Periodicals of Engineering and Natural Sciences ISSN 2303-4521
Vol. 7, No. 3, September 2019, pp.1074-1095
Available online at: *http://pen.ius.edu.ba*

Survey on Intrusion Detection Systems based on Deep Learning

Ali Azawii Abdul Lateef [1], Sufyan T. Faraj Al-Janabi [2], Belal Al-Khateeb [3]
[1] Department of Computer Science\College of Computer Science and IT, University of Anbar.
[2] Department of Information Systems \College of Computer Science and IT, University of Anbar.
[3] Department of Computer Science\College of Computer Science and IT, University of Anbar.

Article Info

Received Feb 27, 2019

Keyword:

Intrusion Detection Systems, Recurrent Neural Network, Deep Learning, Deep Neural Network.

ABSTRACT

Intrusion Detection Systems (IDSs) have a significant role in all networks and information systems in the world to earn the required security guarantee. IDS is one of the solutions used to reduce malicious attacks. As attackers always changing their techniques of attack and find alternative attack methods, IDS must also evolve in response by adopting more sophisticated methods of detection.

The huge growth in the data and the significant advances in computer hardware technologies resulted in the new studies existence in the deep learning field, including intrusion detection. Deep learning is sub-field of Machine Learning (ML) methods that are based on learning data representations. In this paper, a detailed survey of various deep learning methods applied in IDSs is given first. Then, a deep learning classification scheme is presented and the main works that have been reported in the deep learning works is summarized. Utilizing this approach, we have provided a taxonomy survey on the available deep architectures and algorithms in these works and classify those algorithms to three classes, which are: discriminative, hybrid and generative. After that, chosen deep learning applications are reviewed in a wide range of fields of intrusion detection. Finally, popular types of datasets and frameworks are discussed.

Corresponding Author:

Ali Azawii Abdul Lateef,
Department of Computer Science,
College of Computer Science and IT, University of Anbar

Email: aliazawii@uoanbar.edu.iq

1. Introduction

The security of computer and network systems has been in the focal point of research for a long time. All organizations working in the field of information technology have been ratified that the subject of information protection is very critical and important issue that cannot be ignored. It is necessary to achieve the three basic principles that any secure system rests on its (confidentiality, integrity, and availability).

The National Institute of Standards and Technology has defined intrusion detection as "*the process of monitoring the events occurring in a computer system or network and analyzing them for signs of intrusions, defined as attempts to compromise the confidentiality, integrity, availability, or to bypass the security mechanisms of a computer or network*" [1],[2].

Every day there are new types of cyber-attacks that are faced by systems and networks of official and non-official organizations, e-commerce and even people around the world. These attempts aim to obtain certain

information or destroy the information itself to arrive at stopping the operation of these systems which completely rely on this information. Intrusion detection systems (IDS) are one solution to these problems and breakthroughs [3]. Various kinds of computer systems usage and malicious network communications have been detected by IDSs, while this task cannot be implemented by the traditional firewall. The work hypothesis of IDSs is based on the fact that the legal user usage is different from the intruder user [4] .
Commonly, IDSs are classified into two groups 1) anomaly 2) misuse (signature) detection based on their methods of detection methods [5]. Anomaly detection attempts to mark if variation from the determined patterns of normal utilization can be flagged as intrusions. Misuse detection utilizes examples of surely understood attacks or frail spots of the system to recognize if there are intrusions [6]. As of late, deep learning has been significantly used in research and it was widely used with numerous different applications such as images classification, extraction and analysis of data from video files, analysis of social networking data, data mining and information security, including intrusion detection [7].
Deep learning is a type of ML methods, in which numerous information-processing layers in hierarchical architectures are utilized for classifying patterns and for feature or representation learning [8]. Today, deep learning has become a very important and successful research trend in the ML community because of its great success in these fields [9]. In this survey, we give an overview of the most recent papers that have used deep learning approaches in intrusion detection systems.

2. Deep learning approaches

Deep learning (Also referred to as the Deep Neural Network or the Deep Neural Learning) which is a sub-set of ML in the area of Artificial Intelligence (AI), which has networks that can learn in both supervised and unsupervised manners from labelled and unlabelled data. Deep Learning is an AI function which simulates the working of the human brain in the way that it processes data and creates patterns for using in the process of decision making [9]. There is no single definition of deep learning, but most definitions emphasize the following aspects:

- Branch of ML.
- Models are typically nonlinear.
- Uses both supervised and unsupervised approaches to fit models to data.
- Models are graph structures (networks) with numerous layers (deep).

Based on the way structures and methods are designed for utilization, for example, recognition/classification or synthesis/generation, the majority of the study in this field and the applied algorithms in the field of intrusion detection can be broadly categorized to three main categories that are [10]:

(1) Generative (unsupervised).
(2) Discriminative (supervised).
(3) Hybrid deep architecture.

The classification of the deep learning approaches is illustrated in Figure 1.

2.1 Deep networks for unsupervised or generative learning

Unsupervised learning also referred to as the generative architectures utilize unlabeled data. The key idea of applying generative architectures to recognition of patterns is pre-training or unsupervised learning [11]. Due to the difficulty of the lower levels learning of subsequent networks, there is a need for deep generative architectures. This is why, with a limited amount of training data, learning every of the lower layers in layer-

by-layer method without depending on all the higher layers is highly significant [12]. There are numerous approaches, which were classified as unsupervised learning in the following way.

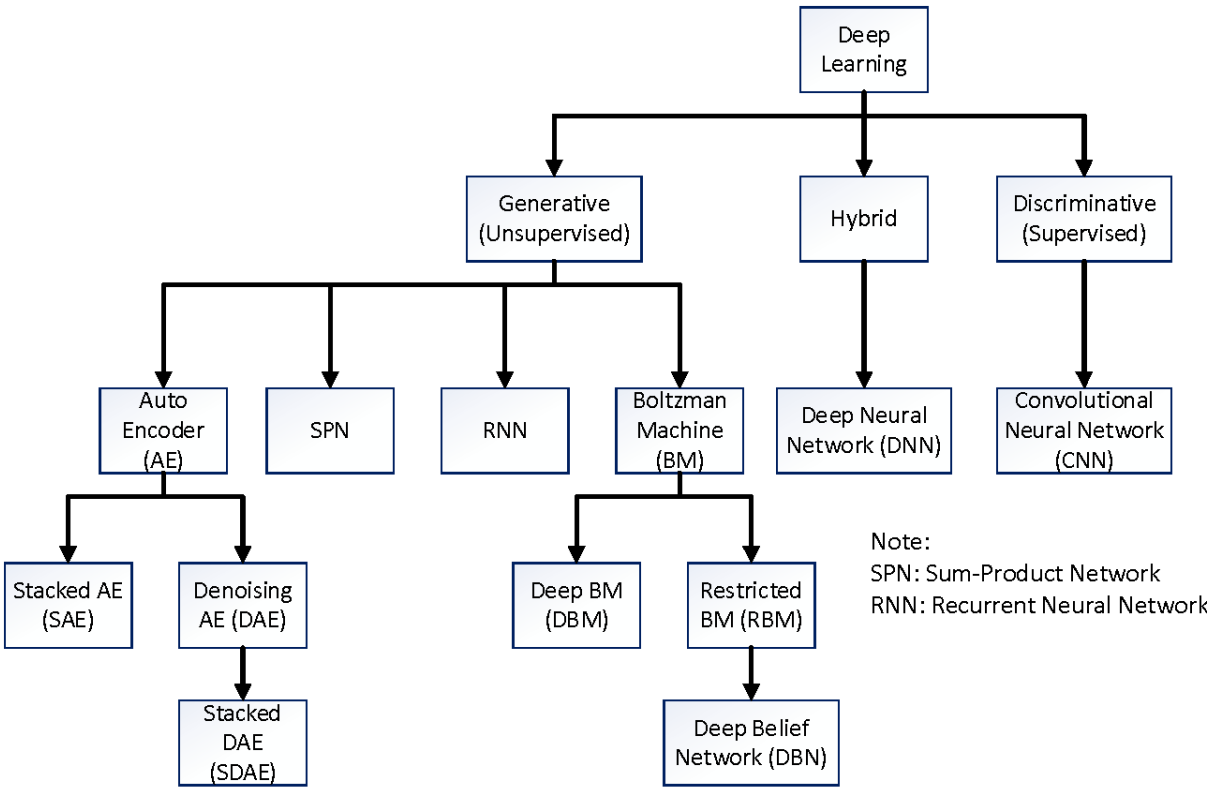

Figure 1. Taxonomy of deep learning methods [13]

2.1.1 Auto encoder (AE)

Deep AE is a specific kind of unsupervised artificial NN whose output is the actual data input. The specific use of the AE is to use a feedforward approach to reconstitute an output from an input. The input is compressed and then sent to be decompressed as output, which is often similar to the original input. That is the nature of an AE – that the similar inputs and outputs get measured and compared for execution results. More explicitly, it's a non-linear approach of feature extraction that does not involve any class labels; therefore generative [14] [15]. When the number of hidden layers is > 1, the AE is considered deep. An AE utilizes three layers or more in the NN:

1. An input data layer to be sufficiently coded (for instance spectra in speech or image pixels);
2. One or more significantly smaller hidden layers that will be forming the encoding.
3. An output layer, in which every one of the neurons has the same meaning as in the input layer.

Figure 2 shows the general structure of an AE, mapping an input $\boldsymbol{x}$ to an output (referred to as the reconstruction) $\boldsymbol{r}$ through an internal representation or code $\boldsymbol{h}$. The auto-encoder has two components: the encoder $\boldsymbol{f}$ (which maps $\boldsymbol{x}$ to $\boldsymbol{h}$) and the decoder $\boldsymbol{g}$ (which maps $\boldsymbol{h}$ to $\boldsymbol{r}$) [15].

