Deep Learning for Cyber Security Applications: A Comprehensive Survey

This paper was downloaded from TechRxiv (<https://www.techrxiv.org>).

LICENSE

CC BY 4.0

SUBMISSION DATE / POSTED DATE

06-10-2021 / 12-10-2021

CITATION

R, vinayakumar; Alazab, Mamoun; KP, Soman; Srinivasan, Sriram; Venkatraman, Sitalakshmi; Pham, Quoc-Viet; et al. (2021): Deep Learning for Cyber Security Applications: A Comprehensive Survey. TechRxiv. Preprint. <https://doi.org/10.36227/techrxiv.16748161.v1>

DOI

[10.36227/techrxiv.16748161.v1](https://doi.org/10.36227/techrxiv.16748161.v1)

Deep Learning for Cyber Security Applications: A Comprehensive Survey

Vinayakumar Ravi, Mamoun Alazab, Soman KP, Sriram Srinivasan, Sitalakshmi Venkatraman, Quoc-Viet Pham, Simran K

Abstract—Deep Learning (DL), a novel form of machine learning (ML) is gaining much research interest due to its successful application in many classical artificial intelligence (AI) tasks as compared to classical ML algorithms (CMLAs). Recently, DL architectures are being innovatively modelled for diverse applications in the area of cyber security. The literature is now growing with DL architectures and their variations for exploring different innovative DL models and prototypes that can be tailored to suit specific cyber security applications. However, there is a gap in literature for a comprehensive survey reporting on such research studies. Many of the survey-based research have a focus on specific DL architectures and certain types of malicious attacks within a limited cyber security problem scenario of the past and lack futuristic review. This paper aims at providing a well-rounded and thorough survey of the past, present, and future DL architectures including next-generation cyber security scenarios related to intelligent automation, Internet of Things (IoT), Big Data (BD), Blockchain, cloud and edge technologies.

This paper presents a tutorial-style comprehensive review of the state-of-the-art DL architectures for diverse applications in cyber security by comparing and analysing the contributions and challenges from various recent research papers. Firstly, the uniqueness of the survey is in reporting the use of DL architectures for an extensive set of cybercrime detection approaches such as intrusion detection, malware and botnet detection, spam and phishing detection, network traffic analysis, binary analysis, insider threat detection, CAPTCHA analysis, and steganography. Secondly, the survey covers key DL architectures in cyber security application domains such as cryptography, cloud security, biometric security, IoT and edge computing. Thirdly, the need for DL based research is discussed for the next generation cyber security applications in cyber physical systems (CPS) that leverage on BD analytics, natural language processing (NLP), signal and image processing and blockchain technology for smart cities and Industry 4.0 of the future. Finally, a critical discussion on open challenges and new proposed DL architecture contributes towards future research directions.

Index Terms—Cyber Security, Machine Learning, Neural Networks, Deep Learning, Communication Networks, Cloud and

Edge Computing.

I. INTRODUCTION

WITH Internet becoming an essential resource for everyone, we are entering Industry 4.0 due to a rapid advancement in cyber physical systems (CPS) geared by technologies such as cloud computing, mobile computing, edge computing, and Internet of Things (IoT). However, the topic of cyber security in CPS is also growing in importance due to the inherent security risks and vulnerabilities as systems become increasingly heterogeneous, complex, and interconnected. The overall number of vulnerabilities has increased by 13% in 2018 [1]. It is predicted that zero-day exploits seen in the wild will grow from one per week in 2015 to one per day by 2021 [2]. While the cyber security job market is growing globally to counter this situation, there is a crisis of skilled cyber security researchers and practitioners with a potential shortage of trained professionals up to 25% [3]. There is a need for a survey that serves as a tutorial for cyber security professionals. It is important to identify the gaps in literature to aid addressing the important problem in cyber security for future ICT systems.

The term cyber security has evolved as a set of concepts and procedures to protect ICT systems and networks with the objective of preserving the confidentiality, integrity, and availability of information in the Cyberspace. Cybercrime deals with criminal activities carried out in the CPS resulting in computer hardware, networks, and software being maliciously attacked. More importantly, risks related to data integrity from unauthorized access, theft, disclosure, and intentional or accidental harm are of growing importance. 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. To counter such threats, various cyber security approaches such as intrusion detection (ID), social network analysis, malware analysis, advanced persistent threats, web application security, and applied cryptography are being adopted. However, with the huge emergence of CPS towards Industry 4.0, there is lack of an adaptive cyber security framework that can react and respond to changes in systems and physical processes proactively.

Current cyber security tools make use of huge amounts of data from network sensors, logs, and endpoint agents that can be efficiently processed using data mining (DM) techniques

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

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

Vinayakumar R was with Division of Biomedical Informatics, Cincinnati Children's Hospital Medical Center, Cincinnati, OH, United States e-mail: (Vinayakumar.Ravi@cchmc.org, vinayakumarr77@gmail.com)

Vinayakumar R, Sriram S, Soman KP and Simran K was with Center for Computational Engineering and Networking(CEN), Amrita School of Engineering, Coimbatore, Amrita Vishwa Vidyapeetham, India e-mail: (r_vinayakumar@cb.amrita.edu, sri27395ram@gmail.com).

Quoc-Viet Pham is with Research Institute of Computer, Information and Communication, Pusan National University, 2 Busandaehakro 63beon-gil, Geumjeong-gu, Busan 46241, South Korea e-mail: (vietpq@pusan.ac.kr).

Sitalakshmi Venkatraman is with Department of IT, Melbourne Polytechnic, Prahran Campus, VIC 3181, Australia e-mail: (sitavenkat@melbournepolytechnic.edu.au)

to provide timely information about malicious activities. DM techniques extract the hidden features to differentiate between normal and malicious activities and are successfully adopted resulting in legacy cyber security solutions in the market such as network-level and host-level firewalls, antivirus software, Intrusion Detection Systems (IDSs), and Intrusion Protection Systems (IPSs). However, such solutions are effective in detecting only the known malicious activities and fail to detect new types of malicious activities associated with Industry 4.0 and big data (BD). Due to the BD characteristics of voluminous data in different formats, types and modes, many challenging issues exist in such cyber security solutions due to their limitations of domain experts. 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 forms a new research direction recently.

With the evolution of new technologies in CPS, data generated by the end-user systems exhibit unknown patterns that are yet to be modelled for classification as malware or benign. Further, the resource constrained IoT devices suffer from several vulnerabilities and security breaches due to their inherent processing limitations. Hence, cyber security specialists and researchers are exploring cognitive technologies using machine learning (ML) and deep learning (DL) for cyber security so that such artificial intelligence (AI) based techniques are incorporated into systems to make judgements as close to domain experts [4]. However, it is understood that there is a potential to mislead ML/DL deployment as discussed in existing literature [5].

ML and DL have become an essential tool for various applications in the field of computer vision and speech processing. The DL architectures have obtained better performance as compared to classical ML algorithms (CMLAs) and more importantly have outperformed domain experts in several computer vision and health related applications [6]. One of the major disadvantages of CMLAs is their reliance on the feature engineering methods that are usually dictated by domain experts. Hence, security researchers are exploring DL methods to handle the dynamically evolving malicious activities. DL scales well for very large amount of data samples compared to CMLAs since it can capture the important features from complex systems including natural language processing (NLP) [5]. As the data of cyber security continuously grows day by day with the evolution of technology, the performance of DL based solutions also improves. Recently, several DL architectures are proposed, and it is important to conduct a survey of their relative suitability to address the growing cyber security issues as we enter Industry 4.0.

A. Existing Surveys on ML and DL on Cyber Security

Classical ML frameworks that were developed in the past decade for solving cyber security problems have been surveyed extensively [7], [8], [9], [10], [11], [12]. However, these do not include DL methodologies. Surveys on DL frameworks have been confined to a very narrow set of applications in cyber security. Majority of research studies focus on a specific

cyber security technique, such as intrusion detection [13], [14], spam detection, anomaly detection, and malware analysis [15]. Recent studies have focused on providing a summary review of the work related to defending CPS [16], and some have discussed various ML and DL methodologies for securing IoT technologies [17]. Certain surveys including short tutorial types have either a specific focus on the application of deep reinforcement learning (DRL) to cyber security or generic DL frameworks in the detection of various attacks included malware, spam, insider threats, network intrusions, false data injection, and malicious domain names used by botnets [18] [19], [20]. However, previous surveys have focused on only certain cyber security scenarios forming limitation in the scope. In particular, they do not discuss the use of DL for securing next-generation communication networks such as autonomous vehicle networks, cloud/edge computing, and blockchain, which are next-generation technologies proceeding towards 5G network and Industry 4.0. Though many surveys on the application of ML and DL to cyber security exist, to the best of our knowledge, a detailed and comprehensive survey on various research undertaken in DL has not yet been performed in such breadth and scope to include the complexity of next generation computing. This paper aims to fill this gap in literature.

B. Research Contribution

Considering the limitations of the state-of-the-art surveys in literature pertaining to DL architectures for next generation computing, our target in this work is to provide an extensive and comprehensive survey on DL for cyber security. The major contributions of our survey can be summarized as follows:

- 1) This paper reviews the DL architectures deployed for various cyber security applications and provides a walk-through of their evolution. Additionally, this survey summarizes, compares and contrasts the various DL architectures providing a detailed understanding of the past, present, and future DL applications in cyber security.
- 2) This survey presents the classification of a variety of studies that have reported on the application of DL to cyber security based on several attributes such as type of architecture and its application, year of study, text representation, type of dataset and performance comparison with CMLA.
- 3) An overview of various issues and major challenges of cyber-security applications involved in off-line and real-time deployment is provided. Further, the importance of shared tasks in the field of cyber security is explained.
- 4) The state-of-the-art reinforcement learning (RL) and adversarial ML applications in cyber security are examined.
- 5) The importance of DL in BD, NLP, signal, and image processing for cyber security is discussed.
- 6) The role of DL architecture-based cyber security in the field of smart cities, pervasive computing, biometric, IoT, fog and cloud computing, and autonomous vehicle is covered.

Fig. 1. Diagrammatic view of the organization of this paper.

- 7) Significance of unsupervised learning for cyber security over semi-supervised and supervised learning is explored. Moreover, importance of explainable AI, transfer learning, visualization and hybrid framework in cyber security is summarized.
- 8) Many publicly available datasets used for various cyber security studies are reviewed and suggestions for future research directions are provided.

C. Paper Organization

The structure of this survey article is organized as shown in Fig. 1. Section II presents the basics of various CMLAs and DL architectures and further discusses the major issues that exist in cyber security. Section III explores the importance of DL architectures in big data, signal and image processing, and natural language processing techniques suitable for cyber security. Section IV presents adversarial DL in cyber security, followed which Section V provides a comprehensive review on the applications of RL in cyber security. Section VI describes the state-of-the-art DL architectures for various cyber security techniques, including intrusion detection, cyber

threat situational awareness using domain generation algorithm (DGA), uniform resource allocator (URL), email and security log data analysis, network traffic analysis, Windows/Android malware analysis, side channel attacks detection, insider threat detection, function recognition, steganalysis and steganography, and social media data for cyber security. In Section VII, we provide an overview of the application of DL for various technologies in next-generation communication networks, smart city, blockchain, cryptography, cloud computing, edge computing, autonomous vehicle networks, pervasive computing, and biometric security. Section VIII discusses miscellaneous issues related to the application of DL to cyber security, including the importance of transfer learning (TL) in cyber security applications, unsupervised learning in cyber security, off-line and real-time deployment, role of explainable AI in cyber security, and causal theory with DL for cyber security. The detailed statistics of DL applications in cyber security is reported in Section IX. Finally, in Section X, we propose a hybrid cyber security framework of submodules of best theories and DL models and in Section XI, we provide conclusions along with future research directions.

II. FUNDAMENTALS OF DL AND ISSUES OF EXISTING CYBER SECURITY SOLUTIONS

A. Basics of Classical ML Algorithms and Deep Learning

In 1955, John McCarthy coined the term Artificial intelligence (AI) and defined AI as “the science and engineering of making intelligent machines”. ML is a subset of AI which was also introduced during the same decade but became more popular in 1990s with the evolution of computing technologies and the exponential increase in digital data. In ML, the mathematical and statistics concepts form the core foundation in the emergence of a variety of complex algorithms for primarily performing pattern discovery, correlations and anomalies of data. The outputs of ML algorithms are represented in terms of probabilities and confidence intervals. Due to the limitations in having experts to analyse huge amounts of data, ML algorithms are employed to successfully automate the learning process for AI based automation.

Over the past decade, ML has matured in its application to cyber security [6]. In general, algorithms in ML can be grouped into five different types namely, supervised, semi-supervised, unsupervised, reinforcement, and active learning. Supervised learning algorithms are task-driven and rely on labelling of the sample files as malware or not. Supervised ML algorithms require preprocessing and feature engineering. Such commonly used CMLAs are Naive Bayes (NB), Logistic regression (LR), Decision Tree (DT), Ada Boost (AB), Random Forest (RF), and Support vector machine (SVM). Unsupervised learning is a data driven approach that requires only the sample data and the algorithm uses implicit learning and labelling based on the distribution of the data. While the performance of unsupervised models is lower as compared to supervised models, they are preferred in real-time cyber security applications as manual labelling of sample data is a tedious task. Semi-supervised learning combines both the supervised and unsupervised learning to get benefits from both approaches. RL is an environment driven approach which works based on rewards and 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 are composed of 3 main steps: 1) raw data collection, 2) feature extraction, and 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. Neural network (NN) is capable of automatic feature extraction and classification without 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. This made the DL to achieve the best performance in long-standing AI applications related to various domains.

DL has become a focal point for both the security researchers and people from security industries. DL is now being employed in various problems existing in cyber security

Fig. 2. Classification of DL architecture.

and performs well in all use cases compared to CMLAs. DL architectures can be classified into generative and discriminative, as shown in Fig. 2. Generative category is composed of deep Boltzmann machine (DBM), deep autoencoder (DAE), deep belief network (DBN), and recurrent structures, whereas discriminative category is composed of recurrent structures and convolutional neural network (CNN). Recurrent structures and CNN most commonly use DL architectures. Both DBN and DBM are based on the restricted Boltzmann machines (RBM). The generative adversarial network (GAN) belongs to both the generative and discriminative DL architecture category. The combination of DL and artificial neural networks (ANN) is called deep neural networks (DNN). While we have provided a pictorial overview of the classification of various DL architectures in Fig. 2, a detailed description and survey reports are available in literature [13], [14], [15].

To evaluate the performance of various DL models available, several statistical measures are used. One of the most important and standard metrics adopted is the confusion matrix which provides the details of the classification results, including the individual distinguished classes arrived at. Classification correctness is measured in terms of true positive (TP) and true negative (TN), where the positive/negative samples are correctly classified by the DL model. In addition, false positive (FP) and false negative (FN) measures indicate the incorrect prediction of the positive and negative samples respectively by the DL model. Using the confusion matrix, some metrics can be estimated such as accuracy, precision, recall/true positive rate (TPR)/sensitivity, F1-measure/F1-score, false positive rate (FPR), true negative rate (TNR), and false negative rate (FNR) [19]. The values of Accuracy, Precision, Recall, F1-score, FPR, TNR and FNR range from 0 to 1 with larger values representing better performance. Since these measures are correlated, any desire to increase one measure such as TPR may result in an undesired increase of another measure such as FPR. Therefore, during the design phase, an optimal detection accuracy is usually assessed based on a discrimi-

nation threshold that reflects the dependency of TPR on FPR, which is represented by the Receiver Operating Characteristics (ROC) curve. For the purpose of benchmarking, the area under the ROC curve (AUC) is estimated. AUC values typically lie between 0.5 to 1.0, and larger AUCs represent better performance.

B. Key Deep Learning Architectures

This section summarises the salient features of four key DL architectures using ANN (DNN) due to their wide applications in cyber security that are reported in recent literature

1) *Deep belief network (DBN) and deep Boltzmann machine (DBM)*: DBN or deep networks is based on generative engineering related to the classical ANN. It contains an input layer, with at least one hidden layer and one output layer. It is to be noted that DBN with one layer is the same as feed forward network (FFN). Both the input and hidden layers should have at least one neuron, scientifically termed for a processing unit. The output layer has one unit for every class that is required for the inputs to be classified by the network. In addition, a network with more than one hidden layer may expend more opportunity for its assembly. For instance, an unsupervised learning component such as the restricted Boltzmann machine (RBM) could take in the minimized element vectors by passing an input vector through at least one of the RBM hidden layers within 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. Each layer of RBM learns features which represent the data in the previous layer with associative memory present at the top layer. Conditional distribution is followed by each hidden layer unit to generate binary form feature vectors that are propagated in reverse direction to reconstruct the training samples. This procedure is followed iteratively for all the training samples.

2) *Autoencoders (AE)*: A set of NN architectures made to learn alias representations of input data via linear or nonlinear operations are called AE which have identical input and output layer dimensions. The main purpose of AE is to achieve dimensionality reduction. In recent literature, researchers are found to 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 is 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 multi-layer perceptron (MLP), i.e., it has one input layer, one or more hidden layers, and one output layer. If AE has multiple hidden layers, then the features extracted from one layer are further processed to different features that are capable of reconstructing the data. During data reconstruction process, AE aims to minimize the error. Therefore, the outputs of intermediate layers are nothing but an encoded version of their inputs capable of reconstructing the input data under specific conditions.

