© 2018 IJSRSET | Volume 4 | Issue 8 | Print ISSN: 2395-1990 | Online ISSN : 2394-4099
Themed Section : Engineering and Technology

A Survey on Network Intrusion Detection

K.Veena
PG Scholar, Akshaya College of Engineering and Technology, Coimbatore, Tamil Nadu, India

ABSTRACT

Network security is any activity designed to protect the usability and integrity of your network and data. It includes both hardware and software technologies. Effective network security manages access to the network. It targets a variety of threats and stops them from entering or spreading on your network. Network security combines multiple layers of defenses at the edge and in the network. Each network security layer implements policies and controls. Authorized users gain access to network resources, but malicious actors is blocked from carrying out exploits and threats. Two types of network includes wired and wireless network. The common vulnerability that exists in both wired and wireless networks is an “unauthorized access” to a network. An attacker can connect his device to a network though unsecure hub/switch port. In this regard, wireless network are considered less secure than wired network, because wireless network can be easily accessed without any physical connection. Network security is a big topic and is growing into a high profile Information Technology (IT) specialty area. Security-related websites are tremendously popular with savvy Internet users. The popularity of security-related certifications has expanded. Esoteric security measures like biometric identification and authentication have become commonplace in corporate America. Many organizations still implement security measures in an almost haphazard way, with no well-thought out plan for making all the parts fit together. Computer security involves many aspects, from protection of the physical equipment to protection of electronic bits and bytes that make up the information that resides on the network.

Keywords : Unauthorized Access, Savvy Internet Users, Intrusion Detection System, NIDS, HIDS

I. INTRODUCTION

INTRUSION DETECTION SYSTEM

An intrusion detection system (IDS) monitors network traffic and monitors for suspicious activity and alerts the system or network administrator. In some cases, the IDS may also respond to anomalous or malicious traffic by taking action such as blocking the user or source IP address from accessing the network. IDS come in a variety and its approach in detecting suspicious traffic in different ways.

There are network based (NIDS) and host based (HIDS) intrusion detection systems. There are IDS that detect based on looking for specific signatures of known threats- similar to the way antivirus software typically detects and protects against malware- and there are IDS that detect based on comparing traffic patterns against a baseline and looking for anomalies. There are IDS that simply monitor and alert and there are IDS that perform an action or actions in response to a detected threat.

1.1 NIDS CHALLENGES

Network monitoring has been used extensively for the purposes of security, forensics and anomaly detection. However, recent advances have created many new obstacles for NIDSs. Some of the most pertinent issues include:

1) Volume - The volume of data both stored and passing through networks continues to increase. It is forecast that by 2020, the amount of data in existence will top 44 ZB [4]. The traffic capacity of modern networks has drastically increased to facilitate the volume of traffic observed. Many modern backbone links are now operating at wirespeeds of 100 Gbps or more. To contextualise the above issue, a 100 Gbps link is capable of handling 148,809,524 packets per second [5]. A NIDS would need to be capable of completing the analysis of a packet within 6.72 ns for operating at wire speed. NIDS at such a speed is difficult and ensuring satisfactory levels of accuracy, effectiveness and efficiency also presents a significant challenge.

2) Accuracy - To maintain the aforementioned levels of accuracy, existing techniques cannot be relied upon. Therefore, greater levels of granularity, depth and contextual understanding are required to provide a more holistic and accurate view. It comes with various financial, computational and time costs.

3) Diversity - Recent years have seen an increase in the number of new or customised protocols being utilised in modern networks. This can be partially attributed to the number of devices with network and/or Internet connectivity. As a result, it is becoming increasingly difficult to differentiate between normal and abnormal traffic and behaviours.

4) Dynamics - Given the diversity and flexibility of modern networks, the behaviour is dynamic and difficult to predict. It leads to difficulty in establishing a reliable behavioural norm. It also raises concerns as to the lifespan of learning models.

5) Low-frequency attacks - These types of attacks have often thwarted previous anomaly detection techniques, including artificial intelligence approaches. The problem stems from imbalances in the training dataset, meaning that NIDS offer weaker detection precision when faced with these types of low frequency attacks.

6) Adaptability - Modern networks have adopted many new technologies to reduce their reliance on static technologies and management styles. Therefore, it becomes more widespread usage of dynamic technologies such as containerisation, virtualisation and Software Defined Networks. NIDSs will need to be able to adapt to the usage of such technologies and the side effects it bring about.

1.2 DEEP LEARNING

Deep learning is an advanced sub-field of machine learning, which advances Machine Learning closer to Artificial Intelligence. It facilitates the modelling of complex relationships and concepts [6] using multiple levels of representation. Supervised and unsupervised learning algorithms are used to construct successively higher levels of abstraction, defined using the output features from lower levels [7].

1) Auto-Encoder: A popular technique currently utilised within deep learning research is auto-encoders, which is utilised by our proposed solution An auto encoder is an unsupervised neural network-based feature extraction algorithm, which learns the best parameters required to reconstruct its output as close to its input as possible. One of it desirable characteristics is the capability to provide more a powerful and non-linear generalisation than Principle Component Analysis (PCA). This is achieved by applying back propagation and setting the target values to be equal to the inputs. In other words, it is trying to learn an approximation to the identity function. An auto-encoder typically has an input layer, output layer (with the same dimension as the input layer) and a hidden layer. This hidden layer normally has a smaller dimension than that of the input (known as an undercomplete or sparse auto-encoder). Most researchers [8]–[10] use auto-encoders as a non-linear transformation to discover interesting data structures, by imposing other constraints on the network and compare the results with those of PCA (linear transformation). These methods are based on the encoder-decoder paradigm. The input is first transformed into a typically lower-dimensional space (encoder) and then expanded to reproduce the initial data (decoder). Once a layer is trained, its code is fed to the next, to better model highly non-linear

dependencies in the input. This paradigm focuses on reducing the dimensionality of input data. A special layer - the code layer [9], at the centre of the deep auto-encoder structure is used to reduce dimensionality. The code layer is used as a compressed feature vector for classification or for combination within a stacked auto-encoder [8]. The hidden layer is used to create a lower dimensionality version of high dimensionality data (known as encoding). By reducing the dimensionality, the auto-encoder is forced to capture the most prominent features of the data distribution. In an ideal scenario, the data features generated by the auto-encoder will provide a better representation of the data points than the raw data itself. The aim of the auto-encoder is to try and learn the function shown in (1).

$$hW,b\ (x) \approx x\ (1)$$

where h = non-linear hypothesis using the parameters
W=weighting
b =bias
x=given data.

The learning process is described as a reconstruction error minimisation function, as shown in (2).

$$L(x, d(f(x)))\ (2)$$

where L = loss function penalising d(f(x)) for being dissimilar to x,
d is a decoding function and
f is an encoding function.

