Generalization of Deep Learning for Cyber-Physical System Security: A Survey

Chathurika S. Wickramasinghe, Daniel L. Marino, Kasun Amarasinghe, Milos Manic
Department of Computer Science
Virginia Commonwealth University
Richmond, USA
brahmanacsw@vcu.edu, marinodl@vcu.edu, amarasinghek@vcu.edu, misko@ieee.org

***Abstract*—Cyber-Physical Systems (CPSs) have become ubiquitous in recent years and has become the core of modern critical infrastructure and industrial applications. Therefore, ensuring security is a prime concern. Due to the success of Deep Learning (DL) in a multitude of domains, development of DL based CPS security applications have received increased interest in the past few years. Developing generalized models is critical since the models have to perform well under threats that they havent trained on. However, despite the broad body of work on using DL for ensuring the security of CPSs, to our best knowledge very little work exists where the focus is on the generalization capabilities of these DL applications. In this paper, we intend to provide a concise survey of the regularization methods for DL algorithms used in security-related applications in CPSs and thus could be used to improve the generalization capability of DL based cyber-physical system based security applications. Further, we provide a brief insight into the current challenges and future directions as well.**

***Index Terms*—Generalization, Deep Neural networks, Regularization, Cyber-Physical Systems, Cyber Security**

I. Introduction

Cyber-Physical Systems (CPSs) have become a common component found in critical infrastructure due to their enormous impact and economic benefits [1]. The increasing dependency of critical infrastructure on cyber-based technologies have made them vulnerable to cyber-attacks such as interception, replacement and removal of information from the communication channels [2] [3] [4]. Therefore, the security of CPSs has become a critical concern. Deep Leaning (DL) has gained significant attention within past years as it has improved the state-of-art performance of many applications, including security-related applications in critical structures, such as intrusion detection, malware detection, access control, anomaly detection, and classifications [1].

Deep learning (DL) was introduced in the late 20th century which was originated from the study of Artificial Neural Networks(ANNs). Deep Neural Networks(DNN) consists of a set of stacked models (layers) that learn a series of hidden representations hierarchically. Higher level representations contain amplified aspects of input samples which are useful for discrimination and suppress of irrelevant features. Deep leaning models have improved the state-of-the-art performance in many tasks including speech recognition, object detection, natural language processing and pattern recognition [5] [6].

Figure 1 illustrates the overall idea of CPS and the use of DL for CPSs. It shows examples of existing CPSs, what kind of features can be extracted from such systems, possible DL models and advantages of using DL. Further, the data collected from CPSs is typically high dimensional. DL models are specifically designed to deal with high-dimensional data. Other characteristics of CPSs include, continues growth of data, data drift and exposure to new system threats. Therefore, it is essential to build DL based security models which are adaptable and extendable with the data drift, continuous discovery of new system threats and vulnerabilities [7]. This concept of "Generalization" is one major problem for building security-based applications in CPSs because developed machine learning models for one scenario is nearly impossible to use in another situation even in the same context. Therefore, it is an essence to focus on generalization of DL models which used in such applications.

Generalization capability is a fundamental problem in designing any kind of artificial neural network [8] [9]. In real-world settings, the performance of ANN mostly depend on its generalization capability which measures the performance of ANN on the actual problem, i.e. the ability of ANN to handle unseen data [10] [11].

In this paper, we present a brief survey on the generalization of deep learning in the context of security of cyber-physical systems. The goal of the paper is to give an idea to the reader about DL methods that can be used in the CPSs, importance of generalization of DL methods in the context of security of CPSs and common regularization techniques on them.

The rest of the paper is organized as follows: Section II discusses DL techniques that have been successfully used in the field of cyber-security/CPSs. Section III provides a brief review of generalization. Section IV discusses regularization methods which have proven increase the generalization of DL methods which are discussed in section II. Section V reviews the most recent challenges in the field whereas Section VI discusses the conclusion and future directions.

II. Deep Learning models in Cyber-Security/ Cyber-Physical Systems

In this section, we talk about the related work where deep learning has been applied in CPSs cyber-security. First, we introduce deep leaning and discusses deep learning methods that