In usual transformations, a particular set of features is selected utilizing the data points, which are then fed to

classification algorithms as input. However, AE follows unsupervised approach, where different features are extracted from different layers and passed on to the other DL layers such as CNN, RNN and hybrid networks that include CNN-RNN and CMLAs. AEs have 3 types of well-known variants: Sparse AE, Denoising AE, and Contractive AE. In sparse AE, there are more hidden nodes than the inputs and outputs in order to encourage sparsity. However, only a portion of the hidden units are activated at a given time. This is accounted for by penalizing the activation of additional nodes. DAE recovers the correct input from a corrupted version to increase the robustness of the model. Contractive AE achieves this by adding an analytic contractive penalty to the reconstruction error function. Overall, the DAE architecture is more robust for noise, while contractive AE can capture the local directions of variation dictated by the data.

3) *Convolutional neural network (CNN)*: CNN is a popular state-of-the-art method adopted for many computer vision applications. CNN is composed of convolutional, pooling, and fully connected layers [21]. A convolutional layer uses kernels or filters to move along various dimensions (1D/2D/3D/4D) of data to extract optimal features, together called as feature maps. These feature maps are then 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 which is nothing but a non-linear downsampling operation. The common pooling operations are maximum, minimum, average, stochastic, spatial pyramid, and deformation value from the partition. The stochastic pooling is similar to maximum pooling but it also prevents overfitting by replacing the conventional deterministic pooling operations with a stochastic procedure determined by the activation within each pooling region according to a multinomial distribution. Generally, CNN network can handle only the fixed length input representations. To handle variable length input representations, spatial pyramid pooling can be used as it can handle input images of variable scales, sizes and aspect ratios. A deformation pooling operation can handle deformation in image efficiently when compared to the max and average pooling. The novelty in the DL architecture could be explored by combining the different pooling layers to boost the performance of the CNN architecture.

Based on CNN, various benchmark architectures are proposed and evaluated on ImageNet Large Scale Visual Recognition Challenge (ILSVRC). Key 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 applied on large datasets. Since obtaining large datasets for all the classes/tasks in real-time is difficult, data augmentation is used to increase the data samples without introducing extra labeling costs. CNN architectures are well-known and can be employed for parameter initialization instead of random parameter value in newer tasks, which represents pretraining. This accelerates the learning process and improves the model generalization.

4) *Recurrent neural network (RNN)*: Recurrent structures are mainly used in sequential and temporal data modeling tasks. RNN is an advanced model of classical NNs which has a self-recurrent connection in the hidden layer that facilitates the network to remember the previous step information. It suffers from vanishing and exploding gradient issue when dealing with long time-steps during backpropagation through time (BPTT). Gradient clipping is one of the prominent strategies to avoid the exploding gradient issues. To alleviate the vanishing issue, research on RNN progressed on three significant directions: i) Hessian-free optimization for improving the optimization methods, ii) Long Short-term Memory (LSTM) or a variant of LSTM network with reduced parameters set, gated recurrent unit (GRU) for introducing complex components in recurrent hidden layer of network structure, and iii) Identity-recurrent neural network (IRNN) for weight initialization with an identity matrix.

C. Major issues in the existing Cyber security solutions

DL serves a best-fit for cyber security due to its BD characteristics, and the availability of multi-core CPUs, GPUs along with the evolution of NNs to train many hidden layers. However, the adoption is still in infancy due to lack of benchmarking in ML and DL algorithms as well as the datasets. Recently, the challenges and issues involved in employing ML/DL techniques in cyber security were discussed [24]. In Table I, we provide a summary of threat detection methods studied and benchmark datasets used. Finding a satisfactory dataset for cyber security use cases is often troublesome due to four main reasons: 1) the vast majority of publicly available datasets are outdated, 2) they are not genuine agent datasets, 3) most researchers follow different splitting methodologies to divide data into train, validate and test categories, and 4) they 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 do not have a standard approach and most enterprises avoid using ML/DL solutions for improving their cyber security applications [23].

Proposed approach to address the issue: Most recent way to enhance the performance of a system is by organizing the shared tasks as a part of a 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 training dataset among the participants. Evaluation of trained models is performed utilizing the testing dataset. Finally, the results are made publicly available with an option for publication. Shared tasks are most familiar in the field of NLP, computer vision and speech recognition. Recently, CDMC¹¹, IWSPA-AP¹², DMD 2018¹³, and AICS 2019¹⁴ are 4 shared tasks in the field of cyber security.

¹¹<http://www.csmining.org/cdmc2018/>

¹²<https://dasavisha.github.io/IWSPA-sharedtask/>

¹³<https://nlp.amrita.edu/DMD2018/>

¹⁴<http://www-personal.umich.edu/~arunesh/AICS2019/index.html>

III. IMAGE/SIGNAL PROCESSING, NATURAL LANGUAGE PROCESSING, AND BIG DATA FOR CYBER SECURITY

A. Image and Signal Processing for Cyber security

Several feature engineering mechanisms available in signal and image processing domain have been successfully employed for malware classification. The malware features are represented in the form of a signal or grayscale image as an alternative to malware binaries represented in Hexadecimal or text forms [115]. The range of each 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. Signal and image-based malware analysis is fast, and does not need disassembly, unpacking, and execution of binary code. Recently, novel feature engineering methods such as spectral flatness, mel-frequency cepstrum coefficients (MFCC), chroma features are proposed to accurately extract important features from signals and images. Current methods of signal and image-based malware detection exhibit two major problems: i) characterization of malware using signal and image-based features does not give much information about the actual behavior of the malware and ii) 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. However, zero-day or new unseen malware attacks cannot be prevented. Hence, feature engineering mechanisms are employed by DL architectures to enhance the performance of malware analysis and detection.

B. Natural Language Processing for Cyber security

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 language in the form of data. The text data in cyber security domain have various sources such as emails, transaction logs from various systems, and online social networks. Leveraging NLP techniques have direct impact in providing situational awareness from various network event logs and user activities. Several methods of text representation in numeric form such as vector space models, distributional representation, and distributed representation can be used to encode text up to word/character level. Word/character level text encoding consists of preprocessing followed by tokenization as the initial step. This involves data cleaning and transformation of unnecessary and unknown words/characters, followed by word/character level tokenization. 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 (TF-IDF) belong to non-sequential representation. N-gram, Keras embedding, Word2vec, Neural-Bag-of-words, and FastText belong to sequential representation which has the capability to extract similarities in word meaning. In cyber security, capturing the sequential information is more important as compared to the similarities in word meaning due to the fact that most data contain time and spatial information. Hence, DL approaches could be adopted for an effective malware detection.

TABLE I
DETAILS OF BENCHMARK DATASETS.

Task	Benchmark Dataset	URL
Intrusion detection	KDDCup-99	http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html
	NSL-KDD	https://www.unb.ca/cic/datasets/nsl.html
	UNSW-NB15	https://www.unsw.adfa.edu.au/unsw-canberra-cyber/cybersecurity/ADFA-NB15-Datasets/
	ADFA-LD	https://www.unsw.adfa.edu.au/unsw-canberra-cyber/cybersecurity/ADFA-NB15-Datasets/
	UNM	https://www.cs.unm.edu/~immsec/systemcalls.htm
	ISCX-IDS-2012	https://www.unb.ca/cic/datasets/ids.html
	CICIDS2017	https://www.unb.ca/cic/datasets/ids-2017.html
	Kyoto	http://www.takakura.com/Kyoto_data/
	UNIBS	http://netweb.ing.unibs.it/~ntw/tools/traces/
	CAIDA	https://www.caida.org/data/
	LBNL	https://commons.lbl.gov/display/cpp/100G+Intrusion+Detection
	CIC DoS	https://www.unb.ca/cic/datasets/dos-dataset.html
	CSE-CIC-IDS2018	https://www.unb.ca/cic/datasets/ids-2018.html
Botnet and DGA Analysis	AWID	http://icsdweb.aegean.gr/awid/index.html
	WSN-DS	https://www.hindawi.com/journals/js/2016/4731953/
	UNB Botnet	https://www.unb.ca/cic/datasets/botnet.html
URL Analysis	DGArchive	https://dgarchive.caad.fkie.fraunhofer.de/
	AmritaDGA	https://vinayakumarr.github.io/AmritaDGA/
	ISCX-URL-2016	https://www.unb.ca/cic/datasets/url-2016.html
Spam and phishing Email detection	Sophos URL	https://github.com/Maddy12/SophosMachineLearningBuildingBlocksTutorial
	CSDMC	http://csmining.org/spam-email-datasets-.html/
	Enron	https://www.cs.cmu.edu/~enron/
Malware Detection	TREC	https://trec.nist.gov/data/spam.html
	SpamAssassin	https://spamassassin.apache.org/old/publiccorpus/
	EMBER	https://github.com/endgameinc/ember
Binary Analysis	BIG 2015	https://www.kaggle.com/c/malware-classification
	Malrec	https://giantpanda.gtisc.gatech.edu/malrec/dataset/
	Microsoft Malware Prediction	https://www.kaggle.com/c/microsoft-malware-prediction
Image spam detection	ByteWeight	http://security.ece.cmu.edu/byteweight/
Android	Image spam hunter	https://www.unb.ca/cic/datasets/android-adware.html
	AAGM	https://www.unb.ca/cic/datasets/android-adware.html
	CICAndMal2017	https://www.unb.ca/cic/datasets/andmal2017.html
Traffic Analysis	Drebin	https://www.sec.cs.tu-bs.de/~danarp/drebin/
	Kharon	http://kharon.gforge.inria.fr/dataset/
	ISCXVPN2016	https://www.unb.ca/cic/datasets/vpn.html
Security in IoT	ISCXTor2016	https://www.unb.ca/cic/datasets/tor.html
	N-BaIoT	archive.ics.uci.edu/ml/datasets/detection_of_IoT_botnet_attacks_N_BaIoT
	Bot-IoT	https://iee-dataport.org/documents/bot-iot-dataset
Side channel attacks	DPA contest	http://www.dpacontest.org/index.php
Insider threat detection	ASCAD	https://www.data.gouv.fr/en/datasets/ascad/
	CERT	https://resources.sei.cmu.edu/library/asset-view.cfm?assetid=508099

C. Big data Analytics for Cyber security

Real-time analysis of BD generated in CPS is important for various cyber security applications with the aim to protect data, computer systems, networks, and IoT from malicious activity. With more and more unstructured and noisy data being generated at an unprecedented rate, it requires advanced technologies in GPU, cluster computing frameworks to process and handle very large amount of data in an efficient way [22]. Such infrastructure is the primary component in BD technologies to store, process, and analyze data using ML techniques. In general, BD technologies are divided into 2 categories namely batch processing such as Hadoop, and stream processing such as InfoSphere. The Hadoop framework consists of Hadoop Distributed File System to store large files and MapReduce programming model to work on large-scale data processing problems. Hadoop tools that adopt ML frameworks include Hive (an SQL-friendly query language), Pig (a platform and a scripting language for complex queries), Mahout and Radoop. New frameworks such as Spark 4 are designed to improve the performance of DM and ML algorithms by repeated reuse of the working dataset. Hence,

databases specifically designed for efficient storage and query of BD include NoSQL databases such as CouchDB, Cassandra, HBase, Greenplum Database, Vertica, and MongoDB. While batch processing has a dominant mature technology such as Hadoop, stream processing is still in its infancy to embrace ML/DL approaches. Complex Event Processing (CEP) is one of the models for stream processing where 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.

Autoencoder (AE) is a generative model which learns the latent representation of different feature sets in an unsupervised manner and is considered suitable method for network traffic analysis due to its significant dimensionality reduction in cyber security. This is advantageous with BD as large amounts of data require to be processed in a fraction of time without any loss of information. Singular value decomposition (SVD) and principal component analysis (PCA) are commonly used classical methods for dimensionality reduction. The main factors limiting the growth of AI and DL are the burden

TABLE II
COMPARISON OF POPULAR DL FRAMEWORKS.

DL Frameworks	Creator	License	Core Languages	Interface Support
Caffe ¹	Berkeley Center	FreeBSD	C++	Python & MATLAB
Torch ²	Ronan Collobert et al.	BSD	C, & Lua	C/C++, Lua, Python
Theano ³	University of Montreal	BSD	Python	Python
Deeplearning4j ⁴	Skymind	Apache 2.0	Java	Java, Scala, & Python
MXNet ⁵	Apache Software Foundation	Apache 2.0	C++	C++, Python, R, Scala, Perl, Julia, etc.
TensorFlow ⁶	Google Brain Team	Apache 2.0	C++, & Python	Python, C/C++, Java, Go
H2O ⁷	H2O.ai	Apache 2.0	Java	R, Python, Scala, Rest API
Chainer ⁸	Japanese venture company Preferred Networks in partnership with IBM, Intel, Microsoft, Nvidia	MIT	Python	Python
CNTK ⁹	Microsoft Research	MIT	C++	Python, C++, BrainScript
Keras ¹⁰ (higher level library for TensorFlow, CNTK, Theano)	François Chollet	MIT	Python	Python

of handling massive datasets and the heavy computational requirements associated with their processing. Traditionally large datasets were handled by clusters of computers using dedicated high-end processors to perform these tasks. Recently, the power of GPU's has been harnessed and has proven to be faster and better for large- scale data processing with a number of platforms tailored for developing ML and DL applications. Such popular platforms implementing DL architectures include TensorFlow, Theano, Torch, Caffe, and DeepDist. A summary of popular DL frameworks is given in Table II. Most of the well-known platforms use C++ that support high-level ML computations at the backend and a Python frontend. In addition, Keras has become the commonly used high-level API which supports TensorFlow, CNTK and Theano.

IV. 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 as they are susceptible to attacks by adversaries. Hackers exploit the vulnerabilities of ML frameworks using adversarial samples much similar to how they are exploiting firewall vulnerabilities. It is important to consider the shortcomings of a ML framework by conducting stress tests in adversarial environments to identify ML vulnerabilities before deployment, and such a study is known as adversarial ML. As many ML frameworks behave as black boxes in critical systems, adversarial ML face constraints. It is exceptionally troublesome for experts and clients to comprehend the model results in such environments as there is no explanation about the decision made by the framework. In the absence of any assurance on the robustness of ML frameworks, their use in secure critical system becomes remote. In Table III, we summarise the state-of-the-art use of adversarial DL and datasets found in cyber security literature.

A. Domain Generation Algorithms

Domain generation algorithms (DGAs) have been adopted by several malware families utilize to establish command and control (C&C) connections resulting in distributed denial of

TABLE III
SUMMARY OF ADVERSARIAL DL AND DATASETS USED.

Reference	Dataset
[179]	Alexa & Private
[180]	Genome, Contagio, VirusShare, Drebin
[181]	Drebin
[182]	BIG 2015
[184]	Private
[183]	Malicious Behavior Windows Audit Logs
[185]	VirusShare, Citadel, APT1
[186]	Spambase
[187]	Wild dataset & BIG 2015
[188]	GTSRB
[189]	Enron Spam Dataset
[190]	Mozilla Common Voice Dataset
[191]	NSL-KDD
[192]	AICS'2019 challenge dataset
[193]	Car-Hacking Dataset from HCRL
[194]	ADFA-LD
[195]	CIDDS-001
[196]	KDDCup-99
[197]	Ember
[198]	AmritaDGA
[199]	CycleGAN & StarGAN

service (DDoS) attacks. Recently, a DL based architecture called generative adversarial network (GAN) was trained to generate adversarial domain names to intentionally bypass a DL based detector [179]. Another study proposed a black box adversarial attack approach for evading DGA without any prior knowledge of the classifier model. [198]. The approach was successfully tested using DMD-2018 dataset to degrade the F1-score from 0.977 to 0.495.

B. Malware Detection Adversary

Malware Re-composition Variation (MRV), a novel method generated adversarial malware examples based on semantic analysis of existing malwares to evade malware detector and to enhance the robustness of the detector by using three defense techniques [180]. Another study employed existing adversarial example generation algorithms to generate malwares without losing its intrusive functionality [181]. A malware detection framework based on transferred GAN (tGAN) was proposed to detect zero-data attacks and demonstrated good learning stability and accuracy of 96.39% [182]. Further, in order

to boost the robustness of a DL model against adversarial attack, a malware detection model was proposed to nullify random features of data [183]. Enhancements to adversarial attacks based on six principles were attempted [192], including visualization methods and API call based adversarial to evade classifiers such as RNN, DNN, and ML classifiers. Such studies aim to train the GAN generator for enhancing DL models as compared to CMLA.

C. Intrusion Detection Systems (IDSs) Adversary

A deep AE based adaptive IDS to test its robustness against adversarial examples resulted in improving accuracy by 15% when compared to PCA based detection system [191]. Some studies have proposed black box attacks based on GAN against IDS by generating adversarial network traffic to evade detection with high success rate including in smart vehicle networks [193]. Host-based ID System (HIDS) based on GAN was proposed by producing adversarial anomalies to train an ANN model [194]. Similarly, three black box adversarial attacks against DNN based Network ID Systems (NIDS) were studied, and two data augmentation modules generated adversarial data for addressing the data insufficiency challenge in NIDS [196].

