APPLICATION OF DEEP LEARNING FOR CYBERSECURITY

ABHISHEK KUMAR[1]

[1]Affiliation not available

March 2, 2022

ABHISHEK KUMAR

er.abhisngh827@gmail.com

School of Computer Science and Engineering Galgotias University, Greater Noida, India

Abstract

Cutting edge Deep Learning (DL)methods are widely applied to areas like image processing and speech appreciation thus far. Likewise, some DL work has been wiped out in the world of cybersecurity. In this survey, we specialize in recent DL methods proposed within cybersecurity, namely intrusion detection, malware detection, phishing/spam detection, and website damage detection. First, initial classifications of popular DL models and algorithms are described. Then, a general framework for cybersecurity applications is planned and explained based on the four main modules. Afterward, related papers are brief and analyzed with the situation to the main target area, methodology, model applicability, and have e various cybersecurity requests using DL models.

Introduction

Today, the Net has become an indispensable need of everyone's life, making this general organized network prone to various threats. Several security threats exist in cyberspace, from jail-breaking to two-faced malware and network impositions. These threats have recognized an arms race in terms of security. Many securities corporations

Around the world are focusing on designing new skills to protect computer devices, networks, and software applications from network interruption attacks and malware infections. Two conventional security systems, network security systems, and host security systems protect the original network and computers from illegal access, destruction, fault, and modification. Both of those systems may contain different integrated security modules, like firewalls, Interruption Detection Systems (IDSs), and anti-viruses that help monitor a system or network and lift a fear when malicious activity happens. Interruption detection is believed to be a necessary security device to affect network attacks and identify malicious activities in network traffics. It plays a vital role in info security technology and helps define, determine, and identify illegal use, copying, change, and destruction of information and information systems.

Generally speaking, IDSs are categorized into three different methods: misuse detection, anomaly detection, and hybrid. Misuse detection methods use pre-defined signs of malicious activity to identify interruptions. Therefore, they have been used to detect known attacks only. On the other hand, anomaly detection methods define normal patterns and identify malicious activities based on deviations from usual designs. So, anomaly-based finding methods have the potential of detecting zero-day attacks. Hybrid methods cash in on both misuse finding and anomaly recognition methods. While reducing the false positives of unidentified attacks, mixed methods aim at growing recognition rates of known intrusions.

Malware has newly posed serious security issues and threats to cyberspace users, recognizing malware as the greatest concern. These intrusive software programs, like worms, viruses, trojans, botnets, ransomware, then on, are broken by fraud actors to arrange the change of security attacks against computer systems and jeopardize the privacy and integrity of connected data and, therefore, the availability of the services offered by the original organization.

Like Kaspersky, Symantec, Microsoft, McAfee, and Invicta, many vendors have developed anti-virus products to defend computers and bonafide users from malware attacks. These vendors normally use signature-based approaches to notice malware. Although signature-based methods are slightly operative, they're powerless to spot zero-day malware, which malware writers can obscure. The unparalleled volume of daily malware manufacture needs planning correct automatic systems to notice and classify malware.

In this survey, we study the application of DL to cybersecurity and several models that have been applied to the areas of intrusion detection, malware detection, phishing/spam detection, and website defacement detection within the literature. To the simplest of our knowledge, this is often the primary survey that presents an in-depth literature review of cybersecurity emphasizing well-known DL algorithm descriptions. Although some research has been conducted so far reviewing ML and Data Mining (DM)techniques for intrusion detection or malware detection.

Basic Concepts

This section presents a brief history of NN's and DL. Also, we provide a comprehensive overview of basic DL and DN architectures.

A brief history

The evolution of NN has involved many ups and downs since the mid20th century. The first inspirations go back to 1943, initiated by McCulloch and Pitts. They borrowed the thought from biological neurons and proposed a computational model for constructing hypothetical nets.

Later at IBM research laboratories, a hypothetical was simulated by Nathaniel Rochester, which was not a successful attempt. Next, in 1958, perceptron was advanced at Cornell Aircraft Laboratory by Frank Rosenblatt, which was the first knowledge contraption.