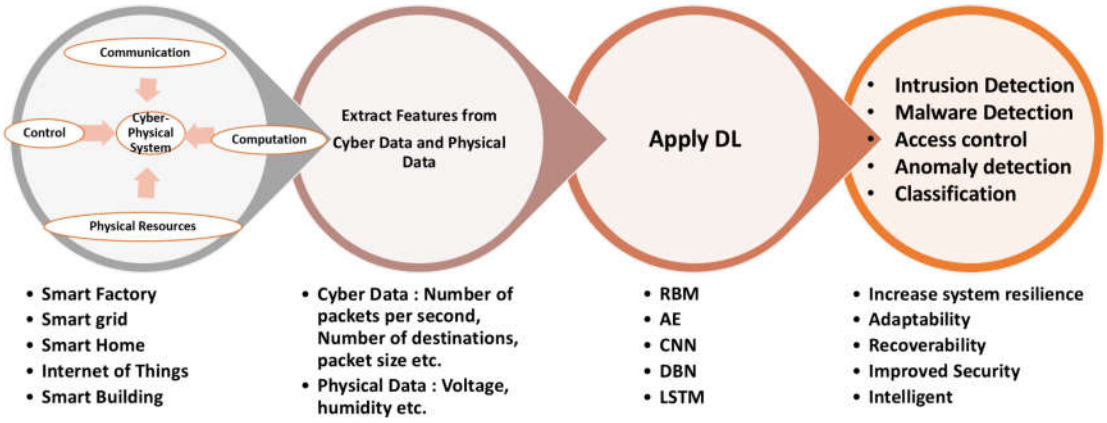

Fig. 1: Application of DL for CPSs

have been successfully applied in security-related applications in the context of Cyber-Physical systems.

Deep learning (DL) has gained high-focus in data science as they have enhanced performance in many applications [12]. DL algorithms consist of hierarchical architectures with many layers where higher level features are defined in terms of lower level features. They have the capability of extracting features and abstractions from underline data [13]. Many researchers have experimentally shown that these architectures are capable of yielding outstanding results in many applications including CPS security [12] [14].

Figure 2 presents important features that characterize the deep NNs and shallow NNs. Typically, Shallow architectures refer to models with only very few (usually one) hidden layer whereas deep architectures are composed of several hidden layers [5]. DL methods are capable of representing more abstract representations of data due to the multi-level architecture. In many practical applications, DL models have shown better generalization capability compared to shallow ANNs. However, the relative simplicity of shallow ANNs translates on a better understanding of shallow architectures compared to DL models.

Few major areas where DL has been successfully applied in CPSs for security related purposes are anomaly detection [15], malware detection and threat hunting [16] [17], vulnerability detection [18], intrusion detection [19], prevention of black-outs, attacks and destructions [20] in cyber-physical systems. In this work, we focus on investigating frequently used deep learning techniques that have been applied in the above areas.

- Deep Feed Forward Neural Networks:
  These are often called Multilayer Perceptrons (MLPs) as they are made with combining many layers of perceptrons (another type of shallow machine learning algorithms) into a deeper structure. These models are called 'feed forward' because there are no feedback connections where the output of the network is fed back to the network. MLPs have been successfully applied in many areas including cyber-security tasks such as malware detection [21], intrusion detection [22] and access control systems [23].
- Convolutional Neural Network (CNNs):
  CNNs are special kind of neural network for processing data with grid-like topology such as images and videos [24]. It combines three architectural ideas: local receptive fields, shared weights, and spatial subsampling to ensure some degree of shift and distortion invariance [25]. In cyber-security, it has been used for tasks like intrusion detection, classification and detection of malware variants [26] [27].
- Long Short-Term Memory:
  Long Short-Term Memory (LSTM) is a type of Recurrent Neural Network (RNN) proposed to solve the problem of vanishing and exploding gradient problem of conventional RNNs [23]. In cyber-security LSTMs have been used for tasks like classification and detection of malware variants [26] and anomaly detection [28] [27].
- Restricted Boltzmann Machines (RBMs):
  An RBM consist of two-layered undirected graphical models [29]. They are a stochastic model used to learn the underlying probability distribution of the dataset. They are used in many applications including image and speech recognition, dimensionality reduction, classification, feature learning, topic modeling and cyber-security. In cyber-security, it has been used for tasks like intrusion detection [30] , malicious code detection [31] and anomaly detection [28].
- Deep Belief Networks (DBNs):
  DBNs consist of a series of unsupervised multi-layered