D. Other Adversarial based Attacks and Defense Techniques in Cyber security

Ensemble defenses are capable of improving the robustness of detection models against adversarial attacks. Such mechanisms against adversarial attack were studied and an ensemble technique was proposed with a weight decay defense to improve its efficacy [184], and by adding few more hidden layers in a outlier based defense technique, the robustness was improved further in linear classifiers against poisoning attacks [186]. However, they fail to detect attacks that are less aggressive such as label flipping. Another study employed generalized distillation learning approach to train the DL based detection model using privileged features [187]. Attacks against computer vision based modules of autonomous vehicles in real-world applications were studied by using adversarial examples to misclassify advertisements and innocuous signs with a success rate of 95% [188]. Another study proposed a novel adversarial attack against DL classifier in a black box environment by modifying the data with small text perturbation to produce adversarial examples that degraded the accuracy by 60% with IMDB and Enron datasets [189]. A DL based automatic speech recognition system was attacked by generating an adversarial example from an input audio waveform by adding a small perturbation with 100% success [190]. A DL based model generated flow-based network traffic by using GANs with three preprocessing methods and evaluated the quality of the generated traffic data [195]. Another study for the detection of fake images generated by GAN was proposed to compute the co-occurrence matrices on the RGB channels of the images by using those matrices to train the CNN model to detect fake GAN images [199]. The proposed framework used two GAN datasets such as cycleGAN and StarGAN to test the model resulting in 99% accuracy for both datasets.

TABLE IV
RL STUDIES AND DATASETS IN CYBER SECURITY LITERATURE.

Reference	Dataset
[153]	KDDCup-99
[154], [155], [156], [157], [160]	Private
[158]	PhishingCorpus, SpamAssassin, PhishTank
[159]	Ember

V. REINFORCEMENT LEARNING FOR CYBER SECURITY

Reinforcement learning (RL) is a revolutionary AI technique that is inspired by the psychological concept of Pavlov's classical conditioning technique and the mathematical concept of Markov decision process [200]. There are three vital elements that construct an RL 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 (DRL) utilizes NNs or approximation methods for finding the optimal value or solution for the problem. DRL based solutions for the applications of cyber security are still in the beginning stage. This methodology can be suitable for cyber security applications such as botnet detection and malware detection. A summary of studies conducted on RL algorithms and the datasets used in cyber security is provided in Table IV.

A. Reinforcement Learning based Intrusion Detection

Recent literature surveys have reported several studies exploring various RL frameworks applied to both public and private datasets. An adaptive IDS based on ML was proposed using multi-class SVM with principal component analysis (PCA) for feature reduction and RL approach for prediction [153]. The model had achieved promising results by adopting training with benchmark datasets available publicly such as the KDDCup-99 dataset. As we embark into Industry 4.0, more DDoS over the network and IoT exploits happen in CPS. To address this, distributed IDS started adopting RL to enhance security. A distributed IDS was proposed using RL sensor agents to differentiate normal and abnormal network states and the decision agents to learn the semantics of the actions agents of sensors [154]. A more autonomous, scalable and secure decentralized DDoS detection and response system for CPS was proposed by using multi RL agent router throttling [155]. Another study detected flooding based DoS and DDoS attacks using RL agents to analyse the data flow information between hosts [157].

B. Other Various Applications in cyber security

A recent study explored the RL based framework to gain insights about the loopholes of the malware detector so as

to pre-empt the malicious attacks [156]. A self-adapting NN and RL based online phishing email detection framework was proposed to detect zero-day phishing attacks with 98.63% accuracy and a very low FPR of 1.81% [158]. A new deep RL (DRL) based malware execution control framework was proposed to emulate a malware execution and to stop after a fixed number of system calls, thereby improving TPR massively by 61.5% while keeping a very low FPR of 1% as compared to other baseline classifiers [160].

VI. DEEP LEARNING APPLICATIONS IN CYBER SECURITY

The application of DL frameworks in cyber security is maturing and research in this direction is of growing interest. However, quality research output in this domain is hampered by lack of labeled data and benchmark datasets that can suit highly complex DL models in the dynamic environment of Industry 4.0. We classify DL methods applied to cyber security falling under main categories and a hierarchy of sub-categories as shown in Fig. 3. In this section, we summarize the key achievements and limitations in literature related to DL-based threat detection methods applied in cyber security.

A. DL in Intrusion Detection

In today's interconnected world of CPS, everyday activities have become autonomous and more vulnerable to attacks that are targeted against critical infrastructures. For instance, autonomous cars can be controlled by devices and smart phones that are prone to cyber-attacks. In Table V, we provide a summary of research studies conducted on DL architectures with different datasets adopted for ID. Since the benchmark dataset plays a major role in the DL training and testing processes, we provide short review of each of the 12 different benchmark datasets and conduct a comparative study on the surveys conducted with DL architectures versus CMLAs.

1) *KDDCup-99 Dataset*: A study used LSTM to detect network intrusion and outperformed other approaches with an accuracy of 93.82% by evaluating its performance with different feature sets in KDDCup-99 challenge [201]. CNN based hybrid architectures for IDS such as CNN-LSTM, CNN-RNN, and CNN-GRU had performed better than any other hybrid model [202]. In another study, a DNN based IDS achieved a maximum accuracy of 99.9% as compared to other traditional ML models.[222].

2) *NSL-KDD Dataset*: An RNN model for both binary and multi-class classification was studied showing that a model with 80 hidden neurons and a learning rate of 0.1 had achieved the best result in binary classification while in multi-class classification, the model with 80 hidden neurons and a learning rate of 0.5 had achieved the best results [203]. Further, an STL framework based on AEs for ID was proposed to learn efficiently from the features and reduces the dimensionality of the features to aid the SVM based detector [208]. A CNN based character level IDS preprocessed the data by considering the network traffic as sequences of character and performed better than traditional ML classifiers with an accuracy of 85.07% [210].

In another study, the effectiveness of CNN, LSTM, and AE models were studied for anomaly based ID system CNN and reported to perform better than AEs and other traditional ML classifiers [211]. In another study, a stacked sparse AE based IDS framework accelerated the detection process to classify normal and malicious traffic using high dimensional sparse features [221].

3) *Private Dataset*: A low speed port scan detection system based on CNN model was proposed to filter the normal packets and group the remaining suspicious packets using its source and destination IP. The CNN model had extracted the interval and sequential features from the input to detect port scan with a precision of 97.4%.[215].

4) *UNSW-NB15 Dataset*: A novel encoding approach using CNN model was proposed for network anomaly detection to enhance its performance as compared to a random forest based detector [212]. The proposed encoding approach gave consistently better results when compared to gray-scale encoding.

5) *Kyoto Dataset*: A hybrid GRU and SVM based IDS was proposed using SVM instead of softmax in the final output layer for detecting intrusion based on network traffic data [205]. The model achieved an accuracy of 81.54 while the accuracy of the traditional softmax approach was only 63.07.

6) *ISCX-IDS-2012 Dataset*: A IDS based on CNN and random forest algorithm were proposed to extract payload features from raw network traffic. The statistical features extracted from the network traffic and payload features from the CNN were used to train the random forest classifier and achieved an accuracy of 99.13% and FAR of 1.18%. The model results were better when compared to other SVM, NN, CNN and Random forest models [218]. The proposed model achieved 99.97% accuracy and a very impressive FAR of 0.02%.

7) *CICIDS2017 Dataset*: In a port scan anomaly detection experiment, a DL model achieved 97.80% accuracy while SVM achieved only 69.79% accuracy [207].

8) *AWID Dataset*: A DL based solution for WiFi NIDS using SAEs and DNN to classify the traffic into 4 classes such as normal, impersonation attack, flooding attack, and injection attack had achieved an accuracy of 98.4%, 98.3%, 73.1% and 99.9%, for the 4 different classes respectively [209].

9) *HTTP DATASET CSIC 2010 Dataset*: A character level CNN based web application firewall was trained using unicode encoded raw http requests and achieved an accuracy of 98.8% with average processing time of 2.35ms [213].

10) *CIDDS-001 Dataset*: The effectiveness of LSTM for flow-based NIDS was studied with different combination of hyperparameters and the experimental analysis compared the performance of the LSTM models with other traditional ML approaches [220]. In another study for anomaly-based IDS, ML and DL models used various techniques to fix the imbalanced dataset and trained DNN, VAE, random forest, voting, and stacking ML models.

11) *CTU-13 Dataset*: Applying a two-level DL based adaptive anomaly detection approach was experimented for 5G networks using flow-based features from the network traffic data and by training a DBN or a SAE model in the first level, and LSTM in the second level [223].

Fig. 3. A hierarchical classification of DL Methods in Cyber Security.

TABLE V
SUMMARY OF STUDIES ON DL ARCHITECTURES AND DATASETS ADOPTED FOR INTRUSION DETECTION.

Reference	Architecture	Dataset	Compared CML
[201]	RS	KDDCup-99	Yes
[211]	DNN	NSL-KDD	No
[202]	CNN, CNN-RS	KDDCup-99	No
[210]	CNN	NSL-KDD	Yes
[208]	Autoencoder	NSL-KDD	Yes
[203]	RS	NSL-KDD	Yes
[222]	DNN	KDDCup-99	Yes
[227]	AutoEncoder	KDDCup-99	No
[204]	CNN & RS	DARPA1998 & ISCX-IDS-2012	Yes
[205]	RS & SVM	Kyoto	No
[206]	CNN	Balabit, & TWOS	No
[207]	DNN	CICIDS2017	Yes
[227]	Autoencoder	NSL-KDD	No
[212]	CNN	UNSW-NB15	No
[213]	CNN	HTTP DATASET CSIC 2010	Yes
[214]	RS	ISCX-IDS-2012 & AWID	Yes
[215]	CNN	Private	No
[216]	DBN	KDDCup-99, NSL-KDD, UNSW-NB15, CICIDS 2017	Yes
[217]	Autoencoder	NSL-KDD & UNSW-NB15	Yes
[218]	CNN	ISCX-IDS-2012	Yes
[219]	DNN	Real-time traffic, KDDCup-99, UNSW-NB15	Yes
[220]	RS & DNN	CIDDS-001	Yes
[221]	Sparse Autoencoder	NSL-KDD	Yes
[223]	DBN, SAE, RS	CTU-13	No
[225]	Autoencoder & DNN	KDDCup-99, & UNSW-NB15	Yes
[226]	CNN & RS	CICIDS2017	No
[264]	DNN, RS, CNN	KDDCup 99 & Kyoto	Yes

12) *Mixed Dataset: DL studies compared with ML:* A novel CNN-LSTM based ID system was proposed with the CNN model to learn the low level spatial features and the LSTM model to learn the high level temporal features from the network traffic data [204]. LSTM architectures for cyber attack detection in fog-of-things environment were proposed to be distributed and scalable to detect cyber-attacks [214]. Another distributed DBN with an ensemble SVM-based detection framework in a large scale network was proposed. The distributed DBN was used for non-linear dimensionality reduction and an apache spark based ensemble SVM model was used as a detection classifier. The model is trained on

4 different datasets and compared with other related models [216]. In another study, anomaly based IDS using DAE and DNN was specifically designed for industrial Internet of things. The DAE model produced optimal parameters by learning the normal behaviour of network which were used to effectively tune the parameters of DNN based classifier [217]. A scalable DNN based framework for routing attack detection was proposed where the attack dataset was extracted from the Cooja IoT simulator [219]. A stack AE based NIDS was proposed with learning of important features from a large quantity of unlabeled data using UNSW-NB15 and KDDCup-99 datasets achieved an accuracy of 89.134% and 99.996%

respectively [225]. A DNN based architecture for host and network level ID and various DL architectures such as LSTM, GRU, RNN, and IRNN were evaluated using NSL-KDD dataset [264].

DL studies without comparing with ML: A two-dimensional (2D) CNN based authentication system using mouse behaviour was proposed and trained on 2 publicly available datasets called Balabit and TWOS and it achieved an average AUC of 0.96 outperforming 1D-CNN and SVM models [206]. The effectiveness of several DL models was studied for DDoS attack detection where MLP, CNN, LSTM, CNN-LSTM models were trained using CICIDS2017 datasets. The proposed CNN-LSTM model outperformed other models by achieving 97.16% accuracy [226]. In another study, a DL based adaptive and scalable misuse IDS was trained using NSL-KDD and KDDCUP 99 datasets and compared with a static IDS scheme [227].

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

With the growing developments in CPS, we have identified the limitations of traditional antimalware systems resulting in the requirement of a more efficient and intelligently adaptive IDS. This could be possible through the analysis of cyber threat situational awareness data related to domain name system (DNS), email, URL, and social media. Timely collection of data from such sources and the application of DL architectures would enable an effective malware detection in real-time. In this section, we provide an overview of key DL techniques that could be adopted successfully.

Domain Name System (DNS): DNS is one of the main Internet protocols for accessing web pages through a browser rendered via Internet. DNS servers have two broad categories: Recursive servers and Non-recursive/Iterative servers. Non-recursive DNS servers work as the Start of Authority (SOA), replying to the queries which are inside their government/local domain only, without considering other DNS servers regardless of whether or not they can cater to the query. 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 passing on 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, and 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 of any possible cyber threat. One of the methods to bypass DNS blacklisting includes an adversary embedding a malware with fixed domain name and IP address by using fluxing to constantly change the IP address and domain name. The most commonly used method for domain fluxing is achieved using a domain generation algorithm (DGA).

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

Fig. 4. Active domain name discovery process employed by recent malwares.

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 that are embedded in the malware binary and triggered after the installation of the malware. The second type is script-based DGAs that are embedded in the Javascript and triggered when the user opens a malicious website. The flow diagram of domain flux attack is illustrated in Fig. 4, where an infected system attempts to access many domains in an attempt to contact the C&C server, which is a growing threat in CPS. 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 to launch the attack. 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 [117]. A compromised device in a network called a bot and a bot master uses these bots to conduct various illegal activities such as DDoS [117], phishing, identity theft, malware distribution, etc. The C&C server is used by the bot master to issue commands to the botnets based on which the botnets perform their assigned tasks and send back the results. Based on the C&C 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 an IRC botnet uses centralized architecture and P2P uses distributed architecture. A hybrid between centralized and distributed architectures for botnet detection is often more complex as compared to that of centralized and distributed architectures.

Spam and Phishing Email Detection: Spam and phishing email attacks have a commercial purpose to obtain passwords, credit card numbers, bank account details and other sensitive information or even to infect the email recipient's computer with malicious code. These can be broadly classified under two

techniques: i) deceptive phishing that uses social engineering schemes and ii) malware-based phishing to trick the victim for capturing personal and financial data [116].

Uniform resource locator (URL): URL is an universal address of documents and other resources on the World Wide Web and plays the important role of locating/accessing documents and other web resources via a web browser on-line. With Internet growth, the URL has become one of the most commonly used tool to host malicious contents by an adversary and a protection is warranted in recent years. The traditional methods of blacklisting and filtering fake URLs are simple but not scalable, even 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 to prove that the user is a human. The common types of CAPTCHA are based on text, image, video, audio, and puzzle image for users to recognize and is an active security technique to protect DDoS attacks from bots. It is a commonly adopted first line of defense.

Next, we provide a survey on DL architectures for analyzing situational awareness using DGA, URL, Email and security log data analysis and is summarized in Table VI.

1) *Email:* DL architectures such as CNN, RNN, LSTM, and MLP were studied for phishing email detection using word embedding and neural bag of n-grams with semantic and syntactic email similarity measures. The experiment results showed that word embedding with LSTM approach performed well with accuracy of 99.1 and 97.1 in two different anti-phishing shared tasks from IWSPA-AP 20181 corpus [244]. In another study, a DL based anti-phishing system was proposed where a distributed representation method had differentiated phishing and legitimate emails with F1 score of 99% achieved for the best case [245].

2) *DGA:* **Recurrent Structures:** An LSTM based real-time DGA domain detection model was evaluated on open datasets and achieved 90% detection rate with AUC of 0.9993 for binary classification [235]. A DL based approach to address the problem of time series deinterleaving was proposed to generate synthetic dataset and evaluated various inference strategies for the problem using Augmented Hidden Markov Model (AHMM) and LSTM. The experimental results showed that the LSTM method outperformed the model based on AHMM [241]. The effectiveness of various DL based approaches were studied for scalable DGA detection using DNS logs. DL models such as RNN and LSTM had extracted useful features from DNS log data and outperformed state-of-the-art ML models [248]. A DL based DGA detection engine was proposed where a 1D-CNN model extracted important features from a large dataset containing URLs from 51 DGA malware families, achieving 97% accuracy and 0.7% FAR [250].

Convolutional Neural Network: A CNN based botnet detection framework for IoT and wearable devices where the network traffic data was converted into image format was trained using CTU-13 Dataset. It achieved the best case accu-

racy of 99.98% while SVM and logistic regression achieved accuracies of 83.15% and 78.56% respectively [253].

Mixed Approaches: A DL based scalable DGA detection framework which works at ISP level used event data from the DNS to detect DGA and provided situational awareness [22]. Another novel CNN-LSTM based approach for DGA and malicious URL detection incorporated NLP technique and the performance was compared with traditional ML classifiers using bi-gram feature representation and character-level CNN model. The results showed that the proposed system achieved an accuracy of 99% for malicious URL detection and 98.3% for DGA detection [249].

Hybrid Recurrent Structures: A DGA botnet detection system using RNN and CNN models was proposed for detecting the malicious domains using DNS traffic data to provide information about the infected host and C&C domain. The results showed that CNN-LSTM achieved the highest accuracy of 98.7% [251]. Various well-known character based short-text classification were modeled for DGA detection and classification using various datasets to show the similarity of performance [263].