A framework of Network Intrusion Detection System (NIDS) based on AE algorithm/stacked Autoencoder (SAE) is proposed in [16], where Muhamad Erza Aminento et al. applied SAE which belongs to deep learning algorithms as a classifier for KDD99 Dataset. The proposed approach showed four different IDS: IDS-A for application layer, IDS-T for transport layer, IDS-N or network layer and IDS-L for data link layer.

Each IDS type is responsible for a variety of network devices which are distributed among computer networks, as shown in Figure (3).

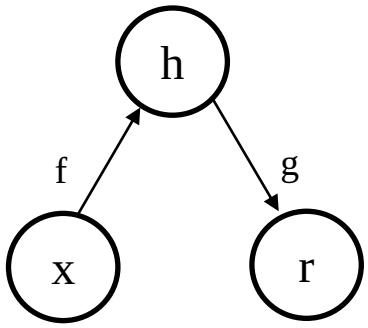

Fig. 2. General Structure of Auto Encoder [15]

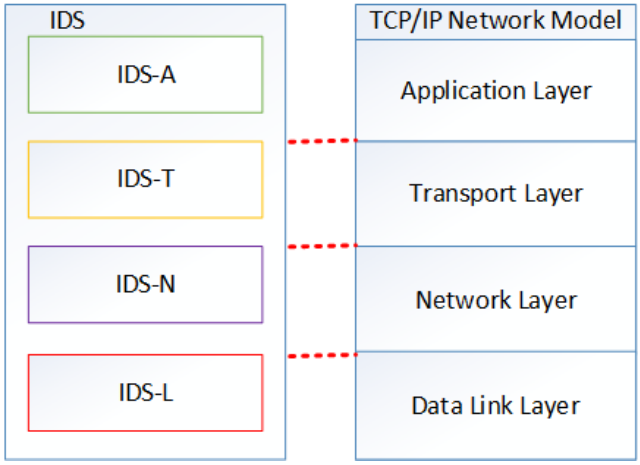

Fig. 3. NIDS architecture [15]

Every one of the IDS types has its unique dataset based on the properties of the TCP/IP layer. As an example, dataset for IDS-A contains data instances with normal and application layer attacks label. Secondly, they applied the feature selection method for every data-set to choose the most important feature set for every one of the IDS types. However, they limited the IDS-T only as a prove of concept (PoC). They did not discuss the details of IDS-N, IDS-A and IDS-L in their paper.

The researchers employed ANN as feature selection method. In the ANN model the value of the hidden layer represents bias value. The researchers used two hidden (encoder) layers. They completed their stacked architecture with the method of supervised learning by SoftMax regression function with the use of labels from training data. The results of this work shown that the lightweight IDS can be done by splitting IDS into smaller parts and reduce feature dimensionality and that the lightweight IDS can achieve comparable detection rate as the ordinary IDS. However, implementing lightweight IDS for a wireless network is still considered to be a challenging issue.

Quamar Niyaz et. al. have utilized Self-taught Learning (STL), a deep learning-based approach, on NSL-KDD benchmark data-set for network intrusion [17]. They used Sparse Auto-Encoder (SAE) based feature learning for their work because it is rather easier to implement and it performs well [18]. An SAE is an NN consisting of an input layer, a hidden layer, and an output layer. The Classification was done using STL in two stages: SAE for Unsupervised Feature Learning and SoftMax regression classifier training for the derived training data. The performance may additionally be improved via applying approaches like SAE and others. They did not apply their approach of real-time NIDS for actual networks. Additionally, they reported that using on-the-

go feature learning on raw network traffic headers rather than derived features may be a research area of high impact in the future.

In the other work, Yisroel Mirsky et al. presented Kitsune: a plug and play NIDS that is capable of learning the detection of attacks on the LAN, with no supervision, and in a sufficient on-line manner. AE used for distinguishing between normal and abnormal patterns of traffic in Kitsune's core algorithm (KitNET).

A feature extraction framework supports KitNET that is effectively specifies a path of the network channels patterns. The main contribution of this work was: 1) novel AE-based NIDS for simple network devices (Kitsune), which is plug-and-play and lightweight. 2) An on-line approach for the automatic construction of the group of auto-encoders (in other words, mapping properties to NN inputs) in an unsupervised manner. This was practically tried on an IoT network, operational IP camera video surveillance network and many different attacks [19].

Fahimeh Farahnakian et al. proposed a Deep Auto Encoder (DAE) method for enhancing the IDS. They have made the argument that the most motivating models for extracting features from the high-dimensional data in deep learning case is AE. [20]. Their suggested Deep Auto Encoder based IDS (DAE-IDS) is made up of four auto encoders, in which the result of the AE at the existing layer is utilized as the AE input in the following layer. Moreover, an AE at the existing layer is trained prior to the AE at the following layer. For the aim of training DAE-IDS, they have used a greedy unsupervised layer-wise training approach which is helpful in improving the efficiency of the deep model. After the 4 auto-encoders are trained, they have utilized a Soft-Max layer for classifying the inputs to normal and attack. They have utilized the KDDCUP 1999 data-set for evaluating the efficiency of DAE-IDS due to the fact that this data-set has been used largely for the evaluation of the IDSs. The suggested method has reached a detection precision equal to 94.71% on the total of 10% KDD-CUP 1999 testing data-set [21]. As their future works, they suggested to explore the way sparsity constraints are forced on AE and the way SAE can be designed for additionally improving the efficiency of the intrusion detection.

Another work by Hongpo Zhang et al suggested a sufficient deep learning-based network IDS approach. The IDS mainly includes a Deep Auto-encoder (DAE) based engine of feature selection and a Multi-Layer Perceptron (MLP) based classifier. A key in the feature selection is the addition of the loss functions weights of various instances and that results in the selector choosing a small group of features which efficiently represent attacks. After selecting, only these useful features are retained and a high performance is reached with a rather compact classifier.

The efficiency of the suggested method has been evaluated by experiments that have been performed on the UNSW-NB data-set, in which 12/202 properties are chosen after feature selection, which results in a selection ratio which is equal to 5.9%. After classification with the use of an MLP with two hidden layers, they have accomplished a high precision of detection of 98.80%. F_score that reflects the efficiency of the attack detection, achieved 0.952 as well. The method has exhibited a promising potential for practical applications in networks of high-speed [22].

Nathan Shone et al. proposed an innovative deep learning approach for enabling NIDS operation in modern networks. The model which they have presented combines deep and shallow learning, which can correctly analyze a wide-range of network traffic. More particularly, due to the classification power of stacked autoencoders with a typical soft-max layer is rather weak when compared with other discriminative approaches such as RF, K-NN and SVMs. They have joined the power of stacking the suggested Non-symmetrical Deep Auto Encoder (NDAE) and Random Forest (RF) precision and time-efficiency (i.e. shallow learning). They have practically made an evaluation of the model with the use of the GPU-enabled Tensor-Flow and reached good results from the analysis of the KDDCup 1999 and NSLKDD data-sets. The model has offered approximately a 5% enhancement in the precision and accelerating the speed of training of up to 98.81% [14]. They have proposed as a future work extending the capability of the model for handling zero-day attacks, and after that attempt at expanding upon their available evaluations with the use of the real-world back-bone network traffic for demonstrating the qualities of the extended model [14].

2.1.2 Boltzmann machine (BM)

BM is a network of symmetrically connected, neuron like units which make stochastic decisions concerning being either on or off. BM has a simple learning algorithm which permits them in discovering interesting properties in data-sets that are comprised of binary vectors [23],[24]. BM is utilized for solving two rather different computational tasks. For a *search* issue, the connections weights are fixed and are utilized for representing the optimization task cost function. Then, BM stochastic dynamics allow it to sample binary state vectors which denote good solutions to the issue of optimization.

For a learning task, the BM is shown a group of vectors of binary data and it has to find weights on the connections in a way that the vectors of data are sufficient solutions to the issue of optimization which is defined by these weights. For solving a learning task, BM makes numerous small alterations to their weights, and every one of the updates obliges them to solve a wide variety of search tasks. BM are primarily divided into two categories: Deep Boltzmann Machines (DBM) and Restricted Boltzmann Machines (RBM). When these RBMs are stacked on top of each other, they are known as Deep Belief Networks (DBN) [25]. Figure 4 shows a graph comparison of BM, RBM and DBM.

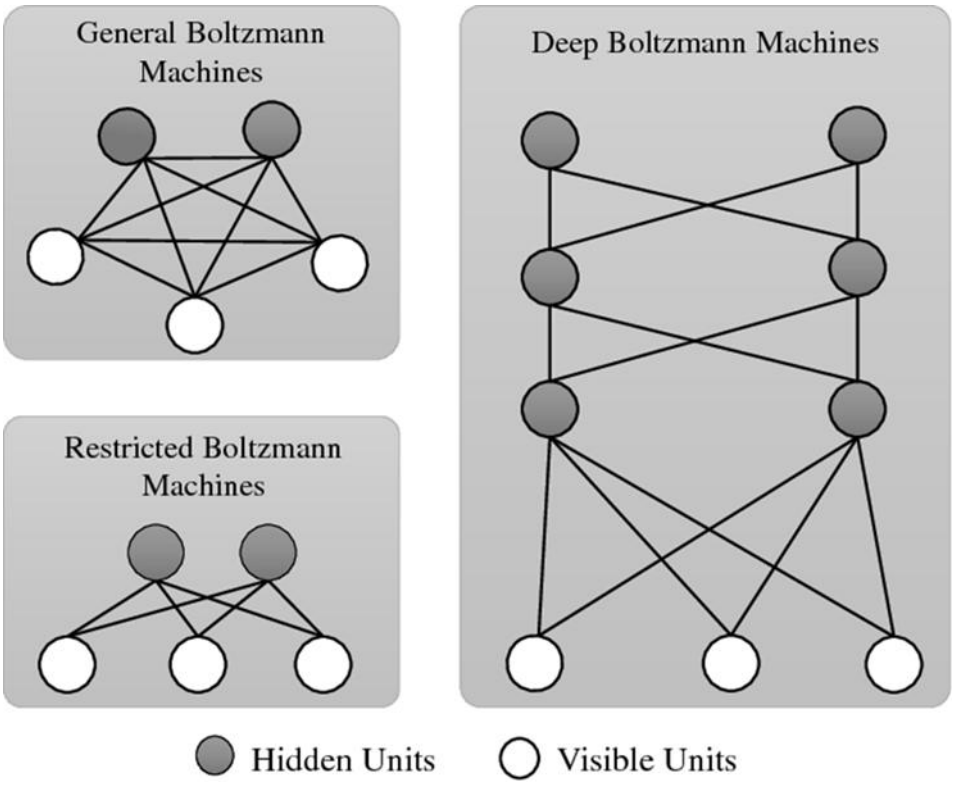

Fig. 4. Comparison of Boltzmann machines, restricted Boltzmann machines and deep Boltzmann machines [26]

A BM is fully connected between and within the layers, while in a RBM, the lateral connections in the hidden and visible layers are eliminated. Therefore, the random variables which are encoded by hidden units are not conditionally dependent taking under consideration the states of the visible units, and the other way around [26].