2) Stacked Auto-Encoder: Unlike a simple auto-encoder, a deep auto-encoder is composed of two symmetrical deep-belief networks, which typically have four or five shallow layers for encoding, and a second set of four or five layers for decoding. The work by Hinton and Salacukhudinov [9] has produced promising results by implementing a deep learning algorithm to convert high dimensional data to low dimensional data by utilising a deep auto-encoder. Deep learning can be applied to auto-encoders, whereby the hidden layers are the simple concepts and multiple hidden layers are used to provide depth, in a technique known as a stacked auto-encoder. This increased depth can reduce computational costs and the amount of required training data, as well as yielding. greater degrees of accuracy [6]. The output from each hidden layer is used as the input for a progressively higher level. Hence, the first layer of a stacked auto-encoder usually learns first order features in raw input. The second layer usually learns second-order features relating to patterns in the appearance of the first-order features. Subsequent higher layers learn higher order features. An illustrative example of a stacked auto-encoder is shown in Fig. 2. Here, the superscript numbers refer to the hidden layer identity and the subscript numbers signify the dimension for that layer.

The findings from our literature review have shown that despite the high detection accuracies being achieved, there is still room for improvement. Such weaknesses include the reliance on human operators, long training times, inconsistent or average accuracy levels and the heavy modification of datasets (e.g. balancing or profiling). The area is still in an infantile stage, with most researchers still experimenting on combining various algorithms (e.g. training, optimisation, activation and classification) and layering approaches to produce the most accurate and efficient solution for a specific dataset. Hence, we believe the model and work presented in this paper will be able to make a valid contribution to the current pool of knowledge.

II. LITERATURE SURVEY

Deep learning is garnering significant interest and its application is being investigated within many research domains, such as: healthcare [11], [12]; automotive design [13], [14]; manufacturing [15] and law enforcement [16], [17]. There are also several existing works within the domain of NIDS.

Comparison Deep Learning Method to Traditional Methods Using for Network Intrusion Detection

Dong and Wang undertook a literary and experimental comparison between the use of specific traditional NIDS techniques and deep learning methods. Deep learning has gained prominence due to the potential it portends for machine learning. Deep learning techniques have been applied in many fields such as recognizing some kinds of patterns or classification. Intrusion detection analyses and get data from monitoring security events to get situation for assessment of network. Lots of traditional machine learning method has been put forward to intrusion detection and it is necessary to improvement the detection performance and accuracy. The current idea discusses different methods which were used to classify network traffic. The above approach use different methods on open data set and did experiment with those methods to find out a best way to intrusion detection.

Deep Learning in Network Security

Traffic identification is a key component in network security since it raises the red flag in case of intrusion into the network. The system has relied on traditional methods of detection that are increasingly becoming ineffective due to the commensurate increase in data. Traditional approaches include port identification for instance, standard HTTP that is failing to perform as envisaged due to less protocols following the system. Another system involves the signature-based method that relies on the payload data was used in several applications.

Detecting intrusion into the system is always the challenge of differentiating between normal and abnormal data on the system. The detection approach should define the characteristics of malicious data on the system. Further, the technique should be able to design a classification system that is able to differentiate between the two sets of information accurately that is genuine and malicious data. This system is known as dimensionality reduction technique that uses automatic encoding approaches to calculate the distance between nodes within the network. Importantly, the technique works on the assumption that the normality of the data is determined by consistency of distance between the nodes. As such, the longer the distance between the nodes is indicative of the abnormality of the information thus acting as a pointer to the presence of malicious data. In relation to this, two measurement systems exist that are Manhattan distance, which is the total distance between the dimensions within the network and the Euclidean distance that primarily is the size of the vector being assessed.

Deep Learning through Analysis of Data Patterns in Network Security

The growth of the information technology field has necessitated the need for newer and better methods of analyzing how these computer systems operate. Several methods in machine learning exist that try investigate the principles behind the devices. The field of deep learning is dynamic due to the development of new techniques in several sub branches that include image recognition, computer security, and speech recognition. The classical methods of deep learning used in network security are increasingly failing to detect intrusions into network systems due to the commensurate increase in data production. As such, big data analysis using deep belief system is the latest innovation that tries to study information patterns with a view of detecting unauthorized entry into computer networks.

The use of deep learning has in recent times gained prominence due to its effectiveness in evaluating network security. The system has enabled the exhaustive and conclusive assessment of network security. Traditional methods of network security are increasingly failing to function effectively due to increased processing of data. Deep learning has revolutionized the evaluation of challenges in network security. The system uses several approaches to detect abnormalities in the system that include anomaly detection and traffic identification. The system faces certain limitations that include sanctity of data used to generate inputs and outputs. Similarly, new methods of deep learning are gaining traction due demand for faster and efficient data assessment. Deep belief and deep coding techniques have enabled the analysis of large data sets and deeper system analysis respectively.

Deep Learning and Its Applications to Machine Health Monitoring: A Survey

Deep learning (DL) has become a rapidly growing research direction, redefining state-of-the-art performances in a wide range of areas such as object recognition, image segmentation, speech recognition and machine translation. In modern manufacturing systems, data-driven machine health monitoring is gaining in popularity due to the widespread deployment of low-cost sensors and their connection to the Internet. Deep learning provides useful tools for processing and analyzing these big machinery data. The main purpose of the method is to review and summarize the emerging research work of deep learning on machine health monitoring. The applications of deep learning in machine health monitoring systems are reviewed mainly from the following aspects: Autoencoder (AE) and its variants, Restricted Boltzmann Machines and its variants including Deep Belief Network (DBN) and Deep Boltzmann Machines (DBM), Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN). Finally, some new trends of DL-based machine health monitoring methods are discussed.

Deep Learning

As a feed-forward neural network, auto-encoder consists of two phases including encoder and decoder. Encoder takes an input x and transforms it to a hidden representation h via a non-linear mapping as follows:

$h = \phi(Wx + b)$

where ϕ is a non-linear activation function. Then, decoder maps the hidden representation back to the original representation in a similar way.

Model parameters are optimized to minimize the reconstruction error between $z = f\theta(x)$ and x. One commonly adopted measure for the average reconstruction error over a collection of N data samples is squared error and the corresponding optimization problem. The hidden representation h can be regarded as a more abstract and meaningful representation for data sample x. The hidden size should be set to be larger than the input size in AE, which is verified empirically. The above method prevent the learned transformation is the identity one and regularize auto-encoders, the sparsity constraint is imposed on the hidden units in addition to sparsity.

Addition of Denoising: Different from conventional AE, denoising AE takes a corrupted version of data as input and is trained to reconstruct/denoise the clean input x from its corrupted sample $x^{\sim}$. The most common adopted noise is dropout noise/binary masking noise, which randomly sets a fraction of the input features to be zero. The variant of AE is denoising auto-encoder (DA), which can learn more robust representation and prevent it from learning the identity transformation.

Stacking Structure

Several DA can be stacked together to form a deep network and learn high-level representations by feeding the outputs of the l-st layer as inputs to the (l + 1)-th layer. The training will be done for one layer greedily at a time.

Convolutional Neural Network

Convolutional neural networks (CNNs) were firstly proposed by LeCun for image processing, which is featured by two key properties: spatially shared weights and spatial pooling. CNN models have shown their success in various computer vision applications where input data is usually 2D data. CNN has also been introduced to address sequential data including Natural Language Processing and Speech Recognition. CNN aims to learn abstract features by alternating and stacking convolutional kernels and pooling operation. In CNN, the convolutional layers (convolutional kernels) convolve multiple local filters with raw input data and generate invariant local features and the subsequent pooling layers extract most significant features with a fixed-length over sliding windows of the raw input data. 2D-CNN have been illustrated extensively in previous research compared to 1D-CNN. Only mathematical details are behind 1D-CNN.