Deep Neural Network: A DL based method for DGA analysis was experimented on various cyber security applications to generalize the DL architecture [15]. A multi-layered neural network based botnet detection system in SDN environment was trained with feature selection and filtering using HogZilla dataset to make it realistic for SDN scenario, achieving 96% accuracy [254].

Bi-directional Recurrent Structures: A bi-directional LSTM based botnet detection engine using word embedding for the conversion of network traffic packets into tokenized integer values was proposed, achieving good accuracy for mirai, udp, and dns attacks [252]. Various DL architectures were proposed for DGA detection and classification, with datasets termed as AmritaDeepDGA, and AmritaDGA for DGA analysis that have been made publicly available for further research [262].

3) *URL:* **Recurrent Structures:** A study on LSTM based scalable model for detecting phishing URL by selecting useful features automatically from a large dataset of URLs compared random forest using features extracted from lexical and statistical analysis. The experimental analysis showed that the LSTM and RF approach achieved an accuracy of 98.7% and 93.5% respectively [247].

Convolutional Neural Network: A CNN architecture was proposed and trained using features from a short character strings such as URLs, registry keys, file paths, named mutexes, and named pipes [236]. A CNN based system used event denoising from proxy logs to extract the sequences to detect URL redirection [239].

Mixed Approach: A recently developed convolutional GRU model used feature extraction from 212 URLs for training to detect malicious URLs with an accuracy of 99.6% [257].

Hybrid Recurrent Structures: A performance evaluation of various DL architectures such as LSTM, RNN, GRU, and CNN for malicious URL detection was conducted by employing Keras embedding and compared with various CMLAs such as RF, DT, MT, AB, and NB [258].

TABLE VI
A SUMMARY OF STUDIES ON DL ARCHITECTURES DATASETS USED FOR SITUATIONAL AWARENESS ANALYSIS

References	Architecture	Task	Data set	Text representation	Compared CML
[235]	RS	Domain name	Publically available sources (PASs)	Keras embedding	Yes
[236]	CNN	URL	Private	One-hot	Yes
[237]	AlexNet, VGGNet, GoogleNet	CAPTCHA	Private	-	No
[238]	DNN & CNN	CAPTCHA	Publically available sources	-	No
[239]	CNN	URL	Publically available sources	One-hot	Yes
[240]	CNN	CAPTCHA	Publically available sources	-	Yes
[241]	RS	Domain name	Private	One-hot	No
[250]	CNN, RS	Domain name	Publically available sources	Keras embedding	No
[258]	RS, CNN, CNN-RS	URL	Publically available sources	Keras embedding	Yes
[242]	VAE	YouTube video address	Private	Sent2vec embedding	Yes
[243]	DNN	Web page	Private	bag-of-words	Yes
[244]	CNN & RS	Email	IWSPA-2018	Word2vec & Neural Bag-of-ngrams	No
[245]	FastText	Email	IWSPA-2018	FastText	No
[246]	DBN	URL	Publically available sources	Manual feature engineering	Yes
[247]	RS	URL	Publically available sources	One-hot	Yes
[248], [249], [251]	RS, CNN, CNN-RS, DNN,	Domain name	Publically available sources	Keras embedding	Yes
[22]	CNN, RS	Domain name	PASs & real-time data set	Keras embedding	Yes
[15]	DNN	Domain name & URL	Publically available sources	n-gram & feature engineering	Yes
[252]	BLSTM	Network flow	Private	Keras embedding	No
[253]	CNN	Network flow	CTU-13 Dataset real-time data set	Traffic to image representation	Yes
[254]	DNN	Network flow	HogZilla dataset, CTU-13 dataset, ISCX-IDS-2012 dataset	-	No
[255]	DNN	URL	Private	Bag-of-words	No
[256]	DNN	URL	Publically available sources	Manual feature engineering	No
[257]	CNN & RS	URL	Private	Keras embedding	Yes
[262]	CNN, RS, CNN-RS	Domain name	AmritaDGA	Keras embedding	No
[263]	Endgame, NYU, MIT, CMU, Invincea, DNN, CNN-RS	Domain name	PASs & real-time dataset	Keras embedding	Yes

Deep Neural Network: A DL based malicious URL detection framework extracted features from static HTML files and used spatial information to yield 97.5% accuracy and low FPR [243]. A DNN framework was trained using real-life datasets of URLs and achieved a detection accuracy of 94.18% [255]. In another study, ANN and DNN based phishing URL detection system was proposed and trained using 73575 URLs with experimental analysis showing 92% accuracy for ANN and 96% accuracy for DNN outperforming ML classifiers [256].

Autoencoder and DBN: A VAE based method was proposed for clickbait problem in Youtube videos [242]. A DL based malicious URL detection system employed greedy multi-layered DBN to extract useful features automatically and was trained using 27,700 URLs to achieve very low FPR [246].

4) **CAPTCHA:** A text-based CAPTCHA technique with amodal completion was compared with a DL based CAPTCHA solver with DL based solver taking more time to solve due to difficulty in emulating the amodal completion [237]. A CNN based CAPTCHA solving technique was proposed and was capable of breaking 11 CAPTCHA schemes with more than 50% accuracy and performed better than MLP, SLP and CNN based solvers [238]. Further, a deep CNN based CAPTCHA breaker was proposed to solve letter-based CAPTCHAs and worked well for single-letter classification

TABLE VII
KEY DL MODELS NETWORK THREAT DETECTION.

Ref.	Architecture	Dataset	Compared CML
[228]	Autoencoder	Private	No
[229]	CNN	Private	No
[230]	CNN, & RS	TIM Big Data Challenge 2015	Yes
[231]	RS	GEANT	No
[232]	CNN, & RS	Moore	Yes
[233]	DNN	Private	Yes
[234]	RS	Private	No
[259]	RS	DARPA 1999, AMP, MAWI, NIMS	Yes

[240].

C. DL in Network Traffic Analysis

Differentiating each stream of information in a network is an important issue with the ever-growing BD. Patterns based on port (e.g. HTTP port 80 and SSL port 443), signature (e.g. strings or hex based payload) and factual highlight (e.g., transmission interim time, parcel time, and rehashing activity) are standard approaches used for network traffic analysis. This section provides a short review of DL architectures for ID using network traffic analysis and is summarized in Table VII.

DL studies compared with ML: A DL based multitask architecture for forecasting mobile Internet traffic was proposed [230]. A CNN-RNN architecture extracted geographical and temporal traffic features, while a DL based SDN architecture was adopted to classify network applications [232]. The hybrid of softmax regression layer and SAE was trained using data from SDN controller achieving high accuracy as compared to SVM classifier. Another study compared different DL techniques to classify mobile network encrypted traffic [233]. Recurrent structures such as RNN and LSTM were evaluated using three different datasets collected from real-time human activities for identifying SSH traffic and Non-SSH traffic [259]. The DL architectures outperformed other classical classifiers such as RF, AB, DT, KNN, NB, and SVM using 4 different types of publicly available network traffic datasets. However, feature engineering could be avoided by passing the entire payload and input to the DL architectures.

DL studies without comparing with ML: A DL framework for network traffic analysis was proposed for applications such as protocol classification, unknown protocol identification, and anomalous protocol detection using real-time data [228]. Further, a DL based model combining RNN and CNN for classifying IoT network traffic was proposed to outperform ML models [229]. Various RNN architectures such as LSTM, GRU, and IRNN were employed for network traffic prediction using real-time data from GEANT backbone networks with LSTM outperforming other RNN architectures [231]. Further, byte segment neural network (BSNN) and RNN based encoders were adopted to classify network traffic [234].

D. DL in Windows Malware Analysis

With the growing prevalence of Windows based applications, they become platforms for cybercrime, espionage and other illegal activities by various types of malware including viruses, trojans, worms, backdoors, rootkits, spyware, ransomware and panic software. To overcome this, DL architectures are being recently adopted for Windows malware analysis. In ransomware, the information on a casualty's PC is bolted, and normally an encrypted financial request is made before the recovered information is decoded and gets back to the casualty [269]. Table VIII summarises on the application of DL in windows malware analysis.

1) *Deep Neural Network (DNN)*: A multi-task DL framework for classifying binary malware was proposed with 4.5 million files for training and 2 million files for testing the DL model [126]. Furthermore, the error rate was reduced significantly using dropout for both deep and shallow neural frameworks, and the number of epochs to train the model were also reduced by using rectified linear activation function. A DL model was trained using BD and was evaluated using a complex dataset resulting in an accuracy of 97% and ROC of 0.99 [138]. Further, a DNN architecture for malware classification was compared to other shallow models such as LR, NB, KNN, DT, AB, and SVM on EMBER benchmark data set [133]. However, the main limitation of this method is that the proposed DNN architecture relies on feature engineering.

2) *Convolutional Neural Network (CNN)*: A convolutional FFN employed hierarchical feature extraction mechanism to detect and classify malwares using metadata of PE files [129]. Malware programs represented as an image were fed into CNN for classification and training of the DL algorithm was performed with different kernel and data size resulting in a high AUC of 0.9973 for malware detection [131]. A DL approach was proposed for classifying malware using two benchmark datasets called Maling and Microsoft malware. Grayscale image features were fed into a CNN and accuracies of 98.52% and 99.97% were achieved with Maling and Microsoft datasets respectively [135]. Further, GoogleNet and ResNet models were analysed for malware detection using a Microsoft dataset [136].

3) *Recurrent Structures (RS)*: A hybrid model of Echo state networks and RNN was trained using unsupervised data and the projection stage employed Half-Frame models and Max Pooling resulting in 98.3% TPR and 0.1% FPR [122]. Further, LSTM and GRU models used semi-supervised learning with attention mechanism and temporal maximum pooling to detect ransomware [130].

4) *Autoencoder (AE) and Deep belief network (DBN)*: Malware signature generation and classification were performed using DBN and deep stack of DAEs were utilized for compact representation of the malware behavior and the model achieved an accuracy of 98.6% [123]. A DL framework employed SAE models with unsupervised feature learning of 4) Windows API calls and fine tuning with the help of supervised parameters to detect malwares efficiently [124]. In another study, malwares were represented as opcode sequences and fed to DBNs for malware classification [127].

5) *Mixed DL architectures*: A two-stage DNN model was proposed for malware detection based on process behavior to check if a terminal is infected or not [125]. It used LSTM to construct the features from API call sequences that represented process behavior and an RNN for extraction of features as images that were fed into a CNN for classification. In another study, a hybrid NN with two convolution layers for extracting hierarchical features having both full sequential modeling and convolution of n-gram features was proposed for classification of malware and outperformed many ML methods such as SVMs and hidden markov models [128]. MalNet, an automatic feature learning model using CNN and LSTM was proposed for detecting malware from raw data of 40000 samples that were converted to grayscale images and achieved 99.88% accuracy for malware classification [132]. Heterogeneous DL architecture was recently proposed consisting of AE, multilayer boltzmann and layers of associate memory for windows API call extraction, pretraining and finetuning to detect malware [137].

6) *Hybrid Recurrent Structures (HRS)*: A DL framework with CNN and BLSTM network utilized data driven approach to identify complex features for classifying nine different types of malware [134]. A neural sequential classification malware model was proposed using API call features to achieve FPR lower than other neural classification models [139].

TABLE VIII
A SUMMARY OF STUDIES ON DL APPLICATIONS FOR MALWARE ANALYSIS.

Reference	Architecture	Dataset	Compared CML
[122]	RS	Private	Yes
[123]	DBN	Private	No
[124]	Autoencoder	Private	Yes
[125]	CNN, RS	Private	
[126]	DNN	Private	No
[127]	DBN	Private	Yes
[128]	CNN, RS	VirusShare, Maltrieve, Private	Yes
[129]	CNN	VirusShare, Maltrieve, Private	Yes
[130]	RS	Private	No
[131]	CNN	VirusShare	No
[135]	CNN	BIG 2015	Yes
[132]	CNN, RS	BIG 2015	Yes
[133]	DNN	Ember	Yes
[134]	CNN, CNN-RS	BIG 2015	No
[136]	CNN	BIG 2015 & privately collected samples	No
[137]	DBN, Autoencoder, Stacked Autoencoder	Private	Yes
[138]	DNN	MalwareTrainingSets	Yes
[139]	LSTM, CNN-RS	Private	No
[135]	CNN	Malimg	Yes

E. DL in Android Malware Detection

Android operating system (OS), an open source with several important financial and personal applications is prone to malicious attacks. Hackers make use of malware to steal the private sensitive data or delete/ alter the existing data with the aim to gain financial benefits. Android applications are hosted in various third-party stores which allow the user to inadvertently repackage Android applications along with malicious code. Android OS automatically assigns a unique Linux user ID during the installation phase to know that each app runs its own instance of virtual machine. This facilitates the creation of a sandbox which isolates the apps from each other. It provides authorization mechanism using Android permissions. Android features are collected via either rooted or unrooted devices that are passed as an input to ML models to learn the characteristics for distinguishing between the benign and malicious apps. However, malicious Apps in disguise from authorized market store that hosts the apps such as Google play and the permission systems could trick the mobile user to receive permissions for installation. Naive users may follow the blind approach in granting permissions during the installation procedure of apps and the impact is less known to the end user. Hence, it is important for Android based permission systems to undergo risk assessment.

Attacks on smart device OS such as Android OS will continue to grow as the technology evolves since signature based and heuristics-based methods are completely failing in zero-day malware detection. Detection techniques of the growing Android malware are increasingly being explored. Self-learning systems composed of DM, ML and DL algorithms could provide new sensing capabilities for Android malware detection which could be enhanced to work to 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 much similar to personal computer environments, namely static analysis

and dynamic analysis [270]. While static analysis collects a set of features from apps by unpacking or disassembling them without the runtime execution, dynamic analysis examines the runtime execution behavior of apps such as system calls, network connections, memory utilization, power consumption, and user interactions. 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, lightweight, and less time-consuming in nature. Hybrid analysis approach is increasingly being used by antivirus providers for the smartphones as it provides higher detection rates. A summary of review on DL applications in Android malware analysis is reported in Table IX.

1) *Dynamic Analysis*: A DL framework for Android malware detection was trained using dynamic analysis of system call graphs and outperformed other traditional detectors when tested with real-world malware samples [164]. A new CNN based android malware detection approach used API call execution paths as graphs and achieved an accuracy of 98.86% [178].

2) *Static and Dynamic Analysis*: A DBN based malware detection framework for Android employed static and dynamic analysis to obtain more than 200 features and outperformed other models such as NB, DT, SVM, MLP, and LR with an accuracy of 96% [161]. A new hybrid DNN Android malware classifier employed a hierarchical multiple kernel learning on combined feature set that improved its detection accuracy to 94.7% [162].

Overall, static analysis resulted in a higher accuracy when compared to dynamic analysis which is more required for online detection of malware. In a comparative study between RNN and LSTM models for detection of Android malware, the LSTM performed better than RNN, and achieved detection accuracy of 93.9% and 97.5% using dynamic and static analysis respectively [174]. A CNN based Android malware identification framework using API call sequence and protection levels as features was proposed [177].

TABLE IX
A SUMMARY OF STUDIES ON DL APPLICATIONS IN ANDROID MALWARE DETECTION.

Reference	Analysis	Features	Architecture	Dataset	Compared CML
[161]	Dynamic, Static analysis	Sensitive API calls	DBN	Publically available sources	Yes
[162]	Static, Dynamic Analysis	10 static, dynamic feature sets	Autoencoder	Publically available sources	No
[163]	Static analysis	API call blocks	DBN	Publically available sources	Yes
[164]	Dynamic analysis	System calls	SAE	Publically available sources	Yes
[165]	Static analysis	More than 1,000 features	DNN	Drebin, collected from PASs	Yes
[166]	Opcode Sequence	Learn to detect sequences of opcode that indicates malware	CNN	Publically available sources	No
[167]	Static analysis	API call sequences	CNN	Publically available sources	Yes
[168]	Transfer classes, dex into RGB color images	Extract features from transferred images	CNN	Collected by the researchers	No
[169]	Static analysis	Byte sequences	CNN	Publically available sources	Yes
[170]	Static analysis	Dangerous API calls, risky permissions	DBN	Drebin, apps collected from PASs	No
[171]	Static analysis	Permissions requested, permissions filtered intents restricted API calls, hardware features, suspicious API calls, code related features	CNN	Publically available sources	Yes
[172]	Static analysis	API sequence calls	CNN	Drebin, apps collected from PASs	No
[173]	Static analysis	Semantic structure of Android bytecode	CNN-RS	Publically available sources	No
[174]	Dynamic, Static analysis	Permissions events generated by Monkey Tool	RS	Publically available sources	Yes
[175]	Static code analysis	Token semantic features from smali	DNN	Private	Yes
[176]	Static analysis	Opcode sequences	CNN	Drebin, collected from PASs	Yes
[177]	Static, Dynamic analysis	API sequences to image	CNN	Publically available sources	No
[178]	Dynamic Analysis	API calls graph	CNN	AMD, AndroZoo, Drebin Malware Collection, ISCX Android Botnet	Yes

3) *Image Processing*: Static image analysis were adopted recently by researchers for Android malware detection. A CNN based model converted the bytecode of classes.dex into images for an effective detection of Android malware [168]. We present a survey of literature on the application of DL algorithms that have adopted image processing techniques for static analysis of Android malware detection.