Ni GAO et al. suggested an approach which has been based on the multilayer DBN for the DoS attacks detection. DBN consists of numerous RBMs. Here in advance in the learning process, the training of the RBM is carried out. Then the trained features of RBM are used as an input data for learning RBM of the next layer

of the DBN stack. The effectiveness of the DBN method is tested on the KDD CUP 1999 data set. The detection precision of the DBN model had shown to be better than the SVM and ANN methods [27].

Sanghyun Seo et al. study compared the rates of intrusion detection between the NIDS with the use of only a classification model and the NIDS trained with data where noise and outliers are eliminated with the use of the RBM. Noise and outliers in KDD Cup '99 Data are eliminated via applying the data to RBM and constructing new data. The study proposed a training approach for classification models to be capable of detecting network intrusions with the use of the data that has been reconstructed based on those RBM features [28].

Xueqin Zhang [24] used two hybrid algorithms that combine SVM, RBM and DB. The algorithms have been utilized for the analysis of the false positive rate, accuracy, false negative rate and testing period with the dataset utilized for the Third International Knowledge Discovery and Data Mining Tools Competition (KDD Cup-99). Compared to one another and the conventional hybrid intrusion detection algorithm, DBN is more sufficient compared to the other, in each of speed and accuracy, and that is because of the unsupervised learning of RBM networks and the combination of the NNs at the bottom.

By comparing the conventional model which is combined by NNs and feature extraction, RBM-DBN has greatly progressed in terms of precision and false positive rate. This is attributed to the fact that the unsupervised learning has solved the drawbacks of the conventional NN and plays the role of feature extraction, and by comparing to conventional ML, RBM-DBN is advantageous in terms of time cost and, in addition, it is less time consuming in training model. RBM-DBN has been capable to solve the possible issue that the large data samples bring to the model training and testing time and that indicates the fact that RBM-DBN is appropriate for intrusion detection of the large data [24].

Khaled Alrawashdeh et al. considered a method of deep learning for detecting anomalies with the use of an RBM and a deep belief network. Their approach made use of a 1-hidden layer RBM for performing unsupervised reduction of features. The resulting weights from this RBM are passed to some other RBM that produces a deep belief network. The pretrained weights are passed to a fine tuning layer that consists of a Logistic Regression (LR) classifier that has multiclass soft-max. Their architecture has performed better than previous approaches of deep learning that have been implemented by Li and Salama [29],[30] in accuracy and speed of detection. They achieved a detection rate equal to 97.9% on the total 10% KDD-CUP 1999 testing data-set. As a future extension, they suggested applying their ML strategy on larger and more challenging data-sets that included wider range of attacks [31].

Yadigar Imamverdiyevet al. provided a comparative study of the accuracy of their suggested approach with Bernoulli-Bernoulli RBM, Gaussian–Bernoulli RBM, deep belief network type deep learning methods on DoS attack detection. Detection accuracy of the methods had been verified on the NSL-KDD data set. Higher accuracy from the proposed multilayer deep Gaussian–Bernoulli type RBM was obtained.

This method also outperforms the results of SVM, radial basis, SVM (epsilon- SVR), decision tree type machine learning methods too. They have shown that their model can improv the detection accuracy on DoS attack detection tasks compared with previous work. As a future work, they suggested to experimenting with LSTM decoders as well as deep and bidirectional LSTM encoders.

2.1.3 Recurrent neural network (RNN)

RNNs are actually NNs that utilize recurrence, and it basically uses information from a preceding forward pass over the NN. Basically, all RNN's can be considered as a recurrence relationship. RNNs are suitable and have had a considerable success when applied to issues in which the input data on which the predictions need to be done is in a form of a sequence (series of entities in which order is of an importance) [8]. Figure 5 represents a general structure of RNN, where h_k denotes the input at time step k, while x_k denotes the output [32].

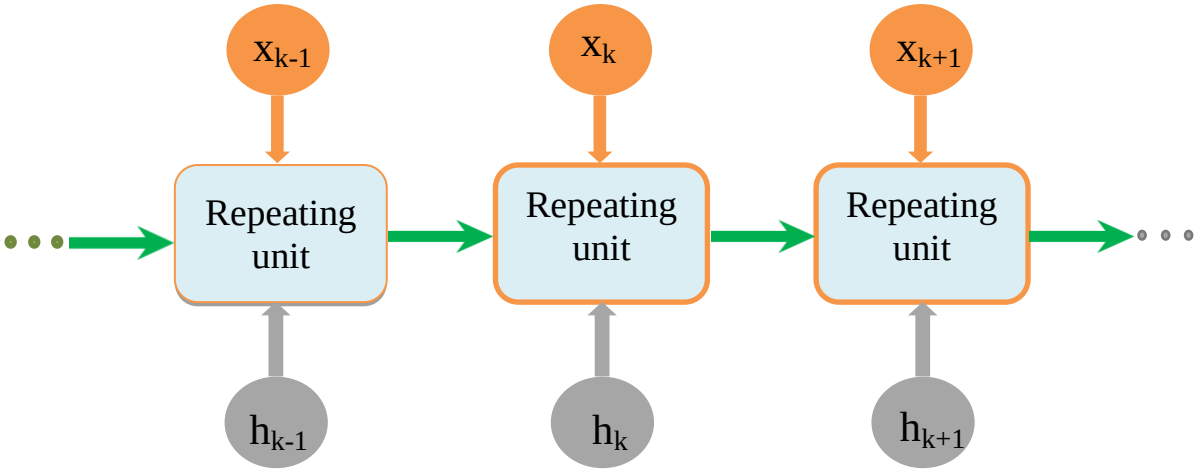

Fig. 5. A general structure of RNN [32]

Jihyun Kim et al. applied recurrent NN to IDS with Hess free optimization which is of a deep learning algorithm for intrusion detection. They have utilized the DARPA data-set for the sake of training and testing their model of intrusion detection. It has been utilized for the KDDCup-99 contest data-set. The experimental results showed that the use of RNN with Hess-free optimizing for intrusion detection is a very efficient method. As a suggestion for future work, they proposed more research for the detection of modern malwares and attacks [33].

Jihyun Kim et al. constructed a model for IDS with deep learning method. They have applied Long Short-Term Memory (LSTM) architecture to an RNN and have trained their IDS with the use of the KDDCup-99 data-set. For the stage of training, they have produced a data-set via the extraction of samples from the KDDCup-99 data-set by comparing it with other IDS classifiers, they have discovered that the attacks are efficiently detected via LSTM-RNN classifier. Due to the fact that they have the best accuracy and Detection Rate although the Rate of False Alarms is a little bit above the others. Through the performance tests, they have confirmed that the method of deep learning is sufficient for the IDS [34].

Yin Chuan-long et al. [35],[36] presented the design and implementation of the detection system based on recurrent NNs. In addition to that, they have investigated the model efficiency in binary and multi-class classifications, the number of neurons and various learning rate effects on the precision. On the other hand, they have investigated the efficiency of the naïve Bayes, multi-layer perceptron, random forest, SVMs and other approaches of ML in multi-class classification on the benchmark KDD-Cup 1999 dataset, and they have performed a comparison of the efficiency of the RNN-IDS with other approaches of ML both in binary and multi-class classifications.

Their experimental results illustrated that RNN-IDS is highly appropriate for IDSs. The efficiency of the RNN-IDS is better than the conventional approach of classification on the KDD-Cup 1999 data-set in each of the binary and multi-class classifications. The model is capable of sufficiently improving each of the precision of IDS and the capability of recognizing the type of intrusion.[35]. On the other hand, there still should be more researches for reducing the time of training with the use of the GPU acceleration, avoid exploding and vanishing gradients and research the efficiency of the LSTM classification, Bi-directional RNNs algorithm in the intrusion detection area[37].

Sara Althubiti et al. implemented a Long Short-Term Memory (LSTM) model for intrusion detection with the use of the CSIC 2010 HTTP data-set. After that, they have compiled the model using an Adam optimizer, aiming to find the best solution for the issue of binary intrusion classification with the use of the accuracy rate as an indicator of performance.

They have utilized an NN which had three layers, input, output, and hidden. They have trained the LSTM model which included an input layer with nine neurons that corresponds to the nine properties of a hidden layer with six neurons and an output layer which gave either normal or abnormal outputs with a single neuron.