A systematic overview of the state-of-the-art DL-based MHMS. Deep learning, as a subfield of machine learning, is serving as a bridge between big machinery data and data-driven MHMS. Deep learning have been applied in various machine health monitoring tasks within past four years. The proposed DL-based MHMS are summarized according to four categories of DL architecture as: Auto-encoder models, Restricted Boltzmann Machines models, Convolutional Neural Networks and Recurrent Neural Networks. Since the momentum of the research of DL-based MHMS is growing fast, the messages about the capabilities of these DL techniques, especially representation learning for complex machinery data and target prediction for various machine health monitoring tasks, can be conveyed to readers. It can be found that DL-based MHMS do not require extensive human labor and expert knowledge, i.e., the end-to-end structure is able to map raw machinery data to targets. Therefore, the application of deep learning models are not restricted to specific kinds of machines, which can be a general solution to address the machine health monitoring problems

Toward an Online Anomaly Intrusion Detection System Based on Deep Learning

Alrawashdeh and Purdy proposed a method called RBM with one hidden layer to perform unsupervised feature reduction.

The progress in intrusion detection has been steady but slow for past twenty years. The biggest challenge is to detect new attacks in real time. A deep learning approach for anomaly detection using a Restricted Boltzmann Machine (RBM) and a deep belief network are implemented. Our method uses a one-hidden layer RBM to perform unsupervised feature reduction. The resultant weights from this RBM are passed to another RBM producing a deep belief network. The pre-trained weights are passed into a fine tuning layer consisting of a Logistic Regression (LR) classifier with multi-class soft-max datas. The deep learning architecture was introduced in C++, Microsoft Visual Studio 2013 using DARPA KDDCUP'99 dataset to evaluate its performance. The proposed architecture outperforms previous deep learning methods implemented by Li and Salama in both detection speed and accuracy. We achieve a detection rate of 97.9% on the total 10% KDDCUP'99 test dataset. By improving the training process of the simulation, low false negative rate of 2.47% is produced. Although the deficiencies in the KDDCUP'99 dataset are well understood, it still presents machine learning approaches for predicting attacks with a reasonable challenge. The future work will include application of machine learning strategy to larger and more challenging datasets, which includes larger classes of attacks.

The weights are passed to another RBM to produce a DBN. The pre-trained weights are passed into a fine tuning layer consisting of a Logistic Regression classifier (trained with 10 epochs) with multi-class soft-max. The proposed solution was evaluated using the KDD Cup '99 dataset. The authors claimed a detection rate of 97.90% and a false negative rate of 2.47%. This is an improvement over other results.

Method of intrusion detection using deep neural network

The work by Kim et al. [19] aspired to specifically target advanced persistent threats. An artificial intelligence (AI) intrusion detection system using a deep neural network (DNN) was investigated and tested with the KDD Cup 99 dataset in response to ever-evolving network attacks in his method. First, the data were preprocessed through data transformation and normalization for input to the DNN model. The DNN algorithm was applied to the data refined through preprocessing to create a learning model and the entire KDD Cup 99 dataset was used to verify it. Finally, the accuracy, detection rate and false alarm rate were calculated to ascertain the detection efficacy of the DNN model which was found to generate good results for intrusion detection.

The new proposed system was Deep Neural Network (DNN) using 100 hidden units, combined with the Rectified Linear Unit activation function and the ADAM optimiser. The approach was implemented on a GPU using TensorFlow and evaluated using the KDD data set. The authors claimed an average accuracy rate of 99% and summarised that both RNN and Long Short-Term Memory (LSTM) models are needed for improving future defences.

A Deep Learning Approach for Network Intrusion Detection System

A Network Intrusion Detection System (NIDS) helps system administrators to detect network security breaches in their organizations. However, many challenges arise while developing a flexible and efficient NIDS for unforeseen and unpredictable attacks. We propose a deep learning based approach for developing such an efficient and flexible NIDS. Self-taught Learning (STL), a deep learning based technique on NSL-KDD - a benchmark dataset for network intrusion is used.

Self-Taught Learning

Self-taught Learning (STL) is a deep learning approach that consists of two stages for the classification. First, a good feature representation is learnt from a large collection of unlabeled data xu, termed as Unsupervised Feature Learning (UFL). In the second stage, this learnt representation is applied to labeled data xl and used for the classification task. Although the unlabeled and labeled data may come from different distributions, there must be relevance among them. There are different approaches used for UFL such as Sparse Autoencoder, Restricted Boltzmann Machine (RBM), K-Means Clustering and Gaussian Mixtures. Sparse autoencoder based feature learning is used for the work due to its relatively easier implementation and good performance. A sparse autoencoder is a neural network consists of an input, a hidden and an output layers. The input and output layers contain N nodes and the hidden layer contains K nodes.

NSL-KDD Dataset

The dataset is an improved and reduced version of the KDD Cup 99 dataset [20] is used. The KDD Cup dataset was prepared using the network traffic captured by 1998 DARPA IDS evaluation program [22]. The network traffic includes normal and different kinds of attack traffic such as DoS, Probing, user-to-root (U2R) and root-to-local (R2L). The network traffic for training was collected for seven weeks followed by two weeks of traffic collection for testing in raw tcpdump format. The test data contains many attacks that were not injected during the training data collection phase to make the intrusion detection task realistic. It is believed that most of the novel attacks can be derived from the known attacks. Finally, the training and test data were processed into the datasets of five million and two million TCP/IP connection records respectively.

The KDD Cup dataset has been widely used as a benchmark dataset for many years in the evaluation of NIDS. One of the major drawbacks with the dataset is that it contains an enormous amount of redundant records both in the training and test data. It was observed that almost 78% and 75% records are redundant in the training and test dataset respectively. The resulted redundancy makes the learning algorithms biased towards the frequent attack records and leads to poor classification results for the infrequent and harmful records. The training and test data were classified with the minimum accuracy of 98% and 86% respectively using a very simple machine learning algorithm. It made the comparison task difficult for various IDSs based on different learning algorithms. NSL-KDD was proposed to overcome the limitation of KDD Cup dataset. The dataset is derived from the KDD Cup dataset. It improved the previous dataset in two ways. First, it eliminated all the redundant records from the training and test data. Second, it partitioned all the records in the KDD Cup dataset into various difficulty levels based on the number of learning algorithms that can correctly classify the records. Further, it selects the records by random sampling of distinct records from different difficulty levels in a fraction that is inversely proportional to their fractions in the distinct records. The multi-steps processing of KDD Cup dataset made the total records statistics reasonable in the NSL-KDD dataset.

A deep learning based approach was developed for an efficient and flexible NIDS. A sparse auto encoder and soft-max regression based NIDS was implemented. The benchmark network intrusion dataset - SL-KDD was used to evaluate anomaly detection accuracy. We observed that the proposed NIDS performed very well compared to previously implemented NIDSs for the normal/anomaly detection when evaluated on the test data. The performance can be further enhanced by applying techniques such as Stacked Auto encoder, an extension of sparse auto encoder in deep belief nets for unsupervised feature learning, and NB-Tree, Random Tree, or J48 for further classification. It was noted that the latter techniques performed well when applied directly on the dataset.