DL models compared with ML: A DBN based malware detector for Android systems used blocks of API calls as features instead of simple API calls and outperformed CMLA [163]. A DNN based android malware detection system used a genetic algorithm to modify the parameters and configurations of the DL model and compared with SVM based models with an accuracy of 91%. [165]. In another study, a CNN model was compared with LSTM model for the android malware detection based on API call sequences and showed that while DL models achieved better accuracy than n-gram based detection models, CNN model surprisingly performed better than LSTM model [167].

New models of user pattern sensing were proposed for detecting Android malware. A CNN based generalized Android malware detection system evaluated user attention map and achieved better detection accuracy than conventional methods [169]. A hybrid large-scale android malware detection system based on CNN and DAE was proposed and was tested with 13000 malicious and 10000 benign Apps and showed considerable reduction in the training period by 83% when compared to simple CNN model [171]. A DNN based android malware detector used token and semantic features of smali files and was trained using smali files from 50 apks to outperform CMLAs with an AUC of 85.98% and 70% in both WPDP and CPDP mode respectively [175]. A CNN based android

malware detection framework used opcode sequences from decompiled APK files for training and achieved a detection accuracy of 99% with very low false positives [176].

DL models without comparing with ML: A CNN based android malware detection framework trained using features from static analysis of raw opcode sequences was computationally more efficient when compared to n-gram based detection models [170]. In another study, a DBN-based android malware detection system extracted features from API calls and permissions to build the detection model. While DREBIN attained accuracy of 90% with 545000 features, the DBN model achieved the same accuracy with just 237 features [166].

A DL based automatic malware detection framework for android systems was proposed for deployment on servers, mobile and IoT devices by exploring raw sequences of API calls from Apps and achieved high F1 score (96%-99%) [172]. Another multi-detection layer based on MLP and LSTM for Android malware detection was trained using xml files in the first layer and bytecode semantics in the second layer and achieved an accuracy of 97.74% [173].

F. Deep Learning in Side Channel Attacks Detection

Kocher introduced Side-Channel attacks in 1996 that were employed by hackers to break into cryptographic devices [121]. Side channel information hardware such as timing information, electromagnetic radiation, power consumption, and statistics of encryption devices could be utilized by hackers to launch a side channel attack. These attacks are very fast and can be implemented easily, posing a great threat to security. Any device from small embedded devices such as RFID to

TABLE X
KEY DL MODELS FOR SIDE CHANNEL ATTACK DETECTION.

Reference	Method	Dataset	Compared CML
[140]	Self-Normalizing Neural Networks, & CNN	ASCAD Database	No
[141]	DNN & its variants	ASCAD Database	No
[142]	CNN	DPA Contest v4	Yes
[143]	CNN	DPA Contest v2 & DPA Contest v4	No
[144]	CNN	MNIST	No
[145]	DNN	Private	No

laptops are vulnerable to side channel attacks. We review studies on DL frameworks for an effective detection of side channel attacks and provide a summary in Table X.

The application of DL for the analysis of side-channel attacks with several hyperparameters, and comparison with benchmark models showed that VGG-16 model outperformed many baseline models [196]. Further, an overview of DL for side channel attacks detection was provided for CNN based models [197]. Further, two DL approaches to enhance the effectiveness of side-channel attacks were proposed. The first approach decreased the training and attack traces to retrieve the key by using new spread layer in NNs. The second approach efficiently corrected the model predictions based on confusion matrix [198]. Several ML and DL models for side-channel attacks were studied to find that CNN performed better when the noise level was low and the number of features were high. RF and XGBoost performed better than CNN with low computational cost in other scenarios. [199]. Further, a comparative study of CNN versus CMLAs on four side channel attacks showed that CNN model achieved better accuracy of 99.3%, while with the DPA contest v2 dataset, SVM outperformed CNN [200]. A novel CNN based model for detecting side-channel attack was trained using the measured power traces achieving high accuracy with MNIST datasets [201]. A DL based side-channel attack was used to retrieve the secret key of AES cryptographic circuit. The relationship between EM noise and power noise was modeled using DNN by analysing the captured EM emission and power dissipation and the secret key was retrieved by analysing a set of 32,500 number of plain texts [202].

G. Deep Learning for Function Recognition

Function recognition is a process of recognizing functions in a binary code that is useful in malware detection. We review the application of DL architectures for function recognition.

A DL approach to recognize functions in binary files of various applications such as language modeling and speech recognition was proposed and compared with MLA showing promising results [50]. EKLAVYA, a RNN based system was introduced to address the problem of function type signature recovery by learning using idioms that match the given domain knowledge and their calling conventions [51]. Gemini, a DNN based approach to generate embedding for binary function was able to identify vulnerable firmware images significantly [52]. MobileFindr, a dynamic strategy for mapping function similarities was developed identify the fine-grained function similari-

ties successfully in mobile Apps [53]. Diff, a DNN augmented solution applied three semantic features namely, inter-module, inter-function and intra-function features to solve cross-version BCSD problem [54]. A DL method employed word embedding along with graph embedding for extracting features from two binary files to find the similarities between them and performed better than ML approaches [55].

Self-attentive function embeddings (SAFE) was introduced for embedding of functions based on self-attentive NN to enhance speed without using CFG [56]. SySeVR, a DL architecture detected several software vulnerabilities by recognising functions based on syntax, semantics, and vectors [57]. INNEREYE-BB, a similarity comparison tool based on NN and word embedding solved the problem of cross-architecture code containment using LSTM [58]. A DL approach was applied to visualize images for binary codes and classify them to solve the problem of binary code similarities [59]. Further, a generic method of decompilation to recover the structure of source code from binary machine code was implemented using RNN architecture [60].

H. Deep Learning for Steganalysis and Steganography

In network communications, steganography technique, an art of sending messages while hiding the existence of the communication have been used to send secret messages such as security keys that are concealed inside ordinary information, making the secret message invisible. Steganalysis, the process of detecting the presence of communication that contains the concealed messages, has been a significant area of research in cyber security recently to recognize covert attacks in public network. Steganography and steganalysis techniques can be applied on different kinds of data such as texts, images and videos [61]. Experiments were performed utilizing ANN to show the capability of ML based steganography [62].

Deep Steganography was proposed using DNN to work as a pair to not only hide but also uncover the concealed messages and experiments were performed using Imagenet database [63]. A two-stage process for hiding information was proposed by applying a DL architecture for information hiding and its performance was studied using multiple steganographic algorithms [64]. Further, the capability of DNN for data hiding was compared with the classical data hiding techniques showing that DNN models were more efficient and robust [65]. In another study, texture synthesizing, a well-known method in computer vision was used for image concealing for achieving steganography and watermarking [66]. In another study, instead of using the classical method of using mathematical functions and features for texture synthesizing, GANs were employed to enhance the robustness of the CNN model in image steganography [67].

Unsupervised GANs were introduced to avoid expert knowledge and complex artificial rules required for steganography and payload capacity by generating the steganographic image from the secret message without the cover image [68]. Further, GANs based method was proposed for hiding the binary data inside an image successfully [69]. Further, an experimental analysis based on DL based steganalysis was studied for to

TABLE XI
KEY DL APPLICATIONS FOR INSIDER THREAT DETECTION.

Reference	Method	Dataset	Compared CML
[146]	DNN & RS	CERT Insider Threat	Yes
[147]	CNN & RS	CERT Insider Threat	No
[148]	RS	Los Alamos National Laboratory (LANL) cyber security	Yes
[149], [151]	RS	CERT Insider Threat	No
[150]	DBN	CERT Insider Threat	No
[152]	CNN	Balabit Mouse Dynamics Challenge	Yes

defeat LSB-based steganography [70]. In another study, deep convolutional generative adversarial networks (DCGANs) was employed to avoid embedding information in steganography [71]. Using DCGAN, a secure steganography method was proposed to automatically generate container for images that enhanced the security against steganalysis method as compared to container derived from original images [72]. A deep residual architecture was introduced as an improvement to previous CNN based models for steganography due to its requirement of good heuristics to identify the value for its various parameters [73]. Further, deep residual multi-scale convolutional network was proposed for steganalysis which outperformed the existing methods based on CNN and other classical steganography methods. In another study, a TL approach was employed for image steganalysis with a detailed experimental analysis using a deep residual NN [74]. Text steganography method using LSTM encoder and de-coder models was proposed to generate Chinese quatrains [75]. To hide information in VoIP streams, Quantization index modulation (QIM) was employed with RNN based linguistic steganography and CNN based text steganalysis for semantic analysis [76]. Further, a DNN based method for steganography in speech signals was proposed [77].

I. DL in Insider Threat Detection

There have been several research studies for the detection of external malware attacking a system. However, only recently researchers have started to consider the possibility of security threats within the system. To address this so-called insider threat, detection systems based on various ML architectures have been explored. The aim of such systems is to identify hostile activities from the behavior of data inside the system. As the network threat evolves with IoT and Industry 4.0, identification of internal threat has become more difficult. In IDS, classical insider threat detection systems have been functioning on the acquired knowledge of past attacks which has been deemed to be inefficient. We present a review of key research studies that have employed DL for insider threat detection is summarised in Table XI.

An online unsupervised DL framework consisting of DNN and LSTM models developed anomaly scores from individual user behavior in real-time using system logs to efficiently identify the insider threat and outperformed the existing anomaly detection baselines such as Isolation Forest, PCA, and SVMs [146]. Another study presented a novel insider threat detection

system based on LSTM for user behaviour feature extraction and CNN for classification and achieved an AUC of 0.9449 [147]. In another study, a flexible unsupervised technique for the detection of anomalous activities using BLSTM was trained on computer security log data and performed significantly better than standard PCA and isolation forest based detection models achieving an AUC of 0.98.[148]. Further, an insider threat detection framework based on LSTM-RNN and PCA evaluated the behaviour abnormality and outperformed SVM, PCA, and Isolation Forest [149]. A novel insider threat detector based on adaptive optimization DBN with multiple hidden layers extracted the behaviour patterns by analysing user logs and achieved detection accuracy of 97.872% [150]. Further, an LSTM based insider threat system used system log to train the model to differentiate anomalous behaviour from normal user behaviour [151]. In another study, a CNN based user authentication technique used the dynamic behaviour of the mouse to authenticate users by checking every 7 seconds and achieved FPR of only 2.94%.[152].

J. Social Media Data for Cyber Security

Social media platforms such as Google Talk, Orkut, Facebook, Twitter and WhatsApp have gained popularity at an exponential rate. With several advantages of social media for real-time communication between two remote locations, there are also challenges to be addressed such as cyber stalking, cyber bullying, hacking, anti-social elements spreading their propaganda and ideologies, and even the spread of fake news. Due to the escalating fake identity profiling on social media platforms, there lies a huge challenge in monitoring relevant text streams in social media and predicting the likelihood of DDoS attacks [118]. A novel application of NLP models to detect DDoS attacks using only social media as a source outperformed the previous state-of-the-art techniques such as FFN and a partially labelled LDA models [119]. Another rare research studied the clusters of Twitter users tweeting about ransomware, virus and other malware since 2010 and used the information for automatic classification of new attacks [120]. A study proposed DNN architectures and employed cyber threat situational awareness techniques for encrypted text classification from Twitter tweets to detect and classify the ransomware events [265]. Further, recurrent structures were employed to convert the data in encrypted form to numeric representation, and using Keras embedding the model was evaluated on both character and word level text representation [260].

VII. DEEP LEARNING FOR SECURING NEXT-GENERATION COMMUNICATION NETWORKS

A. DL in IoT Applications of Smart cities

Developments in IoT and CPS have recently prompted governments of different countries around the world to promote smart city projects as the future of Industry 4.0. Many innovative IoT applications for realizing smart cities such as power grid, water supply, road traffic, and intelligent community services are rapidly growing. IoT networks of various CPS connect various devices with limited storage

capacity and processing power to collaborate, associate, and exchange information in a peer-to-peer manner. The principle objective of IoT is to make secure, reliable, and fully automated interconnected smart environments such as buildings, smart homes, smart vehicles, smart grids, smart cities, smart healthcare, smart agriculture, and so on and this generates BD. As these heterogeneous IoT devices inherently possess low security features, they pose a great threat to smart cities. Different types of attacks including DDoS and botnet could be launched when attackers gain access to these IoT [34]. Compromised IoT devices could then be used to not only invade corporate networks, but also to endanger lives in a smart city environment.

Apart from developing security policies in network and host level systems with secure firewalls, heuristics based approaches using ML and DL can be employed to secure the IoT environment from malicious activities. Recent studies applied DL approaches successfully for IoT botnet detection [35], [36]. Further, an empirically evaluated network-based DL approach was proposed for detecting attacks launched from compromised IoT devices [35]. Anomaly detection was performed by training a DAE with behavioral snapshots of the IoT traffic for each device behavior patterns to gain proficiency. When an AE fails to recreate a snapshot, the IoT device was detected to be compromised. Further, Bot-IoT, a testbed environment, was developed to create a dataset with a combination of various types of attacks with legitimate and simulated IoT traffic that would be useful for future research [36].

B. DL in Blockchain Technology for Cyber Security

State of art DL algorithms have been employed in various applications in wide range of areas such as NLP, speech recognition and computer vision achieving promising results. However, DL is still in its infancy with its applications to cyber security domain due to the heavy importance given to have a very low FPR for practical viability. Further, the DL framework requires 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, a robust and fair decentralized platform for secure collaboration of deep training was proposed utilizing four different aspects in particular, throughput, ciphertext size, training time and training accuracy [39]. If the distributed computer nodes are compromised, it will expose the algorithm and data to various cyber security threats. Alternating Direction Method of Multipliers (ADMM), a solution for the distributed optimization problem was used for the detection of attacks [40]. Another study performed a detailed analysis on potential attack vectors for the generalized distributed optimization problems addressed using ADMM [41].

Blockchain technology can be defined as a realistic solution for cybercrime as it enables decentralized and secure public ledger on multiple computers. The reason behind the popularity of block technology is that its inherent resilience to identity

theft, data breach, and criminal attacks. Recently, a comprehensive survey on using blockchain technology was conducted for various security related services [37]. Furthermore, the ability of blockchain to resolve cyber security challenges was explored and various blockchain based methodologies were compared for providing security services by training a DL model with big datasets on different data servers. The distributed solution offered by the blockchain technology to maintain tamper-resistant system and its ability to provide various solutions to tackle security problems were discussed [38]. Blockchain technology can utilize its decentralized and coordinated platform as the computing power for managing its BD. It will likewise make the DL decisions or outcomes to be more trustworthy, transparent, and explainable. Additionally, it provides secure data sharing environment. In a recent study, a decentralized blockchain-based architecture for DL applications was proposed [42]. Collaborative ID networks have been employed by researchers to enhance the performance of IDS. The main significant issues with such IDS is the availability of data and trust management which impact on the effectiveness of the IDS model. To avoid this, another research study showed the application of blockchain technology in IDS [43].

C. DL for Cryptography

Cryptography is a technique used for secure communication by which only the sender and the receiver can view the contents of the message. In Internet transmission, cryptography is extensively used to convert data into unreadable format (encrypted) that is not understandable by anyone except for the authorized individuals after data is decrypted to the original form. Modern cryptography is used in cyber security to ensure 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).

DL algorithms have been applied to cryptography to achieve a more secure data encryption that cannot be easily broken [78]. Even though DNNs have high computation cost, their application to smart cities is of immense interest among researchers worldwide. With the help of Quantum computing in future, this computation cost will be very minimal and breaking the encryption would become impossible. In recent studies, NNs were trained for encryption and decryption using adversarial methods and their applications to cryptanalysis were considered for a variety of operations such as steganography, pseudorandom-number generation and integrity checks [79] [81]. Another study discussed the theoretical aspects of applying NNs to encrypted data [80]. Various experimental studies showed how an artificial agent can learn a secure encryption method and the use of RNN and CNN based methods for S-box and cipher design as well as their classification [82] [83] [84] [85] [86]. Further, cryptonets, a set of NNs with DL was proposed to allow the user to send the encrypted data to the cloud and the models were trained to predict risks without decrypting the data with various experiments to enhance their performance [87] [88].

D. DL for Cloud Security

Internet is being reformed by cloud computing since it offers a convenient, on-request network accessibility to huge amount of shared, configurable computer system resources available in three main cloud services models, namely Software as a Service (SaaS), Platform as a Service (PaaS), and Infrastructure as a Service (IaaS) [89]. Cloud computing is popular among enterprises as it not only saves cost but also the time and effort involved in active administration and monitoring of the computer system resources and data from security threats. The assumption is that the cloud resources are in a secure environment with the required firewall and IDS to prevent any access by unauthorized users. However, cloud infiltrations have been on the rise and malware attackers are becoming more capable of bypassing the firewall easily. With the recent use of ML and DL techniques in many domains, their application to issues related to trust, privacy, and security in cloud computing were studied [90]. Google and other cloud platform providers have given top priority for security and employ ML and AI techniques to not only block malicious activities but also show warning when there is any suspicious behaviour patterns in their cloud environment.

With cloud computing seeing the era of BD stored in cloud and cloud servers, various DL architectures could be trained with BD to enhance their performance. Cloud providers are introducing new service models such as ML as-a-service (MLaaS) and DL as-a-service (DLaaS) possible. DLaaS facilitates users to design, develop and train DL applications faster and available via commercial services such as Microsoft Azure ML, Google Cloud ML Platform, Amazon ML, and IBM Watson Analytics [91]. Virtual machines being an important component of cloud infrastructure in IaaS, some experimental studies have explored heuristics and the application of DL such as LSTM for VMs network traffic anomalous behavior detection [92], [93]. Further, RNN based ID and other DL approaches such as CNN were proposed for cyberattack detection in mobile cloud computing environment in the context of BD [94].

