv preprint arXiv:1703.00371.

[119] Zhang, Y., Zhang, W., Chen, K., Liu, J., Liu, Y., & Yu, N. (2018, June). Adversarial Examples Against Deep Neural Network based Steganalysis. In Proceedings of the 6th ACM Workshop on Information Hiding and Multimedia Security (pp. 67-72). ACM.

[120] Zou, Y., Zhang, G., & Liu, L. (2019). Research on Image Steganography Analysis Based on Deep Learning. Journal of Visual Communication and Image Representation.

[121] Wu, S., Zhong, S. H., & Liu, Y. (2016, December). Steganalysis via deep residual network. In 2016 IEEE 22nd International Conference on Parallel and Distributed Systems (ICPADS) (pp. 1233-1236). IEEE.

[122] Wu, S., Zhong, S. H., & Liu, Y. (2017, July). Residual convolution network based steganalysis with adaptive content suppression. In 2017 IEEE International Conference on Multimedia and Expo (ICME) (pp. 241-246). IEEE.

[123] Boroumand, M., Chen, M., & Fridrich, J. (2019). Deep residual network for steganalysis of digital images. IEEE Transactions on Information Forensics and Security.

[124] Zhang, S., Zhang, H., Zhao, X., & Yu, H. (2018, October). A Deep Residual Multi-scale Convolutional Network for Spatial Steganalysis. In International Workshop on Digital Watermarking (pp. 40-52). Springer, Cham.

[125] Ozcan, S., & Mustacoglu, A. F. (2018, December). Transfer Learning Effects on Image Steganalysis with Pre-Trained Deep Residual Neural Network Model. In 2018 IEEE International Conference on Big Data (Big Data) (pp. 2280-2287). IEEE.

[126] Huang, X., Wang, S., Sun, T., Liu, G., & Lin, X. (2018, November). STEGANALYSIS OF ADAPTIVE JPEG STEGANOGRAPHY BASED ON RESDET. In Proceedings, APSIPA Annual Summit and Conference (Vol. 2018, pp. 12-15).

[127] Luo, Y., & Huang, Y. (2017, June). Text steganography with high embedding rate: Using recurrent neural networks to generate chinese classic poetry. In Proceedings of the 5th ACM Workshop on Information Hiding and Multimedia Security (pp. 99-104). ACM.

[128] Lin, Z., Huang, Y., & Wang, J. (2018). Rnn-sm: Fast steganalysis of voip streams using recurrent neural network. IEEE Transactions on Information Forensics and Security, 13(7), 1854-1868.

[129] Yang, Z. L., Guo, X. Q., Chen, Z. M., Huang, Y. F., & Zhang, Y. J. (2019). RNN-stega: Linguistic steganography based on recurrent neural networks. IEEE Transactions on Information Forensics and Security, 14(5), 1280-1295.

[130] Yang, Z., Wei, N., Sheng, J., Huang, Y., & Zhang, Y. J. (2018). TS-CNN: Text Steganalysis from Semantic Space Based on Convolutional Neural Network. arXiv preprint arXiv:1810.08136.

[131] Kreuk, F., Adi, Y., Raj, B., Singh, R., & Keshet, J. (2019). Hide and Speak: Deep Neural Networks for Speech Steganography. arXiv preprint arXiv:1902.03083.

[132] Fang, T., Jaggi, M., & Argyraki, K. (2017). Generating steganographic text with LSTMs. arXiv preprint arXiv:1705.10742.

[133] Chen, B., Luo, W., & Li, H. (2017, June). Audio steganalysis with convolutional neural network. In Proceedings of the 5th ACM Workshop on Information Hiding and Multimedia Security (pp. 85-90). ACM.

[134] Ren, Y., Liu, D., Xiong, Q., Fu, J., & Wang, L. (2019). Spec-ResNet: A General Audio Steganalysis scheme based on Deep Residual Network of Spectrogram. arXiv preprint arXiv:1901.06838.

[135] Wang, Y., Yang, K., Yi, X., Zhao, X., & Xu, Z. (2018, June). CNN-based Steganalysis of MP3 Steganography in the Entropy Code Domain. In Proceedings of the 6th ACM Workshop on Information Hiding and Multimedia Security (pp. 55-65). ACM.

[136] Bae, H., Lee, B., Kwon, S., & Yoon, S. (2017). DNA steganalysis using deep recurrent neural networks. arXiv preprint arXiv:1704.08443.

[137] Kinzel, W., & Kanter, I. (2002, November). Neural cryptography. In Proceedings of the 9th International Conference on Neural Information Processing, 2002. ICONIP'02. (Vol. 3, pp. 1351-1354). IEEE.

[138] Abadi, M., & Andersen, D. G. (2016). Learning to protect communications with adversarial neural cryptography. arXiv preprint arXiv:1610.06918.

[139] Xie, P., Bilenko, M., Finley, T., Gilad-Bachrach, R., Lauter, K., & Naehrig, M. (2014). Crypto-nets: Neural networks over encrypted data. arXiv preprint arXiv:1412.6181.

[140] Dylan Modesitt, Tim Henry, Jon Coden, and Rachel Lathe . Neural Cryptography: From Symmetric Encryption to Adversarial Steganography . Avilable at https://courses.csail.mit.edu/6.857/2018/project/Modesitt-Henry-Coden-Lathe-NeuralCryptography.pdf

[141] Coutinho, M., de Oliveira Albuquerque, R., Borges, F., Garca Villalba, L., & Kim, T. H. (2018). Learning Perfectly Secure Cryptography to Protect Communications with Adversarial Neural Cryptography. Sensors, 18(5), 1306.

[142] Arvandi, M., Wu, S., & Sadeghian, A. (2008). On the use of recurrent neural networks to design symmetric ciphers. IEEE computational intelligence magazine, 3(2), 42-53.

[143] Noughabi, M. N. A., & Sadeghiyan, B. (2010, August). Design of S-boxes based on neural networks. In 2010 International Conference on Electronics and Information Engineering (Vol. 2, pp. V2-172). IEEE.

[144] Hill, G. D., & Bellekens, X. J. (2017). Deep Learning Based Cryptographic Primitive Classification. arXiv preprint arXiv:1709.08385.

[145] Hesamifard, E., Takabi, H., & Ghasemi, M. (2017). Cryptodl: Deep neural networks over encrypted data. arXiv preprint arXiv:1711.05189.

[146] Gilad-Bachrach, R., Dowlin, N., Laine, K., Lauter, K., Naehrig, M., & Wernsing, J. (2016, June). Cryptonets: Applying neural networks to encrypted data with high throughput and accuracy. In International Conference on Machine Learning(pp. 201-210).

[147] Chou, E., Beal, J., Levy, D., Yeung, S., Haque, A., & Fei-Fei, L. (2018). Faster CryptoNets: Leveraging Sparsity for Real-World Encrypted Inference. arXiv preprint arXiv:1811.09953.

[148] Bhamare, D., Salman, T., Samaka, M., Erbad, A., & Jain, R. (2016, December). Feasibility of Supervised Machine Learning for Cloud Security. In 2016 International Conference on Information Science and Security (ICISS) (pp. 1-5). IEEE.

[149] R. Barga, V. Fontama, and W. H. Tok, Cortana analytics," in Predictive Analytics With Microsoft Azure Machine Learning. Berkeley, CA, USA: Apress, 2015, pp. 279-283.

[150] Google. (2016). Google Cloud Machine Learning. Available: https://cloud.google.com/products/machine-learning/

[151] Amazon Web Services. (2016). Amazon Machine Learning. Available: https://aws.amazon.com/machine-learning/

[152] IBM. (2014). IBM Watson Ecosystem Program. Available: http://www-03.ibm.com/innovation/us/watson/

[153] Bhattacharjee, B., Boag, S., Doshi, C., Dube, P., Herta, B., Ishakian, V., ... & Muthusamy, V. (2017). IBM deep learning service. IBM Journal of Research and Development, 61(4/5), 10-1.

[154] Ye, X., Chen, X., Wang, H., Zeng, X., Shao, G., Yin, X., & Xu, C. (2016). An anomalous behavior detection model in cloud computing. Tsinghua Science and Technology, 21(3), 322-332.

[155] Vinayakumar, R., Soman, K. P., & Poornachandran, P. (2017, September). Long short-term memory based operation log anomaly detection. In 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI) (pp. 236-242). IEEE.

[156] Loukas, G., Vuong, T., Heartfield, R., Sakellari, G., Yoon, Y., & Gan, D. (2018). Cloud-based cyber-physical intrusion detection for vehicles using deep learning. Ieee Access, 6, 3491-3508.

[157] Glmez, H. G., Tuncel, E., & Angin, P. (2018, June). A Big Data Analytical Approach to Cloud Intrusion Detection. In International Conference on Cloud Computing (pp. 377-388). Springer, Cham.

[158] Nguyen, K. K., Hoang, D. T., Niyato, D., Wang, P., Nguyen, D., & Dutkiewicz, E. (2018, April). Cyberattack detection in mobile cloud computing: A deep learning approach. In 2018 IEEE Wireless Communications and Networking Conference (WCNC) (pp. 1-6). IEEE.

[159] Nguyen, K. K., Hoang, D. T., Niyato, D., Wang, P., Nguyen, D., & Dutkiewicz, E. (2018, April). Cyberattack detection in mobile cloud computing: A deep learning approach. In 2018 IEEE Wireless Communications and Networking Conference (WCNC) (pp. 1-6). IEEE.

[160] Sundararajan, K., & Woodard, D. L. (2018). Deep learning for biometrics: a survey. ACM Computing Surveys (CSUR), 51(3), 65.

[161] Narendra, T., Sankaran, A., Vijaykeerthy, D., & Mani, S. (2018). Explaining Deep Learning Models using Causal Inference. arXiv preprint arXiv:1811.04376.

[162] Deka, R. K., Bhattacharyya, D. K., & Kalita, J. K. Granger Causality in TCP Flooding Attack.

[163] Shi, D., Guo, Z., Johansson, K. H., & Shi, L. (2018). Causality countermeasures for anomaly detection in cyber-physical systems. IEEE Transactions on Automatic Control, 63(2), 386-401.

[164] Liu, Y., Zhang, M., Li, D., Jee, K., Li, Z., Wu, Z., ... & Mittal, P. (2018). Towards a timely causality analysis for enterprise security. In Proceedings of the 25th Network and Distributed System Security Symposium (NDSS). The Internet Society, San Diego, California, USA.

[165] Mugan, J. (2013, May). A developmental approach to learning causal models for Cyber Security. In Machine Intelligence and Bio-inspired Computation: Theory and Applications VII (Vol. 8751, p. 87510A). International Society for Optics and Photonics.

[166] Savas, O., & Deng, J. (2017). Big Data Analytics in Cyber Security. Auerbach Publications.

[167] Prabhu, C. S. R. (2019). Fog Computing, Deep Learning and Big Data Analytics-Research Directions. Springer.

[168] Nassar, A. T., & Yilmaz, Y. (2018). Reinforcement-Learning-Based Resource Allocation in Fog Radio Access Networks for Various IoT Environments. arXiv preprint arXiv:1806.04582.

[169] Guan, Y., Shao, J., Wei, G., & Xie, M. (2018). Data security and privacy in fog computing. IEEE Network, (99), 1-6.

[170] Haque, M., & Ahamed, S. I. (2006). Security in pervasive computing: Current status and open issues. International Journal of Network Security.

[171] Hutter, D., & Ullmann, M. (Eds.). (2005). Security in Pervasive Computing: Second International Conference, SPC 2005, Boppard, Germany, April 6-8, 2005, Proceedings (Vol. 3450). Springer.

[172] Pagter, J. I., & Petersen, M. G. (2007, February). A Sense of Security in Pervasive ComputingIs the Light on When the Refrigerator Door Is Closed?. In International Conference on Financial Cryptography and Data Security (pp. 383-388). Springer, Berlin, Heidelberg.

[173] Zugenmaier, A., & Walter, T. (2007, July). Security in pervasive computing calling for new security principles. In IEEE International Conference on Pervasive Services (pp. 96-99). IEEE.

[174] Stajano, F. (2004). Security in pervasive computing. In Security in Pervasive Computing (pp. 6-8). Springer, Berlin, Heidelberg.

[175] Joshi, A., Finin, T., Kagal, L., Parker, J., & Patwardhan, A. (2008). Security policies and trust in ubiquitous computing. Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, 366(1881), 3769-3780.

[176] Fukushima, Y., Miura, D., Hamatani, T., Yamaguchi, H., & Higashino, T. (2018, June). MicroDeep: In-network Deep Learning by Micro-Sensor Coordination for Pervasive Computing. In 2018 IEEE International Conference on Smart Computing (SMARTCOMP) (pp. 163-170). IEEE.

[177] Lane, N. D., Bhattacharya, S., Mathur, A., Georgiev, P., Forlivesi, C., & Kawsar, F. (2017). Squeezing deep learning into mobile and embedded devices. IEEE Pervasive Computing, 16(3), 82-88.

[178] Meidan, Y., Bohadana, M., Mathov, Y., Mirsky, Y., Shabtai, A., Breitenbacher, D., & Elovici, Y. (2018). N-BaIoTNetwork-Based Detection of IoT Botnet Attacks Using Deep Autoencoders. IEEE Pervasive Computing, 17(3), 12-22.