The number of the iterations has been predefined as 100 epochs, the initialized weights of the network were in the range of (0 - 0.05) and the loss function was logarithmic loss [36].
They have discovered that the Adam optimizer is suitable for the LSTM RNN model in the detection of intrusions, and they have concluded the fact that LSTM RNN model utilizing Adam optimizer is capable of constructing a sufficient IDS binary classifier. In their future work suggestions, they have proposed applying LSTM to more recent intrusion detection data-sets and assess the efficiency of complicated LSTMs with various optimizers [36]. For more details please see [38].
The research of Tuan Tang et al. proposed a Gated Recurrent Unit Recurrent Neural Network (GRU-RNN) which has enabled IDSs for SDNs. The presented method has been tested with the use of the KDDCup-99 data-set, and they have accomplished a precision equal to 89% with only 6 raw features. Their experimental results have also shown that the presented GRU-RNN doesn't degrade the performance of the network.
Their approach has utilized the smallest number of features when compared with other conventional methods. And that raises the computational efficiency of the model for real time detection. Moreover, the evaluation of the efficiency of the network has shown that their method doesn't considerably impact the efficiency of the controller. This work might be further enhanced by optimizing the model and using other features for the aim of increasing the accuracy. It is also possible to attempt to implement their method in a distributed manner for the sake of reducing the overhead on the controller [37].For another works see [39].

2.1.4 Sum-product networks

Sum-product networks (SPNs) are directed cyclic graphs with variables as leaves, summations and products as weighted edges and internal nodes [40]. The summation nodes provide mixture models whereas the nodes of multiplication represent the hierarchy of the features [11]. This is why, it is possible to consider SPN as a combination of mix models and hierarchies of features, as shown in Figure 6.
Table 1 provides brief overview of all works that uses unsupervised deep learning methods, which are discussed above and what is the methods and datasets are used in those works, in addition, a brief description of those methods and the results obtained from this works are showed.

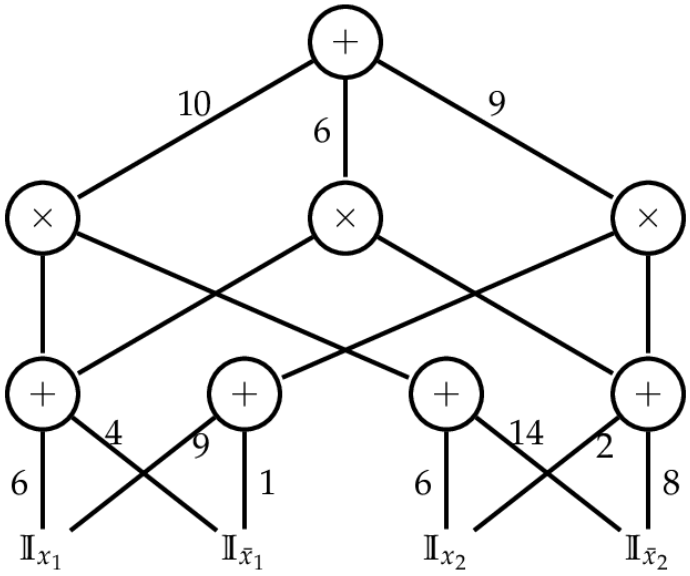

Fig. (6): An example of a sum-product network over two boolean variables X1 and X2 [40]

Table 1. Unsupervised (generative) deep learning methods applied on intrusion detection system

References	Method(s)	Description	Achievement	Dataset
Quamar Niyaz et al. [17]	Self-Taught Learning (STL), Sparse Auto Encoder (SAE)	Using Self-Taught learning as a classification method and Sparse Auto encoder for Unsupervised Feature Learning	STL achieved a classification accuracy rate of more than 98% for all types of classification.	NSL-KDD
Yisroel Mirsky et al. [18]	Auto encoder	Use of autoencoders with or without ensembles for online anomaly detection in computer networks.	the algorithm is efficient enough to run on a single core of a Raspberry PI and has an even greater potential on stronger CPUs.	Mirai Dataset
Fahimeh Farahnakian et al. [21]	Deep Autoencoder((DAE)	Four auto-encoders which the output of the autoencoder at the current layer is used as the input of the autoencoder in the next layer.	The proposed approach achieved detection accuracy of 94.71% on the total 10% KDDCUP99 test dataset	KDD CUP'99
Hongpo Zhang et al. [22]	Deep Autoencoder (DAE)	The IDS mainly consist of a DAE based feature selection engine and an MLP based classifier.	This work achieved a high detection accuracy of 98.80%	UNSW-NB
Nathan Shone et al. [14]	Non-symmetric Deep Auto-Encoder (NDAE) and Random Forest (RF)	Combination the power of stacking our proposed Non-symmetric Deep Auto-Encoder (NDAE) (deep-learning) and the accuracy and speed of Random Forest (RF)	5% improvement in accuracy and training time reduction of up to 98.81%	KDD Cup '99 and NSL-KDD
Ni GAO et al. [27]	Multilayer Deep Boltzman Network (DBN)	DBN consists of numerous RBMs. the trained features of RBM are used as an input data for learning RBM of the next layer of the DBN stack.	This work showed that DBN can learn a better generative model and perform well on intrusion recognition task.	KDD Cup '99
Xueqin Zhang [24]	Restricted Boltzmann machine (RBM) with support vector machine (SVM) and deep belief network (DBN)	Hybrid algorithms, which combine restricted Boltzmann machine (RBM) with support vector machine (SVM) and deep belief network (DBN) respectively, are used to analyze the accuracy, false positive rate, false negative rate and testing time.	The average accuracy of SVM-DBN has passed 97% and both false positive rate and false negative rate are in good performance	KDD Cup '99
Khaled Alrawashdeh et al. [31]	Restricted Boltzmann Machine (RBM) and a Deep Belief Network (DBN)	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.	Achieved a detection rate of 97.9% on the total 10% KDDCUP'99 test dataset and produced a low false negative rate of 2.47%.	KDD Cup '99
Yadigar Imamverdiyevet al. [41]	Deep Belief Network and two types of RBM	Comparative analysis of the accuracy of the proposed method with Bernoulli-Bernoulli RBM, Gaussian–Bernoulli RBM, Deep Belief Network	Their model significantly improvements in the detection accuracy on DoS attack detection tasks	NSL-KDD

Jihyun Kim et al. [33]	Recurrent Neural Network with Hessian Free Optimization	Applied Recurrent Neural Network to Intrusion Detection with Hessian Free Optimization which is one of the deep learning algorithms for intrusion detection.	The detection rate was 95.37% and false alarm rate was only 2.1%.	KDD Cup '99
Jihyun Kim et al. [34]	Long Short-Term Memory (LSTM) architecture to a Recurrent Neural Network (RNN)	Constructed an IDS model with deep learning approach, and applied Long Short-Term Memory (LSTM) architecture to a Recurrent Neural Network (RNN) and trained the IDS model using KDD Cup 1999 dataset	Highest Detection Rate and Accuracy even though the False Alarm Rate is slightly above the other ones.	KDD Cup '99
Yin Chuan-long et al. [35]	Recurrent Neural Network (RNN)	Design and implementation of the detection system based on recurrent neural networks	The model can effectively improve both the accuracy of intrusion detection and the ability to recognize the intrusion type	NSL-KDD
Sara Althubiti et al. [36]	Long Short-Term Memory (LSTM)	Model that contained an input layer with nine neurons, which corresponds to the nine features a hidden layer with six neurons, and an output layer that either produced normal or abnormal results with a single neuron.	Classifier performance is measured with an accuracy rate of 0.9944	CSIC 2010 HTTP
Tuan A Tang et al. [37]	Gated Recurrent Unit Recurrent Neural Network (GRU-RNN)	Gated Recurrent Unit Recurrent Neural Network (GRU-RNN) enabled intrusion detection systems for SDNs.	Achieved an accuracy of 89% with only six raw features.	NSL-KDD
Quamar Niyaz et al. [17]	Self-Taught Learning (STL), Sparse Auto Encoder (SAE)	Using Self-Taught learning as a classification method and Sparse Auto encoder for Unsupervised Feature Learning	STL achieved a classification accuracy rate of more than 98% for all types of classification.	NSL-KDD
Yisroel Mirsky et al. [18]	Auto encoder	Use of autoencoders with or without ensembles for online anomaly detection in computer networks.	the algorithm is efficient enough to run on a single core of a Raspberry PI and has an even greater potential on stronger CPUs.	Mirai Dataset
Fahimeh Farahnakian et al. [21]	Deep Autoencoder((DAE)	Four auto-encoders which the output of the autoencoder at the current layer is used as the input of the autoencoder in the next layer.	The proposed approach achieved detection accuracy of 94.71% on the total 10% KDDCUP99 test dataset	KDD CUP'99

2.2 Deep networks for supervised or discriminative learning

These are intended for the aim of directly providing discriminative power for the purposes of pattern classification, usually through the characterization of the rear class distributions of that are conditioned on the visible data. Target label data are usually available in direct form or in indirect form for this type of supervised learning. They're also referred to as the discriminative deep networks [11]. The most representative example of this type of architecture is the Convolutional Neural Network (CNN).