Accelerated deep neural networks for enhanced Intrusion Detection System

The enhanced intrusion detection method uses 41 features and their DNN has 3 hidden layers (2 auto-encoders and 1 soft-max). The results obtained were mixed and those focusing on fewer classes were more accurate than those with more classes.

Network based communication is more vulnerable to outsider and insider attacks in recent days due to its wide spread applications in many fields. Intrusion Detection System (IDS) a software application or hardware is a security mechanism that is able to monitor network traffic and find abnormal activities in the network. Machine learning techniques which have an important role in detecting the attacks were mostly used in the development of IDS. Due to huge increase in network traffic and different types of attacks, monitoring each and every packet in the network traffic is time consuming and computational intensive. Deep learning acts as a powerful tool by which thorough packet inspection and attack identification is possible. The parallel computing capabilities of the neural network make the Deep Neural Network (DNN) to effectively look through the network traffic with an accelerated performance. An accelerated DNN architecture is developed to identify the abnormalities in the network data. NSL-KDD dataset is used to compute the training time and to analyze the effectiveness of the detection mechanism.

Analyzing flow-based anomaly intrusion detection using Replicator Neural Networks

An unsupervised method to learn models of normal network flows is flow based anomaly detection. The above method use Replicator Neural Networks, auto-encoder and the dropout concepts of deep learning. The exact accuracy of their proposed method evaluated is not fully disclosed. Defending key network infrastructure, such as Internet backbone links or the communication channels of critical infrastructure is yet challenging. The inherent complex nature and quantity of network data impedes detecting attacks in real world settings. The utilization features of network flows, characterized by their entropy, together with an extended version of the original Replicator Neural Network (RNN) and deep learning techniques is used to learn models of normality. The combination allows us to apply anomaly-based intrusion detection on arbitrarily large amounts of data and large networks. The approach is unsupervised and it requires no labeled data. It also accurately detects network-wide anomalies without presuming that the training data is completely free of attacks. The evaluation of intrusion detection method on top of real network data indicates that it can accurately detect resource exhaustion attacks and network profiling techniques of varying intensities. The developed method is efficient because a normality model can be learned by training an RNN within a few seconds only.

Deep learning approach for Network Intrusion Detection in Software Defined Networking

The above method is to monitor network flow data. The paper lacked details about its exact algorithms but does present an evaluation using the NSL-KDD dataset, which the authors claim gave an accuracy of 75.75% using six basic features.

Software Defined Networking (SDN) has recently emerged to become one of the promising solutions for the future Internet. With the logical centralization of controllers and a global network overview, SDN brings us a chance to strengthen our network security. However, SDN also brings us a dangerous increase in potential threats. The deep learning approach for flow-based anomaly detection in an SDN environment is applied. A Deep Neural Network (DNN) model for an intrusion detection system and train the model with the NSL-KDD Dataset. The usage of six basic features (that can be easily obtained in an SDN environment) taken from the forty-one features of NSL-KDD Dataset. Through experiments, we confirm that the deep learning approach shows strong potential to be used for flow-based anomaly detection in SDN environments.

Intrusion Detection System Using Deep Neural Network for In-Vehicle Network Security

A novel intrusion detection system (IDS) using a deep neural network (DNN) is proposed to enhance the security of in-vehicular network. The parameters building the DNN structure are trained with probability-based feature vectors that are extracted from the in-vehicular network packets. For a given packet, the DNN provides the probability of each class discriminating normal and attack packets and thus the sensor can identify any malicious attack to the vehicle. The traditional artificial neural network applied to the IDS and the proposed technique adopts recent advances in deep learning studies such as initializing the parameters through the unsupervised pre-training of deep belief networks (DBN) by improving the detection accuracy. It is demonstrated with experimental results that the proposed technique can provide a real-time response to the attack with a significantly improved detection ratio in controller area network (CAN) bus.

Proposed Intrusion Detection System with Deep Neural Network Structure

The intrusion detection system considers a general type of an attack scenario where malicious data packets are injected into an in-vehicle CAN bus. In-vehicular networks are accessed from the mobile communication links such as 3G, 4G, and WIFI or a self-diagnostic tool such as On-Board Diagnostics paired with the driver's mobile device. Intrusion detection system monitors broadcasting CAN packets in the bus and determines an attack.

CAN Packet Feature

CAN feature is an abstract representation of a CAN packet. The feature is designed by considering computational efficiency. The feature is extracted directly from a bit stream of a CAN packet so that the decoding is not necessary during the extraction. The occurrences of bit-symbols in a data packet are taken into an account. The DATA field chosen includes 64 bit positions (= 8 Bytes) in the CAN syntax and investigate the probability distributions of the bit-symbols.

Training the Deep Neural Network Structure

The learning mechanism of the proposed DNN structure to classify a normal packet and an attack packet is explained. An input layer, multiple hidden layers, and an output layer are used. The feature vector is fed to the input nodes of the structure. Each node computes an output with an activation function using rectified linear unit (ReLU) and the linear combinations of the outputs are linked to the next hidden layers.

Attack Detection

The class of a testing CAN packet is predicted in the detection phase. The output is computed with the trained weight parameters and the feature set extracted from the testing CAN packet as in the training. The classifier provides the logistic value 0 or 1, telling if the sample is normal packet or the attack packet respectively.

An efficient intrusion detection system (IDS) based on a deep neural network (DNN) for the security of in-vehicular network. The parameters of DNN are trained with probability-based feature vectors extracted from the in-vehicular network packets by using unsupervised pre-training method of deep belief networks, followed by the conventional stochastic gradient descent method. The DNN provides the probability of each class to discriminate normal and hacking packets and thus the system can identify any malicious attack to the vehicle as a result. A novel feature vector was proposed comprising the mode information and the value information extracted from the network packets and it was efficiently used in the training and the testing. It was demonstrated with experimental results that the proposed technique could provide a real-time response to the attack with a significantly accurate detection ratio about 98% on average when the computational complexity with the number of the layers is modestly small.

Shallow and Deep Networks Intrusion Detection System: A Taxonomy and Survey

Intrusion detection has attracted a considerable interest from researchers and industries. The community, after many years of research, still faces the problem of building reliable and efficient IDS that are capable of handling large quantities of data, with changing patterns in real time situations. The work presented in this manuscript classifies intrusion detection systems (IDS). Moreover, a taxonomy and survey of shallow and deep networks intrusion detection systems is presented based on previous and current works. The taxonomy and survey reviews machine learning techniques and the performance in detecting anomalies was improved. Feature selection which influences the effectiveness of machine learning (ML) IDS is discussed to explain the role of feature selection in the classification and training phase of ML IDS. A discussion of the false and true positive alarm rates is presented to help researchers model reliable and efficient machine learning based intrusion detection systems.

Anomaly Based Detection

Anomaly Based Detection is a behavioral based intrusion detection system. It observes changes in normal activity within a system by building a profile of the system which is being monitored. The profile is generated over a period of time when the system is established to have behaved normally. One advantage is that it offers the ability to detect attacks which are new to the system.