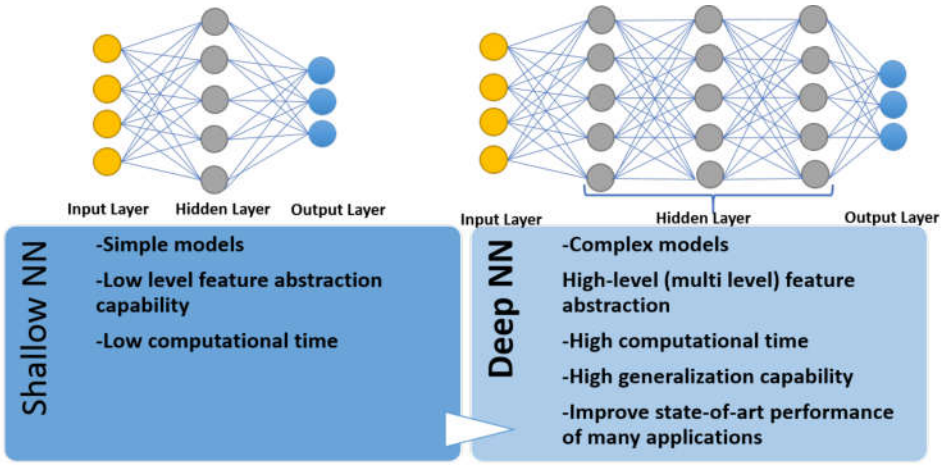

Fig. 2: Shallow vs Deep Neural Networks

RBM networks (stacked RBMs) and a supervised back-propagation network [30] [31]. DBNs are more effective compared to other ANNs specially with unlabeled data [29]. They have been successfully used in many areas including image classification, speech recognition and information retrieval, natural language processing and cyber-security. In cyber-security, DBNs have been used for tasks like malicious code detection [31], intrusion detection [30] and anomaly detection [28].

- Autoencoders: Autoencoder structure is divided into two parts: encoder and decoder. The encoder converts the input data into an abstract representation which is then reconstructed using the decoder. They are widely used for the purpose of dimensionality reduction. In cyber-security, it has been used for tasks like malicious code detection [31], detection of malware variants [26] and anomaly detection [28].

III. Generalization

Since this paper focus on the generalization of DL techniques, it is essential to get a brief idea about what is the generalization, how to measure the generalization capability of a neural network and what are the existing categories of generalization technique. This section briefly discusses each of the above topics.

The typical goal of a machine learning system is to minimize non-computable expected risk by minimizing the computable empirical risk with the aim of obtaining low generalization gap [32]. The difference between empirical risk and expected risk is known as generalization gap. The generalization gap explains the dependency of a trained model on the unseen training dataset. There are several performance measurements have been proposed in the literature to measure the generalization capability of a neural network [33]. These measurements have the ability to control the generalization error. These are the primary candidate measurements that have been proposed in the recent literature:

- Model Complexity: This measure handles the generalization gap by decoupling the model function from training data by considering the worst-case generalization gap in the hypothesis space and by considering different quantities to characterize the set of model functions (such as Rademacher complexity and Vapnik-Chervonenkis (VC) dimension) [32] [33]
- Stability: This measurement deals with the dependence of the model function on the training dataset by considering the stability of the algorithm on different datasets. It measures the change of model output with respect to the change of data points in training dataset [32] [33] [34].
- Robustness: This measure avoids the dependence of the model on the training data, by considering the robustness of the algorithm for all possible datasets i.e, it measures the variation of the amount of loss w.r.t the input space [32] [35].

IV. Regularization

Many strategies have been proposed in the field of machine learning to achieve better generalization. These strategies collectively refer to the term ”regularization”. Regularization is any modification which we make to the algorithm so that it reduces the generalization error, not the training error [24].

Regularization can be categorized into two categories: Implicit and explicit. It has to be noticed that this categorization is subjective [36] [33]. Both control the effective capacity of the network with the purpose of reducing the overfitting. Most recent definitions of them are [36]:

- Explicit Regularization: Regularization methods which are not structural parts of the network architecture, the algorithms or the data and typically can be added or removed easily. Examples: Weight decay, Dropout, Data Augmentation, Stochastic depth [36].
- Implicit Regularization: These regularization methods use characteristics of the network architecture, the learning algorithm or the data in order to control the effective capacity of a neural network Examples: Stochastic

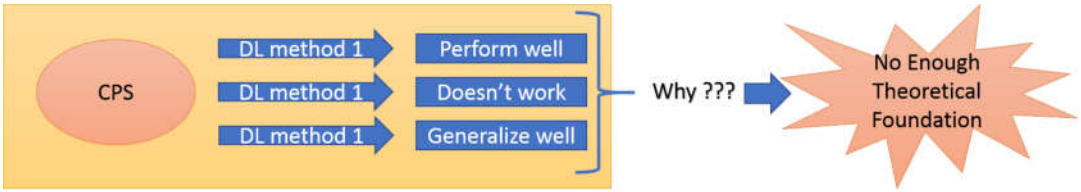

Fig. 3: Challenges of Applying Deep Leaning method in CPSs

Gradient Descent Algorithm, Convolution layers, Batch Normalization. [36]

Table I shows frequently used deep learning techniques for cyber-security and CPSs applications with their commonly used regularization methods. It has to be noticed that these regularization methods are not limited to one DL technique. Some regularization methods prefer some DL techniques based on their architecture and complementary advantages.