CNN is employed as a special architecture which is mainly appropriate for recognition of images. The benefit of the CNN lies in the fact that it takes little time for training, which is attributed to its structure. CNN is capable of training multi-layer nets with gradient descent for learning complicated, non-linear, high-dimensional, mappings from large datasets [42]. CNN utilizes 3 key concepts, which are: pooling, local receptive fields and shared weights (Nielsen 2015). One of the thorough researches which effectively deployed the use of CNN is AlphaGo by Google [43]. Figure 7 shows the CNN architecture:

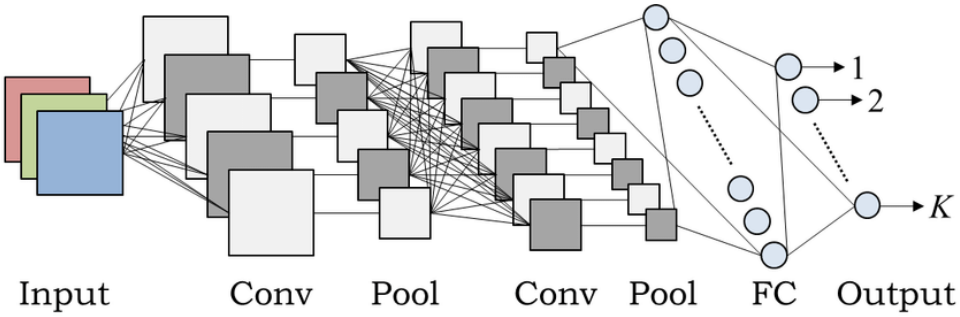

Fig. 7. An example of CNN architecture [44]

A method of anomaly intrusion detection which is based on Hybrid MLP/CNN (Multilayer Perceptron/Chaotic NN) has been presented by YAO Yu et al. A hybrid MLP/CNN NN is generated with the aim of improving the detection rate of time-delayed attacks. The simulation tests have been conducted with the use of the DARPA 98 data-set. The hybrid MLP/CNN NN model takes the result from the MLP as a chaotic neuron input in a way that chaotic neurons number has to be equivalent to the number of output nodes of the MLP. When the result of the classification of an input is analyzed by MLP, it may be forwarded and retained by the CNN which is connected to the MLP output node. They have realized classification with memory of anomaly events with the use of the real-time MLP classification and the memorial CNN functionality. Due to the hybrid NN has flexible time-delay criterion and capability, it can achieve high rates of intrusion detection and low rate of false alarms. The method has a considerable potential of high scalability and the ability of recognizing new patterns of attacks by the detection of the BSM strings [45].

Kehe Wu et al. proposed a NIDS model utilizing CNNs. They have used CNN for automatically selecting traffic properties from raw data-set and they have set the coefficient of the cost function weight of every class, depending on its numbers to solve the issue of imbalanced data-set. The model does not merely reduce the False Alarm Rate (FAR), it also enhances the precision of the class with small numbers. For the aim of additionally reducing the computational costs, they have converted the raw traffic vector format into the image format. They have utilized the original KDDCup-99 data-set for evaluating the efficiency of the suggested CNN model. The experimental results have shown that the precision, FAR and computational cost of the presented model has a better performance compared to the conventional standard algorithms. More improvements can be made for the detection accuracy of this work. It is possible modifying the CNN model structure for the sake of achieving the goal. In addition to that, due to the fact that the detection time is also key to intrusion detection, it is necessary to ensure that the model is capable of meeting the time requirements of the IDS when enhancing the accuracy of detection [46]. For other information please refer to [47].

Table 2 provides a brief overview of all works that uses supervised deep learning methods, which are discussed above and what is the methods and datasets that are used in these references, in addition, a brief description of those methods and the results from this works are showed.

Table 2: Supervised (discriminative) Deep Learning Methods applied on intrusion detection system

References	Method(s)	Description	Achievement	Dataset
YAO Yu et al. [45]	MLP/CNN	A hybrid MLP/CNN neural network was built in order to enhance the detection rate of time-delayed attacks.	High intrusion detection rates and low false alarm rates	DARPA 1998
Kehe Wu et al. [46]	Convolutional Neural Networks (CNNs).	Used CNN to select traffic features from raw dataset automatically, and they set the cost function weight coefficient of each class based on its numbers to solve the imbalanced dataset problem.	Accuracy, FAR and calculation cost of the proposed model performs better than traditional standard algorithms	NSL-KDD

2.3 Hybrid deep networks

Hybrid deep architectures are made up of the combination of each of the generative architecture and the discriminative architecture. Essentially, this architecture has the aim of distinguishing data in addition to a discriminative approach. On the other hand, in the early step, it has been significantly helpful with the results of the generative architectures [11].

Deep Neural Network (DNN) is an example of hybrid deep architectures, it can be defined as a multi-layer network that has cascaded fully connected hidden layers and is usually utilize stacked RBM as a pre training stage. Numerous other generative models which may be thought of as hybrid or discriminative models when classification task is added with class labels. Figure 8 illustrates the basic architecture of a DNN [48].

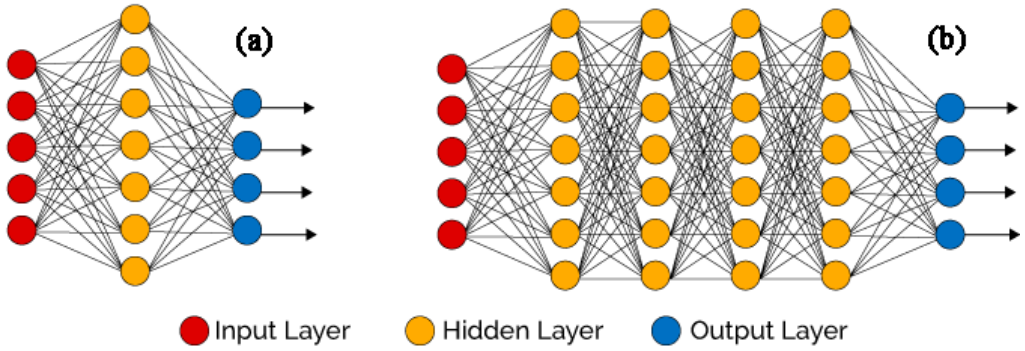

Fig. 8. (a) Simple neural network architecture; (b) Simple architecture of deep neural network (DNN)[49]

Jin Kim et al. proposed a research of an intelligent IDS utilizing the DNN model for effectively detecting attacks. They have utilized the popular KDDCup 1999 data-set for intrusion detection for testing and training. The testing data has been created via data pre-processing and extraction of samples in order to meet the aim of the study. A DNN model which consists of 4 hidden layers and 100 hidden units has been utilized for the proposed IDS of the presented study as its classification algorithm and utilized the ReLU function as the activation function of the hidden layers. In addition to that, this study utilized the adaptive moment (Adam) optimizer, a stochastic approach of optimization for DNN learning. The results showed a considerably high precision and detection rate, which has reached approximately 99%. Moreover, the FAR has reached approximately 0.08% [49].

Tuan A Tang et al. built a flow-based system of anomaly detections with the use of a DNN model for an IDS and trained with using the NSLKDD Data-set. In the work they have proposed, they have utilized only 6 main characteristics (which can easily be obtained in an SDN environment) taken from the 41 features of NSLKDD Data-set. Through the experimental work, they have discovered an optimal hyper-parameter for DNN and confirmed the rates of detection and the false alarms. The model has reached the efficiency with a precision of approximately 75.75% which is rather reasonable from merely utilizing 6 main network features. As a future work, they have proposed implementing this method in a real SDN environment with real network traffic and evaluated the efficiency of the entire network according to latency and throughput [50].

Sasanka Potluri et al. developed an accelerated DNN model for identifying the anomalies in the network data. NSLKDD data-set has been utilized for computing the duration of training and for analyzing the efficiency of the method of detection. The input layer stands for each 41 feature that is fed into the DNN. The hidden layer1 is the 1st auto-encoder that chooses the 20/40 features from the input data. This is why, the AE has 20 neurons inside. The hidden layer2 is another AE that includes 10 neurons and chooses the 10 of 20 features that have been from hidden layer1. The first two hidden layers come into the pre-training procedure of the DNN. The hidden layer3 is the Soft-Max layer that will additionally decrease the number of features to five and, in addition, performs the fine tuning with supervised learning.

The fundamental emphasis was on evaluating the efficiency of the DNN training which is related to various types of processors and numbers of cores. Accelerating the process of training with the use of the multi-core CPU was faster when compared to the approach of serial training. However, the GPU's were not capable of achieving the projected efficiency because of the type of data that has been utilized. It is possible to extend the work via the analysis of the efficiency of the accelerated platforms (each of the multicore CPU and GPU) with much complicated data for the application of intrusion detection. In addition to that, selecting various features out of all 41 can be considered for improving the accuracies of detecting DNN based IDS [51].
Table 3 provide brief overview of all works that uses hybrid deep learning methods, which are discussed above and what its methods and datasets that are used in those works, in addition, a brief description of those methods and the results from this works are showed.

Table 3. Hybrid deep learning methods applied on intrusion detection system

References	Method(s)	Description	Achievement	Dataset
Jin Kim et al. [49]	Deep Neural Network (DNN)	There are four hidden layers and 100 hidden nodes in the DNN model, and used the ReLU activation function, and used the Adam optimizer for DNN learning.	High accuracy and detection rate averaging 99%. FAR achieved 0.08%.	KDD Cup '99
Tuan A Tang et al. [50]	Deep Neural Network (DNN)	Constructed a simple DNN with an input layer, three hidden layers and an output layer. The input dimension is six and the output dimension is two. The hidden layers contain twelve, six and three neurons respectively.	Performance with accuracy of 75.75% for just using six basic network features.	NSL-KDD
Sasanka Potluri et al. [51]	Deep Neural Network (DNN)	41 features are used as input to the DNN. 1st hidden layer is AE is used to select 20 features out of the 41 features. 2nd hidden layer is another AE (with 10 neurons) are used to select the 10 features out of 20. The (1st and 2nd) hidden layers are fed to the pre-training process of the DNN. 3rd hidden layer (SoftMax) is used to select 5 features out of 10 and also used as a fine tuner with supervised learning.	The detection accuracies were reliable on NSL-KDD dataset by generalizing the attack classes to fewer types.	NSL-KDD

3. Popular intrusion detection datasets for deep learning

Several research groups now put together many types of data both for their own study purposes and to provide data to community repositories. Here the most popular intrusion detection datasets used in DL research are explained.