Today, DL is working dominantly in almost every application. It is simply a new classical Multilayer Perceptron (MLP) variation. The goal of DL algorithms is to supply high-level and versatile features from the raw input file that help generalization within the classification. DL addresses compound requests with millions of data that need a large number of neurons and hidden layers. To this end, countless DL frameworks, like TensorFlow, Caffe, or Theano, are advanced in new years that offer the building blocks for applying DN constructions professionally and eliminate the need for coding from scrape.

Deep learning in a nutshell

DL allows computational illustrations composed of multiple processing layers to find out knowledge images with various concept levels. DL constructions are usually created as multilayer networks so that more abstract features are calculated as nonlinear purposes of lower-level components. The most current kinds of DL models are CNN, RNN, and DBN, which have been generally applied to large-scale image appreciation tasks, natural language processing, bioinformatics, and speech appreciation, to name a few.

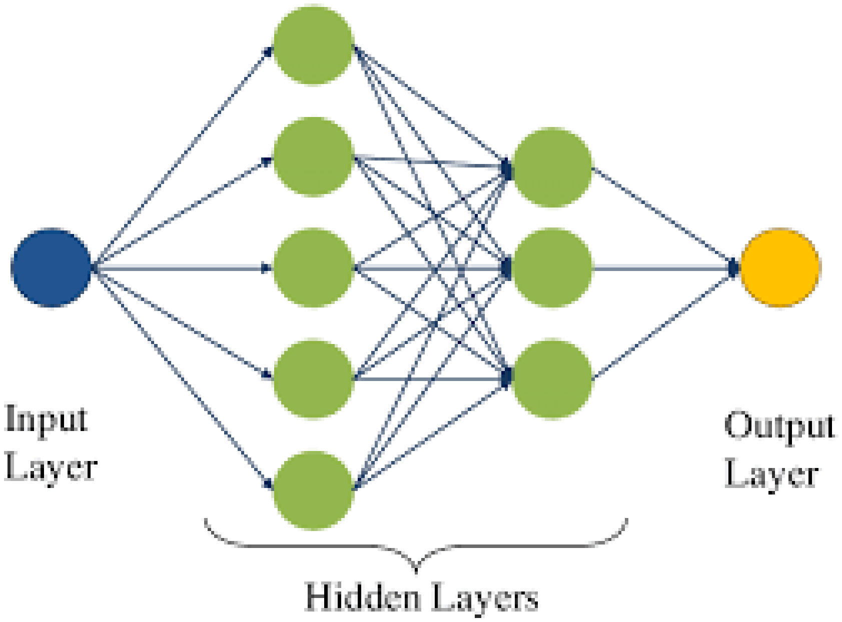

Fig.1 A DNN with two hidden layers

Feature engineering is the main difference between traditional ML and DL algorithms. While standard ML algorithms handcraft the features, DL algorithms aim at extracting the features automatically, leading to more accurate ML models. DL techniques can address large-scale data problems using many processing layers compared with the shallow ML algorithms.

One of the primary mutual methods of DL construction is FFNN, a.k.a Deep Neural Network (DNN). A DNN typically contains an input layer tracked by several hidden layers and an output layer. The input layer receives an input feature vector on behalf of the thing to be confidential. The output layer is liable for producing the group prospect vector related to the input vector.

Fig.1 shows a characteristic DNA with two hidden layers. Each node in the classical, temporary as a neuron, consumes the output of the last layer plus a bias from a singular neuron. It then computes a weighted average of the total information mentioned by its inputs. For each hidden unit j, the output yj produces a numerical output Yj = f (ni=1wjixi + bj) where bj is the bias term, and wji are the elements of a layer's weight matrix. Presenting non-linearities to the DNN model, the purpose f(.)is called the beginning function that describes the output of a hidden unit.

Deep network architectures

This subsection explains some state-of-the-art DN architectures that have attracted much attention among researchers because of their incomparable accuracy in modelling and classifying complex data.