TABLE I: Deep Learning Techniques and Their Commonly used Regularization Methods

Deep Learning Technique	Regularization Methods
LSTM-RNN	• Drop Out • Mixed-Norm Regularization • Weight Sharing • DropConnect • L2 regularizer • Zoneout
CNN	• Drop Out • Weight Decay • Pooling • Drop Connect • Data Augmentation
RBM	• L2 Regularization • Weight Decay • Sparsity Regularization
DBN	• Drop Out • L1 Regularization • Sparsity Regularization
AE	• Weight Decay • Drop Out • Sparsity Regularization • Max-Pooling • Data Augmentation • On-line Gradient Decent
DFFNN	• Weight Decay • Dropout • Bayesian Regularization • Data Augmentation • Principal CA • Sparse Regularization

Table I shows that the most commonly used regularization methods are Dropout and Weight decay for the presented Deep Learning techniques. Regularization methods that are presented in Table I are discussed below:

- Weight decay / L2 regularization:
  Weight decay is a explicit regularization method which has the ability to constrain a network, thus capable of decreasing the complexity of a neural network [37]. It limits the growth of the weight, preventing the weights growing into too large values unless it is really necessary.
- Dropout:
  In [38], researcher have proposed this regularization method in order to address two major issues of large neural networks. They are: avoiding over-fitting and provide a way to combine many different neural network architectures efficiently. Dropout technique randomly drop units, i.e. temporarily removing them from the network with their incoming and outgoing connections during training process. This regularization method has proven to reduce over-fitting and to give major improvement over other regularization methods such as weight decay and data augmentation [38].
- Sparsity Regularization
  Classic Multilayer perceptron are composed by fully-connected layers, often also referred as dense layers given that every unit has an independent weighted connection with all units in the next layer. In contrast to dense layers, sparsity reefers to representations with most coefficients being zero. Deep learning often is designed with the objective of obtaining such sparse representations in its hidden layers. The idea of sparsity regularization is based on the assumption that the output of a model can be learned by reduced number of variables. Sparsity can be enforced using implicit regularization (e.g. using convolution layers) and explicit regularization (e.g. including a loss that penalizes non-zero weights like in sparse autoencoders) Dense connections often waste valuable resources, often adding capacity that is used inefficiently [39]. Enforcing sparsity is attractive given that it leads to a reduction on computational and memory requirements.
- Weight Sharing:
  In this method, a single weight is shared among many nodes in the neural network, i.e, groups of neuron nodes share common set of weight values where each group only processes only a local region of the input [40]. This technique implements receptive field in the network which makes the neural network models which are shift invariant which help with generalization capability of the model. Since neuron nodes are sharing weights, the number of weights in the neural network model is less than the total number of connections in the network. Therefore, it reduces the capacity/complexity of the network resulting better generalization [40].
- DropConnect:
  DropConnect is a recently introduced regularization method for regularizing large fully connected layers within neural networks. It has introduced as regularization of dropout in order to prevent the co-adaptation of feature detectors [41]. In DropConnect, randomly selected subset of weights within the network to zero. Therefore it introduces dynamic sparsity within the neural networks which helps with generalization ability of the model. Researchers has used this regularization technique on small datasets and has found that performance is sometimes better compared to dropout, but not always [42].
- Pooling:

Pooling is an operation used in almost all the convolution neural networks [24]. Pooling operations make output representations approximately invariant to small translations of the input image, i.e. translate the input by a small amount, the values of most of the pooled outputs do not change. It is performed by modifying the output of a neural network layer by replacing the output of a particular location in with a summery of the nearby outputs [24].

- Data Augmentation:
  One of the best ways to generalized model is by training it on more data [24]. In case of limited data, new data can be generated as fake data and add it to the training set [24]. This technique is called Data Augmentation. It is a regularization method which has shown the capability of achieving higher or same performance without any other explicit regularization techniques [24].
- Stochastic Gradient Decent (SGD): SGD is the most common optimization algorithm used to minimizes the objective function of a neural network [43]. SGD can be considered an implicit regularization method for DL models [44]. SGD methods often have these properties [45]: guarantees of convergence to minimizers of strong-convex functions [46] and to stationary points for non convex functions [47], saddle point avoidance and robustness to input data [48]. However, in [45], researchers have pointed out some limitation of SGD. They are: requires small batch sizes, sequential nature of the iteration and limited capability for parallelization. But there are some efforts that have proposed in literature to parallelize SAD for deep leaning applications [49] [44] [50] [45].
- Adversarial Training:
  In many cases, machine learning models are vulnerable to adversarial example, i.e. malicious inputs designed to fool the machine learning model [24] [51]. Adversarial training is the process of explicitly training a neural network on adversarial examples in order to make the model more robust to attack or to reduce the test error on clean input examples [24]. This idea have become popular in the context of regularization because it can reduce the error rate on the original test set via adversarial training [24] [51] [52].