[179] KDDCup 99, Avilable at http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html

[180] NSL-KDD, Avilable at https://www.unb.ca/cic/datasets/nsl.html

[181] UNSW-NB15, Avilable at https://www.unsw.adfa.edu.au/unsw-canberra-cyber/CyberSecurity/ADFA-NB15-Datasets/

[182] ADFA-LD, Avilable at https://www.unsw.adfa.edu.au/unsw-canberra-cyber/CyberSecurity/ADFA-IDS-Datasets/

[183] UNM, Avilable at https://www.cs.unm.edu/~immsec/systemcalls.htm

[184] ISCX-IDS-2012, Avilable at https://www.unb.ca/cic/datasets/ids.html

[185] CICIDS2017, Avilable at https://www.unb.ca/cic/datasets/ids-2017.html

[186] Kyoto, Avilable at http://www.takakura.com/Kyoto_data/

[187] UNIBS, Avilable at http://netweb.ing.unibs.it/~ntw/tools/traces/

[188] CAIDA, Avilable at https://www.caida.org/data/

[189] LBNL, Avilable at https://commons.lbl.gov/display/cpp/100G+Intrusion+Detection

[190] CIC DoS, Avilable at https://www.unb.ca/cic/datasets/dos-dataset.html

[191] CSE-CIC-IDS2018, Avilable at https://www.unb.ca/cic/datasets/ids-2018.html

[192] AWID, Avilable at http://icsdweb.aegean.gr/awid/index.html

[193] WSN-DS, Avilable at https://www.hindawi.com/journals/js/2016/4731953/

[194] UNB Botnet, Avilable at https://www.unb.ca/cic/datasets/botnet.html

[195] DGArchive, Avilable at https://dgarchive.caad.fkie.fraunhofer.de/

[196] AmritaDGA, Avilable at https://vinayakumarr.github.io/AmritaDGA/

[197] ISCX-URL-2016, Avilable at https://www.unb.ca/cic/datasets/url-2016.html

[198] Sophos URL, Avilable at https://github.com/Maddy12/SophosMachineLearningBuildingBlocksTutorial

[199] CSDMC, Avilable at http://csmining.org/spam-email-datasets-.html/

[200] Enron, Avilable at https://www.cs.cmu.edu/~./enron/

[201] TREC, Avilable at https://trec.nist.gov/data/spam.html

[202] SpamAssassin, Avilable at https://spamassassin.apache.org/old/publiccorpus/

[203] EMBER, Avilable at https://github.com/endgameinc/ember

[204] Microsoft malware classification challenge, Avilable at https://www.kaggle.com/c/malware-classification

[205] Microsoft Malware Prediction, Avilable at https://www.kaggle.com/c/microsoft-malware-prediction

[206] Malrec, Avilable at http://security.ece.cmu.edu/byteweight/

[207] ByteWeight, Avilable at http://www.cs.northwestern.edu/~yga751/ML/ISH.htm

[208] Image spam hunter, Avilable at https://www.unb.ca/cic/datasets/android-adware.html

[209] Android Adware and General Malware Dataset, Avilable at https://www.unb.ca/cic/datasets/andmal2017.html

[210] CICAndMal2017, Avilable at https://www.sec.cs.tu-bs.de/~danarp/drebin/

[211] Drebin, Avilable at http://kharon.gforge.inria.fr/dataset/

[212] Kharon, Avilable at https://www.unb.ca/cic/datasets/vpn.html

[213] ISCXVPN2016, Avilable at https://www.unb.ca/cic/datasets/tor.html

[214] ISCXTor2016, Avilable at https://archive.ics.uci.edu/ml/datasets/detection_of_IoT_botnet_attacks_N_BaIoT

[215] N-BaIoT, Avilable at https://archive.ics.uci.edu/ml/datasets/detection_of_IoT_botnet_attacks_N_BaIoT

[216] Bot-IoT, Avilable at https://www.unsw.adfa.edu.au/unsw-canberra-cyber/cybersecurity/ADFA-NB15-Datasets/bot_iot.php

[217] DPA contest, Avilable at http://www.dpacontest.org/index.php

[218] ASCAD, Avilable at https://www.data.gouv.fr/s/resources/ascad/20180530-163000/ASCAD_data.zip

[219] CERT, Avilable at https://resources.sei.cmu.edu/library/asset-view.cfm?assetid=508099

[220] Nataraj, L., Karthikeyan, S., Jacob, G., & Manjunath, B. S. (2011, July). Malware images: visualization and automatic classification. In Proceedings of the 8th international symposium on visualization for Cyber Security (p. 4). ACM.

[221] Nataraj, L., Yegneswaran, V., Porras, P., & Zhang, J. (2011, October). A comparative assessment of malware classification using binary texture analysis and dynamic analysis. In Proceedings of the 4th ACM Workshop on Security and Artificial Intelligence (pp. 21-30). ACM.

[222] Kirat, D., Nataraj, L., Vigna, G., & Manjunath, B. S. (2013, December). Sigmal: A static signal processing based malware triage. In Proceedings of the 29th Annual Computer Security Applications Conference (pp. 89-98). ACM.

[223] Nataraj, L., & Manjunath, B. S. (2016). Spam: Signal processing to analyze malware

[224] . IEEE Signal Processing Magazine, 33(2), 105-117.

[225] Nataraj, L., Kirat, D., Manjunath, B. S., & Vigna, G. (2013, December). Sarvam: Search and retrieval of malware. In Proceedings of the Annual Computer Security Conference (ACSAC) Worshop on Next Generation Malware Attacks and Defense (NGMAD).

[226] Nataraj, L., Karthikeyan, S., & Manjunath, B. S. (2015, June). SATTVA: SpArsiTy inspired classificaTion of malware VAriants. In Proceedings of the 3rd ACM Workshop on Information Hiding and Multimedia Security (pp. 135-140). ACM.

[227] Nataraj, L. (2015). A signal processing approach to malware analysis. University of California, Santa Barbara.

[228] Almomani, A., Gupta, B. B., Atawneh, S., Meulenberg, A., & Almomani, E. (2013). A survey of phishing email filtering techniques. IEEE communications surveys & tutorials, 15(4), 2070-2090.

[229] G. Dini, F. Martinelli, A. Saracino, and D. Sgandurra, "Madam: a multilevel anomaly detector for android malware.

[230] Nguyen-Vu, L., Chau, N.T., Kang, S. and Jung, S., 2017. Android Rooting: An Arms Race between Evasion and Detection. Security and Communication Networks, 2017.

[231] Ingo Bente, Bastian Hellmann, Joerg Vieweg, Josef von Helden, Gabi Dreo, "TCADS: Trustworthy, Context-Related Anomaly Detection for Smartphones", 2013 16th International Conference on Network-Based Information Systems, vol. 00, no. , pp. 247-254, 2012, doi:10.1109/NBiS.2012.53

[232] Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., ... & Darrell, T. (2014, November). Caffe: Convolutional architecture for fast feature embedding. In Proceedings of the 22nd ACM international conference on Multimedia (pp. 675-678). ACM.

[233] Collobert, R., Bengio, S., & Marithoz, J. (2002). Torch: a modular machine learning software library (No. EPFL-REPORT-82802). Idiap.

[234] Al-Rfou, R., Alain, G., Almahairi, A., Angermueller, C., Bahdanau, D., Ballas, N., ... & Bengio, Y. (2016). Theano: A Python framework for fast computation of mathematical expressions. arXiv preprint.

[235] Skymind. 2017. Deeplearning4j deep learning framework. Retrieved from https://deeplearning4j.org. Accessed April 18, 2017.

[236] Chen, T., Li, M., Li, Y., Lin, M., Wang, N., Wang, M., ... & Zhang, Z. (2015). Mxnet: A flexible and efficient machine learning library for heterogeneous distributed systems. arXiv preprint arXiv:1512.01274.

[237] Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., ... & Kudlur, M. (2016, November). TensorFlow: a system for large-scale machine learning. In OSDI (Vol. 16, pp. 265-283).

[238] Intel Nervana Systems. 2017. Neon deep learning framework. Retrieved from https://www.nervanasys.com/technology/neon. Accessed April 4, 2017.

[239] Candel, A., Parmar, V., LeDell, E., & Arora, A. (2016). Deep learning with H2O. H2O. ai Inc.

[240] Tokui, S., Oono, K., Hido, S., & Clayton, J. (2015, December). Chainer: a next-generation open source framework for deep learning. In Proceedings of workshop on machine learning systems (LearningSys) in the twenty-ninth annual conference on neural information processing systems (NIPS) (Vol. 5, pp. 1-6).

[241] Seide, F., & Agarwal, A. (2016, August). CNTK: Microsoft's open-source deep-learning toolkit. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 2135-2135). ACM.

[242] Chollet, F. (2015). Keras.

[243] Silva, S. S., Silva, R. M., Pinto, R. C., & Salles, R. M. (2013). Botnets: A survey. Computer Networks, 57(2), 378-403.

[244] Wang, Z., & Zhang, Y. (2017, August). DDoS event forecasting using Twitter data. In Proceedings of the 26th International Joint Conference on Artificial Intelligence (pp. 4151-4157). AAAI Press.

[245] Chambers, N., Fry, B., & McMasters, J. (2018). Detecting Denial-of-Service Attacks from Social Media Text: Applying NLP to Computer Security. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers) (Vol. 1, pp. 1626-1635).

[246] Puliga, M., Caldarelli, G., Chessa, A., & De Nicola, R. (2018). Understanding the Twitter User Networks of Viruses and Ransomware Attacks. In ITASEC.

[247] Alguliyev, R. M., Aliguliyev, R. M., & Abdullayeva, F. J. (2019). The Improved LSTM and CNN Models for DDoS Attacks Prediction in Social Media. International Journal of Cyber Warfare and Terrorism (IJCWT), 9(1), 1-18.

[248] Alguliyev, R. M., Aliguliyev, R. M., & Abdullayeva, F. J. (2019). Deep Learning Method for Prediction of DDoS Attacks on Social Media. Advances in Data Science and Adaptive Analysis.

[249] Vinayakumar R, Mamoun Alazab, Alireza Jolfaei, Soman KP, Prabaharan Poornachandran. Ransomware Triage Using Deep Learning: Twitter as a Case Study, The 9th IEEE International Conference on Cyber Security and Communication Systems at Melbourne, Australia.

[250] Kocher, P. C. (1996, August). Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems. In Annual International Cryptology Conference (pp. 104-113). Springer, Berlin, Heidelberg.

[251] Dahl, G. E., Stokes, J. W., Deng, L., Yu, D. (2013, May). Large-scale malware classification using random projections and neural networks. In Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on (pp. 3422-3426). IEEE.

[252] Saxe, J., Berlin, K. (2015, October). Deep neural network based malware detection using two dimensional binary program features. In Malicious and Unwanted Software (MALWARE), 2015 10th International Conference on (pp. 11-20). IEEE.

[253] Pascanu, R., Stokes, J. W., Sanossian, H., Marinescu, M., Thomas, A. (2015, April). Malware classification with recurrent networks. In Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on (pp. 1916-1920). IEEE.

[254] David, O. E., Netanyahu, N. S. (2015, July). Deepsign: Deep learning for automatic malware signature generation and classification. In Neural Networks (IJCNN), 2015 International Joint Conference on (pp. 1-8). IEEE.

[255] Hardy, W., Chen, L., Hou, S., Ye, Y., Li, X. (2016, January). DL4MD: A deep learning framework for intelligent malware detection. In Proceedings of the International Conference on Data Mining (DMIN) (p. 61). The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp).

[256] Tobiyama, S., Yamaguchi, Y., Shimada, H., Ikuse, T., Yagi, T. (2016, June). Malware detection with deep neural network using process behavior. In Computer Software and Applications Conference (COMPSAC), 2016 IEEE 40th Annual (Vol. 2, pp. 577-582). IEEE.

[257] Huang, W., Stokes, J. W. (2016, July). MtNet: a multi-task neural network for dynamic malware classification. In International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment (pp. 399-418). Springer, Cham.

[258] Ding, Y., Chen, S., & Xu, J. (2016, July). Application of Deep Belief Networks for opcode based malware detection. In Neural Networks (IJCNN), 2016 International Joint Conference on (pp. 3901-3908). IEEE.

[259] Gibert, D. (2016). Convolutional neural networks for malware classification (Doctoral dissertation, MS Thesis, Dept. of Computer Science, UPC).

[260] Kolosnjaji, B., Zarras, A., Webster, G., Eckert, C. (2016, December). Deep learning for classification of malware system call sequences. In Australasian Joint Conference on Artificial Intelligence (pp. 137-149). Springer, Cham.

[261] Aragorn, T., YunChun, C., YiHsiang, K., & Tsungnan, L. (2016). Deep Learning for Ransomware Detection. IEICE Technical Report; IEICE Tech. Rep., 116(282), 87-92.

[262] Shibahara, T., Yagi, T., Akiyama, M., Chiba, D., & Yada, T. (2016, December). Efficient dynamic malware analysis based on network behavior using deep learning. In Global Communications Conference (GLOBECOM), 2016 IEEE (pp. 1-7). IEEE.

[263] Raff, E., Barker, J., Sylvester, J., Brandon, R., Catanzaro, B., Nicholas, C. (2017). Malware detection by eating a whole exe. arXiv preprint arXiv:1710.09435.

[264] Kolosnjaji, B., Eraisha, G., Webster, G., Zarras, A., Eckert, C. (2017, May). Empowering convolutional networks for malware classification and analysis. In Neural Networks (IJCNN), 2017 International Joint Conference on (pp. 3838-3845). IEEE.

[265] Athiwaratkun, B., Stokes, J. W. (2017, March). Malware classification with LSTM and GRU language models and a character-level CNN. In Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on (pp. 2482-2486). IEEE.

[266] Yuxin, D., Siyi, Z. (2017). Malware detection based on deep learning algorithm. Neural Computing and Applications, 1-12.

[267] Guo, W., Wang, T., Wei, J. (2017, August). Malware Detection with Convolutional Neural Network Using Hardware Events. In CCF National Conference on Compujter Engineering and Technology (pp. 104-115). Springer, Singapore.

[268] Agarap, A. F., Pepito, F. J. H. (2017). Towards Building an Intelligent Anti-Malware System: A Deep Learning Approach using Support Vector Machine (SVM) for Malware Classification. arXiv preprint arXiv:1801.00318.

[269] Yousefi-Azar, M., Varadharajan, V., Hamey, L., Tupakula, U. (2017, May). Autoencoder-based feature learning for Cyber Security applications. In Neural Networks (IJCNN), 2017 International Joint Conference on (pp. 3854-3861). IEEE.

[270] Vinayakumar, R., Soman, K. P., Velan, K. S., Ganorkar, S. (2017,September). Evaluating shallow and deep networks for ransomware detection and classification. In Advances in Computing, Communications and Informatics (ICACCI), 2017 International Conference on (pp. 259-265). IEEE.