Self-learning – The self-learning system operate by example with a baseline set for normal operation. This is achieved by building a model for the underlying processes with the observed system traffic built up over a period of time. Self-learning systems are sub-divided into the following main categories: time series model and machine learning.

Programmed – A programmed model is when a system needs either a user or an external person to teach the system to detect changes in behavior. The user decides the extent of abnormal behavior in the system and flags an intrusion threat. The programmed models are grouped into four categories: threshold, simple rule based and statistical models.

Signature Based Detection defines a set of rules used to match the patterns in the network traffic. If a mismatch is detected it raises an alarm. It has an advantage of being able to detect attacks giving a low false positive detection ratio. It has a drawback of being able to detect only attacks known to the database. Signature based detection systems are programmed with distinct decision rules. The rules set for detection are coded in a straight forward manner to detect intrusion.

State modeling is the encoding of attacks as a number of different states in a finite automaton. Each of these attacks has to be observed in the traffic profile to be considered as an intrusion. It occurs in sub-classes as time series models: the first is state transition which was proposed by Porras et al. which uses a state transition diagram to represent intrusion. The approach in models intrusion as a series of state transitions which are described as signature action and states descriptions.

String matching is a process of knowledge acquisition just as Expert system but has a different approach in exploiting the knowledge. It deals with matching the patterns in the audit event generated by the attack but not involved in the decision making process. The above technique has been used effectively used as IDS.

Machine Learning Techniques

Machine Learning (ML) can provide IDS methods to detect current, new and subtle attacks without extensive human-based training or intervention. It is defined as a set of methods that can automatically detect patterns to predict future data trends. A large number of machine learning techniques exist; the fundamental operation of all of them relies upon optimal feature selection. The features are the metrics which will be used to detect patterns and trends. If one feature of a network is the packet size: machine learning techniques may monitor the packet size over time and generate distributions from which conclusions may be drawn regarding an intrusion.

Shallow and deep networks intrusion detection systems have gained a considerable interest commercially and amongst the research community. With advancement in data sizes, intrusion detection systems should have the characteristics to handle noisy data with high accuracy in detection with high computational speed. The proposed method gives an overview of the general classification of intrusion detection systems and taxonomy with recent and past works. The taxonomy gives a clear description of intrusion detection system and its complexity. Current studies of deep learning intrusion detection systems have been reviewed to help address the challenges in the new technique still in its early stages in intrusion detection. The scope of the work on classifying intrusion detection systems, reviewing the various methods of detecting anomaly, performance of these methods were based on past and recent works revealing the advantages and disadvantages of each of them.

The focus of the research on shallow and deep networks described experiments is to compare the performance of these learning algorithms. The experiments demonstrated deep networks significantly outperformed the shallow network in detection of attacks. CNN has not been exploited in the field of intrusion detection but proven to be a good classifier. DBN is also new in its exploitation in this field and experimental works are still in progress to determine the reliability of these learning algorithms to detect attacks. Signature based technique have been in use commercially but have not been able to detect all types of attacks especially if the IDS signature list did not contain the right signature.

A Deep Learning Based DDoS Detection System in Software-Defined Networking (SDN)

Distributed Denial of Service (DDoS) is one of the most prevalent attacks in an organizational network infrastructure. A deep learning based multi-vector DDoS detection system is used in a software-defined network (SDN) environment. SDN provides flexibility to program network devices for different objectives and eliminates the need for third-party vendor-specific hardware. A system as a network application was implemented on top of an SDN controller. Deep learning for feature reduction of a large set of features derived from network traffic headers was used. The system based on different performance metrics by applying it on traffic traces collected from different scenarios was evaluated. The high accuracy with a low false-positive for attack detection was observed in our proposed system.

Software-Defined Networking (SDN)

SDN architecture decouples the control plane and data plane from network devices, also termed as 'switches' and makes them simple packet forwarding elements. The decoupling of control logic and its unification to a centralized controller offers several advantages compared to the current network architecture that integrates both the planes tightly. Administrators can implement policies from a single point, i.e. controller and observe their effects on the entire network that makes management simple, less error-prone and enhances security. Switches become generic and vendor-agnostic. Applications that run inside a controller can program these switches for different purposes such as layer 2/3 switch, firewall, IDS, load balancer using API offered by a controller to them.

Stacked Auto encoder (SAE)

Stacked sparse auto encoders and soft-max classifier for unsupervised feature learning and classification respectively. A sparse auto encoder is a neural network that consists of three layers in which the input and output layers contain M nodes and the hidden layer contains N nodes. The M nodes at the input represent a record with M features, i.e., X = {x1, x2, ..., xm}. The output layer is made an identity function of the input layer for training purpose, i.e., X^ = X .

A deep learning based DDoS detection system for multi-vector attack detection in an SDN environment. The proposed system identifies individual DDoS attack class with an accuracy of 95.65%. It classifies the traffic in normal and attack classes with an accuracy of 99.82% with very low false-positive compared to other works. Future enhancement is to reduce the controller's bottleneck and implement an NIDS that can detect different kinds of network attacks in addition to DDoS attack.

A deep learning-based RNNs model for automatic security audit of short messages

The traditional text classification methods usually follow this process: first, a sentence can be considered as a bag of words (BOW), then transformed into sentence feature vector which can be classified by some methods, such as maximum entropy (ME), Naive Bayes (NB), support vector machines (SVM) and so on. An ideal result is not obtained by the application of the above methods. The most important reason is that the semantic relations between words are very important for text categorization however, the traditional method cannot capture it. Sentiment classification, as a special case of text classification, is binary classification (positive or negative). Inspired by the sentiment analysis, a novel deep learning-based recurrent neural networks (RNNs)model for automatic security audit of short messages from prisons is used to classify short messages(secure and non-insecure). The feature of short messages is extracted by word2vec which captures word order information, and each sentence is mapped to a feature vector. In particular, words with similar meaning are mapped to a similar position in the vector space and then classified by RNNs. RNNs are now widely used and the network structure of RNNs determines that it can easily process the sequence data. Short messages are preprocessed to extract typical features from existing security and non-security short messages via word2vec and classify short messages through RNNs which accept a fixed-sized vector as input and produce a fixed-sized vector as output. The experimental results show that the RNNs model achieves an average 92.7% accuracy which is higher than SVM.

An automatic security auditing tool for short messages (SMS) is developed. The application is based upon the RNN model. The authors claimed that their evaluations resulted in an accuracy rate of 92.7%, thus improving existing classification methods (e.g. SVM and Naive Bayes).

A deep learning approach for detecting malicious JavaScript code

Malicious JavaScript code in web pages on the Internet is an emergent security issue because of its universality and potentially severe impact. Because of its obfuscation and complexities, detecting it has a considerable cost. Over the last few years, several machine learning-based detection approaches have been proposed; most of them use shallow discriminating models with features that are constructed with artificial rules. However, with the advent of the big data era for information transmission, the existing methods already cannot satisfy actual needs. A new deep learning framework for detection of malicious JavaScript code is proposed from the where the highest detection accuracy is compared with the control group. The architecture is composed of a sparse random projection, deep learning model and logistic regression. Stacked denoising auto-encoders were used to extract high-level features from JavaScript code; logistic regression as a classifier was used to distinguish between malicious and benign JavaScript code.