V. Current Challenges

In this section, we briefly discuss the current challenges of machine learning community on choosing DL and regularization techniques for CPSs.

Understandability: one of the major problems with current DL approaches is lack of theoretical background [35]. Many researchers rely on empirical studies to show the impressive performance of DL methods without explaining why and how they generalize well. Since current DL approaches are not that transparent to users, their black-box behavior and lack of theoretical background reduce the human trust on DL approaches. Further, the lack of theoretical background has made it difficult to determine DL architectures, their hyper-parameters and proper regularization (generalization) techniques on them.

Regularization: Selecting a regularization technique is not a simple process when it comes to DL due to its lack of theoretical background [35]. For example, classic theories of machine learning have suggested that when the number of parameters are larger than the number of training samples, some form or regularization is required to ensure good generalization [33]. But deep neural networks have shown good generalization even with such over-parameterized settings [35].I.e, DL methods show good generalization even without any regularization technique. This makes the problem of selecting DL method or regularization method for a particular application is very difficult. Figure 3 illustrates the current view and challenges when applying DL for many applications.

*Selection:*The efficient use and development of GPUs have been increased the practicality of DL methods [36]. This has led many researchers to focus on training broader and deeper networks of larger capacity [53] [54] [36]. However, in [24], researchers have pointed out that network with large capacity reduce the practical usage of other explicit regularizations such as dropout and weight decay take longer training time [36]. Therefore researchers need to find the balance between implementing deeper architectures and practical usage of regularization methods. In [24], researchers have pointed out that there is no best machine learning algorithm or best form of regularization for a specific task that needed to be solved. Instead, it is essential to pick a form of regularization that is well suited for a particular task such that it will result in better generalization [24]. Therefore the selection of proper DL methods and appropriate regularization method is crucial to gain their optimal advantages.

Robustness: Deep learning models have been shown to be extremely sensitive to adversarial samples: samples with small perturbations that result in incorrect estimations with high confidence [55]. This is an excellent example of how the lack of understanding of DL models lead to unexpected vulnerabilities.

VI. Discussion and Future Directions

This paper briefly discussed regularization techniques that can be used to improve the generalization capabilities of DL based security applications of Cyber-Physical Systems. Attackers are becoming more smarter every day, making new system attacks and finding new system vulnerabilities. Therefore, the development of security related applications which are generalized to new system threats is essential. If we mention the conclusions of this paper lightly, in terms of generalization techniques for CPSs, we can say that the most popular DL techniques used in the domain of CPS security are LSTMs, CNNs, RBMs, DBNs, AE and DFFNN, most popular regularization techniques on them are dropout, weight decay and sparse regularization. But as we discussed in section V, selecting a regularization is not that straightforward.

Why deep learning for CPS: Deep Learning methods have been specifically designed to handle large datasets with a large

number of features. DL provides a rich class of models that can approximate any function. These attributes are desirable when applying DL methods in cyber-physical for the following reasons: 1) Data collected from CPSs is typically high dimensional as data coming from a large number of physical and cyber-sensors, 2) There is a constant growth of data due to improvements and exposure to new vulnerabilities 3) The models must be constantly updated with new data in order to account for drifting of the system and new vector attacks.

Challenges: Despite the generalization of DL models The lack of theoretical foundation in Deep Learning is one of the causes for the poor understanding of their generalization capabilities. Practitioners have to rely on trial and error without a clear path to improve the performance of the model. This is a huge issue when applying DL into critical systems such as CPSs. Currently, collecting more data (data augmentation) seems to be one of the promising approaches that guarantees an improvement of model generalization [36]. This is a huge drawback of using DL models in CPS given that collecting data may be expensive or even not possible because of physical constraints or safety concerns.

Future: In order to alleviate the challenges mentioned above, improving understandability with the help of explainable AI [56] will be a fundamental step for the application of DL models into CPS [57] [58]. The main problem with current DL approaches is that they are not transparent to the user, reducing the human trust on the system. The development of explainable AI will help to increase the transparency of ML models, eventually leading to an increase of generalization of ML models as well as human trust on DL methods.