3. Deep learning framework for cybersecurity applications

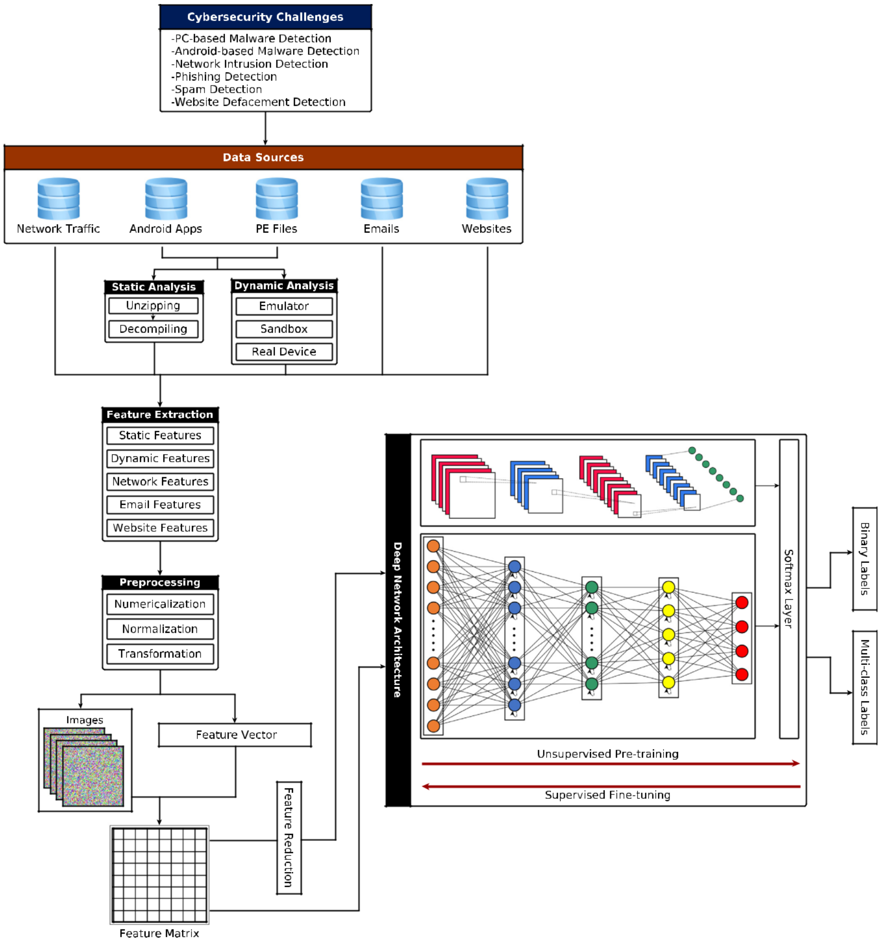

Fig. 2 Conceptual DL framework for cybersecurity applications.

In this section, we current an overall DL framework for cybersecurity requests maintained by the papers we've scrutinized during this survey. The framework is taken into account to be as generic as possible to hide various cybersecurity challenges, including interruption discovery, malware discovery and analysis, phishing/spam finding, and website destruction detection. The conceptual model of the proposed framework is illustrated in Fig. 2. The problematic domain justifies the type of data source we select for our framework. If we aim to detect or classify malware, dependent on the structure, the mobile malware, or the PC malware, we would have an Android request package, i.e.,.apk or Portable Executable (PE)files as the input data.

On the other hand, the input file would be the network traffic records if we try to find intrusion detection applications. The network traffic input data are either coming from publicly available intrusion detection datasets, such as KDD'99 [and NSL-KDD, or real-world traffic captured in the form of cap files. E-mails are another data source for a phishing e-mail or spam recognition, and websites are the input file for phishing website and website defacement recognition. The general framework contains four major modules: analysis, feature removal, pre-processing, and DL-based classifier.

3.1. Analysis

The framework workflow starts by analyzing .apk files or PE files during a static or dynamic mode. The input data is first decompressed to abstract matching features in the stationary analysis phase. Suppose the input data are apk. They are open to nearly original form counting Machine apparent (*.XML) and classes. Dex files. The dex files are stripped to produce the Dalvik VM assembly code *. XML files are parsed to extract meta-data information, like the package name, permissions, libraries to be linked, and components definitions. The PE file is from a non-mobile atmosphere like Windows and is typically compressed by a binary density tool like UPX or AS Pack Shell, so first, it should be discharged.PE file is then decompiled into its matching assembly code. The dynamic analysis phase executes the .apk or PE file in an emulator, sandbox, or a real device, i.e., smartphone or PC. At the same time, the required information like network packets, Application Programming Interface (API)call, or system call traces is recorded. Communication with the request is also required to mimic the real atmosphere for the proposal, which is either conducted by users or done automatically using ADT Monkey, for example.