[271] Maniath, S., Ashok, A., Poornachandran, P., Sujadevi, V. G., Sankar, A. P., Jan, S. (2017, October). Deep learning LSTM based ransomware detection. In Control, Automation Power Engineering (RDCAPE), 2017 Recent Developments in (pp. 442-446). IEEE.

[272] Rahul, R. K., Anjali, T., Menon, V. K., Soman, K. P. (2017, September). Deep learning for network flow analysis and malware classification. In International Symposium on Security in Computing and Communication (pp. 226-235). Springer, Singapore.

[273] Kabanga, E. K., Kim, C. H. (2017). Malware images classification using convolutional neural network. Journal of Computer and Communications, 6(01), 153.

[274] Yue, S. (2017). Imbalanced malware images classification: a CNN based approach. arXiv preprint arXiv:1708.08042.

[275] Kim, H. J. (2017). Image-based malware classification using convolutional neural network. In Advances in Computer Science and Ubiquitous Computing (pp. 1352-1357). Springer, Singapore.

[276] Liu, L., & Wang, B. (2017, September). Automatic Malware Detection Using Deep Learning Based on Static Analysis. In International Conference of Pioneering Computer Scientists, Engineers and Educators (pp. 500-507). Springer, Singapore.

[277] Krl, M., vec, O., Blek, M., Jaek, O. (2018). Deep Convolutional Malware Classifiers Can Learn from Raw Executables and Labels Only.

[278] Ni, S., Qian, Q., & Zhang, R. (2018). Malware identification using visualization images and deep learning. Computers & Security, 77, 871-885.

[279] Yajamanam, S., Selvin, V. R. S., Di Troia, F., Stamp, M. (2018). Deep Learning versus Gist Descriptors for Image-based Malware Classification. In ICISSP (pp. 553-561).

[280] Yan, J., Qi, Y., Rao, Q. (2018). Detecting malware with an ensemble method based on deep neural network. Security and Communication Networks, 2018.

[281] Cakir, B., Dogdu, E. (2018, March). Malware classification using deep learning methods. In Proceedings of the ACMSE 2018 Conference (p. 10). ACM.

[282] Yi, H., Kim, G., Lee, J., Ahn, S., Lee, Y., Yoon, S., Paek, Y. (2018). Mimicry Resilient Program Behavior Modeling with LSTM based Branch Models. arXiv preprint arXiv:1803.09171.

[283] Su, J., Vargas, D. V., Prasad, S., Sgandurra, D., Feng, Y., Sakurai, K. (2018). Lightweight Classification of IoT Malware based on Image Recognition. arXiv preprint arXiv:1802.03714.

[284] Vinayakumar, R., & Soman, K. P. (2018). DeepMalNet: Evaluating shallow and deep networks for static PE malware detection. ICT Express, 4(4), 255-258.

[285] Le, Q., Boydell, O., Mac Namee, B., & Scanlon, M. (2018). Deep learning at the shallow end: Malware classification for non-domain experts. Digital Investigation, 26, S118-S126.

[286] Sun, G., & Qian, Q. (2018). Deep Learning and Visualization for Identifying Malware Families. IEEE Transactions on Dependable and Secure Computing.

[287] Kalash, M., Rochan, M., Mohammed, N., Bruce, N. D., Wang, Y., & Iqbal, F. (2018, February). Malware Classification with Deep Convolutional Neural Networks. In New Technologies, Mobility and Security (NTMS), 2018 9th IFIP International Conference on (pp. 1-5). IEEE.

[288] Rathore, H., Agarwal, S., Sahay, S. K., & Sewak, M. (2018, December). Malware Detection Using Machine Learning and Deep Learning. In International Conference on Big Data Analytics (pp. 402-411). Springer, Cham.

[289] Kim, J. Y., & Cho, S. B. (2018, November). Detecting Intrusive Malware with a Hybrid Generative Deep Learning Model. In International Conference on Intelligent Data Engineering and Automated Learning (pp. 499-507). Springer, Cham.

[290] Khan, R. U., Zhang, X., & Kumar, R. (2018). Analysis of ResNet and GoogleNet models for malware detection. Journal of Computer Virology and Hacking Techniques, 1-9.

[291] Kim, C. H., Kabanga, E. K., & Kang, S. J. (2018, February). Classifying malware using convolutional gated neural network. In 2018 20th International Conference on Advanced Communication Technology (ICACT) (pp. 40-44). IEEE.

[292] Ye, Y., Chen, L., Hou, S., Hardy, W., & Li, X. (2018). DeepAM: a heterogeneous deep learning framework for intelligent malware detection. Knowledge and Information Systems, 54(2), 265-285.

[293] Sewak, M., Sahay, S. K., & Rathore, H. (2018, August). An investigation of a deep learning based malware detection system. In Proceedings of the 13th International Conference on Availability, Reliability and Security (p. 26). ACM.

[294] Agrawal, R., Stokes, J. W., Marinescu, M., & Selvaraj, K. (2018, October). Robust neural malware detection models for emulation sequence learning. In MILCOM 2018-2018 IEEE Military Communications Conference (MILCOM) (pp. 1-8). IEEE.

[295] Alrawashdeh, K., & Purdy, C. (2018, July). Ransomware detection using limited precision deep learning structure in fpga. In NAECON 2018-IEEE National Aerospace and Electronics Conference (pp. 152-157). IEEE.

[296] Yakura, H., Shinozaki, S., Nishimura, R., Oyama, Y., & Sakuma, J. (2018, March). Malware Analysis of Imaged Binary Samples by Convolutional Neural Network with Attention Mechanism. In Proceedings of the Eighth ACM Conference on Data and Application Security and Privacy (pp. 127-134). ACM.

[297] Ghalaty, N. F., & Salem, M. B. (2018, December). A Hierarchical Framework to Detect Targeted Attacks using Deep Neural Network. In 2018 IEEE International Conference on Big Data (Big Data) (pp. 5021-5026). IEEE.

[298] Sang, D. V., Cuong, D. M., & Cuong, L. T. B. (2018, December). An Effective Ensemble Deep Learning Framework for Malware Detection. In Proceedings of the Ninth International Symposium on Information and Communication Technology (pp. 192-199). ACM.

[299] Sewak, M., Sahay, S. K., & Rathore, H. (2018, June). Comparison of deep learning and the classical machine learning algorithm for the malware detection. In 2018 19th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD) (pp. 293-296). IEEE.

[300] Abdelsalam, M., Krishnan, R., Huang, Y., & Sandhu, R. (2018, July). Malware detection in cloud infrastructures using convolutional neural networks. In 2018 IEEE 11th International Conference on Cloud Computing (CLOUD) (pp. 162-169). IEEE.

[301] Masabo, E., Kaawaase, K. S., & Sansa-Otim, J. (2018, May). Big data: deep learning for detecting malware. In 2018 IEEE/ACM Symposium on Software Engineering in Africa (SEiA) (pp. 20-26). IEEE.

[302] Wang, Y., Stokes, J. W., & Marinescu, M. (2018). NEURAL MALWARE CONTROL WITH DEEP REINFORCEMENT LEARNING.

[303] Chen, L. (2018). Deep Transfer Learning for Static Malware Classification. arXiv preprint arXiv:1812.07606.

[304] Agrawal, R., Stokes, J. W., Marinescu, M., & Selvaraj, K. (2018, April). Neural Sequential Malware Detection with Parameters. In 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 2656-2660). IEEE.

[305] Cui, Z., Xue, F., Cai, X., Cao, Y., Wang, G. G., & Chen, J. (2018). Detection of malicious code variants based on deep learning. IEEE Transactions on Industrial Informatics, 14(7), 3187-3196.

[306] Benadjila, R., Prouff, E., Strullu, R., Cagli, E., & Dumas, C. Study of Deep Learning Techniques for Side-Channel Analysis and Introduction to ASCAD Database.

[307] Perin, G., Ege, B., & van Woudenberg, J. (2018). Lowering the Bar: Deep Learning for Side Channel Analysis.

[308] Pfeifer, C., & Haddad, P. Spread: a new layer for profiled deep-learning side-channel attacks. Analysis, 6, 3.

[309] Picek, S., Samiotis, I. P., Kim, J., Heuser, A., Bhasin, S., & Legay, A. (2018, December). On the performance of convolutional neural networks for side-channel analysis. In International Conference on Security, Privacy, and Applied Cryptography Engineering (pp. 157-176). Springer, Cham.

[310] Timon, B. Non-Profiled Deep Learning-Based Side-Channel Attacks.

[311] Samiotis, I. P. (2018). Side-Channel Attacks using Convolutional Neural Networks: A Study on the performance of Convolutional Neural Networks on side-channel data.

[312] Wei, L., Luo, B., Li, Y., Liu, Y., & Xu, Q. (2018, December). I know what you see: Power side-channel attack on convolutional neural network accelerators. In Proceedings of the 34th Annual Computer Security Applications Conference (pp. 393-406). ACM.

[313] Yu, W., & Chen, J. (2018). Deep learning-assisted and combined attack: a novel side-channel attack. Electronics Letters, 54(19), 1114-1116.

[314] Hua, W., Zhang, Z., & Suh, G. E. (2018, June). Reverse engineering convolutional neural networks through side-channel information leaks. In Proceedings of the 55th Annual Design Automation Conference (p. 4). ACM.

[315] Tuor, A., Kaplan, S., Hutchinson, B., Nichols, N., & Robinson, S. (2017). Deep learning for unsupervised insider threat detection in structured cybersecurity data streams. arXiv preprint arXiv:1710.00811.

[316] Yuan, F., Cao, Y., Shang, Y., Liu, Y., Tan, J., & Fang, B. (2018, June). Insider Threat Detection with Deep Neural Network. In International Conference on Computational Science (pp. 43-54). Springer, Cham.

[317] Tuor, A. R., Baerwolf, R., Knowles, N., Hutchinson, B., Nichols, N., & Jasper, R. (2018, June). Recurrent Neural Network Language Models for Open Vocabulary Event-Level Cyber Anomaly Detection. In Workshops at the Thirty-Second AAAI Conference on Artificial Intelligence.

[318] Meng, F., Lou, F., Fu, Y., & Tian, Z. (2018, June). Deep learning based attribute classification insider threat detection for data security. In 2018 IEEE Third International Conference on Data Science in Cyberspace (DSC) (pp. 576-581). IEEE.

[319] Zhang, D., Zheng, Y., Wen, Y., Xu, Y., Wang, J., Yu, Y., & Meng, D. (2018, October). Role-based log analysis applying deep learning for insider threat detection. In Proceedings of the 1st Workshop on Security-Oriented Designs of Computer Architectures and Processors (pp. 18-20). ACM.

[320] Zhang, J., Chen, Y., & Ju, A. (2018). Insider threat detection of adaptive optimization DBN for behavior logs. Turkish Journal of Electrical Engineering & Computer Sciences, 26(2), 792-802.

[321] Lu, J., & Wong, R. K. (2019, January). Insider Threat Detection with Long Short-Term Memory. In Proceedings of the Australasian Computer Science Week Multiconference (p. 1). ACM.

[322] Hu, T., Niu, W., Zhang, X., Liu, X., Lu, J., & Liu, Y. (2019). An Insider Threat Detection Approach Based on Mouse Dynamics and Deep Learning. Security and Communication Networks, 2019.

[323] Xu, X. (2006). Adaptive intrusion detection based on machine learning: feature extraction, classifier construction and sequential pattern prediction. International Journal of Web Services Practices, 2(1-2), 49-58.

[324] Servin, A., Kudenko, D. (2008). Multi-agent reinforcement learning for intrusion detection. In Adaptive Agents and Multi-Agent Systems III. Adaptation and Multi-Agent Learning (pp. 211-223). Springer, Berlin, Heidelberg.

[325] Malialis, K., Kudenko, D. (2013). Large-scale DDoS response using cooperative reinforcement learning. In 11th EuropeanWorkshop on Multi-Agent Systems (EUMAS).

[326] Malialis, K. (2014). Distributed Reinforcement Learning for Network Intrusion Response (Doctoral dissertation, University of York).

[327] Anderson, H. S., Kharkar, A., Filar, B., Roth, P. (2017). Evading machine learning malware detection. Black Hat.

[328] Servin, A. Towards Traffic Anomaly Detection via Reinforcement Learning and Data Flow.

[329] Anderson, H. S., Kharkar, A., Filar, B., Evans, D., Roth, P. (2018). Learning to Evade Static PE Machine Learning Malware Models via Reinforcement Learning. arXiv preprint arXiv:1801.08917.

[330] Bttinger, K., Godefroid, P., Singh, R. (2018). Deep Reinforcement Fuzzing. arXiv preprint arXiv:1801.04589.

[331] Liu, M. Dong, K. Ota, J. Li and J. Wu, "Deep Reinforcement Learning based Smart Mitigation of DDoS Flooding in Software-Defined Networks," 2018 IEEE 23rd International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD), Barcelona, 2018, pp. 1-6. doi: 10.1109/CAMAD.2018.8514971

[332] Smadi, S., Aslam, N., & Zhang, L. (2018). Detection of online phishing email using dynamic evolving neural network based on reinforcement learning. Decision Support Systems, 107, 88-102.

[333] Wu, C., Shi, J., Yang, Y., & Li, W. (2018, November). Enhancing Machine Learning Based Malware Detection Model by Reinforcement Learning. In Proceedings of the 8th International Conference on Communication and Network Security (pp. 74-78). ACM.

[334] Wang, Y., Stokes, J. W., & Marinescu, M. (2018). NEURAL MALWARE CONTROL WITH DEEP REINFORCEMENT LEARNING.

[335] Z. Yuan, Y. Lu, Z. Wang, and Y. Xue, "Droid-Sec: Deep Learning in Android Malware Detection, Sigcomm 2014, pp. 371-372, 2014.

[336] Yuan, Z., Lu, Y., Xue, Y. (2016). Droiddetector: android malware characterization and detection using deep learning. Tsinghua Science and Technology, 21(1), 114-123.

[337] Nix, R. A. (2016). Applying Deep Learning Techniques to the Analysis of Android APKs.