3.1 DARPA, KDD99, and NSL-KDD datasets

DARPA 1998 has gathered and deal out the first standard data by MIT Lincoln Lab under "Defense Advanced Research Projects Agency" (DARPA) and "Air Force Research Lab" (AFRL) sponsorship for evaluating computer network IDS. Due to the fact that DARPA data-set is made up of raw files, scholars must obtain characteristics from those files for using them in ML algorithms [52].

The KDDCup 1999 data-set has been utilized in DARPA's IDS evaluation program [53]. The data is made up of 4 GB-worth of compressed tcp dump data which has resulted from seven weeks of network traffic. Which might be processed is approximately 5 m. connection records, every one of which is with nearly 100 bytes. It is made up of about 4,900,000 single connection vectors, every of one of those vectors includes 41 features. Which include Basic features (such as packet size and Protocol type), Domain knowledge features (such as the Number of unsuccessful logins) and timed observation features (such as the percentage of connections having SYN errors). Every one of the vectors is either labelled as normal or an attack (there are 22 defined types of attacks) [14].

The newer NSLKDD data-set has been created by Tavallaee et al. for overcoming the intrinsic issues of the KDD 1999 dataset that have been discussed in [54]. It is an enhanced version from the KDD-99 dataset [55]. In the NSL-KDD dataset, three main issues have been solved. First, duplicate records in the training and test sets have been reduced for eliminating them from biasing classification systems toward the most redundant records. Second, the training and test sets have been created via selecting many different records from various parts of the traditional KDD-99 dataset for achieving authentic results at the same time as applying classification systems. Finally, the unbalanced issue between the number of the testing and the training has been addressed for being reduced. Nevertheless, this new version of the data-set is still suffering from some issues that have been discussed by Mc-Hugh in [56] and might not be an optimal representation of the available real networks. The newest research of NIDS still utilizes this data-set, therefore, there is a belief that it is still an efficient benchmark which helps scholars in comparing various approaches.

The NSLKDD data-set is essentially the same structure as the KDDCup 1999 data-set (in other words it has 22 patterns of attacks or normal traffic, and fields for 41 properties). Figure 9 gives a general summary for these correlated data-sets (DARPA, KDD-99, and NSLKDD). DARPA is a base raw data-set. KDD-99 is the feature extracted version of DARPA data-set. NSLKDD is the size reduced and duplicates eliminated version of the KDD-99 data-set.

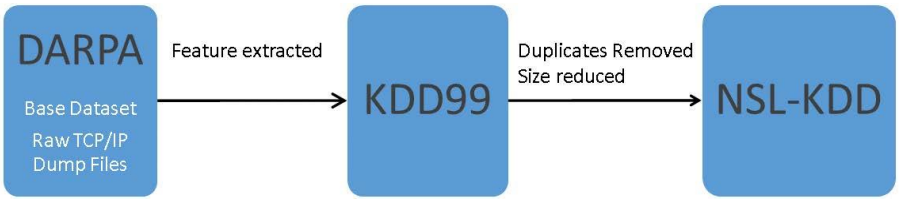

Fig. 9. The correlation between the main and the extracted datasets [57].

3.2 ECML-PKDD 2007 dataset

The ECML-PKDD 2007 data-set has been created for the European Conference on ML and Knowledge Discovery in the year of 2007. The ECML/PKDD Discovery Challenge was a data mining competition that has been held combined with the 18th European Conference on Machine Learning (ECML). The dataset is described in extensible markup language (XML). All of the sample is represented by a unique id and consists of the three main parts that are context, class and query [58][59].

3.3 ISOT (information security and object technology) dataset

ISOT dataset is a combination of openly available different botnets and normal data-sets containing 1,675,424 total traffic flow. For malicious traffic in ISOT, it was collected from French chapter of honeynet project that consist of Storm and Waledac botnets. Non malicious traffic has been obtained from Traffic Lab Ericson Research in Hungary. After that, this traffic was combined with another dataset that is generated by L. Berkeley National Laboratory (LBNL). This compilation includes general traffic from several types of

application besides that HTTP website browsing, World of Warcraft traffic, and traffic from Azureus bit-torent client. This is why, this traffic is a considerably huge data-set for Ericson Laboratory [58].

3.4 HTTP CSIC 2010 dataset

The HTTP CSIC2010 data-set involves several thousands of web requests that are automatically produced and developed at Information Security Institute of CSIC (Spanish Research National Council). The dataset may be utilized to test systems of web attack protection. This data consist of 6,000 normal requests and over 25,000 abnormal requests and HTTP requests are categorized as normal or abnormal [60].

3.5 CTU-13 (czech technical university) dataset

CTU-13 dataset is the combination of seizures of 13 different malware in a nonfictional network environment. The aim of this data-set is capturing real mixed botnet traffic. Infected hosts generated botnet traffic and verified normal hosts generated normal traffic. Lastly, Background traffic is a remainder of the traffic that we do not know what it is for sure[61]. The UNSW-NB15 dataset has lately been released. This dataset includes nine distinct modern types of attacks and many different real normal activities. This dataset includes 49 features with the class label which involves network traffic characteristics utilizing the flow based between hosts (in other words, server-to-client or client-to-server) and the packet header [62][63].

3.6 The ADFA dataset

In 2013, Australian Defence Force Academy Linux Data-set was released by the Defence Force Academy in Australia in New South Wale University. In order to assess host-based IDS, ADFA dataset (Linux dataset) was generated on an Ubuntu Linux 11.04 host OS with Apache 2.2.17 running PHP 5.3.5. FTP, SSH, MySQL 14.14 and TikiWiki were started. This dataset involves normal and attack Linux based system calls traces. The aim of ADFA dataset is to take the place of existing benchmark data sets, because these benchmark datasets have been unsuccessful in reflecting the properties of current computer Systems [64].

3.7 ISCX IDS 2012

The data-set which is utilized for testing the classifiers is the Information Security Centre of Excellence (ISCX 2012) data-set generated by Sh. Ali et. al. from University of Brunswick ISCX [65]. The data-set has been particularly created in order to develop, test and evaluate algorithms of network intrusion and anomaly detection. The data-set includes 17 properties and the tag value means whether the flow is normal or abnormal. The whole ISCX labeled data-set includes approximately 1512000 packets with 19 features and gathered over a week of network activity (normal and attack).

4. Frameworks for deep learning implementation

Deep learning architecture combine implementing the algorithms of modularized deep learning, methods of optimization, methods of distribution and support to infrastructures. In this section, we briefly introduce the most popular frameworks used for implementing deep learning algorithms.

4.1 NVIDIA cuDNN

The NVIDIA CUDA DNN library (cuDNN) is a GPU-accelerated library of primitives for DNNs. The cuDNN ensured highly tuned standard routine implementations like backward and forward convolutions, normalizing, pooling and activation layers. Numerous frameworks such as TensorFlow, Theano, Caffe and Torch are dependent on the acceleration of high-performance GPU [64]. NVidia-1 is now a driving force in

developing hardware technologies like Graphical Processing Unit (GPU) and other processors which are capable of accelerating learning and improving the efficiency of the approaches of DL [8][66].

4.2 Tensor-flow

Tensor-Flow (TF), is the successor of Dist-Belief, is the distributed system for training NNs which Google has been using since the year of 2011. TF is an open source library for numerical calculation, which has been created by Google brain team [67]. TF has been programmed with a Python API over a C/C++ engine, which makes it operate faster. TF has CUDA support. Nearly every type of networks may be built with the use of TensorFlow, even though it allows no hyper-parameter configuration of deep networks. Tensor-Flow provides an interface for C++ as well.

Moreover, the Tensor-Flow team has released TF-Slim which is a high-level library for defining complicated models in Tensor-Flow. The library TF-Slim offers common abstractions that give users the ability of quickly and concisely defining models, at the same time as keeping the transparency of the model architecture in addition to maintaining its hyper-parameters explicit. TF has the maximum number of community support to implement models of deep learning. TF is quite popular in researching deep learning because it is flexible for many different algorithms. In addition to that, it supports low level and high-level network training with numerous GPU, robust and provides consistency of the updates of parameters [66].

4.3 Theano

Theano has been produced by the ML team at Montreal University. It is a cross platform open source python library. Theano is a Python library which is utilized for defining, optimizing and evaluating the multidimensional array mathematical expression. Theano offers high network modelling capability, dynamic code creation and speed with numerous GPU support. Nevertheless, Theano provides low-level API and includes many complicated compilations which are usually rather time consuming. At the same time, Theano has many different resources of learning and is still utilized by a considerable number of scholars and developers [68].

4.4 Keras

Keras has been developed for implementing deep learning in Theano and TF written in Python. It gives the ability of high-level NN API for fast implementation of deep learning algorithms. The fundamental key point of Keras is the fact that it supports Theano and TF and can run on top of either Theano or TF, widely used deep learning implementation framework and allows extensible, modular and user platform utilizing Python [66]. Because of the Theano and TF design, it is possible writing high level libraries such as Keras that might run on any of the backends. Programs of TF and Theano are, in general, larger than the equivalent programs of Keras [66]. Figure 10 shows the Keras model.

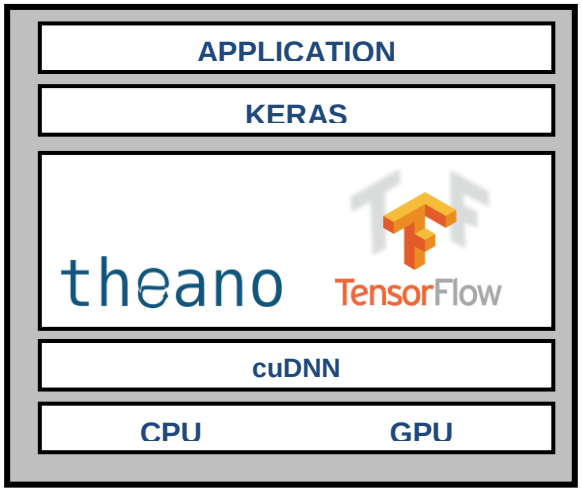

Fig. 10. The architecture of Keras (Parvat et al. 2017)

4.5 Caffe

Caffe is an open source deep learning toolkit, which has been developed by Berkeley center for vision and learning together with community contributors. It has an expressive architecture with modularity, expression and speed. Caffe is a framework that is utilized to express algorithms in a modular form. It ensures C++ core language and binding support in MATLAB and Python. This toolkit provides a complete architecture used to train, test and deploy the deep learning model. In addition to that, NVidiaGPU provides Caffe support for accelerated learning of deep learning [69].