E. DL based Cyber Security in Edge Computing

Edge and fog computing are new emerging concepts introduced to complement cloud computing to bring computing closer to the data location for enhancing response time by optimizing IoT devices and Web applications. Fog computing is the standard adopted for edge computing where the information exchanged between the end users is through the fog layers which process the input information by changing structure, size and validity of the data which can be very large. These layers have fog nodes which are from various providers which leads to more issues of trust and cyber security that need much attention due to their impact on smart city, e-Health, smart homes, mobile applications, etc. Recent studies have explored new privacy and security problems in fog computing, analysis of BD for fog security using DL architectures [107], [100].

Similar to fog computing, the concept of multi-access edge computing (MEC), has been introduced by the European Telecommunications Standards Institute (ETSI) Mobile Edge

Computing Industry Specification Group (MEC ISG) in late 2014, the ETSI defines MEC as a new technology that provide the cloud-computing capabilities and functionalities within radio access networks and in close proximity to end users [101], [102]. MEC differs from cloud computing as it possesses advantageous features such as on-premise data, low latency, location awareness, and network contextualization but suffers from limited computing power, caching, and storage capabilities, and security and privacy of personal user data [89], [102]. Since the typical deployment of MEC lies between the cloud and end users, it is vulnerable to severe malicious attacks. Hence, the security of MEC physical layer was studied and evaluated in terms of energy consumption over several baseline schemes, including secure partial offloading, secure full offloading, local computing, and partial offloading without eavesdropping [103].

Security threats and challenges that can affect edge paradigms were analyzed and reviewed [104], [105]. Further, as MEC is becoming integrated with forthcoming technologies such as 5G 6G, non-orthogonal multiple access (NOMA), cloud radio access network (C-RAN), and unmanned aerial vehicle (UAV) communications security and privacy. challenges of MEC systems are much exacerbated. Reviews on MEC threat landscapes, security vulnerabilities, and potential solutions were conducted that opened up future research directions in the application of DL architectures for secured edge computing [89], [106].

F. DL and Cyber Security in Autonomous Vehicle Technology

One of the most important of applications of AI for smart cities of the future is autonomous vehicle technology. Due to the recent innovation by automobile companies such as Tesla, Waymo and other startups companies, autonomous driving cars on streets are becoming the foreseeable future. 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 controls these autonomous cars. In recent research studies, cybercrime in CPS of automotive Controller Area Network (CAN) and the possible attacks, vulnerabilities and exploitations of autonomous vehicles were identified and statistical methods to detect the anomalies in the CAN traffic and data broadcasts were tested [261], [266]. Any error in the built-in DNNs of a typical modern-day autonomous vehicle system would result in potentially fatal outcome. DeepTest, a tool for automated testing of DNN-driven autonomous cars was proposed to systematically explore various parts of the logic of DNN for building robust systems that could be trained based on DNNs adversarial DRL algorithm in order to make the system more robust [267]. Further, another study discussed the importance of testing the safety of autonomous driving suites within in-vehicular networks and implemented an intrusion detection system (IDS) using deep neural networks [268].

G. DL and Cyber Security in Pervasive computing

Pervasive computing, also called ubiquitous computing, is another new technology developing with CPS advancement for

embedded systems to be available anywhere at any time. While it is trying to increase embedding in IoT with computational capability in everyday activities to interconnect, communicate and perform more efficiently, pervasive security becomes a vital concern. Comprehensive studies on the security of this new technology and the current stage of open problems in pervasive security were reported [108][111]. Experiments using falsification and singleton invariant were conducted to identify if the light in refrigerator was off when the door was closed to ascertain methods of monitoring, evidence gathering and reconciliation for security in pervasive computing are a better way for security [109][110]. experiments were. Another study proposed a method to combine decentralized trust and reputation management, network-level observations and Semantic Web languages declarative policies to address the challenges faced by pervasive security [112]. MicroDeep, a CNN over a distributed sensor network was proposed in a DL framework to predict the data coming from multiple sensors of IoT devices and demonstrated that MicroDeep performed better than simple CNN [113]. Further, DL frameworks were embedded in IoT devices, and DAEs were utilized to detect suspicious network traffic of compromised IoT devices to identify IoT botnet attacks [35], [114].

H. DL for Biometric Security

Biometrics security deals with identifying a person by their unique physiological characteristics and behavioral characteristics. Physiological characteristics include face, fingerprints, palmprints, iris etc., whereas behaviors characteristics involve voice, signature, gait, keystroke, etc. By using such biometric measurements for personal identification, it becomes extremely difficult for an intruder to break into any system. Hence, biometric security could be enforced when access to confidential data is required. Since DL algorithms can learn hierarchical features, they have become highly popular in biometric related fields such as speech, natural language processing, and computer vision. A recent study reviewed DL models for biometric security [95].

VIII. ML AND DL FOR MISCELLANEOUS CYBER SECURITY ISSUES

A. Transfer Learning in Cyber Security Applications

Transfer learning (TL) is a method of making use of an existing model for a particular task for 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 a new output layer.

The key advantages of TL are saving of time and enhanced performance as the model need not be created again. As illustrated in Fig. 5, TL could be employed on a DL model trained on network log dataset to detect malicious activities 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. A new clustering-enhanced hierarchical TL technique which finds the relation between known and new

Fig. 5. Transfer Learning.

attack was proposed and the TL technique was evaluated using combination of different traditional ML classifiers such as DT, random forest, KNN, SVM, and NB with various existing TL approaches [224].

B. Unsupervised ML leads to build better Cyber Security system for an organization

Unsupervised learning is a ML approach for 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. In such circumstances, supervised learning and CMLAs are not dependable. To address this disadvantage, unsupervised ML is employed and a classical example of unsupervised learning is ANN. A novel approach for IDS using unsupervised learning in the field of cyber security was proposed to demonstrate its advantage when the data is almost always unlabeled [25]. A neural language model using unsupervised learning was proposed for signature extraction which is the key part in forensic log analysis and it outperformed other signature extraction techniques [26]. In another study, an enterprise-grade framework used a divide and conquer strategy by combining the analytics of behavior and modeling of time series with an AUC of 0.943 [27]. Further, a real-time collective anomaly detection architecture based on NN learning was developed and tested using time series version of KDDCup-99 dataset [28].

In ML and DL, tensors which are multi-dimensional arrays that contain numerical values are employed to 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. A joint probabilistic tensor factorization method to derive the latent tensor subspace was developed to extract common behaviors in network traffic that vary in time across multi-views to detect inconsistency [29]. When the tensor is complex with high order dimension, the decomposition techniques used for finding dense blocks were not satisfactory

with respect to accuracy, speed, and flexibility. To address this, M-ZOOM was developed providing promising results in terms of scalability, accuracy, flexibility, and effectiveness with an AUC score of 0.98 [30]. TensorDet, was developed to enhance real-time computational efficiency of tensor decomposition by exploiting the factorization structures with novel methodologies such as sequential tensor truncation and two-phase anomaly detection [31].

C. Cyber Security Applications in Off-line and Real-time Deployment

The application of DL architectures in cyber security for both off-line and real-time deployment requires important factors to be considered during designing the underlying models. An understanding of DL architectures, interpreting 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 established. Generally, the simple models (linear models) are easier to interpret than the complex models (non-linear models). Interpretability is the ability to understand what the predictions of features are, such as features of texts or images, while non-interpretability refers to hidden layer features, vectors spaces produced by say text representations and word embedding. Heat map is one of the most commonly used approaches to understand the classification decision. The pixel of a 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 misclassifies 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. A method to identify the optimal number of feature maps was proposed and it worked well for extremely small receptive fields [32]. A visualization approach was proposed that facilitated for intermediate feature visualization [33].

Unavailability of well-known labeled benchmark datasets: Due to privacy and security reasons, the labeled datasets are not publicly available for research purpose. Labeling data samples by using manual approach is one of the daunting tasks. Most commonly used solution to label data sample is based on vendor provided blacklist and whitelist. Basically, there are three 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 three 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 preferred method in the domain of cyber security. The main important factors to be considered during dataset collection are 1) different qualities of measurement, 2) different subjects, 3) evolution of technology over time, 4) different ways of labeling examples, 5) different level of concentration, 6) different environments, 7) different protocols, and 8) time of the day.

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 learn the different types of new patterns used by an adversary simply following pertaining method.

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.

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.

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.

D. Role of Explainable Artificial Intelligence in Cyber Security

Today, AI-based systems can perform specific tasks quicker and better than human intelligence, which motivate humans to rely on these systems for making complex decisions in real-time. However, while human based decisions are justified through explanations, the fundamental principles of AI based reasoning in each step are usually not explainable in terms of what the model did, how the choice was made, etc. Due to this limitation, AI based decisions carry low trust and confidence among decision makers. Trust is one of the major acceptance factors when safety comes into picture. To overcome this limitation, trust on decisions made by AI based models could be improved over time by Explainable AI (XAI) [48]. Explainable AI is a possible and desirable concept where humans can access the decision-making procedure of the AI as it gives reasons and explanations for everything that is happening by the algorithms including the reason behind the outcome. The biggest challenge here is to create AI based systems which humans can trust. A recent study undertook a detailed survey on explainable AI [49].

DL is a state-of-the-art technique where models are composed of multiple layers that has similarities with the human brain composition. While real-time deployment of DL in cyber security is still growing, XAI was applied using an adversarial approach for the incorrect classification in an ID system and the reason behind the incorrect classification was explained visually with graphical displays [44]. In another study, a survey to improve the generalization capability of DL based CPS using regularization techniques was conducted [16]. Further, Critical Infrastructure Security and Resilience (CISR) were discussed [45]. LEMNA, a method to treat a DL model as a black box and derive explanations for each and every classification outcome in cyber security was proposed [46]. Another study demonstrated XAI by the mapping outcomes to 3 different tasks with detailed analysis using various interpretation and visualization methods [47].

E. Casual Theory with DL for Cyber Security

State of art DL algorithms have been applied to cyber security with enhanced accuracies. However, they are not widely used commercially due to the fact that there is no explicit reasoning to understand the inference mechanism used by these models. Causal inference is one of the methods used for this purpose as it has the ability to answer questions related to data distribution and intervention changes. Causal theory could be applied to cyber security for understanding DL models and in answering what-if type of questions that involves changes to the existing framework. A framework to understand a DL architecture using causal inference was proposed to show the effectiveness of this model [96]. Another study used causal theory in network traffic to confirm the TCP-SYN flooding DDoS attacks [97]. Further, causality countermeasures were utilized for detection of attacks [98]. PRIOTRACKER was proposed for tracking processes by prioritizing the investigation of abnormal causal dependencies. [99].

Fig. 6. Statistics of the DL Architectures in cyber security.

IX. A STATISTICAL SUMMARY OF DL APPLICATIONS IN CYBER SECURITY

In recent years, applying novel DL methods to cyber security as well as evaluating their performance to arrive at an optimal DL framework have become key research directions for security researchers. Fig. 6 provides a statistical summary of the percentage of contribution by popular DL architectures for cyber security. 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 and other variants of Autoencoder.
- 4) CNN and variant architectures such as AlexNet, VGG16, SqueezeNet, Inception-BN-21k, Inception-BN-1k, Inception V4, and 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 LSTM, Bidirectional gated recurrent unit, and other variants.
- 8) Reinforcement learning and variants.
- 9) Adversarial ML.

From Fig. 6, it is evident that DL architectures based on recurrent structures, CNN and DNN are largely used with surpassing both CNN and DNN. This may be due to the reason that most of the cyber security datasets involves the sequencing and time series information which forms a good fit with recurrent structures of DL architectures.

In Fig. 7 we present the summary of statistics of various studies of cyber security applications that have adopted DL approaches. We have considered the following 13 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)

Fig. 7. Statistics of the DL based Cyber Security Applications.

Fig. 8. Statistics of DL approach for Cyber Security Applications based on comparison with classical ML algorithms.

insider threat detection, 11) attacks detection in autonomous vehicles, 12) event detection in social media, 13) cryptography applications. Among these, we find that DL architectures are largely used in intrusion detection with highest number of studies reported in literature. Next comes Windows malware detection, Android malware detection and DGA, Email, URL and Security log data analysis.

Among the research works on DL based cyber security applications, the main significant published studies have not compared the performance of DL architecture with the existing CMLAs. This is very much required because for certain cyber security applications the CMLAs are more sufficient than the DL architectures. To identify this, we have shown the statistics of the various DL based cyber security applications in Fig. 8. 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, and 7) insider threat detection. The figure indicates that fewer published research works based on DL architectures have compared the results with CMLAs. Most importantly, the DL architectures outperformed the CMLAs in most of the research works justifying that DL architectures are more efficient and robust than the CMLAs.

Fig. 9. Statistics of DL approach for DGA, Email, URL and Security log Data analysis based on comparison with NLP text representation.

NLP is an important domain which has many important applications in cyber security. It deals with conversion of text to numerical representation. There exist many text representations and their performance implicitly depend 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 as shown in Fig. 9. Most of the research works have utilized Keras embedding as it helps to preserve the sequence information of words or characters in the texts. However, not much work exists based on the word embedding models because most of the cyber security text data does not involve semantic and contextual representation. We have considered the following text representations: 1) Bag of words, 2) n-grams, 3) One hot, 4) ASCII representations, 5) Keras embedding, 6) Word2vec (Sent2vec), 7) FastText, 8) Characters converted into image, and 9) Manual feature engineering. Fig. 9 shows that the Keras embedding text representation is largely used in published research studies 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 datasets are important and play an important role in the development of DL based cyber security applications. We have discussed that major issues exist in the various available datasets in detail in the above section. The datasets used in the existing DL based cyber security applications typically are categorized based on the following: 1) benchmark datasets can be used for performance evaluation of existing and as well as newly introduced algorithms, 2) datasets collected from publicly available sources, 3) real-time datasets are collected from real-time environment are considered as real-time datasets, and 4) private datasets are ones that do not

Fig. 10. Statistics of DL based Cyber Security Applications based on the dataset classification.

belong to these three types. The detailed statistics of the DL application in cyber security based on dataset type is shown in Fig. 10. While most of the research works have used the benchmark datasets, some research works use datasets which belong to the remaining categories. 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 in future research studies.

X. SUGGESTED HYBRID SYSTEMS FOR ORGANIZATIONAL CYBER SECURITY

Cybercriminals are in constant improvement of their attack skills and hence huge amount of data about user behaviors from different sources over the Internet is processed using DL methods to monitor and trace them in real-time. In order to preprocess and apply DL methods on the massive amount of data, new tools and technologies are warranted. To achieve this, we propose a highly scalable distributed computing platform with a DL based solution to collect data in a distributed manner and use distributed algorithms to analyze data with the aim to accurately detect and classify security events. This facilitates the user to know whether the administrator has to take an action. In this solution, 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 in each threat. The proposed solution monitors the network and the connected devices to identify variations from normal events that are associated with malicious attacks. The proposed model is designed to provide organizations with the situational awareness needed to deal with their most pressing issues of trust and security.

Based on the knowledge obtained from the surveyed papers, our proposed general DL framework for cyber security applications consists of 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 DL models with the aim to meet

today's cyber security challenges. Primarily the proposed DL framework contains data collection, data preprocessing, and DL based classification modules. In the Data collection phase, the data samples are collected passively from various sensors and stored in NoSQL database. These raw data samples are 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 as well as classify the malicious activities. The proposed framework is more general to handle various cyber security challenges in the modern society. It contains the following sub modules;

- Cyber threat situational awareness sub module based on DNS data analysis using DL.
- A sub module to analyze the global BGP updates for cyber security threat detection using DL.
- DL based sub module for Spam and Phishing detection using URL, Email, and social media data analysis.
- DL based hybrid intrusion detection sub module which can detect attacks at network and host level.
- DL 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 DL.
- DL based sub module for ransomware identification.
- DL and Visualization sub module for Botnet Detection in the Internet of Things of Smart Cities.
- DL based sub module for Android malware analysis.
- DL based anomaly detection sub module using operational logs in cloud applications.
- Malware spread modeling sub module using DL and scientific computing models.
- A Casual approach with DL sub module for the network anomaly detection.
- Malware visualization sub module using image processing and DL.
- Privacy preserved DL sub module using block chain technologies.
- DL based cyber security sub module for Cloud environment.
- DL based cyber security sub module for fog computing.
- DL based cyber security sub module for Cryptography.
- GAN based sub module for enhancement of robustness of DL models in an adversarial environment.
- Reinforcement learning based sub module for enhancing the system performance through pretraining.

An organisation's cyber security system composed of the above mentioned sub modules collectively is capable of detecting attacks more accurately and results in blocking the communication point between the malicious activities and the target host very quickly.

XI. CONCLUSION AND FUTURE RESEARCH

With technological advancements of BD and IoT resulting in Industry 4.0 of the future, the need for a well-rounded review of studies to overcome malicious attacks in CPS was discussed in this paper leading to the main focus of the study.