3.2. Feature extraction

In the feature abstraction unit, the standing features, the production of static analysis, are removed from decoded resources. Static features include API calls, strings, URL-based features, raw opcode sequences, file-related features, authorizations, intents, suspicious calls, app components, to name a few. The dynamic characteristics are extracted from the log files or the cap files created and captured during the execution of the applications in the underlying environment. Functional features include but are not limited to API call traces, system call sequences, domain-based elements, machine activity, network traffic, file creation, deletion, and registry keys are written. The network feature extractor may be a flow-based feature extractor that will extract network traffic features from a pcap file.

4. Deep learning-based cybersecurity-related work

Scholars have recently planned several methods that have practical DL algorithms to detect or classify malware, detect network intrusions and phishing/spam attacks, and inspect website mutilations. In this section, we review these studies in three main groups: malware recognition and analysis, intrusion detection, and other, which include phishing detection, spam detection, and website disfigurement recognition. Summarizes the most branches of applying DL to cybersecurity.

4.1. Malware detection and analysis

Normally speaking, malware recognition methods are classified into three groups:(a) static, (b)dynamic, and (c)hybrid. Static methods disassemble and analyze the source code without executing it. Though they are fast, they suffer from constructing high false-positive rates. In calculation, they fail in contradiction of the detection of complicated malware. Dynamic analysis techniques monitor the connections of the executed code in a virtual atmosphere and address malware confusion. At the same time, intense lots of time and memory resources, whereas hybrid methods employ the returns of both static and dynamic ways.

4.2. Phishing detection

Taking working some of the basic structures, such as operational features, link structures part structures, and word list features that capture the characteristics of phishing e-mails, Zhang et al. Aimed at detecting phishing e-mail attacks done a 3-layer FFNN. The planned FFNN contains 1 input layer, one hidden layer, and one output layer, and so the number of neurons within the hidden layer is learned by testing different sceneries. Toft the used dataset, tanh and sigmoidal are used as beginning functions, and Resilient Broadcast (RPROP)training is used to train the FFNN. To the new assessments, a real dataset of4,202 ham e-mails and 4,560 phishing e-mails are used. A pre-processing stage is run to conduct the trial to abstract the structures mentioned above from the e-mails using Perl scripts and control the dataset between the range. Finally, the dataset is trained using the exercise set to urge the parameter estimations then verified on the testing set to estimate the presentation using cross-validation. This process is repeated 20 times for many sizes of the exercise and trying datasets. Once the assessment metrics are calculated, the results are related to NN settings, i.e., the number of units in the hidden layer and beginning functions.

Moreover, the presentation of the NN is related to that of other well-known ML algorithms, such as DT, k-NN, NB, and SVM, getting 95.51% truth and 95.71% F1 score for NN. From the arithmetical analysis, we can arrange that the NN offers a useful fact even when the training examples are scarce. However, the authors have not examined the effect of adding more hidden layers to the FFNN.

4.3. Spam detection

The increasing number of spam messages sent daily resulted in designing many anti-spam filters. Many ML and a few DL techniques have been employed to improve e-mail spam detection. RBM has shown to be effective in this area, though fine-tuning its parameters is a big challenge. Da Silva et al. Presented an approach to learn the intrinsic features of e-mail messages by RBMs to identify malicious or benign content. To adjust the RBMs boundaries, Harmony Search-based Optimization technique was employed to gauge the parameters' robustness within the context of spam detection. The RBM parameters are learning rate, weight decay, penalty parameter, and the number of hidden units. The extracted features are then fed into the OPF classifier to evaluate the model's accuracy. OPF algorithm employs the path-cost function for estimating prototypes, i.e., key samples that best represent the classes.