Adversarial machine learning has shown to provide valuable insights into the vulnerabilities of ML models. It can be used to find vulnerabilities and week points of machine learning applications so that they can be used to increase the adaptability of the system for new cyber-physical data or new attacks, i.e., capable of increasing the generalization of models [24]. Explainable AI and adversarial ML provide tools that can be used to understand why a particular model is not working [59] [60]. This is an attractive approach for diagnosing and debugging DL models, providing the practitioner with insights on how to improve the generalization of the model.

Ultimately, a better theoretical background will be essential to increase understanding of DL models and provide clear guidelines to apply sound regularization methodologies for critical systems such as CPSs.

REFERENCES

[1] D. Kwon, H. Kim, J. Kim, S. C. Suh, I. Kim, and K. J. Kim, "A survey of deep learning-based network anomaly detection," *Cluster Computing*, Sep 2017. [Online]. Available: https://doi.org/10.1007/s10586-017-1117-8

[2] B. S. Sridhar, A. Hahn, and M. Govindarasu, "Cyber Physical System Security for the Electric Power Grid," vol. 100, no. 1, 2012.

[3] R. Raj, I. Lee, and J. Stankovic, "Cyber-Physical Systems : The Next Computing Revolution," pp. 731–736, 2010.

[4] A. C. Alvaro, "Challenges for Securing Cyber Physical Systems."

[5] J. Schmidhuber, "Deep Learning in neural networks: An overview," *Neural Networks*, vol. 61, pp. 85–117, 2015. [Online]. Available: http://dx.doi.org/10.1016/j.neunet.2014.09.003

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

[7] A. Buczak and E. Guven, "A survey of data mining and machine learning methods for cyber security intrusion detection," *IEEE Communications Surveys & Tutorials*, vol. PP, no. 99, p. 1, 2015.

[8] W. Wan, S. Mabu, K. Shimada, K. Hirasawa, and J. Hu, "Enhancing the generalization ability of neural networks through controlling the hidden layers," *Applied Soft Computing Journal*, vol. 9, no. 1, pp. 404–414, 2009.

[9] C. S. Wickramasinghe, K. Amarasinghe, and M. Manic, "Parallalizable deep self-organizing maps for image classification," *2017 IEEE Symposium Series on Computational Intelligence, SSCI 2017 - Proceedings*, vol. 2018-January, pp. 1–7, 2018.

[10] S. Urolagin, K. V. Prema, and N. V. S. Reddy, "Generalization Capability of Artificial Neural Network," pp. 171–178, 2012.

[11] V. Moyo and K. Sibanda, "Training Set Size for Generalization Ability of Artificial Neural Networks in Forecasting TCP/IP Traffic Trends," *International Journal of Computer Applications*, vol. 113, no. 13, pp. 975–8887, 2015.

[12] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle, "Greedy layer-wise training of deep networks," in *Proceedings of the 19th International Conference on Neural Information Processing Systems*, ser. NIPS'06. Cambridge, MA, USA: MIT Press, 2006, pp. 153–160. [Online]. Available: http://dl.acm.org/citation.cfm?id=2976456.2976476

[13] M. M. Najafabadi, F. Villanustre, T. M. Khoshgoftaar, N. Seliya, R. Wald, and E. Muharemagic, "Deep learning applications and challenges in big data analytics," *Journal of Big Data*, vol. 2, no. 1, p. 1, Feb 2015. [Online]. Available: https://doi.org/10.1186/s40537-014-0007-7

[14] I. Goodfellow, H. Lee, Q. V. Le, A. Saxe, and A. Y. Ng, "Measuring invariances in deep networks," in *Advances in Neural Information Processing Systems 22*, Y. Bengio, D. Schuurmans, J. D. Lafferty, C. K. I. Williams, and A. Culotta, Eds. Curran Associates, Inc., 2009, pp. 646–654. [Online]. Available: http://papers.nips.cc/paper/3790-measuring-invariances-in-deep-networks.pdf

[15] J. Goh, S. Adepu, M. Tan, and Z. S. Lee, "Anomaly Detection in Cyber Physical Systems Using Recurrent Neural Networks," *2017 IEEE 18th International Symposium on High Assurance Systems Engineering (HASE)*, pp. 140–145, 2017. [Online]. Available: http://ieeexplore.ieee.org/document/7911887/

[16] W. Hardy, L. Chen, S. Hou, Y. Ye, and X. Li, "Dl4md: A deep learning framework for intelligent malware detection." Athens: The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp), 2016, pp. 61–67, copyright - Copyright The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp) 2016; Document feature - Diagrams; Tables; Equations; Illustrations; Graphs; Last updated - 2016-07-25. [Online]. Available: http://proxy.library.vcu.edu/login?url=https://search.proquest.com/docview/1806428874?accountid=14780

[17] B. Kolosnjaji, A. Zarras, G. Webster, and C. Eckert, "Deep learning for classification of malware system call sequences," in *AI 2016: Advances in Artificial Intelligence*, B. H. Kang and Q. Bai, Eds. Cham: Springer International Publishing, 2016, pp. 137–149.