The ability of DL techniques to capture patterns from large volume of data to distinguish the legitimate and malicious activities formed the key motivation of the study to conduct an extensive review of DL architectures suitable for cyber security. Hence, this survey paper was presented in a tutorial-style, initially with a description of existing classical ML algorithms and DL architectures to address zero-day attacks. We summarized the major objectives, achievements and limitations of studies on DL applications in cyber security that were reported in literature. In addition, the paper uniquely reviewed the DL based cyber security applications according to various criteria such as the type of DL architecture, datasets used and the type of cyber security applications. Next, the importance of NLP, signal and image processing, and big data analytics in cyber security applications were discussed. Then, we carried out a comprehensive literature review of various DL architectures applied in cyber security, including state-of-the-art studies conducted with explainable AI, transfer learning, reinforcement learning, and adversarial deep learning. Recently, since attackers follow adversarial ML to bypass the DL based models and due to the paucity of studies in literature, in this paper we reviewed the robustness of DL methods in adversarial environments towards significant avenues for future research work.

We considered the importance of cyber security in emerging areas such as smart cities, IoT, cloud and edge computing, biometrics, pervasive computing, blockchain, and causal theory and discussed DL applications for such next generation technologies. We proposed a DL based hybrid framework with different layers of security to learn the characteristics of malware and legitimate activities more accurately. The proposed framework would evolve in real-time to detect and prevent from advanced attacks.

To conclude, DL is an important method for all the cyber security applications due to the rise of big data and IoT in CPS. The data distribution in big data is highly non-linear, noisy and dirty. Classical ML algorithms are not sufficient to deal with big data and the performance is of concern to be addressed in cyber security applications. Finally, through our review we find that DL applications for cyber security problems are suitable as they are well-suited to learn complex non-linear hypotheses with large number of features and high-order polynomial terms due to the dimensions of big data. In summary, this detailed survey on DL applications in cyber security provides a deep insight into a wide spectrum of research studies that would motivate researchers to advance the state of DL applications in cyber security.

When several types of learning approaches emerge, good research studies in future are required to identify an optimal DL architecture. Shared tasks are one of the prominent ways to push DL applications in cyber security research forward and organisation of shared tasks is anticipated to be more in the near future. Another emerging aspect is that due to limited availability of benchmark datasets, few of the published research studies have utilized the private datasets and various modified versions of benchmark datasets. Since private and modified version of benchmark datasets are not publicly available for further research, most of their solutions

are not directly comparable. This can undermine possible future research towards experimental evaluation in arriving at the best-fit DL architecture for any real-life environment. When choosing an effective and practical DL architecture, several checklists have to be considered such as accuracy, space and time complexity, ability to detect new malwares, easy integration and deployment in real-time system, and robustness in an adversarial environment. Another important factor to be considered is that most of the existing DL based models for cyber security applications are trained using supervised learning approaches, which require the labeled datasets and it is expensive to collect them. Therefore, there is a dire need for the learning methodology of contemporary DL based cyber security applications to be at least semi-supervised and at most unsupervised. In order to self-adapt to detect new attacks, DL architectures require retraining in cyber security domain, which implies that it relies on incremental learning and continual lifelong learning. Exploring the types of incremental learning and continual lifelong learning with DL applications in cyber security is another significant direction of future research towards enforcing security for Industry 4.0.

ACKNOWLEDGMENT

The authors would like to thank NVIDIA India, for the GPU hardware support to research grant. They would like to thank Computational Engineering and Networking (CEN) department for encouraging the research. Quoc-Viet Pham would like to thank the National Research Foundation of Korea (NRF) for financial support (No.NRF-2019R1C1C1006143). The authors are thankful to all the reviewers and editors for their valuable suggestions to enhance the quality of the article.

REFERENCES

- [1] 2018 Internet Security Threat Report, <https://www.symantec.com/content/dam/symantec/docs/reports/istr-23-executive-summary-en.pdf>
- [2] S. Morgan, "2017 Cybercrime report, Cybercrime damages will cost the world us\$6 trillion by 2021," Cybersecurity Ventures, Herjavec Group, Online Report, 2017.
- [3] J. Trull, "Top 5 best practices to automate security operations," Microsoft Secure, Enterprise Cybersecurity Group, Online Blog, August 2017.
- [4] M. Armstrong, "The future of a.i.," Statista Infographics, Statista, Online Report, November 2016.
- [5] 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.
- [6] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep Learning. *nature*, 521(7553), 436.
- [7] Buczak, L.; Guven, E. A Survey of Data Mining and Machine Learning Methods for Cybersecurity. *IEEE Commun. Surv. Tutor.* 2016, 18, 1153–1176.
- [8] Nguyen, T.T.T.; Armitage, G. A survey of techniques for internet traffic classification using machine learning. *IEEE Commun. Surv. Tutor.* 2008, 10, 56–76.
- [9] Garcia-Teodoro, P., Diaz-Verdejo, J., Maciá-Fernández, G., & Vázquez, E. (2009). Anomaly-based network intrusion detection: Techniques, systems and challenges. *computers & security*, 28(1-2), 18-28.
- [10] 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, 343–356.
- [11] Wu, S.X.; Banzhaf, W. The use of computational intelligence in intrusion detection systems: A review. *Appl. Soft Comput.* 2010, 10, 1–35.
- [12] Torres, J.M.; Comesaña, C.I.; García-Nieto, P.J. Machine Learning techniques applied to cybersecurity. *Int. J. Mach. Learn. Cybern.* 2019, 1–14.

- [13] 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, 35365-35381.
- [14] Ferrag, M.A., Maglaras, L., Moschogiannis, S. and Janicke, H., 2020. Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study. *Journal of Information Security and Applications*, 50, p.102419.
- [15] Apruzzese, G., Colajanni, M., Ferretti, L., Guido, A. and 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.
- [16] 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, 21–23 October 2018; pp. 745–751.
- [17] Amanullah, M.A., Habeeb, R.A.A., Nasaruddin, F.H., Gani, A., Ahmed, E., Nainar, A.S.M., Akim, N.M. and Imran, M., 2020. Deep learning and big data technologies for IoT security. *Computer Communications*.
- [18] Mahdaviifar, S., & Ghorbani, A. A. (2019). Application of Deep Learning to Cybersecurity: A Survey. *Neurocomputing*.
- [19] Berman, D. S., Buczak, A. L., Chavis, J. S., & Corbett, C. L. (2019). A Survey of Deep Learning Methods for Cybersecurity. *Information*, 10(4), 122.
- [20] Lopez-Martin, M., Carro, B. and Sanchez-Esguevilas, A., 2020. Application of deep reinforcement learning to intrusion detection for supervised problems. *Expert Systems with Applications*, 141, p.112963.
- [21] Huynh-The, T., Hua, C.H., Pham, Q.V. and Kim, D.S., 2020. MCNet: An Efficient CNN Architecture for Robust Automatic Modulation Classification. *IEEE Communications Letters*.
- [22] 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.
- [23] 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.
- [24] 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.
- [25] Alom, M. Z., & Taha, T. M. (2017, June). Network intrusion detection for Cybersecurity using unsupervised deep learning approaches. In *Aerospace and Electronics Conference (NAECON), 2017 IEEE National* (pp. 63-69). IEEE.
- [26] 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.
- [27] Arnaldo, I., Cuesta-Infante, A., Arun, A., Lam, M., Bassias, C., & Veeramachaneni, K. (2017, June). Learning Representations for Log Data in Cybersecurity. In *International Conference on Cybersecurity Cryptography and Machine Learning* (pp. 250-268). Springer, Cham.
- [28] 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.
- [29] 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.
- [30] 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.
- [31] 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.
- [32] J.L. Chu, A. Krzyzak, Analysis of feature maps selection in supervised learning using convolutional neural networks. *Advances in Artificial Intelligence*, Springer International Publishing, 2014, pp. 59-70.
- [33] M.D. Zeiler, R. Fergus, Visualizing and understanding convolutional neural networks, in: *Proceedings of the ECCV*, 2014.
- [34] Angrishi, K. (2017). Turning internet of things (iot) into internet of vulnerabilities (iov): Iot botnets. *arXiv preprint arXiv:1702.03681*.
- [35] Meidan, Yair, et al. "N-BalIoT—Network-Based Detection of IoT Botnet Attacks Using Deep Autoencoders." *IEEE Pervasive Computing* 17.3 (2018): 12-22.
- [36] 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).
- [37] Salman, Tara, et al. "Security services using blockchains: A state of the art survey." *IEEE Communications Surveys & Tutorials* (2018).
- [38] Dinh, Thang N., and My T. Thai. "AI and blockchain: A disruptive integration." *Computer* 51.9 (2018): 48-53.
- [39] Weng, J., et al. "Deepchain: Auditable and privacy-preserving deep learning with blockchain-based incentive." *Cryptology ePrint Archive, Report 2018/679* (2018).
- [40] Zhang, C., Ahmad, M., & Wang, Y. (2019). Admm based privacy-preserving decentralized optimization. *IEEE Transactions on Information Forensics and Security*, 14(3), 565-580.
- [41] Munsing, E., & Moura, S. (2018). Cybersecurity in Distributed and Fully-Decentralized Optimization: Distortions, Noise Injection, and ADMM. *arXiv preprint arXiv:1805.11194*.
- [42] 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*.
- [43] 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.
- [44] 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.
- [45] Rieger, Craig, and Milos Manic. "On Critical Infrastructures, Their Security and Resilience-Trends and Vision." *arXiv preprint arXiv:1812.02710* (2018).
- [46] 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.
- [47] Samek, W., Wiegand, T., & Müller, K. R. (2017). Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models. *arXiv preprint arXiv:1708.08296*.
- [48] Gunning, D. (2017). Explainable artificial intelligence (XAI). *Defense Advanced Research Projects Agency (DARPA)*, nd Web.
- [49] Adadi, A., & Berrada, M. (2018). Peeking inside the black-box: A survey on Explainable Artificial Intelligence (XAI). *IEEE Access*, 6, 52138-52160.
- [50] 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).
- [51] 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).
- [52] 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.
- [53] 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.
- [54] 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.
- [55] 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*.
- [56] 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*.
- [57] Li, Z., Zou, D., Xu, S., Jin, H., Zhu, Y. and Chen, Z., 2018. Sysevr: A framework for using deep learning to detect software vulnerabilities. *arXiv preprint arXiv:1807.06756*.
- [58] 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*.
- [59] 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.
- [60] Katz, D. S., Ruchti, J., & Schulte, E. (2018, March). Using recurrent neural networks for decompilation. In *2018 IEEE 25th International Con-*

- ference on Software Analysis, Evolution and Reengineering (SANER) (pp. 346-356). IEEE.
- [61] Krenn, R. (2004). Steganography and steganalysis. Retrieved September, 8, 2007.
- [62] 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.
- [63] Baluja, S. (2017). Hiding images in plain sight: Deep steganography. In *Advances in Neural Information Processing Systems* (pp. 2069-2079).
- [64] 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.
- [65] Zhu, J., Kaplan, R., Johnson, J., & Fei-Fei, L. (2018). Hidden: Hiding data with deep neural networks. In *Proceedings of the European Conference on Computer Vision (ECCV)* (pp. 657-672).
- [66] 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.
- [67] Zhang, R., Dong, S., & Liu, J. (2018). Invisible steganography via generative adversarial networks. *Multimedia Tools and Applications*, 1-17.
- [68] Wang, Z., Gao, N., Wang, X., Qu, X., & Li, L. (2018, December). SSteganGAN: Self-learning Steganography Based on Generative Adversarial Networks. In *International Conference on Neural Information Processing* (pp. 253-264). Springer, Cham.
- [69] Zhang, K. A., Cuesta-Infante, A., & Veeramachaneni, K. (2019). SteganoGAN: Pushing the Limits of Image Steganography. arXiv preprint arXiv:1901.03892.
- [70] 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.
- [71] 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.
- [72] Volkhonskiy, D., Borisenko, B., & Burnaev, E. (2016). Generative adversarial networks for image steganography.
- [73] 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.
- [74] 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.
- [75] 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.
- [76] 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.
- [77] 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.
- [78] 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.
- [79] Abadi, M., & Andersen, D. G. (2016). Learning to protect communications with adversarial neural cryptography. arXiv preprint arXiv:1610.06918.
- [80] 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.
- [81] Dylan Modesitt, Tim Henry, Jon Coden, and Rachel Lathe. *Neural Cryptography: From Symmetric Encryption to Adversarial Steganography*. Available at <https://courses.csail.mit.edu/6.857/2018/project/Modesitt-Henry-Coden-Lathe-NeuralCryptography.pdf>
- [82] Coutinho, M., de Oliveira Albuquerque, R., Borges, F., García Villalba, L., & Kim, T. H. (2018). Learning Perfectly Secure Cryptography to Protect Communications with Adversarial Neural Cryptography. *Sensors*, 18(5), 1306.
- [83] 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.
- [84] 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.
- [85] Hill, G. D., & Bellekens, X. J. (2017). Deep Learning Based Cryptographic Primitive Classification. arXiv preprint arXiv:1709.08385.
- [86] Hesamifard, E., Takabi, H., & Ghasemi, M. (2017). Cryptodl: Deep neural networks over encrypted data. arXiv preprint arXiv:1711.05189.
- [87] 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).
- [88] 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.
- [89] Pham, Quoc-Viet, Fang Fang, Vu Nguyen Ha, Mai Le, Zhiguo Ding, Long Bao Le, and Won-Joo Hwang. "A survey of multi-access edge computing in 5G and beyond: Fundamentals, technology integration, and state-of-the-art." arXiv preprint arXiv:1906.08452 (2019).
- [90] 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.
- [91] Bhattacharjee, B., Boag, S., Doshi, C., Dube, P., Herta, B., Ishakian, V., Jayaram, K.R., Khalaf, R., Krishna, A., Li, Y.B. and Muthusamy, V., 2017. IBM deep learning service. *IBM Journal of Research and Development*, 61(4/5), pp.10-1.
- [92] 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.
- [93] 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.
- [94] Gülmez, 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.
- [95] Sundararajan, K., & Woodard, D. L. (2018). Deep learning for biometrics: a survey. *ACM Computing Surveys (CSUR)*, 51(3), 65.
- [96] Narendra, T., Sankaran, A., Vijaykeerthy, D., & Mani, S. (2018). Explaining Deep Learning Models using Causal Inference. arXiv preprint arXiv:1811.04376.
- [97] Deka, R. K., Bhattacharyya, D. K., & Kalita, J. K. Granger Causality in TCP Flooding Attack.
- [98] 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.
- [99] Liu, Y., Zhang, M., Li, D., Jee, K., Li, Z., Wu, Z., Rhee, J. and 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.
- [100] Prabhu, C. S. R. (2019). *Fog Computing, Deep Learning and Big Data Analytics-Research Directions*. Springer.
- [101] Pham, Quoc-Viet, Tuan Leanh, Nguyen H. Tran, Bang Ju Park, and Choong Seon Hong (2018). "Decentralized computation offloading and resource allocation for mobile-edge computing: A matching game approach." *IEEE Access*, 6, 75868-75885.
- [102] Pham, Q.V., Le, L.B., Chung, S.H. and Hwang, W.J., 2019. Mobile edge computing with wireless backhaul: Joint task offloading and resource allocation. *IEEE Access*, 7, pp.16444-16459.
- [103] Xu, J. and Yao, J., 2018. Exploiting physical-layer security for multiuser multicarrier computation offloading. *IEEE Wireless Communications Letters*, 8(1), pp.9-12.
- [104] Roman, R., Lopez, J. and Mambo, M., 2018. Mobile edge computing, fog et al.: A survey and analysis of security threats and challenges. *Future Generation Computer Systems*, 78, pp.680-698.
- [105] Xiao, Y., Jia, Y., Liu, C., Cheng, X., Yu, J. and Lv, W., 2019. Edge computing security: State of the art and challenges. *Proceedings of the IEEE*, 107(8), pp.1608-1631.
- [106] Ahmad, I., Shahabuddin, S., Kumar, T., Okwuibe, J., Gurtov, A. and Ylianttila, M., 2019. Security for 5G and Beyond. *IEEE Communications Surveys & Tutorials*, 21(4), pp.3682-3722.
- [107] Guan, Y., Shao, J., Wei, G., & Xie, M. (2018). Data security and privacy in fog computing. *IEEE Network*, (99), 1-6.
- [108] Haque, M., & Ahamed, S. I. (2006). Security in pervasive computing: Current status and open issues. *International Journal of Network Security*.