[338] Xu, L., Zhang, D., Jayasena, N. and Cavazos, J., 2016, September. Hadm: Hybrid analysis for detection of malware. In Proceedings of SAI Intelligent Systems Conference (pp. 702-724). Springer, Cham.

[339] Xu, L. (2016). Android malware classification using parallelized machine learning methods (Doctoral dissertation, University of Delaware).

[340] X. Su, D. Zhang, W. Li, K. Zhao, H. P. Academy, and E. Engineering, "A Deep Learning Approach to Android Malware Feature Learning and Detection, 2016.

[341] S. Hou, A. Saas, Y. Ye, and L. Chen, "Droiddelver: An Android malware detection system using deep belief network based on API call blocks, Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 9998 LNCS. pp. 54-66, 2016.

[342] Z. Yuan, Y. Lu, and Y. Xue, "Droiddetector: Android malware characterization and detection using deep learning, Tsinghua Sci. Technol., vol. 21, no. 1, pp. 114-123, 2016.

[343] S. Hou, A. Saas, L. Chen, and Y. Ye, "Deep4MalDroid: A deep learning framework for Android malware detection based on Linux kernel system call graphs, Proc. - 2016 IEEE/WIC/ACM Int. Conf. Web Intell. Work. WIW 2016, pp. 104-111, 2017.

[344] Martn, A., Fuentes-Hurtado, F., Naranjo, V., Camacho, D. (2017, June). Evolving deep neural networks architectures for Android malware classification. In Evolutionary Computation (CEC), 2017 IEEE Congress on (pp. 1659-1666). IEEE.

[345] Hou, S., Ye, Y., Song, Y., Abdulhayoglu, M. (2017, August). Hindroid: An intelligent android malware detection system based on structured heterogeneous information network. In Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 1507-1515). ACM.

[346] Vinayakumar, R., Soman, K. P., Poornachandran, P. (2017, September). Deep android malware detection and classification. In Advances in Computing, Communications and Informatics (ICACCI), 2017 International Conference on (pp. 1677-1683). IEEE.

[347] Z. Wang, J. Cai, S. Cheng, and W. Li, "DroidDeepLearner: Identifying Android malware using deep learning, 37th IEEE Sarnoff Symp. Sarnoff 2016, pp. 160-165, 2017.

[348] M. Ganesh, P. Pednekar, P. Prabhuswamy, D. S. Nair, Y. Park, and H. Jeon, "CNN-Based Android Malware Detection, 2017 Int. Conf. Softw. Secur. Assur., pp. 60-65, 2017.

[349] D. Zhu, M. H. Jin, D. Wu, M. Y. Yang, and W. Chen, "DeepFlow: Deep Learning-Based Malware Detection by Mining Android Application for Abnormal Usage of Sensitive Data, pp. 0-5, 2017.

[350] S. Hou, A. Saas, L. Chen, Y. Ye, and T. Bourlai, "Deep neural networks for automatic Android malware detection, Proc. 2017 IEEE/ACM Int. Conf. Adv. Soc. Networks Anal. Mining, ASONAM 2017, pp. 803-810, 2017.

[351] N. McLaughlin et al., "Deep Android Malware Detection, Proc. Seventh ACM Conf. Data Appl. Secur. Priv. - CODASPY '17, pp. 301-308, 2017.

[352] R. Nix and J. Zhang, "Classification of Android apps and malware using deep neural networks, 2017 Int. Jt. Conf. Neural Networks, pp. 1871-1878, 2017.

[353] Huang, TonTon Hsien-De, and Hung-Yu Kao. "R2-D2: color-inspired convolutional neural network (cnn)- based android malware detections." arXiv preprint arXiv:1705.04448 (2017).

[354] F. Martinelli, F. Marulli, and F. Mercaldo, "Evaluating Convolutional Neural Network for Effective Mobile Malware Detection, Procedia Comput. Sci., vol. 112, pp. 2372-2381, 2017.

[355] H. Liang, Y. Song, and D. Xiao, "An end-To-end model for Android malware detection, 2017 IEEE Int. Conf. Intell. Secur. Informatics Secur. Big Data, ISI 2017, pp. 140-142, 2017.

[356] Yan, J., Qi, Y., Rao, Q. (2018). LSTM-Based Hierarchical Denoising Network for Android Malware Detection. Security and Communication Networks, 2018.

[357] Yakura, H., Shinozaki, S., Nishimura, R., Oyama, Y., Sakuma, J. (2018, March). Malware Analysis of Imaged Binary Samples by Convolutional Neural Network with Attention Mechanism. In Proceedings of the Eighth ACM Conference on Data and Application Security and Privacy (pp. 127-134). ACM.

[358] Saif, D., El-Gokhy, S. M., & Sallam, E. (2018). Deep Belief Networks-based framework for malware detection in Android systems. Alexandria engineering journal, 57(4), 4049-4057.

[359] W. Li, Z. Wang, J. Cai, and S. Cheng, "An Android Malware Detection Approach Using Weight-Adjusted Deep Learning, 2018 Int. Conf. Comput. Netw. Commun., pp. 437-441, 2018.

[360] Zhang, Yi, Yuexiang Yang, and Xiaolei Wang. "A Novel Android Malware Detection Approach Based on Convolutional Neural Network." Proceedings of the 2nd International Conference on Cryptography, Security, and Privacy. ACM, 2018.

[361] L. Shiqi, T. Shengwei, Y. Long, Y. Jiong, and S. Hua, "Android malicious code Classification using Deep Belief Network, KSII Trans. Internet Inf. Syst., vol. 12, no. 1, pp. 454-475, 2018.

[362] W. Wang, M. Zhao, and J. Wang, "Effective Android malware detection with a hybrid model based on deep autoencoder and convolutional neural network, J. Ambient Intell. Humaniz. Comput., vol. 0, no. 0, pp. 1-9, 2018

[363] E. M. B. Karbab, M. Debbabi, A. Derhab, and D. Mouheb, "MalDozer: Automatic framework for Android malware detection using deep learning, Digit. Investig., vol. 24, no. March, pp. S48-S59, 2018.

[364] K. Xu, Y. Li, R. H. Deng, and K. Chen, "DeepRefiner: Multi-layer Android Malware Detection System Applying Deep Neural Networks, 2018 IEEE Eur. Symp. Secur. Priv., pp. 473-487, 2018.

[365] D. Li, Z. Wang, and Y. Xue, "Fine-grained Android Malware Detection based on Deep Learning, 2018 IEEE Conf. Commun. Netw. Secur., vol. 1, no. L, pp. 1-2, 2018.

[366] C. Hasegawa and H. Iyatomi, "One-dimensional convolutional neural networks for Android malware detection, in 2018 IEEE 14th International Colloquium on Signal Processing & Its Applications (CSPA), 2018, no. March, pp. 99-102.

[367] H. Alshahrani, H. Mansourt, S. Thorn, A. Alshehri, A. Alzahrani, and H. Fu, "DDefender: Android application threat detection using static and dynamic analysis, 2018 IEEE Int. Conf. Consum. Electron., pp. 1-6, 2018

[368] R. Vinayakumar, K. P. Soman, P. Poornachandran, and S. Sachin Kumar, "Detecting Android malware using Long Short-term Memory (LSTM), J. Intell. Fuzzy Syst., vol. 34, no. 3, pp. 1277-1288, 2018.

[369] Dong, F., Wang, J., Li, Q., Xu, G., & Zhang, S. (2018). Defect prediction in android binary executables using deep neural network. Wireless Personal Communications, 102(3), 2261-2285.

[370] Duc, N. V., & Giang, P. T. (2018, December). NADM: Neural Network for Android Detection Malware. In Proceedings of the Ninth International Symposium on Information and Communication Technology (pp. 449-455). ACM.

[371] Zhao, L., Li, D., Zheng, G., & Shi, W. (2018, October). Deep Neural Network Based on Android Mobile Malware Detection System Using Opcode Sequences. In 2018 IEEE 18th International Conference on Communication Technology (ICCT) (pp. 1141-1147). IEEE.

[372] Xu, Z., Ren, K., Qin, S., & Craciun, F. (2018, November). CDGDroid: Android Malware Detection Based on Deep Learning Using CFG and DFG. In International Conference on Formal Engineering Methods (pp. 177-193). Springer, Cham.

[373] Zegzhda, P., Zegzhda, D., Pavlenko, E., & Ignatev, G. (2018, September). Applying deep learning techniques for Android malware detection. In Proceedings of the 11th International Conference on Security of Information and Networks (p. 7). ACM.

[374] Kim, T., Kang, B., Rho, M., Sezer, S., & Im, E. G. (2019). A Multimodal Deep Learning Method for Android Malware Detection Using Various Features. IEEE Transactions on Information Forensics and Security, 14(3), 773-788.

[375] Naway, A., & Li, Y. (2019). Android Malware Detection Using Autoencoder. arXiv preprint arXiv:1901.07315.

[376] Pekta, A., & Acarman, T. (2019). Deep learning for effective Android malware detection using API call graph embeddings. Soft Computing, 1-17.

[377] Anderson, H. S., Woodbridge, J., Filar, B. (2016, October). DeepDGA: Adversarially-tuned domain generation and detection. In Proceedings of the 2016 ACM Workshop on Artificial Intelligence and Security (pp. 13-21). ACM.

[378] Yang, W., Kong, D., Xie, T., Gunter, C. A. (2017, December). Malware detection in adversarial settings: Exploiting feature evolutions and confusions in android apps. In Proceedings of the 33rd Annual Computer Security Applications Conference (pp. 288-302). ACM.

[379] Rigaki, M. (2017). Adversarial Deep Learning Against Intrusion Detection Classifiers.

[380] Grosse, K., Papernot, N., Manoharan, P., Backes, M., McDaniel, P. (2017, September). Adversarial examples for malware detection. In European Symposium on Research in Computer Security (pp. 62-79). Springer, Cham.

[381] Kim, J. Y., Bu, S. J., & Cho, S. B. (2017, November). Malware detection using deep transferred generative adversarial networks. In International Conference on Neural Information Processing (pp. 556-564). Springer, Cham.

[382] Hu, W., & Tan, Y. (2017). Generating adversarial malware examples for black-box attacks based on GAN. arXiv preprint arXiv:1702.05983.

[383] Wang, Q., Guo, W., Zhang, K., Ororbia II, A. G., Xing, X., Liu, X., & Giles, C. L. (2017, August). Adversary resistant deep neural networks with an application to malware detection. In Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 1145-1153). ACM.

[384] Stokes, J. W., Wang, D., Marinescu, M., Marino, M., & Bussone, B. (2017). Attack and defense of dynamic analysis-based, adversarial neural malware classification models. arXiv preprint arXiv:1712.05919.

[385] Huang, A., Al-Dujaili, A., Hemberg, E., O'Reilly, U. M. (2018). Adversarial Deep Learning for Robust Detection of Binary Encoded Malware. arXiv preprint arXiv:1801.02950.

[386] Kolosnjaji, B., Demontis, A., Biggio, B., Maiorca, D., Giacinto, G., Eckert, C., Roli, F. (2018). Adversarial Malware Binaries: Evading Deep Learning for Malware Detection in Executables. arXiv preprint arXiv:1803.04173.

[387] Paudice, A., Muoz-Gonzlez, L., Gyorgy, A., Lupu, E. C. (2018). Detection of Adversarial Training Examples in Poisoning Attacks through Anomaly Detection. arXiv preprint arXiv:1802.03041.

[388] Celik, Z. B., McDaniel, P. (2018, May). Extending Detection with Privileged Information via Generalized Distillation. In 2018 IEEE Security and Privacy Workshops (SPW) (pp. 83-88). IEEE.

[389] Sitawarin, C., Bhagoji, A. N., Mosenia, A., Mittal, P., Chiang, M. (2018). Rogue signs: Deceiving traffic sign recognition with malicious ads and logos. arXiv preprint arXiv:1801.02780.

[390] Gao, J., Lanchantin, J., Soffa, M. L., Qi, Y. (2018). Black-box Generation of Adversarial Text Sequences to Evade Deep Learning Classifiers. arXiv preprint arXiv:1801.04354.

[391] Carlini, N., Wagner, D. (2018). Audio adversarial examples: Targeted attacks on speech-to-text. arXiv preprint arXiv:1801.01944.

[392] Aghakhani, H., Machiry, A., Nilizadeh, S., Kruegel, C., Vigna, G. (2018). Detecting Deceptive Reviews using Generative Adversarial Networks. arXiv preprint arXiv:1805.10364.

[393] Rigaki, M., Garcia, S. Bringing a GAN to a Knife-fight: Adapting Malware Communication to Avoid Detection.

[394] Intrator, Y., Katz, G., & Shabtai, A. (2018). Mdgan: Boosting anomaly detection using
multi-discriminator generative adversarial networks. arXiv preprint arXiv:1810.05221.

[395] Yin, C., Zhu, Y., Liu, S., Fei, J., & Zhang, H. (2018, May). An enhancing framework for botnet detection using generative adversarial networks. In 2018 International Conference on Artificial Intelligence and Big Data (ICAIBD) (pp. 228-234). IEEE.

[396] Kim, J. Y., Bu, S. J., & Cho, S. B. (2018). Zero-day malware detection using transferred generative adversarial networks based on deep autoencoders. Information Sciences, 460, 83-102.

[397] Madani, P., & Vlajic, N. (2018, April). Robustness of deep autoencoder in intrusion detection under adversarial contamination. In Proceedings of the 5th Annual Symposium and Bootcamp on Hot Topics in the Science of Security (p. 1). ACM.

[398] Suciu, O., Coull, S. E., & Johns, J. (2018). Exploring adversarial examples in malware detection. arXiv preprint arXiv:1810.08280.

[399] Li, D., Li, Q., Ye, Y., & Xu, S. (2018). Enhancing Robustness of Deep Neural Networks Against Adversarial Malware Samples: Principles, Framework, and AICS'2019 Challenge. arXiv preprint arXiv:1812.08108.

[400] Bhaskara, V. S., & Bhattacharyya, D. (2018). Emulating malware authors for proactive protection using GANs over a distributed image visualization of the dynamic file behavior. arXiv preprint arXiv:1807.07525.