[18] N. Papernot, P. Mcdaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, "The limitations of deep learning in adversarial settings," *Proceedings - 2016 IEEE European Symposium on Security and Privacy, EURO S and P 2016*, pp. 372–387, 2016.

[19] M.-J. Kang and J.-W. Kang, "Intrusion Detection System Using Deep Neural Network for In-Vehicle Network Security," *Plos One*, vol. 11, no. 6, p. e0155781, 2016. [Online]. Available: http://dx.plos.org/10.1371/journal.pone.0155781

[20] A. Anwar and A. N. Mahmood, "Cyber security of smart grid infrastructure," *CoRR*, vol. abs/1401.3936, 2014. [Online]. Available: http://arxiv.org/abs/1401.3936

[21] M. Z. Mas'ud, S. Sahib, M. F. Abdollah, S. R. Selamat, and R. Yusof, "Analysis of Features Selection and Machine Learning Classifier in Android Malware Detection," *2014 International Conference on Information Science & Applications (ICISA)*, pp. 1–5, 2014. [Online]. Available: http://ieeexplore.ieee.org/document/6847364/

[22] B. Subba, S. Biswas, and S. Karmakar, "A Neural Network based system for Intrusion Detection and attack classification," *2016 Twenty Second National Conference on Communication (NCC)*, pp. 1–6, 2016. [Online]. Available: http://ieeexplore.ieee.org/document/7561088/

[23] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[24] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

[25] Y. LeCun and Y. Bengio, "Convolutional networks for images, speech, and time series," *The handbook of brain theory and neural networks*, vol. 3361, no. April 2016, pp. 255–258, 1995. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.32.9297{&}rep=rep1{&}type=pdf

[26] W. Wang, M. Zhao, and J. Wang, "Effective android malware detection with a hybrid model based on deep autoencoder and convolutional neural network," *Journal of Ambient Intelligence and Humanized Computing*, vol. 0, no. 0, p. 0, 123. [Online]. Available: https://doi.org/10.1007/s12652-018-0803-6

[27] B. Kolosnjaji, A. Zarras, G. Webster, and C. Eckert, "Deep Learning for Classification of Malware System Call Sequences," vol. 9992, pp. 137–149, 2016. [Online]. Available: http://link.springer.com/10.1007/978-3-319-50127-7

[28] W. Liu, Z. Wang, X. Liu, N. Zeng, Y. Liu, and F. E. Alsaadi, "A survey of deep neural network architectures and their applications," *Neurocomputing*, vol. 234, no. October 2016, pp. 11–26, 2017. [Online]. Available: http://dx.doi.org/10.1016/j.neucom.2016.12.038

[29] ——, "A survey of deep neural network architectures and their applications," *Neurocomputing*, vol. 234, pp. 11 – 26, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0925231216315533

[30] N. Gao, L. Gao, Q. Gao, and H. Wang, "An Intrusion Detection Model Based on Deep Belief Networks," *2014 Second International Conference on Advanced Cloud and Big Data*, pp. 247–252, 2014. [Online]. Available: http://ieeexplore.ieee.org/document/7176101/

[31] Y. Li, R. Ma, and R. Jiao, "A hybrid malicious code detection method based on deep learning," *International Journal of Security and its Applications*, vol. 9, no. 5, pp. 205–216, 2015.

[32] K. Kawaguchi, L. P. Kaelbling, and Y. Bengio, "Generalization in Deep Learning," 2017. [Online]. Available: http://arxiv.org/abs/1710.05468

[33] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, "Understanding deep learning requires rethinking generalization," 2016. [Online]. Available: http://arxiv.org/abs/1611.03530

[34] I. Kuzborskij and C. H. Lampert, "Data-Dependent Stability of Stochastic Gradient Descent," pp. 1–22, 2017. [Online]. Available: http://arxiv.org/abs/1703.01678

[35] B. Neyshabur, S. Bhojanapalli, D. McAllester, and N. Srebro, "Exploring Generalization in Deep Learning," no. Nips, 2017. [Online]. Available: http://arxiv.org/abs/1706.08947

[36] A. Hernández-garcía and P. König, "Data augmentation instead of explicit regularization," 2016.

[37] A. Krogh and J. A. Hertz, "A Simple Weight Decay Can Improve Generalization," *Advances in Neural Information Processing Systems*, vol. 4, pp. 950–957, 1992. [Online]. Available: http://0-citeseerx.ist.psu.edu.innopac.up.ac.za/viewdoc/summary?doi=10.1.1.41.2305

[38] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," *Journal of Machine Learning Research*, vol. 15, pp. 1929–1958, 2014.