- [109] Pagter, J. I., & Petersen, M. G. (2007, February). A Sense of Security in Pervasive Computing—Is the Light on When the Refrigerator Door Is Closed?. In *International Conference on Financial Cryptography and Data Security* (pp. 383-388). Springer, Berlin, Heidelberg.
- [110] 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.
- [111] Stajano, F. (2004). Security in pervasive computing. In *Security in Pervasive Computing* (pp. 6-8). Springer, Berlin, Heidelberg.
- [112] 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.
- [113] 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.
- [114] 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.
- [115] Kirat, D., Nataraj, L., Vigna, G., & Manjunath, B. S. (2013, December). Signal: A static signal processing based malware triage. In *Proceedings of the 29th Annual Computer Security Applications Conference* (pp. 89-98). ACM.
- [116] 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.
- [117] Silva, S. S., Silva, R. M., Pinto, R. C., & Salles, R. M. (2013). Botnets: A survey. *Computer Networks*, 57(2), 378-403.
- [118] 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.
- [119] 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).
- [120] Puliga, M., Caldarelli, G., Chessa, A., & De Nicola, R. (2018). Understanding the Twitter User Networks of Viruses and Ransomware Attacks. In *ITASEC*.
- [121] 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.
- [122] 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.
- [123] 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.
- [124] 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).
- [125] 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.
- [126] 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.
- [127] 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.
- [128] 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.
- [129] 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.
- [130] 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.
- [131] Guo, W., Wang, T., Wei, J. (2017, August). Malware Detection with Convolutional Neural Network Using Hardware Events. In *CCF National Conference on Computer Engineering and Technology* (pp. 104-115). Springer, Singapore.
- [132] Yan, J., Qi, Y., Rao, Q. (2018). Detecting malware with an ensemble method based on deep neural network. *Security and Communication Networks*, 2018.
- [133] Vinayakumar, R., & Soman, K. P. (2018). DeepMalNet: Evaluating shallow and deep networks for static PE malware detection. *ICT Express*, 4(4), 255-258.
- [134] Le, Q., Boydel, O., Mac Namee, B., & Scanlon, M. (2018). Deep learning at the shallow end: Malware classification for non-domain experts. *Digital Investigation*, 26, S118-S126.
- [135] 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.
- [136] 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.
- [137] 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.
- [138] 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.
- [139] 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.
- [140] Benadjila, R., Prouff, E., Strullu, R., Cagli, E., & Dumas, C. Study of Deep Learning Techniques for Side-Channel Analysis and Introduction to ASCAD Database.
- [141] Pfeifer, C., & Haddad, P. Spread: a new layer for profiled deep-learning side-channel attacks. *Analysis*, 6, 3.
- [142] 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.
- [143] Samiotis, I. P. (2018). Side-Channel Attacks using Convolutional Neural Networks: A Study on the performance of Convolutional Neural Networks on side-channel data.
- [144] 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.
- [145] Yu, W., & Chen, J. (2018). Deep learning-assisted and combined attack: a novel side-channel attack. *Electronics Letters*, 54(19), 1114-1116.
- [146] 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*.
- [147] 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.
- [148] 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*.
- [149] 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.
- [150] 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.
- [151] 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.
- [152] 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.
- [153] 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.

- [154] 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.
- [155] Malialis, K., Kudenko, D. (2013). Large-scale DDoS response using cooperative reinforcement learning. In *11th European Workshop on Multi-Agent Systems (EUMAS)*.
- [156] Anderson, H. S., Kharkar, A., Filar, B., Roth, P. (2017). *Evading Machine Learning malware detection*. Black Hat.
- [157] Servin, A. Towards Traffic Anomaly Detection via Reinforcement Learning and Data Flow.
- [158] 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.
- [159] 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.
- [160] Wang, Y., Stokes, J. W., & Marinescu, M. (2018). Neural Malware Control with Deep Reinforcement Learning.
- [161] Z. Yuan, Y. Lu, Z. Wang, and Y. Xue, "Droid-Sec: Deep Learning in Android Malware Detection," *Sigcomm 2014*, pp. 371-372, 2014.
- [162] 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.
- [163] 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.
- [164] 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.
- [165] 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.
- [166] N. McLaughlin et al., 2017, March. Deep android malware detection. In *Proceedings of the Seventh ACM on Conference on Data and Application Security and Privacy* (pp. 301-308).
- [167] Nix, R. and Zhang, J., 2017, May. Classification of android apps and malware using deep neural networks. In *2017 International joint conference on neural networks (IJCNN)* (pp. 1871-1878). IEEE.
- [168] Huang, TonTon Hsien-De, and Hung-Yu Kao. "R2-D2: color-inspired convolutional neural network (CNN)- based android malware detections." 2018 IEEE International Conference on Big Data, Dec. 2018. IEEE.
- [169] 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.
- [170] 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.
- [171] Wang, W., Zhao, M. and Wang, J., 2019. Effective android malware detection with a hybrid model based on deep autoencoder and convolutional neural network. *Journal of Ambient Intelligence and Humanized Computing*, 10(8), pp.3035-3043.
- [172] E. M. B. Karbab, M. Debbabi, A. Derhab, and D. Mouheb, "Mal-Dozer: Automatic framework for Android malware detection using Deep Learning," *Digit. Investig.*, vol. 24, no. March, pp. S48-S59, 2018.
- [173] 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.
- [174] 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.
- [175] 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.
- [176] 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.
- [177] 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.
- [178] Pektas, A., & Acarman, T. (2019). Deep learning for effective Android malware detection using API call graph embeddings. *Soft Computing*, 1-17.
- [179] 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.
- [180] 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.
- [181] 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.
- [182] 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.
- [183] 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.
- [184] 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*.
- [185] 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*.
- [186] Paudice, A., Muoz-Gonzalez, L., Gyorgy, A., Lupu, E. C. (2018). Detection of Adversarial Training Examples in Poisoning Attacks through Anomaly Detection. *arXiv preprint arXiv:1802.03041*.
- [187] 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.
- [188] 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*.
- [189] 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*.
- [190] Carlini, N., Wagner, D. (2018). Audio adversarial examples: Targeted attacks on speech-to-text. *arXiv preprint arXiv:1801.01944*.
- [191] Madani, P., & Vljajic, 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.
- [192] 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*.
- [193] 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.
- [194] Salem, M., Taheri, S., & Yuan, J. S. (2018). Anomaly Generation using Generative Adversarial Networks in Host Based Intrusion Detection. *arXiv preprint arXiv:1812.04697*.
- [195] Ring, M., Schlör, D., Landes, D., & Hotho, A. (2019). Flow-based network traffic generation using generative adversarial networks. *Computers & Security*, 82, 156-172.
- [196] 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*.
- [197] 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*.
- [198] Sidi, L., Nadler, A., & Shabtai, A. (2019). MaskDGA: A Black-box Evasion Technique Against DGA Classifiers and Adversarial Defenses. *arXiv preprint arXiv:1902.08909*.
- [199] Nataraj, L., Mohammed, T. M., Manjunath, B. S., Chandrasekaran, S., Flenner, A., Bappy, J. H., & Roy-Chowdhury, A. K. (2019). Detect-

- ing GAN generated Fake Images using Co-occurrence Matrices. arXiv preprint arXiv:1903.06836.
- [200] 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.
- [201] Staudemeyer, R. C. (2015). Applying long short-term memory recurrent neural networks to intrusion detection. *South African Computer Journal*, 56(1), 136-154.
- [202] 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.
- [203] 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.
- [204] 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.
- [205] 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.
- [206] Chong, P., Tan, Y. X. M., Guarnizo, J.Elovici, , Y., Binder, A. (2018, May). Mouse Authentication Without the Temporal Aspect-What Does a 2D-CNN Learn?. In *2018 IEEE Security and Privacy Workshops (SPW)* (pp. 15-21).
- [207] 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.
- [208] 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.
- [209] 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.
- [210] 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.
- [211] 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.
- [212] 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.
- [213] 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.
- [214] Diro, A., & Chilamkurti, N. (2018). Leveraging LSTM networks for attack detection in fog-to-things communications. *IEEE Communications Magazine*, 56(9), 124-130.
- [215] 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.
- [216] 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.
- [217] 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.
- [218] 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.
- [219] Yavuz, F. Y., Ünal, D., & Gül, E. (2018). Deep learning for detection of routing attacks in the internet of things. *International Journal of Computational Intelligence Systems*, 12(1), 39-58.
- [220] 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.
- [221] Yan, B., & Han, G. (2018). Effective feature extraction via stacked sparse autoencoder to improve intrusion detection system. *IEEE Access*, 6, 41238-41248.
- [222] Jia, Y., Wang, M., & Wang, Y. (2018). Network intrusion detection algorithm based on deep neural network. *IET Information Security*,
- [223] Maimó, L. F., Gómez, Á. L. P., Clemente, F. J. G., Pérez, M. G., & Pérez, G. M. (2018). A self-adaptive deep learning-based system for anomaly detection in 5G networks. *IEEE Access*, 6, 7700-7712.
- [224] 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.
- [225] Khan, F. A., Gumaei, A., Derhab, A., & Hussain, A. (2019). TSDL: A TwoStage Deep Learning Model for Efficient Network Intrusion Detection. *IEEE Access*.
- [226] Roopak, M., Tian, G. Y., & Chambers, J. (2019, January). Deep Learning Models for Cybersecurity in IoT Networks. In *2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC)* (pp. 0452-0457). IEEE.
- [227] Papamartzivanos, D., Mármol, F. G., & Kambourakis, G. (2019). Introducing Deep Learning Self-Adaptive Misuse Network Intrusion Detection Systems. *IEEE Access*.
- [228] Wang, Z. (2015). The applications of Deep Learning on traffic identification. *BlackHat USA*.
- [229] 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.
- [230] 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.
- [231] 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.
- [232] 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.
- [233] Aceto, G., Ciunzo, D., Montieri, A., Pescap, A. (Jun. 2019). Mobile Encrypted Traffic Classification Using Deep Learning. *IEEE Transactions on Network and Service Management*, 16, 445-458.
- [234] 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.
- [235] 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*.
- [236] 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*.
- [237] 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.
- [238] 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*.
- [239] Shibahara, T., Yamanishi, K., Takata, Y., Chiba, D., Akiyama, M., Yagi, T., Ohsita, Y. and Murata, M., 2017, May. Malicious URL sequence detection using event de-noising convolutional neural network. In *2017 IEEE International Conference on Communications (ICC)* (pp. 1-7). IEEE.
- [240] Zhao, N., Liu, Y., Jiang, Y. CAPTCHA Breaking with Deep Learning.
- [241] 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.
- [242] Zannettou, S., Chatzis, S., Papadamou, K. and Sirivianos, M., 2018, May. The good, the bad and the bait: Detecting and characterizing clickbait on YouTube. In *2018 IEEE Security and Privacy Workshops (SPW)* (pp. 63-69).
- [243] 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*.
- [244] Ra, V., HBa, B. G., Ma, A. K., KPa, S., Poornachandran, P. DeepAnti-PhishNet:Applying Deep Neural Networks for Phishing Email Detection.
- [245] HBa, B. G., Ra, V., Ma, A. K., KPa, S. Distributed Representation using Target Classes: Bag of Tricks for Security and Privacy Analytics.

- [246] 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.
- [247] Bahnsen, A. C., Bohorquez, E. C., Villegas, S., Vargas, J., & Gonzalez, F.A. (2017, April). Classifying phishing URLs using recurrent neural networks. In *Electronic Crime Research (eCrime)*, 2017 APWG Symposium on (pp. 1-8). IEEE.
- [248] 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
- [249] 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.
- [250] 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.
- [251] 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.
- [252] 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.
- [253] 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.
- [254] 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.
- [255] Shima, K., Miyamoto, D., Abe, H., Ishihara, T., Okada, K., Sekiya, Y., Asai, H. and Doi, Y., 2018, February. Classification of URL bitstreams using bag of bytes. In *2018 21st Conference on Innovation in Clouds, Internet and Networks and Workshops (ICIN)* (pp. 1-5). IEEE.
- [256] Sahingoz, O. K., Baykal, S. I., & Bulut, D. (2017). Phishing Detection from URLs by using Neural Networks. In *Computer Science & Information Technology (CS & IT) CSCP 2018*.
- [257] Yang, W., Zuo, W., & Cui, B. (2019). Detecting Malicious URLs via a Keyword-based Convolutional Gated-recurrent-unit Neural Network. *IEEE Access*.
- [258] Vinayakumar, R., Soman, K. P., & Poornachandran, P. (2018). Evaluating deep learning approaches to characterize and classify malicious URL's. *Journal of Intelligent & Fuzzy Systems*, 34(3), 1333-1343.
- [259] 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.
- [260] 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.
- [261] 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*.
- [262] 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]
- [263] Vinayakumar, R., Soman, K.P., Poornachandran, P., Alazab, M. and Jolfaei, A., 2019. DBD: Deep Learning DGA-Based Botnet Detection. In *Deep Learning Applications for Cyber Security* (pp. 127-149). Springer, Cham.
- [264] Vinayakumar, R., Alazab, M., Soman, K.P., Poornachandran, P., Al-Nemrat, A. and Venkatraman, S., 2019. Deep learning approach for intelligent intrusion detection system. *IEEE Access*, 7, pp.41525-41550.
- [265] Vinayakumar, R., Alazab, M., Jolfaei, A., Soman, K.P. and Poornachandran, P., 2019, May. Ransomware triage using deep learning: twitter as a case study. In *2019 Cybersecurity and Cyberforensics Conference (CCC)* (pp. 67-73). IEEE.
- [266] 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.
- [267] 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.
- [268] Kang, M. J., & Kang, J. W. (2016). Intrusion detection system using deep neural network for in-vehicle network security. *PloS one*, 11(6), e0155781.
- [269] 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.
- [270] 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.

Vinayakumar Ravi received the Ph.D. degree in computer science from Computational Engineering & Networking, Amrita School of Engineering, Coimbatore, Amrita Vishwa Vidyapeetham, India. He is currently a Postdoctoral research fellow in developing and implementing novel computational and machine learning algorithms and applications for big data integration and data mining with Cincinnati Children's Hospital Medical Center, Cincinnati, OH, USA. He has received MCA from Amrita Vishwa Vidyapeetham, Mysore in 2014 and BCA from JSS College of Arts, Commerce and Sciences, Ooty road, Mysore in 2011. He has several papers in Machine Learning applied to Cyber Security. His Ph.D. work centers on Application of Machine learning (some times Deep learning) for Cyber Security and discusses the importance of Natural language processing, Image processing and Big data analytics for Cyber Security. He has participated in several international shared tasks and organized a shared task on detecting malicious domain names (DMD 2018) as part of SSCC'18 and ICACCI'18. His research interests include machine learning and deep learning applications with natural language processing and image processing for Cyber Security. More details available at <https://vinayakumarr.github.io/>.

Mamoun Alazab received his PhD degree in Computer Science from the Federation University of Australia, School of Science, Information Technology and Engineering. He is an Associate Professor in the College of Engineering, IT and Environment at Charles Darwin University, Australia. He is a Cyber Security researcher and practitioner with industry and academic experience. Alazab's research is multi-disciplinary that focuses on Cyber Security and digital forensics of computer systems including current and emerging issues in the cyber environment like cyber-physical systems and internet of things with a focus on cyber crime detection and prevention. He has more than 100 research papers. He delivered many invited and keynote speeches, 22 events in 2018 alone. He convened and chaired more than 50 conferences and workshops. He works closely with government and industry on many projects. He is an editor on multiple editorial boards including Associate Editor of IEEE Access, Editor of the Security and Communication Networks Journal, and Book Review Section Editor: Journal of Digital Forensics, Security and Law (JDFSL). He is a Senior Member of the IEEE.

K. P. Soman has 25 years of research and teaching experience at Amrita School of Engineering, Coimbatore. He has around 150 publications in national and international journals and conference proceedings. He has organized a series of workshops and summer schools in Advanced signal processing using wavelets, Kernel Methods for pattern classification, Deep Learning, and Big-data Analytics for industry and academia. He authored books on "Insight into Wavelets", "Insight into Data mining", "Support Vector Machines and Other Kernel Methods" and "Signal and Image processing-the sparse way", published by Prentice Hall, New Delhi, and Elsevier. More details available at <https://nlp.amrita.edu/somankp/>.

Simran K is a M.Tech student in the Computational Engineering & Networking, Amrita School of Engineering, Coimbatore, Amrita Vishwa Vidyapeetham, India. She has received B.E in Information Technology from MVSR engineering college, Hyderabad, India in 2018. Her areas of interests are Machine Learning, Deep Learning, Natural language processing, Cyber Security, Image processing and Internet of Things (IoT). More details available at <https://simranketha.github.io/>.

Sriram Srinivasan is a M.Tech student in the Computational Engineering & Networking, Amrita School of Engineering, Coimbatore, Amrita Vishwa Vidyapeetham, India. He has received B.Tech in Computer Science and Engineering from Sastra University, Srinivasa Ramanujan Center, Kumbakonam, India in 2016. His areas of interests are Machine Learning, Deep Learning, Cyber Security and Internet of Things (IoT). More details available at <https://thecodinglama.github.io/>.

Dr. Sitalakshmi Venkatraman has more than 30 years of work experience both in industry and academics within India, Singapore, New Zealand, and in Australia since 2007. She specialises in applying efficient computing models and data mining techniques for various industry problems and recently in the e-health, e-security and e-business domains through collaborations with industry and universities in Australia. Currently she leads the Business Analytics team for teaching and research in Melbourne Polytechnic. She has published nine book chapters and more than 130 research papers in internationally well-known refereed journals and conferences. She is a Senior Member of professional societies and editorial boards of international journals and serves as Program Committee Member of several international conferences.

Quoc-Viet Pham (M'18) received the B.S. degree in electronics and telecommunications engineering from Hanoi University of Science and Technology, Vietnam, in 2013, and the M.S. and Ph.D. degrees, both in telecommunications engineering, from Inje University, South Korea, in 2015 and 2017 respectively. He is currently a research professor at Research Institute of Computer, Information and Communication, Pusan National University, South Korea. From Sept. 2017 to Dec. 2019, he was with Kyung Hee University, Changwon National University, and Inje University on various academic positions. He received the best PhD thesis award in Engineering from Inje University in 2017. His research interests include convex optimization, game theory, and machine learning to mobile edge/cloud computing, and resource allocation for 5G wireless networks and beyond. He is a member of the IEEE.