[401] Kreuk, F., Barak, A., Aviv-Reuven, S., Baruch, M., Pinkas, B., & Keshet, J. (2018). Deceiving end-to-end deep learning malware detectors using adversarial examples. In CoRR.

[402] Anand, A., Gorde, K., Moniz, J. R. A., Park, N., Chakraborty, T., & Chu, B. T. (2018, December). Phishing URL Detection with Oversampling based on Text Generative Adversarial Networks. In 2018 IEEE International Conference on Big Data (Big Data) (pp. 1168-1177). IEEE.

[403] Lin, Z., Shi, Y., & Xue, Z. (2018). Idsgan: Generative adversarial networks for attack generation against intrusion detection. arXiv preprint arXiv:1809.02077.

[404] Seo, E., Song, H. M., & Kim, H. K. (2018, August). GIDS: GAN based Intrusion Detection System for In-Vehicle Network. In 2018 16th Annual Conference on Privacy, Security and Trust (PST) (pp. 1-6). IEEE.

[405] Salem, M., Taheri, S., & Yuan, J. S. (2018). Anomaly Generation using Generative Adversarial Networks in Host Based Intrusion Detection. arXiv preprint arXiv:1812.04697.

[406] Huang, C. H., Lee, T. H., Chang, L. H., Lin, J. R., & Horng, G. (2018, June). Adversarial Attacks on SDN-Based Deep Learning IDS System. In International Conference on Mobile and Wireless Technology (pp. 181-191). Springer, Singapore.

[407] Yang, K., Liu, J., Zhang, C., & Fang, Y. (2018, October). Adversarial Examples Against the Deep Learning Based Network Intrusion Detection Systems. In MILCOM 2018-2018 IEEE Military Communications Conference (MILCOM) (pp. 559-564). IEEE.

[408] Ye, G., Tang, Z., Fang, D., Zhu, Z., Feng, Y., Xu, P., ... & Wang, Z. (2018, October). Yet Another Text Captcha Solver: A Generative Adversarial Network Based Approach. In Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security (pp. 332-348). ACM.

[409] Tramr, F., Dupr, P., Rusak, G., Pellegrino, G., & Boneh, D. (2018). Ad-versarial: Defeating Perceptual Ad-Blocking. arXiv preprint arXiv:1811.03194.

[410] Liu, F., Li, Z., Li, X., & Lv, T. (2018, December). A Text-Based CAPTCHA Cracking System with Generative Adversarial Networks. In 2018 IEEE International Symposium on Multimedia (ISM) (pp. 192-193). IEEE.

[411] Rosenberg, I., Shabtai, A., Rokach, L., & Elovici, Y. (2018, September). Generic Black-Box End-to-End Attack Against State of the Art API Call Based Malware Classifiers. In International Symposium on Research in Attacks, Intrusions, and Defenses (pp. 490-510). Springer, Cham.

[412] Li, D., Baral, R., Li, T., Wang, H., Li, Q., & Xu, S. (2018). HashTran-DNN: A Framework for Enhancing Robustness of Deep Neural Networks against Adversarial Malware Samples. arXiv preprint arXiv:1809.06498.

[413] Kim, J. Y., & Cho, S. B. (2018, November). Detecting Intrusive Malware with a Hybrid Generative Deep Learning Model. In International Conference on Intelligent Data Engineering and Automated Learning (pp. 499-507). Springer, Cham.

[414] Ring, M., Schlr, D., Landes, D., & Hotho, A. (2019). Flow-based network traffic generation using generative adversarial networks. Computers & Security, 82, 156-172.

[415] Zhang, H., Yu, X., Ren, P., Luo, C., & Min, G. (2019). Deep Adversarial Learning in Intrusion Detection: A Data Augmentation Enhanced Framework. arXiv preprint arXiv:1901.07949.

[416] Demetrio, L., Biggio, B., Lagorio, G., Roli, F., & Armando, A. (2019). Explaining Vulnerabilities of Deep Learning to Adversarial Malware Binaries. arXiv preprint arXiv:1901.03583.

[417] Yan, Q., Wang, M., Huang, W., Luo, X., & Yu, F. R. (2019). Automatically synthesizing DoS attack traces using generative adversarial networks. International Journal of Machine Learning and Cybernetics, 1-10.

[418] Sidi, L., Nadler, A., & Shabtai, A. (2019). MaskDGA: A Black-box Evasion Technique Against DGA Classifiers and Adversarial Defenses. arXiv preprint arXiv:1902.08909.

[419] Nataraj, L., Mohammed, T. M., Manjunath, B. S., Chandrasekaran, S., Flenner, A., Bappy, J. H., & Roy-Chowdhury, A. K. (2019). Detecting GAN generated Fake Images using Co-occurrence Matrices. arXiv preprint arXiv:1903.06836.

[420] Ni, S., Qian, Q., & Zhang, R. (2018). Malware identification using visualization images and deep learning. Computers & Security, 77, 871-885.

[421] Williams, G., Baxter, R., He, H., Hawkins, S., Gu, L. (2002). A comparative study of RNN for outlier detection in data mining. In Data Mining, 2002. ICDM 2003. Proceedings. 2002 IEEE International Conference on (pp. 709-712). IEEE.

[422] Staudemeyer, R. C., Omlin, C. W. (2013, October). Evaluating performance of long short-term memory recurrent neural networks on intrusion detection data. In Proceedings of the South African Institute for Computer Scientists and Information Technologists Conference (pp. 218-224). ACM.

[423] Gao, N., Gao, L., Gao, Q., & Wang, H. (2014, November). An intrusion detection model based on deep belief networks. In Advanced Cloud and Big Data (CBD), 2014 Second International Conference on (pp. 247-252). IEEE.

[424] Staudemeyer, R. C. (2015). Applying long short-term memory recurrent neural networks to intrusion detection. South African Computer Journal, 56(1), 136-154.

[425] Li, Y., Ma, R., & Jiao, R. (2015). A hybrid malicious code detection method based on deep learning. methods, 9(5).

[426] Potluri, S., & Diedrich, C. (2016, September). Accelerated deep neural networks for enhanced Intrusion Detection System. In Emerging Technologies and Factory Automation (ETFA), 2016 IEEE 21st International Conference on (pp. 1-8). IEEE.

[427] Alrawashdeh, K., & Purdy, C. (2016, December). Toward an online anomaly intrusion detection system based on deep learning. In Machine Learning and Applications (ICMLA), 2016 15th IEEE International Conference on (pp. 195-200). IEEE.

[428] Javaid, A., Niyaz, Q., Sun, W., Alam, M.: A deep learning approach for network intrusion detection system. In: Proceedings of the 9th EAI International Conference on Bio-Inspired Information and Communications Technologies (Formerly BIONETICS), pp. 21-26. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering) (2016)

[429] Niyaz, Q., Sun, W., Javaid, A. Y. (2016). A deep learning based DDoS detection system in software-defined networking (SDN). arXiv preprint arXiv:1611.07400.

[430] Tang, T. A., Mhamdi, L., McLernon, D., Zaidi, S. A. R., Ghogho, M. (2016, October). Deep learning approach for network intrusion detection in software defined networking. In Wireless Networks and Mobile Communications (WINCOM), 2016 International Conference on (pp. 258-263). IEEE.

[431] Kim, G., Yi, H., Lee, J., Paek, Y., Yoon, S. (2016). LSTM-based system-call language modeling and robust ensemble method for designing host-based intrusion detection systems. arXiv preprint arXiv:1611.01726.

[432] Kim, J., Kim, J., Thu, H. L. T., Kim, H. (2016, February). Long short term memory recurrent neural network classifier for intrusion detection. In Platform Technology and Service (PlatCon), 2016 International Conference on (pp. 1-5). IEEE.

[433] Vinayakumar, R., Soman, K. P., Poornachandran, P. (2017, September). Applying convolutional neural network for network intrusion detection. In Advances in Computing, Communications and Informatics (ICACCI), 2017 International Conference on (pp. 1222-1228). IEEE.

[434] Li, Z., Qin, Z., Huang, K., Yang, X., Ye, S. (2017, November). Intrusion detection using convolutional neural networks for representation learning. In International Conference on Neural Information Processing (pp. 858-866). Springer, Cham

[435] Yu, Y., Long, J., Cai, Z. (2017). Network intrusion detection through stacking dilated convolutional autoencoders. Security and Communication Networks, 2017.

[436] Yousefi-Azar, M., Varadharajan, V., Hamey, L., Tupakula, U. (2017, May). Autoencoder-based feature learning for Cyber Security applications. In Neural Networks (IJCNN), 2017 International Joint Conference on (pp. 3854-3861). IEEE.

[437] Dymshits, M., Myara, B., Tolpin, D. (2017, October). Process monitoring on sequences of system call count vectors. In Security Technology (ICCST), 2017 International Carnahan Conference on (pp. 1-5). IEEE.

[438] Kim, J., Kim, H. (2017, February). An Effective Intrusion Detection Classifier Using Long Short-Term Memory with Gradient Descent Optimization. In Platform Technology and Service (PlatCon), 2017 International Conference on (pp. 1-6). IEEE.

[439] Thi, N. N., Le-Khac, N. A. (2017). One-Class Collective Anomaly Detection Based on LSTM-RNNs. In Transactions on Large-Scale Dataand Knowledge-Centered Systems XXXVI (pp. 73-85). Springer, Berlin, Heidelberg.

[440] Bediako, P. K. (2017). Long Short-Term Memory Recurrent Neural Network for detecting DDoS flooding attacks within TensorFlow Implementation framework.

[441] Vinayakumar, R., Soman, K. P., & Poornachandran, P. (2017). Evaluation of Recurrent Neural Network and its Variants for Intrusion Detection System (IDS). International Journal of Information System Modeling and Design (IJISMD), 8(3), 43-63.

[442] Meng, F., Fu, Y., Lou, F., & Chen, Z. (2017, December). An Effective Network Attack Detection Method Based on Kernel PCA and LSTM-RNN. In 2017 International Conference on Computer Systems, Electronics and Control (ICCSEC) (pp. 568-572). IEEE.

[443] Ding, S., & Wang, G. (2017, December). Research on intrusion detection technology based on deep learning. In Computer and Communications (ICCC), 2017 3rd IEEE International Conference on (pp. 1474-1478). IEEE.

[444] Yin, C., Zhu, Y., Fei, J., & He, X. (2017). A deep learning approach for intrusion detection using recurrent neural networks. IEEE Access, 5, 21954-21961.

[445] Kaynar, O., Yksek, A. G., Grmez, Y., & Iik, Y. E. (2017, May). Intrusion detection with autoencoder based deep learning machine. In Signal Processing and Communications Applications Conference (SIU), 2017 25th (pp. 1-4). IEEE.

[446] Li, Z., Qin, Z., Huang, K., Yang, X., & Ye, S. (2017, November). Intrusion detection using convolutional neural networks for representation learning. In International Conference on Neural Information Processing (pp. 858-866). Springer, Cham.

[447] Van, N. T., Thinh, T. N., & Sach, L. T. (2017, July). An anomaly-based network intrusion detection system using deep learning. In System Science and Engineering (ICSSE), 2017 International Conference on (pp. 210-214). IEEE.

[448] Yu, Y., Long, J., & Cai, Z. (2017, October). Session-Based Network Intrusion Detection Using a Deep Learning Architecture. In Modeling Decisions for Artificial Intelligence(pp. 144-155). Springer, Cham.

[449] Huang, H., Khalid, R.S., Liu, W., Yu, H.: Work-in-progress: a fast online sequential learning accelerator for IoT network intrusion detection. In: 2017 International Conference on Hardware/Software Codesign and System Synthesis (CODES+ ISSS), pp. 1-2. IEEE (2017)

[450] Alrawashdeh, K., & Purdy, C. (2017, June). Reducing calculation requirements in FPGA implementation of deep learning algorithms for online anomaly intrusion detection. In Aerospace and Electronics Conference (NAECON), 2017 IEEE National (pp. 57-62). IEEE.

[451] Blanco, R., Cilla, J. J., Malagn, P., Penas, I., Moya, J. M. (2018, June). Tuning CNN Input Layout for IDS with Genetic Algorithms. In International Conference on Hybrid Artificial Intelligence Systems (pp. 197-209). Springer, Cham.

[452] Zhu, M., Ye, K., Xu, C. Z. (2018, June). Network Anomaly Detection and Identification Based on Deep Learning Methods. In International Conference on Cloud Computing (pp. 219-234). Springer, Cham.

[453] Wang, W., Sheng, Y., Wang, J., Zeng, X., Ye, X., Huang, Y., Zhu, M. (2018). HAST-IDS: learning hierarchical spatial-temporal features using deep neural networks to improve intrusion detection. IEEE Access, 6, 1792-1806.

[454] Agarap, A. F. M. (2018, February). A Neural Network Architecture Combining Gated Recurrent Unit (GRU) and Support Vector Machine (SVM) for Intrusion Detection in Network Traffic Data. In Proceedings of the 2018 10th International Conference on Machine Learning and Computing (pp. 26-30). ACM.

[455] Radford, B. J., Apolonio, L. M., Trias, A. J., Simpson, J. A. (2018). Network Traffic Anomaly Detection Using Recurrent Neural Networks. arXiv preprint arXiv:1803.10769.

[456] Ponkarthika, M., Saraswathy, V. R. Network Intrusion Detection Using Deep Neural Networks.

[457] Lee, B., Amaresh, S., Green, C., Engels, D. (2018). Comparative Study of Deep Learning Models for Network Intrusion Detection. SMU Data Science Review, 1(1), 8.

[458] Doshi, R., Apthorpe, N., Feamster, N. (2018). Machine Learning DDoS Detection for Consumer Internet of Things Devices. arXiv preprint arXiv:1804.04159.

[459] Chong, P., Tan, Y. X. M., Guarnizo, J., Elovici, Y., Binder, A. (2018, May). Mouse Authentication Without the Temporal AspectWhat Does a 2D-CNN Learn?. In 2018 IEEE Security and Privacy Workshops (SPW) (pp. 15-21). IEEE

[460] Potluri, S., Ahmed, S., & Diedrich, C. (2018, December). Convolutional Neural Networks for Multi-class Intrusion Detection System. In International Conference on Mining Intelligence and Knowledge Exploration (pp. 225-238). Springer, Cham.