[39] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

[40] S. J. Nowlan and G. E. Hinton, "Simplifying Neural Networks by Soft Weight-Sharing," *Neural Computation*, vol. 4, no. 4, pp. 473–493, 1992. [Online]. Available: http://www.mitpressjournals.org/doi/10.1162/neco.1992.4.4.473

[41] L. Wan, M. Zeiler, S. Zhang, Y. LeCun, and R. Fergus, "Regularization of neural networks using dropconnect," *Icml*, no. 1, pp. 109–111, 2013. [Online]. Available: http://machinelearning.wustl.edu/mlpapers/papers/icml2013{_}wan13

[42] E. A. Smirnov, D. M. Timoshenko, and S. N. Andrianov, "Comparison of regularization methods for imagenet classification with deep convolutional neural networks," *AASRI Procedia*, vol. 6, pp. 89 – 94, 2014, 2nd AASRI Conference on Computational Intelligence and Bioinformatics. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S2212671614000146

[43] L. Bottou, "LNCS 7700 - Stochastic Gradient Descent Tricks," pp. 421–436, 2012.

[44] D. Das, S. Avancha, D. Mudigere, K. Vaidyanathan, S. Sridharan, D. D. Kalamkar, B. Kaul, and P. Dubey, "Distributed deep learning using synchronous stochastic gradient descent," *CoRR*, vol. abs/1602.06709, 2016. [Online]. Available: http://arxiv.org/abs/1602.06709

[45] N. S. Keskar, D. Mudigere, J. Nocedal, M. Smelyanskiy, and P. T. P. Tang, "On Large-Batch Training for Deep Learning: Generalization Gap and Sharp Minima," pp. 1–16, 2016. [Online]. Available: http://arxiv.org/abs/1609.04836

[46] Y. Dauphin, R. Pascanu, Ç. Gülçehre, K. Cho, S. Ganguli, and Y. Bengio, "Identifying and attacking the saddle point problem in high-dimensional non-convex optimization," *CoRR*, vol. abs/1406.2572, 2014. [Online]. Available: http://arxiv.org/abs/1406.2572

[47] R. Ge, F. Huang, C. Jin, and Y. Yuan, "Escaping from saddle points - online stochastic gradient for tensor decomposition," *CoRR*, vol. abs/1503.02101, 2015. [Online]. Available: http://arxiv.org/abs/1503.02101

[48] M. Hardt, B. Recht, and Y. Singer, "Train faster, generalize better: Stability of stochastic gradient descent," *CoRR*, vol. abs/1509.01240, 2015. [Online]. Available: http://arxiv.org/abs/1509.01240

[49] S. Zhang, A. Choromanska, and Y. LeCun, "Deep learning with elastic averaging SGD," *CoRR*, vol. abs/1412.6651, 2014. [Online]. Available: http://arxiv.org/abs/1412.6651

[50] J. Dean, G. S. Corrado, R. Monga, K. Chen, M. Devin, Q. V. Le, M. Z. Mao, M. Ranzato, A. Senior, P. Tucker, K. Yang, and A. Y. Ng, "Large scale distributed deep networks," in *NIPS*, 2012.

[51] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. J. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," *CoRR*, vol. abs/1312.6199, 2013. [Online]. Available: http://arxiv.org/abs/1312.6199

[52] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2014, pp. 2672–2680. [Online]. Available: http://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf

[53] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *ArXiv e-prints*, Sep. 2014.

[54] S. Zagoruyko and N. Komodakis, "Learning to compare image patches via convolutional neural networks," *CoRR*, vol. abs/1504.03641, 2015. [Online]. Available: http://arxiv.org/abs/1504.03641

[55] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," *arXiv preprint arXiv:1607.02533*, 2016.

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

[57] D. Marino, M. Anderson, K. Kenney, and M. Manic, "Interpretable data-driven modeling in biomass preprocessing," in *2018 11th International Conference on Human System Interactions (HSI)*, July 2018.

[58] K. Amarasinghe and M. Manic, "Toward explainable deep neural network based anomaly detection," in *2018 11th International Conference on Human System Interactions (HSI)*, July 2018.

[59] W. Samek, T. Wiegand, and K. Müller, "Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models," *CoRR*, vol. abs/1708.08296, 2017. [Online]. Available: http://arxiv.org/abs/1708.08296

[60] D. L. Marino, C. S. Wickramasinghe, and M. Manic, "An adversarial approach for explainable ai in intrusion detection systems," *IECON 2018*, 2018.