4.6 Deeplearning-4j

Deeplearning-4j is an open source deep learning model which has been programmed with Java, CUDA, C, Scala, and C++. It has been released under the Apache license 2.0. It has been developed by an ML team and supported by a start-up company which is known as the Skymind. This framework operates on OSs such as Linux, Windows, Android, and OS X [70]. Dl4j is an open source, distributed and commercial ML tool-kits for implementing deep learning, which has been developed by Skymind. The framework integrates Spark and Hadoop with CPU and GPU-enabled for simple and fast prototyping of the implementation of DNN [71].

4.7 Torch/PyTorch

Torch is an open source deep learning framework which has been based on the scripting language of Lua which is simple, fast, and portable. This framework is a scientific calculation framework offering wide support for ML mechanisms. Lately, Py-Torch has witnessed a high degree of adoption in the deep learning framework community and it's considered as an opponent to Tensor-Flow. Py-Torch is essentially a port to the Torch framework which is utilized for the construction of deep NNs and the execution of tensor calculation high are high regarding complexity.

Py-Torch has been recently developed at Facebook and is a front-end Torch integration for sufficient performance deep learning development with considerable GPU support. It ensures Python front-end which enables constructing dynamic NN. On the other hand, the tool-kit has been newly released and no much community support, learning resources and evaluating its efficiency [72].

4.8 Cognitive network toolkit (CNTK)

It has been developed by Microsoft Research for the sake of providing a unified frame-work for popular deep learning mechanisms. It offers multi-GPU parallelizing of learning approaches and performs an implementation of automatic differentiation and stochastic gradient descent. This tool-kit has been released in the year of 2015 and described as Visual Studio (VS) for ML. For the ones who have utilized VS for programming, it might be a thin and more sufficient way of getting into deep learning. The efficiency is, in general, quite good. It is a relatively new addition to the publicly presented tool-kits and utilization is presently less than numerous others[66][72].

4.9 MX-net

MX-Net is a deep learning framework that has been programmed using C++ with numerous language bindings, and it offers support for distributed computing, which includes multi GPU. It offers accessing both lower-level constructs in addition to the higher/symbolic level API. The efficiency is considered on par with other efficient systems, which include Tensor-Flow, Caffe and some others [72].

4.10 DIGITS

DIGITS has been developed by NVIDIA and is a web-based tool for the development of deep networks. In numerous ways, it's similar to Caffe, and it utilizes a text file instead of a programming language, for describing the parameters and the network. It has a tool for network visualization, which is why, errors in the text can be more easily spotted. Moreover, it has tools for the visualization of the learning process and has multiple GPU support [72].

5. Conclusion

In this paper, we have presented an overview of deep learning and what the most definitions emphasize on. We have reviewed the latest papers of deep learning in the intrusion detection domain. Some widely used deep learning architectures are investigated and selected applications to intrusion detection are highlighted. More specifically, three classes of deep learning architectures, namely the Generative (unsupervised), Discriminative (supervised) and Hybrid deep architecture are discussed with its methods in details. Those three classes provide a lot of flexibility and have proven themselves over decades to be useful and reliable in a wide range of problems. As an example, the unsupervised architecture can be classified into AE, the sum-product Network (SPN), BM and RNN. We have viewed the related works for each class and methods mentioned above that are applied in intrusion detection domain. After that we have pointed out the most popular intrusion detection Datasets used for deep learning and the most popular deep learning implementation frameworks. Regarding the comparative results of the related works, the supervised learning algorithms deals with labeled data, since it is rough to obtain labeled data while dealing with big data, it cannot provide satisfactory performance in these cases, therefore the unsupervised learning algorithms is used to process the unlabeled data and if we don't have any idea about the output data, we can also use unsupervised learning algorithms to predicate the optimal solutions (outputs) on the obverse if we have the input and output we use supervised learning algorithms.

Datasets for intrusion detection are very important for training and testing systems. Dataset always contains a huge number of features where most of them are redundant or irrelevant. Deep learning methods are preferably used as feature extraction or reducing complex features. We may use deep learning methods if we have no idea about the correlation between raw input data and targeted classification output.

Based on previous works, it was found that AE and RNN are used more than CNN in the classification, also the performance of AE and RNN are better than CNN, while CNN is faster than AE and RNN. It is worth mentioning here if researchers need to use CNN method then they may convert the raw input into image file first before using this approach. This is due to the fact that the CNN algorithm is very effective in dealing with files that are images, for example Facebook uses CNN for automatic tagging algorithms, Amazon for generating product recommendations and Google for search through among users' photos.

In summary, it can be said that most of the discussed techniques have shown a capability of obtaining high accuracy levels in a more automatic way. As a possible direction for future work, AE and the RNN based methods can be combined in models for accuracy improvements and it is recommended to use feature extraction and feature selection as a hybrid approach to increase the accuracy of intrusion detection.

6. References

[1] R. Bace and P. Mell, "NIST special publication on intrusion detection systems," BOOZ-ALLEN AND HAMILTON INC MCLEAN VA, 2001.

[2] A. Lazarevic, V. Kumar, and J. Srivastava, "Intrusion detection: A survey," in *Managing Cyber Threats*, Springer, 2005, pp. 19–78.

[3] S. K. Wagh, V. K. Pachghare, and S. R. Kolhe, "Survey on intrusion detection system using machine learning techniques," *Int. J. Comput. Appl.*, vol. 78, no. 16, 2013.

[4] W. Stallings, "Cryptography and network security: principles and practice," *Pract. (6th Ed.*, vol. 9, p. 9685, 1998.

[5] M. H. Aghdam and P. Kabiri, "Feature Selection for Intrusion Detection System Using Ant Colony Optimization.," *IJ Netw. Secur.*, vol. 18, no. 3, pp. 420–432, 2016.

[6] C.-F. Tsai, Y.-F. Hsu, C.-Y. Lin, and W.-Y. Lin, "Intrusion detection by machine learning: A review," *Expert Syst. Appl.*, vol. 36, no. 10, pp. 11994–12000, 2009.

[7] B. Durakovic, "Design of experiments application, concepts, examples: State of the art," *Period. Eng. Nat. Sci.*, vol. 5, no. 3, 2017.

[8] S. Pouyanfar *et al.*, "A Survey on Deep Learning: Algorithms, Techniques, and Applications," *ACM*

Comput. Surv., vol. 51, no. 5, p. 92, 2018.
[9] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning. nature 521 (7553): 436," *Google Sch.*, 2015.
[10] L. Deng and X. Li, "Machine learning paradigms for speech recognition: An overview," *IEEE Trans. Audio. Speech. Lang. Processing*, vol. 21, no. 5, pp. 1060–1089, 2013.
[11] L. Deng and D. Yu, "Deep learning: methods and applications," *Found. Trends® Signal Process.*, vol. 7, no. 3–4, pp. 197–387, 2014.
[12] Y. Bengio, N. Boulanger-Lewandowski, and R. Pascanu, "Advances in optimizing recurrent networks," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2013, pp. 8624–8628.
[13] E. Aminanto and K. Kim, "Deep learning in intrusion detection system: An overview," in *2016 International Research Conference on Engineering and Technology (2016 IRCET)*, 2016.
[14] N. Shone, T. N. Ngoc, V. D. Phai, and Q. Shi, "A deep learning approach to network intrusion detection," *IEEE Trans. Emerg. Top. Comput. Intell.*, vol. 2, no. 1, pp. 41–50, 2018.
[15] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep learning*, vol. 1. MIT press Cambridge, 2016.
[16] M. E. Aminanto and K. Kim, "Deep learning-based feature selection for intrusion detection system in transport layer." 2016.
[17] A. Javaid, Q. Niyaz, W. Sun, and M. Alam, "A deep learning approach for network intrusion detection system," in *Proceedings of the 9th EAI International Conference on Bio-inspired Information and Communications Technologies (formerly BIONETICS)*, 2016, pp. 21–26.
[18] R. Raina, A. Battle, H. Lee, B. Packer, and A. Y. Ng, "Self-taught learning: transfer learning from unlabeled data," in *Proceedings of the 24th international conference on Machine learning*, 2007, pp. 759–766.
[19] Y. Mirsky, T. Doitshman, Y. Elovici, and A. Shabtai, "Kitsune: an ensemble of autoencoders for online network intrusion detection," *arXiv Prepr. arXiv1802.09089*, 2018.
[20] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural Comput.*, vol. 18, no. 7, pp. 1527–1554, 2006.
[21] F. Farahnakian and J. Heikkonen, "A deep auto-encoder based approach for intrusion detection system," in *Advanced Communication Technology (ICACT), 2018 20th International Conference on*, 2018, pp. 178–183.
[22] H. Zhang, C. Q. Wu, S. Gao, Z. Wang, Y. Xu, and Y. Liu, "An Effective Deep Learning Based Scheme for Network Intrusion Detection," in *2018 24th International Conference on Pattern Recognition (ICPR)*, 2018, pp. 682–687.
[23] L. Deng, G. Hinton, and B. Kingsbury, "New types of deep neural network learning for speech recognition and related applications: An overview," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, 2013, pp. 8599–8603.
[24] X. Zhang and J. Chen, "Deep learning based intelligent intrusion detection," in *Communication Software and Networks (ICCSN), 2017 IEEE 9th International Conference on*, 2017, pp. 1133–1137.
[25] R. Salakhutdinov, A. Mnih, and G. Hinton, "Restricted Boltzmann machines for collaborative filtering," in *Proceedings of the 24th international conference on Machine learning*, 2007, pp. 791–798.
[26] I. Bozcan, Y. Oymak, I. Z. Alemdar, and S. Kalkan, "What is (missing or wrong) in the scene? A Hybrid Deep Boltzmann Machine For Contextualized Scene Modeling," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 1–6.
[27] N. Gao, L. Gao, Q. Gao, and H. Wang, "An intrusion detection model based on deep belief networks," in *Advanced Cloud and Big Data (CBD), 2014 Second International Conference on*, 2014, pp. 247–252.
[28] S. Seo, S. Park, and J. Kim, "Improvement of Network Intrusion Detection Accuracy by Using Restricted Boltzmann Machine," in *Computational Intelligence and Communication Networks (CICN), 2016 8th International Conference on*, 2016, pp. 413–417.
[29] M. A. Salama, H. F. Eid, R. A. Ramadan, A. Darwish, and A. E. Hassanien, "Hybrid intelligent intrusion detection scheme," in *Soft computing in industrial applications*, Springer, 2011, pp. 293–303.
[30] Y. Li, R. Ma, and R. Jiao, "A hybrid malicious code detection method based on deep learning," *methods*, vol. 9, no. 5, 2015.
[31] K. Alrawashdeh and C. Purdy, "Toward an online anomaly intrusion detection system based on deep