Deep learning

Deep learning is an emerging research field of machine learning; it attempts to hierarchically learn high-level representation of data with deep neural networks. Each layer of the network is first layer-wise pre-trained via unsupervised learning and then the entire network carries out fine-tuning in a supervised mechanism. In this manner, high-hierarchy features can be learned from low-hierarchy ones and the appropriate features can be applied to pattern classification in the end. According to the universal approximation theorem of neural networks, deep models have a better ability to represent the nonlinear functions than shallow ones; therefore, deep models can achieve better results on large-scale training data. Furthermore, from a feature recognition and classification point of view, the deep learning framework incorporates a feature extractor and classifier into one framework, which can automatically learn feature representations (often from unlabeled data), thus avoiding spending substantial effort to manually design features. Typical deep neural networks include convolutional neural networks, stacked auto-encoders, Stacked denoising Autoencoders (SdA), deep Boltzmann machines and deep belief networks. In this research, in order to learn more useful features with unsupervised pre-training, SdAs is focused for two reasons: first of all, SdA is suitable for the application of text classification, especially when the input is binary form. Secondly, according to the results described by Vincent et al. , the SdA would yield better results than other unsupervised methods when the input is binary high-dimensional data.

Denoising auto-encoder (dA)

The dA is an extension of a classical auto-encoder. In real applications, many problems like missing data and data noise would cause the theoretical method to be impractical. In order to force the hidden layer to learn more robust features, we trained the auto-encoder to reconstruct the input from corrupted input data. To convert the auto encoder into a dA, we needed to introduce a stochastic noise (i.e., a stochastic corruption step operation) into the input layer. A dA is an extension of auto-encoder, which is used to reconstruct the inputs from a noisy version of it by minimizing the reconstruction loss. A dA first encodes x to a hidden representation y through a deterministic mapping, where y ∈ Rh , and then decodes representation y back into a reconstruction z of the same shape as x through a similar transformation, where z ∈ Rd . Hence, the dA is attempting to reconstruct the original inputs x from the corrupted values.

Stacked denoising auto-encoders (SdA)

An SdA model is created by integrating multiple dAs to form a deep learning. Every hidden layer in the network is trained as a dA by optimizing equation through unsupervised pretraining and we can take the output of a dA on a previous layer as the input of the next. More clearly, the first hidden layer is trained as a dA with JavaScript code vectors as input, and after finishing training the first hidden layer uses the output of the first hidden layer as the input of the second hidden layer. Similarly, once the k-th hidden layer is trained, (k + 1)- th layer can be trained using the output of the k-th hidden layer as the input of the (k + 1) - th hidden layer to compute the latent representation. Multiple dAs can be stacked hierarchically. To use the SdA model for malicious JavaScript code detection, a logistic regression classifier is applied to the output of the last hidden layer to distinguish between malicious and benign JavaScript code. The parameters are adjusted throughout the entire network by making use of target class labels.

A method using deep features extracted by SdA for classification, to detect JavaScript on webpages as either malicious or not. Experimental results indicated that features extracted by our SdAs were useful for pattern classification and the SdA helped to improve the accuracy of both logistic regression and the SVM classifiers. When compared with other feature extraction methods such as principle component analysis, independent component analysis and FA, the SdA has the highest classification accuracy. The proposed SdA-LR model was verified to have higher statistical evaluations than the existing methods. It was shown in our experimental results that building the deep architecture network with three layers of auto-encoders and 250 hidden units in each layer was the optimal choice for the JavaScript code classification. The limitation of the SdA-LR is the long training time; however, the high-speed testing time makes up for this deficiency. The classifier has some other potential drawbacks. One obvious drawback is that the classifier is likely to identify a small number of good JavaScript codes as latently bad. JavaScript codes in some websites have been packed to compress as well as to obfuscate; the reason for doing this is to reduce the size of the initial JavaScript code and to make it more difficult for one to find out what happened in the code and to steal their source code. Some benign packed JavaScript codes are the most likely to be categorized as malicious by the classifier. Namely, benign packed JavaScript code might yield a false positive to a large extent.

Deep4MalDroid: A Deep Learning Framework for Android Malware Detection Based on Linux Kernel System Call Graphs

Commercial Android malware detection framework was Deep4MalDroid. The method involves the use of stacked auto-encoders with best accuracy resulting from 3 layers. The 10-fold cross validation was used, showing that in comparison to shallow learning, the approach offers improved detection performance.

With explosive growth of Android malware and due to its damage to smart phone users (e.g., stealing user credentials, resource abuse), Android malware detection is one of the cyber security topics that are of great interests. Currently, the most significant line of defense against Android malware is anti-malware software products, such as Norton, Lookout and Comodo Mobile Security, which mainly use the signature-based method to recognize threats. However, malware attackers increasingly employ techniques such as repackaging and obfuscation to bypass signatures and defeat attempts to analyze their inner mechanisms.

The increasing sophistication of Android malware calls for new defensive techniques that are harder to evade and are capable of protecting users against novel threats. A novel dynamic analysis method named Component Traversal was proposed that can automatically execute the code routines of each given Android application (app) as completely as possible. Based on the extracted Linux kernel system calls, construct the weighted directed graphs are constructed and then apply a deep learning framework resting on the graph based features for newly unknown Android malware detection.

A comprehensive experimental study on a real sample collection from Comodo Cloud Security Center is performed to compare various malware detection approaches. Promising experimental results demonstrate that the proposed method outperforms other alternative Android malware detection techniques. The developed system Deep4MalDroid has also been integrated into a commercial Android anti-malware software.

Deep Neural Network Self-training Based on Unsupervised Learning and Dropout

In supervised learning methods, a large amount of labeled data is necessary to find reliable classification boundaries to train a classifier. However, it is hard to obtain a large amount of labeled data in practice and it is time-consuming with a lot of cost to obtain labels of data. Although unlabeled data is comparatively plentiful than labeled data, most of supervised learning methods are not designed to exploit unlabeled data. Self-training is one of the semi supervised learning methods that alternatively repeat training a base classifier and labelling unlabeled data in training set. Most self-training methods have adopted confidence measures to select confidently labeled examples because high-confidence usually implies low error. A major difficulty of self-training is the error amplification.

If the classifier misclassifies some examples and the misclassified examples are included in the labeled training set, the next classifier may learn improper classification boundaries and generate more misclassified examples. Since base classifiers are built with small labeled dataset and are hard to earn good generalization performance due to the small labeled dataset. Although improving training procedure and the performance of classifiers, error occurrence is inevitable, so corrections of self-labeled data are necessary to avoid error amplification in the following classifiers. A deep neural network based approach was proposed for alleviating the problems of self-training by combining schemes: pre-training, dropout and error forgetting. By applying combinations of these schemes to various dataset, a trained classifier using the above approach shows improved performance than trained classifier using common self-training.