[461] Roy, B., & Cheung, H. (2018, November). A Deep Learning Approach for Intrusion Detection in Internet of Things using Bi-Directional Long Short-Term Memory Recurrent Neural Network. In 2018 28th International Telecommunication Networks and Applications Conference (ITNAC) (pp. 1-6). IEEE.

[462] Aksu, D., & Aydin, M. A. (2018, December). Detecting Port Scan Attempts with Comparative Analysis of Deep Learning and Support Vector Machine Algorithms. In 2018 International Congress on Big Data, Deep Learning and Fighting Cyber Terrorism (IBIGDELFT) (pp. 77-80). IEEE.

[463] Tchakoucht, T. A., & Ezziyyani, M. (2018). Multilayered Echo-State Machine: A Novel Architecture for Efficient Intrusion Detection. IEEE Access, 6, 72458-72468.

[464] Amarasinghe, K., & Manic, M. (2018, October). Improving User Trust on Deep Neural Networks based Intrusion Detection Systems. In IECON 2018-44th Annual Conference of the IEEE Industrial Electronics Society (pp. 3262-3268). IEEE.

[465] Nguyen, K. K., Hoang, D. T., Niyato, D., Wang, P., Nguyen, D., & Dutkiewicz, E. (2018, April). Cyberattack detection in mobile cloud computing: A deep learning approach. In Wireless Communications and Networking Conference (WCNC), 2018 IEEE (pp. 1-6). IEEE.

[466] Madani, P., & Vlajic, N. (2018, April). Robustness of deep autoencoder in intrusion detection under adversarial contamination. In Proceedings of the 5th Annual Symposium and Bootcamp on Hot Topics in the Science of Security (p. 1). ACM.

[467] Al-Qatf, M., Lasheng, Y., Al-Habib, M., & Al-Sabahi, K. (2018). Deep Learning Approach Combining Sparse Autoencoder With SVM for Network Intrusion Detection. IEEE Access, 6, 52843-52856.

[468] Rajkumar, N., D'Souza, A., Alex, S., & Kathrine, G. J. W. (2018, May). Long Short-Term Memory-Based Recurrent Neural Network Approach for Intrusion Detection. In International Conference on ISMAC in Computational Vision and Bio-Engineering (pp. 837-846). Springer, Cham.

[469] Hsu, C. M., Hsieh, H. Y., Prakosa, S. W., Azhari, M. Z., & Leu, J. S. (2018, October). Using Long-Short-Term Memory Based Convolutional Neural Networks for Network Intrusion Detection. In International Wireless Internet Conference (pp. 86-94). Springer, Cham.

[470] Wang, S., Li, B., Yang, M., & Yan, Z. (2018, October). Intrusion Detection for WiFi Network: A Deep Learning Approach. In International Wireless Internet Conference (pp. 95-104). Springer, Cham.

[471] Lin, S. Z., Shi, Y., & Xue, Z. (2018, July). Character-Level Intrusion Detection Based On Convolutional Neural Networks. In 2018 International Joint Conference on Neural Networks (IJCNN) (pp. 1-8). IEEE.

[472] Naseer, S., Saleem, Y., Khalid, S., Bashir, M. K., Han, J., Iqbal, M. M., & Han, K. (2018). Enhanced network anomaly detection based on deep neural networks. IEEE Access, 6, 48231-48246.

[473] Wang, W., Sheng, Y., Wang, J., Zeng, X., Ye, X., Huang, Y., & Zhu, M. (2018). HAST-IDS: learning hierarchical spatial-temporal features using deep neural networks to improve intrusion detection. IEEE Access, 6, 1792-1806.

[474] Rahul, V. K., Vinayakumar, R., Soman, K. P., & Poornachandran, P. (2018, July). Evaluating Shallow and Deep Neural Networks for Network Intrusion Detection Systems in Cyber Security. In 2018 9th International Conference on Computing, Communication and Networking Technologies (ICCCNT) (pp. 1-6). IEEE.

[475] Kim, T., Suh, S. C., Kim, H., Kim, J., & Kim, J. (2018, December). An Encoding Technique for CNN-based Network Anomaly Detection. In 2018 IEEE International Conference on Big Data (Big Data) (pp. 2960-2965). IEEE.

[476] Pekta, A., & Acarman, T. A deep learning method to detect network intrusion through flowbased features. International Journal of Network Management, e2050.

[477] Wu, K., Chen, Z., & Li, W. (2018). A Novel Intrusion Detection Model for a Massive Network Using Convolutional Neural Networks. IEEE Access, 6, 50850-50859.

[478] Ito, M., & Iyatomi, H. (2018, March). Web application firewall using character-level convolutional neural network. In Signal Processing & Its Applications (CSPA), 2018 IEEE 14th International Colloquium on (pp. 103-106). IEEE.

[479] Zhu, M., Ye, K., Wang, Y., & Xu, C. Z. (2018, November). A Deep Learning Approach for Network Anomaly Detection Based on AMF-LSTM. In IFIP International Conference on Network and Parallel Computing (pp. 137-141). Springer, Cham.

[480] Shone, N., Ngoc, T. N., Phai, V. D., & Shi, Q. (2018). A deep learning approach to network intrusion detection. IEEE Transactions on Emerging Topics in Computational Intelligence, 2(1), 41-50.

[481] Mirsky, Y., Doitshman, T., Elovici, Y., & Shabtai, A. (2018). Kitsune: an ensemble of autoencoders for online network intrusion detection. arXiv preprint arXiv:1802.09089.

[482] Farahnakian, F., & Heikkonen, J. (2018, February). A deep auto-encoder based approach for intrusion detection system. In Advanced Communication Technology (ICACT), 2018 20th International Conference on (pp. 178-183). IEEE.

[483] Kwon, D., Natarajan, K., Suh, S. C., Kim, H., & Kim, J. (2018, July). An empirical study on network anomaly detection using convolutional neural networks. In 2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS) (pp. 1595-1598). IEEE.

[484] Kim, K., Aminanto, M. E., & Tanuwidjaja, H. C. (2018). Deep Learning-Based IDSs. In Network Intrusion Detection using Deep Learning (pp. 35-45). Springer, Singapore.

[485] Li, Z., & Qin, Z. (2018, December). A Semantic Parsing Based LSTM Model for Intrusion Detection. In International Conference on Neural Information Processing (pp. 600-609). Springer, Cham.

[486] Ieracitano, C., Adeel, A., Gogate, M., Dashtipour, K., Morabito, F. C., Larijani, H., ... & Hussain, A. (2018, July). Statistical Analysis Driven Optimized Deep Learning System for Intrusion Detection. In International Conference on Brain Inspired Cognitive Systems (pp. 759-769). Springer, Cham.

[487] Diro, A., & Chilamkurti, N. (2018). Leveraging LSTM networks for attack detection in fog-to-things communications. IEEE Communications Magazine, 56(9), 124-130.

[488] Yan, B., & Han, G. (2018). LA-GRU: Building Combined Intrusion Detection Model Based on Imbalanced Learning and Gated Recurrent Unit Neural Network. Security and Communication Networks, 2018.

[489] Kunang, Y. N., Nurmaini, S., Stiawan, D., Zarkasi, A., & Jasmir, F. (2018, October). Automatic Features Extraction Using Autoencoder in Intrusion Detection System. In 2018 International Conference on Electrical Engineering and Computer Science (ICECOS) (pp. 219-224). IEEE.

[490] Yang, X., Gao, L., Wang, H., Zheng, J., & Cao, R. (2018, August). A Cooperative Deep Belief Network for Intrusion Detection. In 2018 Sixth International Conference on Advanced Cloud and Big Data (CBD) (pp. 230-236). IEEE.

[491] Mubalaike, A. M., & Adali, E. (2018, September). Deep Learning Approach for Intelligent Financial Fraud Detection System. In 2018 3rd International Conference on Computer Science and Engineering (UBMK) (pp. 598-603). IEEE.

[492] Wang, Y., & Zhang, J. (2018, November). DeepPort: Detect Low Speed Port Scan Using Convolutional Neural Network. In International Conference on Bio-Inspired Computing: Theories and Applications (pp. 368-379). Springer, Singapore.

[493] Li, L., Xie, L., Li, W., Liu, Z., & Wang, Z. (2018). Improved Deep Belief Networks (IDBN) Dynamic Model-Based Detection and Mitigation for Targeted Attacks on Heavy-Duty Robots. Applied Sciences (2076-3417), 8(5).

[494] Marir, N., Wang, H., Feng, G., Li, B., & Jia, M. (2018). Distributed abnormal behavior detection approach based on deep belief network and ensemble svm using spark. IEEE Access, 6, 59657-59671.

[495] Al Rawashdeh, K. (2018). Toward a Hardware-assisted Online Intrusion Detection System Based on Deep Learning Algorithms for Resource-Limited Embedded Systems(Doctoral dissertation, University of Cincinnati).

[496] Muna, A. H., Moustafa, N., & Sitnikova, E. (2018). Identification of malicious activities in industrial internet of things based on deep learning models. Journal of Information Security and Applications, 41, 1-11.

[497] Alrawashdeh, K., & Purdy, C. (2018, May). Fast activation function approach for deep learning based online anomaly intrusion detection. In 2018 IEEE 4th International Conference on Big Data Security on Cloud (BigDataSecurity), IEEE International Conference on High Performance and Smart Computing,(HPSC) and IEEE International Conference on Intelligent Data and Security (IDS) (pp. 5-13). IEEE.

[498] Min, E., Long, J., Liu, Q., Cui, J., & Chen, W. (2018). TR-IDS: Anomaly-based intrusion detection through text-convolutional neural network and random forest. Security and Communication Networks, 2018.

[499] Cui, J., Long, J., Min, E., & Mao, Y. (2018, October). WEDL-NIDS: Improving Network Intrusion Detection Using Word Embedding-Based Deep Learning Method. In International Conference on Modeling Decisions for Artificial Intelligence(pp. 283-295). Springer, Cham.

[500] Mighan, S. N., & Kahani, M. (2018, May). Deep Learning Based Latent Feature Extraction for Intrusion Detection. In Electrical Engineering (ICEE), Iranian Conference on (pp. 1511-1516). IEEE.

[501] Wang, H., Wen, Y., & Zhao, D. Identifying localization attacks in wireless sensor networks using deep learning. Journal of Intelligent & Fuzzy Systems, (Preprint), 1-13.

[502] Anani, W. (2018). Recurrent Neural Network Architectures Toward Intrusion Detection (Doctoral dissertation, The University of Western Ontario).

[503] Imamverdiyev, Y., & Abdullayeva, F. (2018). Deep Learning Method for Denial of Service Attack Detection Based on Restricted Boltzmann Machine. Big Data, 6(2), 159-169.

[504] Diro, A. A., & Chilamkurti, N. (2018). Distributed attack detection scheme using deep learning approach for Internet of Things. Future Generation Computer Systems, 82, 761-768.

[505] Al Najada, H., Mahgoub, I., & Mohammed, I. (2018, November). Cyber Intrusion Prediction and Taxonomy System Using Deep Learning And Distributed Big Data Processing. In 2018 IEEE Symposium Series on Computational Intelligence (SSCI) (pp. 631-638). IEEE.

[506] Yavuz, F. Y., nal, D., & Gl, E. (2018). Deep learning for detection of routing attacks in the internet of things. International Journal of Computational Intelligence Systems, 12(1), 39-58.

[507] Nicholas, L., Ooi, S. Y., Pang, Y. H., Hwang, S. O., & Tan, S. Y. (2018). Study of long short-term memory in flow-based network intrusion detection system. Journal of Intelligent & Fuzzy Systems, (Preprint), 1-11.

[508] Abdulhammed, R., Faezipour, M., Abuzneid, A., & AbuMallouh, A. (2018). Deep and Machine Learning Approaches for Anomaly-Based Intrusion Detection of Imbalanced Network Traffic. IEEE Sensors Letters.

[509] Yan, B., & Han, G. (2018). Effective feature extraction via stacked sparse autoencoder to improve intrusion detection system. IEEE Access, 6, 41238-41248.

[510] Jia, Y., Wang, M., & Wang, Y. (2018). Network intrusion detection algorithm based on deep neural network. IET Information Security, 13(1), 48-53.

[511] Maim, L. F., Gmez, . L. P., Clemente, F. J. G., Prez, M. G., & Prez, G. M. (2018). A self-adaptive deep learning-based system for anomaly detection in 5G networks. IEEE Access, 6, 7700-7712.

[512] Maim, L. F., Celdrn, A. H., Prez, M. G., Clemente, F. J. G., & Prez, G. M. (2018). Dynamic management of a deep learning-based anomaly detection system for 5G networks. Journal of Ambient Intelligence and Humanized Computing, 1-15.

[513] Zhang, H., Wu, C. Q., Gao, S., Wang, Z., Xu, Y., & Liu, Y. (2018, August). An Effective Deep Learning Based Scheme for Network Intrusion Detection. In 2018 24th International Conference on Pattern Recognition (ICPR) (pp. 682-687). IEEE.

[514] Yudhana, A., Riadi, I., & Ridho, F. (2018). DDoS Classification Using Neural Network and Nave Bayes Methods for Network Forensics. INTERNATIONAL JOURNAL OF ADVANCED COMPUTER SCIENCE AND APPLICATIONS, 9(11), 177-183.

[515] Zhao, J., Shetty, S., Pan, J. W., Kamhoua, C., & Kwiat, K. (2019). Transfer learning for detecting unknown network attacks. EURASIP Journal on Information Security, 2019(1), 1.

[516] Khan, F. A., Gumaei, A., Derhab, A., & Hussain, A. (2019). TSDL: A TwoStage Deep Learning Model for Efficient Network Intrusion Detection. IEEE Access.

[517] Moore, M. R., & Vann, J. M. (2019, January). Anomaly Detection of Cyber Physical Network Data Using 2D Images. In 2019 IEEE International Conference on Consumer Electronics (ICCE) (pp. 1-5). IEEE.