learning," in *Machine Learning and Applications (ICMLA), 2016 15th IEEE International Conference on*, 2016, pp. 195–200.

[32] A. Khan and F. Zhang, "Using recurrent neural networks (RNNs) as planners for bio-inspired robotic motion," in *Control Technology and Applications (CCTA), 2017 IEEE Conference on*, 2017, pp. 1025–1030.

[33] J. Kim and H. Kim, "Applying recurrent neural network to intrusion detection with hessian free optimization," in *International Workshop on Information Security Applications*, 2015, pp. 357–369.

[34] J. Kim, J. Kim, H. L. T. Thu, and H. Kim, "Long short term memory recurrent neural network classifier for intrusion detection," in *Platform Technology and Service (PlatCon), 2016 International Conference on*, 2016, pp. 1–5.

[35] C. Yin, Y. Zhu, J. Fei, and X. He, "A deep learning approach for intrusion detection using recurrent neural networks," *IEEE Access*, vol. 5, pp. 21954–21961, 2017.

[36] S. Althubiti, W. Nick, J. Mason, X. Yuan, and A. Esterline, "Applying Long Short-Term Memory Recurrent Neural Network for Intrusion Detection," in *SoutheastCon 2018*, 2018, pp. 1–5.

[37] T. A. Tang, S. Ali, R. Zaidi, D. Mclernon, L. Mhamdi, and M. Ghogho, "Deep Recurrent Neural Network for Intrusion Detection in SDN-based Networks," in *2018 4th IEEE Conference on Network Softwarization and Workshops (NetSoft)*, 2018, pp. 25–29.

[38] B. Durakovic and H. Basic, "Continuous Quality Improvement in Textile Processing by Statistical Process Control Tools: A Case Study of Medium-Sized Company," *Period. Eng. Nat. Sci.*, vol. 1, no. 1, 2013.[39]

[40] H. Poon and P. Domingos, "Sum-product networks: A new deep architecture," in *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, 2011, pp. 689–690.

[41] Y. Imamverdiyev and F. Abdullayeva, "Deep Learning Method for Denial of Service Attack Detection Based on Restricted Boltzmann Machine," *Big Data*, vol. 6, no. 2, pp. 159–169, 2018.

[42] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[43] D. Silver *et al.*, "Mastering the game of Go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, p. 484, 2016.

[44] A. Hidaka and T. Kurita, "Consecutive dimensionality reduction by canonical correlation analysis for visualization of convolutional neural networks," in *Proceedings of the ISCIE International Symposium on Stochastic Systems Theory and its Applications*, 2017, vol. 2017, pp. 160–167.

[45] Y. Yao, Y. Wei, F. Gao, and G. Yu, "Anomaly intrusion detection approach using hybrid MLP/CNN neural network," in *Intelligent Systems Design and Applications, 2006. ISDA'06. Sixth International Conference on*, 2006, vol. 2, pp. 1095–1102.

[46] K. Wu, Z. Chen, and W. Li, "A Novel Intrusion Detection Model for a Massive Network Using Convolutional Neural Networks," *IEEE Access*, vol. 6, pp. 50850–50859, 2018.[47]

[48] J. Thanaki, *Python Natural Language Processing*. Packt Publishing Ltd, 2017.

[49] J. Kim, N. Shin, S. Y. Jo, and S. H. Kim, "Method of intrusion detection using deep neural network," in *Big Data and Smart Computing (BigComp), 2017 IEEE International Conference on*, 2017, pp. 313–316.

[50] 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 *Wireless Networks and Mobile Communications (WINCOM), 2016 International Conference on*, 2016, pp. 258–263.

[51] S. Potluri and C. Diedrich, "Accelerated deep neural networks for enhanced Intrusion Detection System," in *Emerging Technologies and Factory Automation (ETFA), 2016 IEEE 21st International Conference on*, 2016, pp. 1–8.

[52] K. K. R. Kendall, "A database of computer attacks for the evaluation of intrusion detection systems." Massachusetts Institute of Technology, 1999.

[53] 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," COLUMBIA UNIV NEW YORK DEPT OF COMPUTER SCIENCE, 2000.

[54] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in *Computational Intelligence for Security and Defense Applications, 2009. CISDA 2009. IEEE Symposium on*, 2009, pp. 1–6.

[55] L. Dhanabal and S. P. Shantharajah, "A study on NSL-KDD dataset for intrusion detection system

based on classification algorithms," *Int. J. Adv. Res. Comput. Commun. Eng.*, vol. 4, no. 6, pp. 446–452, 2015.

[56] 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. Secur.*, vol. 3, no. 4, pp. 262–294, 2000.

[57] A. Özgür and H. Erdem, "A review of KDD99 dataset usage in intrusion detection and machine learning between 2010 and 2015," *PeerJ Prepr.*, vol. 4, p. e1954v1, 2016.

[58] B. Gallagher and T. Eliassi-Rad, "Classification of http attacks: a study on the ECML/PKDD 2007 discovery challenge," Lawrence Livermore National Lab.(LLNL), Livermore, CA (United States), 2009.

[59] K. Kato and V. Klyuev, "An intelligent DDoS attack detection system using packet analysis and Support Vector Machine," *Int. J. Intell. Comput. Res. IJICR*, vol. 14, no. 5, p. 3, 2014.

[60] C. Torrano-Gimenez, A. Pérez-Villegas, G. Álvarez, E. Fernández-Medina, M. Malek, and J. Hernando, "An Anomaly-based Web Application Firewall.," in *SECRYPT*, 2009, pp. 23–28.

[61] M. Xie, J. Hu, and J. Slay, "Evaluating host-based anomaly detection systems: Application of the one-class svm algorithm to adfa-ld," in *Fuzzy Systems and Knowledge Discovery (FSKD), 2014 11th International Conference on*, 2014, pp. 978–982.

[62] J. Moustafa, Nour Slay, "UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," in *Military Communications and Information Systems Conference (MilCIS), 2015*, 2015, pp. 1–6.

[63] N. Moustafa and J. Slay, "The significant features of the UNSW-NB15 and the KDD99 data sets for network intrusion detection systems," in *Building Analysis Datasets and Gathering Experience Returns for Security (BADGERS), 2015 4th International Workshop on*, 2015, pp. 25–31.

[64] O. Yavanoglu and M. Aydos, "A review on cyber security datasets for machine learning algorithms," in *Big Data (Big Data), 2017 IEEE International Conference on*, 2017, pp. 2186–2193.

[65] A. Shiravi, H. Shiravi, M. Tavallaee, and A. A. Ghorbani, "Toward developing a systematic approach to generate benchmark datasets for intrusion detection," *Comput. Secur.*, vol. 31, no. 3, pp. 357–374, 2012.

[66] H. F. Nweke, Y. W. Teh, M. A. Al-Garadi, and U. R. Alo, "Deep learning algorithms for human activity recognition using mobile and wearable sensor networks: State of the art and research challenges," *Expert Syst. Appl.*, 2018.

[67] M. Abadi *et al.*, "Tensorflow: a system for large-scale machine learning.," in *OSDI*, 2016, vol. 16, pp. 265–283.

[68] J. Bergstra *et al.*, "Theano: A CPU and GPU math compiler in Python," in *Proc. 9th Python in Science Conf*, 2010, vol. 1.

[69] Y. Jia *et al.*, "Caffe: Convolutional architecture for fast feature embedding," in *Proceedings of the 22nd ACM international conference on Multimedia*, 2014, pp. 675–678.

[70] A. Parvat, J. Chavan, S. Kadam, S. Dev, and V. Pathak, "A survey of deep-learning frameworks," in *Inventive Systems and Control (ICISC), 2017 International Conference on*, 2017, pp. 1–7.

[71] S.-M. Lee, S. M. Yoon, and H. Cho, "Human activity recognition from accelerometer data using Convolutional Neural Network," in *Big Data and Smart Computing (BigComp), 2017 IEEE International Conference on*, 2017, pp. 131–134.

[72] B. J. Erickson, P. Korfiatis, Z. Akkus, T. Kline, and K. Philbrick, "Toolkits and libraries for deep learning," *J. Digit. Imaging*, vol. 30, no. 4, pp. 400–405, 2017.