In self-training, the error amplification is a major problem. If the current base classifier mislabels some examples, the mislabelled examples may provide inaccurate information to base classifier in the next phase. Then the next base classifier may learn the inaccurate information and generate more mislabelled examples. The error of the base classifier is reinforced because of error viscosity. The base classifier needs ability to learn accurate but generalized class boundaries even from small training data for the reduction of error amplification. The misclassified examples in the proceeding labelling processes are inevitable and it needs to have a filtering step to remove errors in the classifier or the training dataset. A deep neural network based approach for self-training was used by adopting pre-training, dropout and error forgetting.

The self-training scheme adopting unsupervised pre-training based on restricted Boltzmann machine learning methods was dropped out. The target of our approach is alleviating error amplification that is major problem of self-training. The classification performance of self-training is improved in various datasets by the applying the combination of above schemes and it is showed and confirmed in experimental result. The combination of the error forgetting and example re-evaluation shows the performance degradation in our experiments. It is assumed that the reason of the performance degradation is the over-regularization by the combination of the error forgetting

III. LIMITATIONS

3.1 5-CLASSES KDD CUP '99 CLASSIFICATION

The results involving KDD Cup '99 dataset evaluation showed that the model is able to offer an average accuracy of 97.85%. More specifically, the results show that the accuracy is better than or comparable in 3 out of 5 classes. It is noted that the results for "Remote to local" and "User to roots" attack classes are anomalous. The stacked NDAE model requires greater amounts of data to learn from. Unfortunately, due to the smaller number of training datum available, the results achieved are less stable. It is evident from the performance analysis that the model can offer improved precision, recall and F-score, especially for larger classes. Furthermore, the proposed model managed to produce these comparable performance results, whilst consistently reducing the required training time by an average of 97.72%.

3.2 5-CLASS NSL-KDD CLASSIFICATION

The results that throughout all of the measures the model yields superior level of performance by using NSL-KDD dataset in 3 of the 5 classes. The model offered a total accuracy rate of 85.42% which improves upon the DBN model by just fewer than 5%. It also offered a 4.84% reduction in the false alarm rate. The results also re-emphasize the point made, that the model doesn't handle smaller classes as well. Another important factor is that the time required to train the model is drastically reduced, yielding an average time saving of 78.19% against DBN. It is of critical importance particularly for application in a NIDS.

3.3 13-CLASS NSL-KDD CLASSIFICATION

The results from the 13-Class classification evaluate demonstrate that the model was able to offer a 3.8% improvement on its own accuracy simply by using a more granular dataset. This supports a claim that the model is able to work more effectively with larger and complex datasets. Furthermore, the larger dataset gives a better insight into the weakness in the model. As it can be seen from the results, there is a direct correlation between the size of the training datasets for each label and the accuracy/error rates. The smaller classes yield lower levels of accuracy using our model, the larger classes yielded consistently high rates throughout all of the performance measures

3.4 COMPARISON WITH RELATED WORKS

The results are compared from our stacked NDAE model against the results obtained from similar deep learning based NIDSs. In [26], the authors claim their 5-class classification of the NSL-KDD dataset produced an f-score of 75.76%. Their recall and precision results are not listed but the bar charts show them to be around 69% and 83% respectively. The proposed model has produced superior results by offering f-score of 87.37%, recall of 85.42% and precision of 100.00%. Tang et al. [23] claim that their Deep Neural Network (DNN) approach achieved an accuracy of 75.75% when performing a 5-class classification of the NSL-KDD dataset. The result is lower than our achieved accuracy of 85.42%. Whilst classifying the KDD Cup '99 dataset, Kim et al. [37] claim they have achieved an accuracy of 96.93%. Gao et al. [38] claim their deep learning DBN model achieved an accuracy of 93.49%. Both of these results are less than the 97.85% accomplished by the model. The comparisons show that the proposed model's results are very promising when compared to other current deep learning-based methods.

IV. CONCLUSION & FUTURE WORK

There are several problems faced by existing NIDS techniques and a novel NDAE method was produced for unsupervised feature learning. A novel classification model constructed from stacked NDAEs and the RF classification algorithm. Proposed Model in TensorFlow and performed extensive evaluations on its capabilities. Evaluations utilized the benchmark KDD Cup '99 and NSL-KDD datasets and achieved very promising results. The results have demonstrated that the approach offers high levels of accuracy, precision and recall together with reduced training time. The stacked NDAE model was compared against the mainstream DBN technique. These comparisons have demonstrated that the model offers up to a 5% improvement in accuracy and training time reduction of up to 98.81%. Unlike most previous work, project was evaluated with the capabilities of the model based on both benchmark datasets, revealing a consistent level of classification accuracy. Although the model has achieved the above promising results, result that acknowledges that it is not perfect and there is further room for improvement. The first avenue of exploration for improvement will be to assess and extend the capability of the model to handle zero-day attacks was the future enhancement. The idea to expand upon the existing evaluations by utilizing real-world backbone network traffic to demonstrate the merits of the extended model was also be in future work

V. REFERENCES

1. B Dong and X. Wang, "Comparison deep learning method to traditional methods using for network intrusion detection," in Proc. 8th IEEE Int. Conf. Commun. Softw. Netw., Beijing, China, Jun. 2016, pp. 581-585.
2. R Zhao, R. Yan, Z. Chen, K. Mao, P. Wang, and R. X. Gao, "Deep learning and its applications to machine health monitoring: A survey,"Submitted to IEEE Trans. Neural Netw. Learn. Syst., 2016. [Online].Available: http://arxiv.org/abs/1612.07640
3. S Hou, A. Saas, L. Chen, and Y. Ye, "Deep4MalDroid: A Deep learningframework for android malware detection based on linux kernel systemcall graphs," in Proc. IEEE/WIC/ACM Int. Conf. Web Intell. Workshops,Omaha, NE, USA, Oct. 2016, pp. 104-111.
4. IDC, "Executive summary: Data growth, business opportunities, and the IT imperatives. The digital universe of opportunities: Rich data and the increasing value of the internet of things," IDC, Framingham, MA, USA,Tech. Rep. IDC_1672, 2014. [Online]. Available: https://www.emc.com/leadership/digital-universe/2014iview/executive-summary.htm
5. Juniper Networks, "Juniper Networks, How many packets per secondper port are needed to achieve Wire-Speed?," 2015. [Online]. Available: https://kb.juniper.net/InfoCenter/index?page=content&id=KB14737
6. I Goodfellow, Y. Bengio, and A. Courville, Deep Learning. Cambridge,MA, USA: MIT Press, 2016. [Online]. Available: http://www.deeplearningbook.org
7. L Deng, "Deep learning: Methods and applications," Found. Trends Signal Process., vol. 7, no. 3/4, pp. 197-387, Aug. 2014.
8. P Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol,"Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," J. Mach. Learn. Res.,vol. 11, pp. 3371-3408, 2010.
9. G E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," Science, vol. 313, no. 5786, pp. 504-507,2006.
10. Y. Wang, H. Yao, and S. Zhao, "Auto-encoder based dimensionality reduction," Neurocomputing, vol. 184, pp. 232-242, 2016.
11. Z. Liang, G. Zhang, J. X. Huang, and Q. V. Hu, "Deep learning for healthcare decision making with EMRs," in Proc. IEEE Int. Conf. Bioinformat.Biomed., Nov. 2014, pp. 556-559.
12. S. P. Shashikumar, A. J. Shah, Q. Li, G. D. Clifford, and S. Nemati,"A deep learning approach to monitoring and detecting atrial fibrillation using wearable technology," in Proc. IEEE EMBS

Int. Conf. Biomed.Health Informat, FL, USA, 2017, pp. 141-144.