4.4. Website defacement detection

Bergolte et al. Addressed website defacement as a disruptive attack that may cause serious financial damage to companies and organizations and ruin their reputation. They proposed MEERKAT as a monitoring system that combines SAEs and DNNs to identify defacements. Getting help from the screenshot regions (windows)of the websites, MEERKAT automatically learns high-level features from the visual representation. Unlike previous approaches, it relies neither on additional information provided by the website's operator like its source code, content, or structure nor on manually-crafted features. Still, it only requires the URL of the website. Applying MEERKAT on the largest website

5. Analysis and discussion

This section analyzes all studies concentrating on emerging areas of DL and cybersecurity, intrusion detection, and malware detection/analysis from four different aspects: focus area, methodology, model applicability, and feature granularity.

5.1. Focus area

All related work focuses on detection, classification, or analysis. These studies detect PC/Android malware and categorize them into miscellaneous families using static, dynamic, or hybrid techniques. However, a few malware-related studies concentrate on efficient malware analysis only. Another group of studies detect intrusion attacks and classify them into different attack types.

5.2. Model applicability

DN architectures have been used in two different ways, either as a dimensionality reduction method before applying a classifier, such as SVM, LR, NB, and DT or as a classifier by itself. Some of the DN models are used for both tasks. DBN and SAE, for example, could be employed for both feature reduction and classification tasks.

5.3. Feature granularity

Typically, DNNs use several hidden layers to learn a high-level representation of the input features hierarchically. Suppose that we intend to classify a dog image. In the first layer, we might detect the edges of the dog. The second layer might see curves associated with the dog image. Finally, m might detect the whole dog image in the third layer. Practically, the DL model does here bridge the gap between high-level representation and low-level features.

6. Conclusion, concerns, and open directions

In this survey, we presented the history of NN and DL by focusing on the basics of DL models and algorithms. We further proposed a general DL framework for cybersecurity applications and elaborated its four major modules, including analysis, feature extraction, pre-processing, and DL-based classifier. Discussed a series of related studies in intrusion detection, malware detection, phishing/spam detection, and website defacement detection using DL methods along with their achievements and limitations. All intrusion and malware-related studies were further analyzed and compared from four perspectives: focus area, methodology, model applicability, and feature granularity. Categorizing the DN architectures into three generative, discriminative, and hybrid classes, a taxonomy was presented for the types of deep models falling into each category. Concerning public desire towards DL applications to cybersecurity, there are still many open directions for progress. DL today is mostly about purely supervised learning. A major drawback of supervised learning is that it requires many labelled data, and it is quite expensive to collect them. So, DL is expected to be unsupervised more human-like in the future.

References

[1] A.L.Buczak, E.Guven, A survey of data mining and machine learning methods for cyber security intrusion detection, IEEE Communications Surveys & Tutorials 18 (2)(2016)1153- 1176.

[2] P.Faruki, A.Bharmal, V.Laxmi, V.Ganmoor, M.S.Gaur, M.Conti, M.Rajarajan, Android security: A survey of issues, malware penetration, and defenses, IEEE Communications Surveys & Tutorials 17 (2)(2015)998–1022,

[3] Y.Ye, T.Li, D.Adjeroh, S.S.Iyengar, A survey on malware detection using data mining techniques, ACM Computing Surveys (CSUR)50 (3)(2017)1–40,

[4] D.Kwon, H.Kim, J.Kim, S.C.Suh, I.Kim, K.J.Kim, in A survey of deep learning-based network anomaly detection, Cluster Computing,2017, pp.1–13,

[5] W.S.McCulloch, W.Pitts, A logical calculus of the ideas immanent in nervous activity, The Bulletin of Mathematical Biophysics 5 (4)(1943)115–133,

[6] F.Rosenblatt, The perceptron: A probabilistic model for information storage

and organization in the brain, Psychological Review 65 (6) (1958)65–386,

[7] C.Van Der Malsburg, Frank Rosenblatt: Principles of neurodynamics: Perceptron's and the theory of brain mechanisms, in: Brain Theory,Springer,1986, pp.245–248

[8] S.Dreyfus, The computational solution of optimal control problems with time lag, IEEE Transactions on Automatic Control18 (4)(1973)383–385