[518] Roopak, M., Tian, G. Y., & Chambers, J. (2019, January). Deep Learning Models for Cyber Security in IoT Networks. In 2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC) (pp. 0452-0457). IEEE.

[519] Dawoud, A., Shahristani, S., & Raun, C. (2019, March). Dimensionality Reduction for Network Anomalies Detection: A Deep Learning Approach. In Workshops of the International Conference on Advanced Information Networking and Applications (pp. 957-965). Springer, Cham.

[520] Liu, H., Lang, B., Liu, M., & Yan, H. (2019). CNN and RNN based payload classification methods for attack detection. Knowledge-Based Systems, 163, 332-341.

[521] Rezvy S., Petridis M., Lasebae A., Zebin T. (2019) Intrusion Detection and Classification with Autoencoded Deep Neural Network. In: Lanet JL., Toma C. (eds) Innovative Security Solutions for Information Technology and Communications. SECITC 2018. Lecture Notes in Computer Science, vol 11359. Springer, Cham

[522] Yang, Y., Zheng, K., Wu, C., Niu, X., & Yang, Y. (2019). Building an Effective Intrusion Detection System Using the Modified Density Peak Clustering Algorithm and Deep Belief Networks. Applied Sciences, 9(2), 238.

[523] Papamartzivanos, D., Mrmol, F. G., & Kambourakis, G. (2019). Introducing Deep Learning Self-Adaptive Misuse Network Intrusion Detection Systems. IEEE Access.

[524] Wang, Z. (2015). The applications of deep learning on traffic identification. BlackHat USA.

[525] Oliveira, T. P., Barbar, J. S., & Soares, A. S. (2016). Computer network traffic prediction: a comparison between traditional and deep learning neural networks. International Journal of Big Data Intelligence, 3(1), 28-37.

[526] Lotfollahi, M., Shirali, R., Siavoshani, M. J., Saberian, M. (2017). Deep Packet: A Novel Approach For Encrypted Traffic Classification Using Deep Learning. arXiv preprint arXiv:1709.02656.

[527] Smit, D., Millar, K., Page, C., Cheng, A., Chew, H. G., Lim, C. C. (2017). Looking deeper: Using deep learning to identify internet communications traffic. Macquarie Matrix: Special edition, ACUR, 1,1318-1323.

[528] Lopez-Martin, M., Carro, B., Sanchez-Esguevillas, A., Lloret, J.(2017). Network traffic classifier with convolutional and recurrent neural networks for Internet of Things. IEEE Access, 5, 18042-18050.

[529] Huang, C. W., Chiang, C. T., Li, Q. (2017, October). A study of deep learning networks on mobile traffic forecasting. In Personal, Indoor, and Mobile Radio Communications (PIMRC),2017 IEEE 28th Annual International Symposium on (pp. 1-6). IEEE.

[530] Vinayakumar, R., Soman, K. P., Poornachandran, P. (2017, September). Applying deep learning approaches for network traffic prediction. In Advances in Computing, Communications and Informatics (ICACCI),2017 International Conference on (pp. 2353-2358). IEEE.

[531] Zhang, C., Wang, X., Li, F., He, Q., Huang, M. (2018). Deep learning based network application classification for SDN. Transactions on Emerging Telecommunications Technologies, 29(5), e3302.

[532] Aceto, G., Ciuonzo, D., Montieri, A., Pescap, A. Mobile Encrypted Traffic Classification Using Deep Learning.

[533] Dargenio, R., Srikant, S., Hemberg, E., O'Reilly, U. M. (2018, May). Exploring the Use of Autoencoders for Botnets Traffic Representation. In 2018 IEEE Security and Privacy Workshops (SPW) (pp. 57-62). IEEE.

[534] Li, R., Xiao, X., Ni, S., Zheng, H., & Xia, S. (2018, June). Byte Segment Neural Network for Network Traffic Classification. In 2018 IEEE/ACM 26th International Symposium on Quality of Service (IWQoS) (pp. 1-10). IEEE.

[535] Lin, D., & Tang, B. (2018, December). Detecting unmanaged and unauthorized devices on the network with long short-term memory network. In 2018 IEEE International Conference on Big Data (Big Data) (pp. 2980-2985). IEEE.

[536] Zhou, H. (2018, August). Malware Detection with Neural Network Using Combined Features. In China Cyber Security Annual Conference (pp. 96-106). Springer, Singapore.

[537] Tzortzis, G., & Likas, A. (2007, October). Deep belief networks for spam filtering. In ictai (pp. 306-309). IEEE.

[538] Woodbridge, J., Anderson, H. S., Ahuja, A., Grant, D. (2016). Predicting domain generation algorithms with long short-term memory networks. arXiv preprint arXiv:1611.00791.

[539] Saxe, J., Berlin, K. (2017). eXpose: A character-level convolutional neural network with embeddings for detecting malicious URLs, file paths and registry keys. arXiv preprint arXiv:1702.08568.

[540] Abdi, F. D., Wenjuan, L. MALICIOUS URL DETECTION USING CONVOLUTIONAL NEURAL NETWORK.

[541] Lison, P., Mavroeidis, V. (2017). Automatic Detection of Malware-Generated Domains with Recurrent Neural Models. arXiv preprint arXiv:1709.07102.

[542] Feng, Z., Shuo, C., Xiaochuan,W. (2017, December). Classification for DGA-Based Malicious Domain Names with Deep Learning Architectures. In 2017 Second International Conference on Applied Mathematics and information technology (p. 5).

[543] Azakami, T., Uda, R. (2017, March). Deep Learning Analysis of Amodal Completion CAPTCHA with Colors and Hidden Positions. In Advanced Information Networking and Applications Workshops (WAINA), 2017 31st International Conference on (pp. 341-346). IEEE.

[544] Kopp, M., Nikl, M., Holena, M. (2017). Breaking CAPTCHAs with Convolutional Neural Networks. In Proceedings of the 17th Conference on Information Technologies-Applications and Theory.

[545] Shibahara, T., Yamanishi, K., Takata, Y., Chiba, D., Akiyama, M., Yagi, T., ... & Murata, M. (2017, May). Malicious URL sequence detection using event de-noising convolutional neural network. In Communications (ICC), 2017 IEEE International Conference on (pp. 1-7). IEEE.

[546] Jiang, J., Chen, J., Choo, K. K. R., Liu, C., Liu, K., Yu, M., & Wang, Y. (2017, October). A Deep Learning Based Online Malicious URL and DNS Detection Scheme. In International Conference on Security and Privacy in Communication Systems (pp. 438-448). Springer, Cham.

[547] Yu, B., Gray, D. L., Pan, J., De Cock, M., and Nascimento, A. C. (2017). Inline DGA detection with deep networks. In IEEE International Conference on Data Mining Workshops, pp. 683- 692.

[548] Mac, H., Tran, D., Tong, V., Nguyen, L. G., and Tran, H. A. (2017). DGA Botnet Detection Using Supervised Learning Methods. In Proceedings of the Eighth ACM International Symposium on Information and Communication Technology, pp. 211-218.

[549] Zhao, N., Liu, Y., Jiang, Y. CAPTCHA Breaking with Deep Learning.

[550] Asiaee, A., Goel, H., Ghosh, S., Yegneswaran, V., Banerjee, A. (2018, May). Time Series Deinterleaving of DNS Traffic. In 2018 IEEE Security and Privacy Workshops (SPW) (pp. 103-108). IEEE.

[551] Yu, B., Pan, J., Hu, J., Nascimento, A., De Cock, M. (2018). Character Level Based Detection of DGA Domain Names.

[552] Zannettou, S., Chatzis, S., Papadamou, K., Sirivianos, M. (2018). The Good, the Bad and the Bait: Detecting and Characterizing Clickbait on YouTube. In meeting of the 1st Deep Learning and Security Workshop, co-located with the 39th IEEE Symposium on Security and Privacy, San Francisco.

[553] Saxe, J., Harang, R., Wild, C., Sanders, H. (2018). A Deep Learning Approach to Fast, Format-Agnostic Detection of Malicious Web Content. arXiv preprint arXiv:1804.05020.

[554] Ra, V., HBa, B. G., Ma, A. K., KPa, S., Poornachandran, P. DeepAnti-PhishNet:Applying Deep Neural Networks for Phishing Email Detection.

[555] HBa, B. G., Ra, V., Ma, A. K., KPa, S. Distributed Representation using Target Classes: Bag of Tricks for Security and Privacy Analytics.

[556] El Aassal, A., Moraes, L., Baki, S., Das, A., Verma, R. Anti-Phishing Pilot at ACM IWSPA 2018.

[557] Le, H., Pham, Q., Sahoo, D., Hoi, S. C. (2018). URLNet: Learning a URL Representation with Deep Learning for Malicious URL Detection.arXiv preprint arXiv:1802.03162

[558] Selvaganapathy, S., Nivaashini, M., & Natarajan, H. (2018). Deep belief network based detection and categorization of malicious URLs. Information Security Journal: A Global Perspective, 27(3), 145-161.

[559] Bahnsen, A. C., Bohorquez, E. C., Villegas, S., Vargas, J., & Gonzlez, F. A. (2017, April). Classifying phishing URLs using recurrent neural networks. In Electronic Crime Research (eCrime), 2017 APWG Symposium on (pp. 1-8). IEEE.

[560] Bahnsen, A. C., Torroledo, I., Camacho, D., & Villegas, S. (2018). DeepPhish: Simulating Malicious AI. In 2018 APWG Symposium on Electronic Crime Research (eCrime) (pp. 1-8).

[561] Zhao, J., Wang, N., Ma, Q., & Cheng, Z. (2018, July). Classifying Malicious URLs Using Gated Recurrent Neural Networks. In International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (pp. 385-394). Springer, Cham.

[562] Kumar, A. D., & KP, S. (2018). DeepImageSpam: Deep Learning based Image Spam Detection. arXiv preprint arXiv:1810.03977.

[563] Vinayakumar, R., Soman, K. P., Poornachandran, P., and Sachin Kumar, S. (2018). Evaluating deep learning approaches to characterize and classify the DGAs at scale. Journal of Intelligent and Fuzzy Systems, vol. 34, no. 3, pp. 1265-1276.

[564] Vinayakumar, R., Soman, K. P., and Poornachandran, P. (2018). Detecting malicious domain names using deep learning approaches at scale. Journal of Intelligent and Fuzzy Systems, vol. 34, no. 3, 1355-1367

[565] Vinayakumar, R., Poornachandran, P., and Soman, K. P. (2018). Scalable Framework for Cyber Threat Situational Awareness Based on Domain Name Systems Data Analysis. In Big Data in Engineering Applications, pp. 113-142.

[566] Mohan, V. S., Vinayakumar, R., Soman, K. P., and Poornachandran, P. (2018). Spoof net: Syntactic patterns for identification of ominous online factors. In IEEE Security and Privacy Workshops, pp. 258-263.

[567] Yu, B., Pan, J., Hu, J., Nascimento, A., and De Cock, M. (2018). Character level based detection of DGA domain names. In IEEE World Congress on Computational Intelligence, pp. 41684175.

[568] Curtin, R. R., Gardner, A. B., Grzonkowski, S., Kleymenov, A., and Mosquera, A. (2018). Detecting DGA domains with recurrent neural networks and side information. arXiv preprint arXiv:1810.02023

[569] Tran, D., Mac, H., Tong, V., Tran, H. A., and Nguyen, L. G. (2018). A LSTM based framework for handling multiclass imbalance in DGA botnet detection. Neurocomputing, vol. 275, pp. 2401-2413.

[570] Chen, Y., Zhang, S., Liu, J., & Li, B. (2018, September). Towards a Deep Learning Approach for Detecting Malicious Domains. In 2018 IEEE International Conference on Smart Cloud (SmartCloud) (pp. 190-195). IEEE.

[571] Sivaguru, R., Choudhary, C., Yu, B., Tymchenko, V., Nascimento, A., & De Cock, M. (2018, December). An Evaluation of DGA Classifiers. In 2018 IEEE International Conference on Big Data (Big Data) (pp. 5058-5067). IEEE.

[572] Khehra, G., & Sofat, S. (2018, July). BotScoop: Scalable Detection of DGA Based Botnets Using DNS Traffic. In 2018 9th International Conference on Computing, Communication and Networking Technologies (ICCCNT) (pp. 1-6). IEEE.

[573] Apruzzese, G., Colajanni, M., Ferretti, L., Guido, A., & Marchetti, M. (2018, May). On the effectiveness of machine and deep learning for Cyber Security. In 2018 10th International Conference on Cyber Conflict (CyCon) (pp. 371-390). IEEE.

[574] McDermott, C. D., Majdani, F., & Petrovski, A. V. (2018, July). Botnet detection in the internet of things using deep learning approaches. In 2018 International Joint Conference on Neural Networks (IJCNN) (pp. 1-8). IEEE.

[575] Taheri, S., Salem, M., & Yuan, J. S. (2018). Leveraging Image Representation of Network Traffic Data and Transfer Learning in Botnet Detection. Big Data and Cognitive Computing, 2(4), 37.

[576] McDermott, C. D., Petrovski, A. V., & Majdani, F. (2018, June). Towards Situational Awareness of Botnet Activity in the Internet of Things. In 2018 International Conference On Cyber Situational Awareness, Data Analytics And Assessment (Cyber SA) (pp. 1-8). IEEE.

[577] Letteri, I., Della Penna, G., & De Gasperis, G. (2018, October). Botnet Detection in Software Defined Networks by Deep Learning Techniques. In International Symposium on Cyberspace Safety and Security (pp. 49-62). Springer, Cham.

[578] Shima, K., Miyamoto, D., Abe, H., Ishihara, T., Okada, K., Sekiya, Y., ... Doi, Y. Classification of URL bitstreams using Bag of Bytes.

[579] Sahingoz, O. K., Baykal, S. I., & Bulut, D. PHISHING DETECTION FROM URLS BY USING NEURAL NETWORKS.

[580] Catania, C., Garca, S., & Torres, P. An Analysis of Convolutional Neural Networks for detecting DGA.