13. F. Falcini, G. Lami, and A. M. Costanza, "Deep learning in automotive software," IEEE Softw., vol. 34, no. 3, pp. 56-63, May 2017. [Online]. Available: http://ieeexplore.ieee.org/document/7927925/
14. A. Luckow, M. Cook, N. Ashcraft, E. Weill, E. Djerekarov, and B. Vorster,"Deep learning in the automotive industry: Applications and tools," in Proc. IEEE Int. Conf. Big Data, Dec. 2016, pp. 3759-3768. [Online].Available: http://ieeexplore.ieee.org/document/7841045/
15. H. Lee, Y. Kim, and C. O. Kim, "A deep learning model for robust wafer fault monitoring with sensor measurement noise," IEEE Trans. Semicond.Manuf., vol. 30, no. 1, pp. 23-31, Feb. 2017.
16. L. You, Y. Li, Y. Wang, J. Zhang, and Y. Yang, "A deep learning based RNNs model for automatic security audit of short messages," in Proc. 16th Int. Symp. Commun. Inf. Technol., Qingdao, China, Sep. 2016,pp. 225-229.
17. R. Polishetty, M. Roopaei, and P. Rad, "A next-generation secure cloudbased deep learning license plate recognition for smart cities," in Proc.15th IEEE Int. Conf. Mach. Learn. Appl., Anaheim, CA, USA, Dec. 2016,pp. 286-293.
18. K. Alrawashdeh and C. Purdy, "Toward an online anomaly intrusion detection system based on deep learning," in Proc. 15th IEEE Int. Conf. Mach. Learn. Appl., Anaheim, CA, USA, Dec. 2016,pp. 195-200.
19. J. Kim, N. Shin, S. Y. Jo, and S. H. Kim, "Method of intrusion detection using deep neural network," in Proc. IEEE Int. Conf. Big Data Smart Comput., Hong Kong, China, Feb. 2017, pp. 313-316.
20. A. Javaid, Q. Niyaz, W. Sun, and M. Alam, "A deep learning approach for network intrusion detection system," in Proc. 9th EAI Int.Conf. BioInspired Inf. Commun. Technol., 2016, pp. 21-26. [Online]. Available:http://dx.doi.org/10.4108/eai.3-12-2015.2262516
21. S. Potluri and C. Diedrich, "Accelerated deep neural networks for enhanced intrusion detection system," in Proc. IEEE 21st Int. Conf. Emerg. Technol. Factory Autom., Berlin, Germany, Sep. 2016,pp. 1-8.
22. C. Garcia Cordero, S. Hauke, M. Muhlhauser, and M. Fischer, "Analyzing flow-based anomaly intrusion detection using replicator neural networks," in Proc. 14th Annu. Conf. Privacy, Security. Trust, Auckland, New Zeland,Dec. 2016, pp. 317-324.
23. T. A. Tang, L. Mhamdi, D. McLernon, S. A. R. Zaidi, and M. Ghogho, "Deep learning approach for network intrusion detection in software defined networking," in Proc. Int. Conf. Wireless Netw. Mobile Commun.,Oct. 2016, pp. 258-263.
24. 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,Jun. 2016, Art. no. e0155781.
25. E. Hodo, X. J. A. Bellekens, A. Hamilton, C. Tachtatzis, and R. C. Atkinson, Shallow and deep networks intrusion detection system: A taxonomy and survey, Submitted to ACM Survey, 2017, [Online]. Available:http://arxiv.org/abs/1701.02145
26. Q. Niyaz, W. Sun, and A. Y. Javaid, A deep learning based DDOS detection system in software-defined networking (SDN), Submitted to EAI Endorsed Transactions on Security and Safety, In Press, 2017, [Online].Available: http://arxiv.org/abs/1611.07400
27. Y. Wang, W.-D. Cai, and P.-C. Wei, "A deep learning approach for detecting malicious JavaScript code," Security Commun. Netw., vol. 9, no. 11,pp. 1520-1534, Jul. 2016.
28. H.-W. Lee, N.-R. Kim, and J.-H. Lee, "Deep neural network self-training based on unsupervised learning and dropout," Int. J. Fuzzy Logic Intell.Syst., vol. 17, no. 1, pp. 1-9, Mar. 2017. [Online].

Available:http://www.ijfis.org/journal/view.html?doi=10.5391/IJFIS.2017.17.1.1

29. S. Choudhury and A. Bhowal, "Comparative analysis of machine learning algorithms along with classifiers for network intrusion detection," in Proc. Int. Conf. Smart Technol. Manage. Comput., Commun., Controls, Energy Mater., May 2015, pp. 89-95.
30. M. Anbar, R. Abdullah, I. H. Hasbullah, Y. W. Chong, and O. E. Elejla, "Comparative performance analysis of classification algorithms for intrusion detection system," in Proc. 14th Annu. Conf. Privacy, Security Trust,Dec. 2016, pp. 282-288.
31. Y. Chang, W. Li, and Z. Yang, "Network intrusion detection based on random forest and support vector machine," in Proc. IEEE Int. Conf. Comput.Sci. Eng./IEEE Int. Conf. Embedded Ubiquitous Comput., Jul. 2017,pp. 635-638.
32. Y. Y. Aung and M. M. Min, "An analysis of random forest algorithm based network intrusion detection system," in Proc. 18th IEEE/ACIS Int.Conf. Softw. Eng., Artif. Intell., Netw. Parallel/Distrib. Comput., Jun. 2017,pp. 127-132.
33. L. Breiman, "Random forests," Mach. Learn., vol. 45, no. 1, pp. 5-32,2001.
34. S. J. Stolfo, W. Fan, W. Lee, A. Prodromidis, and P. K. Chan, "Cost-based modeling for fraud and intrusion detection: Results from the JAM project," in Proc. DARPA Inf. Survivability Conf. Expo., 2000, pp. 130-144.
35. M. Tavallaee, E. Bagheri, W. Lu, and A.-A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in Proc. 2nd IEEE Symp. Comput. Intell. Security Defence Appl., 2009, pp. 53-58.
36. J. McHugh, "Testing intrusion detection systems: A critique of the 1998 and 1999 DARPA intrusion detection system evaluations as performed by Lincoln Laboratory," ACM Trans. Inf. Syst. Security, vol. 3, no. 4,pp. 262-294, 2000.
37. J. Kim, J. Kim, H. L. T. Thu, and H. Kim, "Long short term memory recurrent neural network classifier for intrusion detection," in Proc. Int.Conf. Platform Technol. Service, Feb. 2016, pp. 1-5.
38. N. Gao, L. Gao, Q. Gao, and H. Wang, "An intrusion detection model based on deep belief networks," in Proc. 2nd Int. Conf. Adv. Cloud Big Data, Nov. 2014, pp. 247