[581] Choudhary, C., Sivaguru, R., Pereira, M., Yu, B., Nascimento, A. C., & De Cock, M. (2018). Algorithmically generated domain detection and malware family classification. In Proceedings of the Sixth International Symposium on Security in Computing and Communications (SSCC18), ser. Communications in Computer and Information Science Series (CCIS). Springer.

[582] Rajalakshmi R., Ramraj S., Ramesh Kannan R. (2019) Transfer Learning Approach for Identification of Malicious Domain Names. In: Thampi S., Madria S., Wang G., Rawat D., Alcaraz Calero J. (eds) Security in Computing and Communications. SSCC 2018. Communications in Computer and Information Science, vol 969. Springer, Singapore.

[583] Bharathi B., Bhuvana J. (2019) Domain Name Detection and Classification Using Deep Neural Networks. In: Thampi S., Madria S., Wang G., Rawat D., Alcaraz Calero J. (eds) Security in Computingand Communications. SSCC 2018. Communications in Computer and Information Science, vol 969. Springer, Singapore.

[584] Attardi G., Sartiano D. (2019) Bidirectional LSTM Models for DGA Classification. In: Thampi S., Madria S., Wang G., Rawat D., Alcaraz Calero J. (eds) Security in Computing and Communications. SSCC 2018. Communications in Computer and Information Science, vol 969. Springer, Singapore.

[585] Jyothsna P.V., Prabha G., Shahina K.K., Vazhayil A. (2019) Detecting DGA Using Deep Neural Networks (DNNs). In: Thampi S., Madria S., Wang G., Rawat D., Alcaraz Calero J. (eds) Security in Computing

and Communications. SSCC 2018. Communications in Computer and Information Science, vol 969. Springer, Singapore.

[586] Vinayakumar, R., Soman, K. P., Poornachandran, P., Mohan, V. S., and Kumar, A. D. (2019). ScaleNet: Scalable and Hybrid Framework for Cyber Threat Situational Awareness Based on DNS, URL, and Email Data Analysis. Journal of Cyber Security and Mobility, vol. 8, no. 2, pp. 189-240

[587] Spooren, J., Preuveneers, D., Desmet, L., Janssen, P., & Joosen, W. (2019). Detection of algorithmically generated domain names used by botnets: a dual arms race. In Proceedings of the 34rd ACM/SIGAPP Symposium On Applied Computing (pp. 1902-1910). Association for Computing Machinery.

[588] Li, Y., Xiong, K., Chin, T., & Hu, C. (2019). A Machine Learning Framework for Domain Generation Algorithm (DGA)-Based Malware Detection. IEEE Access.

[589] Vinayakumar, R., Soman, K. P., Poornachandran, P., & Menon, P. A deep-dive on Machine learning for Cybersecurity use cases. Machine Learning for Computer and Cyber Security: Principle, Algorithms, and Practices. CRC Press, USA.

[590] Yang, W., Zuo, W., & Cui, B. (2019). Detecting Malicious URLs via a Keyword-based Convolutional Gated-recurrent-unit Neural Network. IEEE Access.

[591] Lennan, C., Naber, B., Reher, J., Weber, L. End-To-End Spam Classification With Neural Networks.

[592] Hiransha, M., Unnithan, N. A., Vinayakumar, R., Soman, K. P. Deep Learning Based Phishing E-mail Detection.

[593] Vinayakumar, R., Soman, K. P., & Poornachandran, P. (2018). Evaluating deep learning approaches to characterize and classify malicious URLs. Journal of Intelligent & Fuzzy Systems, 34(3), 1333-1343.

[594] Vinayakumar, R., Soman, K. P., & Poornachandran, P. (2017, September). Evaluating shallow and deep networks for secure shell (ssh) traffic analysis. In 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI) (pp. 266-274). IEEE.

[595] Vinayakumar, R., Soman, K. P., & Poornachandran, P. (2017, September). Deep encrypted text categorization. In 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI) (pp. 364-370). IEEE.

[596] Vinayakumar, R., Soman, K. P., & Poornachandran, P. (2017, September). Evaluating effectiveness of shallow and deep networks to intrusion detection system. In 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI) (pp. 1282-1289). IEEE.

[597] Vinayakumar, R., Soman, K. P., & Poornachandran, P. (2017, September). Secure shell (ssh) traffic analysis with flow based features using shallow and deep networks. In 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI) (pp. 2026-2032). IEEE.

[598] Vinayakumar, R., & Soman, K. P. (2018). DeepMalNet: Evaluating shallow and deep networks for static PE malware detection. ICT Express, 4(4), 255-258.

[599] Kumar, A. D., Chebrolu, K. N. R., & KP, S. (2018). A Brief Survey on Autonomous Vehicle Possible Attacks, Exploits and Vulnerabilities. arXiv preprint arXiv:1810.04144.

[600] HB, B. G., Poornachandran, P., & KP, S. (2018). Deep-Net: Deep Neural Network for Cyber Security Use Cases. arXiv preprint arXiv:1812.03519.

[601] Vazhayil, A., Vinayakumar, R., & Soman, K. P. (2018, July). Comparative Study of the Detection of Malicious URLs Using Shallow and Deep Networks. In 2018 9th International Conference on Computing, Communication and Networking Technologies (ICCCNT) (pp. 1-6). IEEE.

[602] KP, S. (2019). RNNSecureNet: Recurrent neural networks for Cyber Security use-cases. arXiv preprint arXiv:1901.04281.

[603] Vinayakumar R, Soman KP, Prabaharan Poornachandran, Mamoun Alazab and Sabu M. Thampi. "AmritaDGA: a comprehensive data set for domain generation algorithms (DGAs)", Big Data Recommender Systems, IET. [In Press]

[604] N.B. Harikrishnan1, R. Vinayakumar, K.P. Soman, B. Annappa, and Mamoun Alazab, "Deep learning architecture for big data analytics in detecting intrusions and malicious URL", Big Data Recommender Systems, IET. []In Press]

[605] Vinayakumar R, Soman KP, Prabaharan Poornachandran, Mamoun Alazab, and Alireza Jolfaei. "DBD: Deep Learning DGA-based Botnet Detection", Deep Learning Applications for Cyber Security, Springer. [In Press]

[606] Amara Dinesh Kumar, Harish Thodupunoori, R. Vinayakumar, K. P. Soman, Prabaharan Poornachandran, Mamoun Alazab, and Sitalakshmi Venkatraman. "Enhanced Domain Generating Algorithm Detection Based on Deep Neural Networks" , Deep Learning Applications for Cyber Security, Springer. [In Press]

[607] Vinayakumar R, Soman kp, Prabaharan poornachandran, Akarsh S and Mohamed Elhoseny, Deep learning Framework for Cyber Threat Situational Awareness based on Email and URL Data Analysis, Cybersecurity and Secure Information Systems, Springer. [In Press]

[608] Vinayakumar R, Soman kp, Prabaharan poornachandran and Akarsh S, Application of Deep Learning Architectures for Cyber Security, Cybersecurity and Secure Information Systems, Springer. [In Press]

[609] Vinayakumar R, Soman kp, Prabaharan poornachandran, Akarsh S and Mohamed Elhoseny, Improved DGA Domain Detection and Categorization using Deep learning Architectures with Classical Machine learning Algorithms, Cybersecurity and Secure Information Systems, Springer. [In Press]

[610] Harikrishnan NB, Vinayakumar R, and Soman Kp, Time Split based pre-processing for Malicious URL Detection, Cybersecurity and Secure Information Systems, Cybersecurity and Secure Information Systems, Springer. [In Press]

[611] Akarsh S, Prabaharan Poornachandran, Vijay Krishna Menon, and Soman K P. "A Detailed Investigation and Analysis of Deep learningArchitectures and Visualization Techniques for MalwareFamily Identification", Cybersecurity and Secure Information Systems, Springer. [In Press]

[612] Vinayakumar R, Soman kp and Prabaharan poornachandran. "DeepDGA-MINet: Cost-Sensitive Deep Learning based Framework for Handling Multiclass Imbalanced DGA Detection", Handbook of Computer Networks and Cyber Security: Principles and Paradigms, Springer. [In Press]

[613] Vinayakumar R, Mamoun Alazab, Soman KP, Prabaharan Poornachandran, and Sitalakshmi Venkatraman. "Robust Intelligent Malware DetectionUsing Deep Learning", IEEEAccess [In Press]

[614] Vinayakumar R, Mamoun Alazab, Soman KP, Prabaharan Poornachandran, Ameer Al-Nemrat, and Sitalakshmi Venkatraman. "Deep Learning Approach for IntelligentIntrusion Detection System", IEEEAccess [In Press]

[615] Vinayakumar R, Mamoun Alazab, Alireza Jolfaei, Soman Kp and Prabaharan Poornachandran. "Ransomware Triage Using Deep Learning: Twitter as a Case Study", The 9th IEEE International Conference on Cyber Security and Communication, Springer. [In Press]

[616] Tomlinson, A., Bryans, J., Shaikh, S. A., & Kalutarage, H. K. (2018, June). Detection of Automotive CAN Cyber-Attacks by Identifying Packet Timing Anomalies in Time Windows. In 2018 48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W) (pp. 231-238). IEEE.

[617] Tian, Y., Pei, K., Jana, S., & Ray, B. (2018, May). Deeptest: Automated testing of deep-neural-network-driven autonomous cars. In Proceedings of the 40th International Conference on Software Engineering (pp. 303-314). ACM.

[618] Ferdowsi, A., Challita, U., Saad, W., & Mandayam, N. B. (2018). Robust Deep Reinforcement Learning for Security and Safety in Autonomous Vehicle Systems. arXiv preprint arXiv:1805.00983.

[619] Kang, M. J., & Kang, J. W. (2016). Intrusion detection system using deep neural network for in-vehicle network security. PloS one, 11(6), e0155781.

[620] Rahul, R. K., Anjali, T., Menon, V. K., & Soman, K. P. (2017, September). Deep learning for network flow analysis and malware classification. In International Symposium on Security in Computing and Communication (pp. 226-235). Springer, Singapore.

[621] Akarsh S, Simran K, Prabaharan Poornachandran, Vijay Krishna Menon and Soman KP. "Deep Learning Framework and Visualization for Malware Classification", IEEEXplore [In Press]

[622] Akarsh S, Sriram S, Prabaharan Poornachandran, Vijay Krishna Menon, Soman KP "Deep Learning Framework for Domain Generation Algorithms Prediction Using Long Short-term Memory", IEEEXplore [In Press]

[623] Maniath, S., Poornachandran, P., & Sujadevi, V. G. (2018, September). Survey on Prevention, Mitigation and Containment of Ransomware Attacks. In International Symposium on Security in Computing and Communication (pp. 39-52). Springer, Singapore.

[624] Faruki, P., Bharmal, A., Laxmi, V., Ganmoor, V., Gaur, M. S., Conti, M., & Rajarajan, M. (2015). Android security: a survey of issues, malware penetration, and defenses. IEEE communications surveys & tutorials, 17(2), 998-1022.

[625] Vinayakumar R, Soman KP, and Prabaharan Poornachandran. A Comparative Analysis of Deep learning Approaches for Network Intrusion Detection Systems (N-IDSs), International Journal of Digital Crime and Forensics (IJDCF), 11(3). [In Press]

[626] Buczak, L.; Guven, E. A Survey of Data Mining and Machine Learning Methods for Cyber Security. IEEE Commun. Surv. Tutor. 2016, 18, 11531176.
[627] Nguyen, T.T.T.; Armitage, G. A survey of techniques for internet traffic classification using machine learning. IEEE Commun. Surv. Tutor. 2008, 10, 5676.
[628] Garcia-Teodoro, P.; Diaz-Verdejo, J.; Maci-Fernndez, G.; Vzquez, E. Anomaly-based network intrusion detection: Techniques, systems and challenges. Comput. Secur. 2009, 28, 1828.
[629] Sperotto, A.; Schaffrath, G.; Sadre, R.; Morariu, C.; Pras, A.; Stiller, B. An overview of IP flow-based intrusion detection. IEEE Commun. Surv. Tutor. 2010, 12, 343356.
[630] Wu, S.X.; Banzhaf, W. The use of computational intelligence in intrusion detection systems: A review. Appl. Soft Comput. 2010, 10, 135.
[631] Torres, J.M.; Comesaa, C.I.; Garca-Nieto, P.J. Machine learning techniques applied to cybersecurity. Int. J. Mach. Learn. Cybern. 2019, 114.
[632] Xin, Y.; Kong, L.; Liu, Z.; Chen, Y.; Li, Y.; Zhu, H.; Gao, M.; Hou, H.; Wang, C. Machine Learning and Deep Learning Methods for Cybersecurity. IEEE Access 2018, 6, 3536535381.
[633] Apruzzese, G.; Colajanni, M.; Ferretti, L.; Guido, A.; Marchetti, M. On the effectiveness of machine and deep learning for cyber security. In Proceedings of the 2018 10th IEEE International Conference on Cyber Conflict (CyCon), Tallinn, Estonia, 29 May1 June 2018; pp. 371390.
[634] Wickramasinghe, C.S.; Marino, D.L.; Amarasinghe, K.; Manic, M. Generalization of Deep Learning for Cyber-Physical System Security: A Survey. In Proceedings of the IECON 2018-44th Annual Conference of the IEEE Industrial Electronics Society, Washington, DC, USA, 2123 October 2018; pp. 745751.
[635] Al-Garadi, M.A.; Mohamed, A.; Al-Ali, A.; Du, X.; Guizani, M. A Survey of Machine and Deep Learning Methods for Internet of Things (IoT) Security. arXiv 2018, arXiv:1807.11023.
[636] Mahdavifar, S., & Ghorbani, A. A. (2019). Application of Deep Learning to Cybersecurity: A Survey. Neurocomputing.
[637] Berman, D. S., Buczak, A. L., Chavis, J. S., & Corbett, C. L. (2019). A Survey of Deep Learning Methods for Cyber Security. Information, 10(4), 122.
[638] Arulkumaran, K., Deisenroth, M. P., Brundage, M., & Bharath, A. A. (2017). Deep reinforcement learning: A brief survey. IEEE Signal Processing Magazine, 34(6), 26-38.