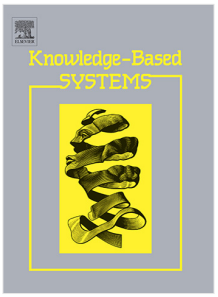
Journal Pre-proof



Deep learning approaches for anomaly-based intrusion detection systems: A survey, taxonomy, and open issues

Arwa Aldweesh, Abdelouahid Derhab, Ahmed Z. Emam

PII: S0950-7051(19)30489-7
DOI: https://doi.org/10.1016/j.knosys.2019.105124
Reference: KNOSYS 105124

To appear in: *Knowledge-Based Systems*

Received date : 7 February 2019
Revised date : 7 September 2019
Accepted date : 11 October 2019

Please cite this article as: A. Aldweesh, A. Derhab and A.Z. Emam, Deep learning approaches for anomaly-based intrusion detection systems: A survey, taxonomy, and open issues, *Knowledge-Based Systems* (2019), doi: https://doi.org/10.1016/j.knosys.2019.105124.

*Conflict of Interest Form

AUTHOR DECLARATION

Manuscript no.: KNOSYS-D-19-00370R1
Manuscript title: Deep Learning Approaches for Anomaly-Based Intrusion Detection Systems: A Survey, Taxonomy, and Open Issues

We wish to confirm that there are no known conflicts of interest associated with this publication and there has been no significant financial support for this work that could have influenced its outcome.

We confirm that the manuscript has been read and approved by all named authors and that there are no other persons who satisfied the criteria for authorship but are not listed. We further confirm that the order of authors listed in the manuscript has been approved by all of us.

We confirm that we have given due consideration to the protection of intellectual property associated with this work and that there are no impediments to publication, including the timing of publication, with respect to intellectual property. In so doing we confirm that we have followed the regulations of our institutions concerning intellectual property.

We understand that the Corresponding Author is the sole contact for the Editorial process (including Editorial Manager and direct communications with the office). She is responsible for communicating with the other authors about progress, submissions of revisions and final approval of proofs. We confirm that we have provided a current, correct email address which is accessible by the Corresponding Author and which has been configured to accept email from (aldeweesh.ar@gmail.com).

Signed by all authors as follows:

Arwa Aldweesh	Dr. Abdelouahid Derhab	Prof. Ahmed Emam
24/05/2019	24/05/2019	24/05/2019



Deep Learning Approaches for Anomaly-Based Intrusion Detection Systems: A Survey, Taxonomy, and Open Issues

Arwa Aldweesh[a], Abdelouahid Derhab[b], Ahmed Z. Emam[c]
King Saud University, Riyadh, 12372, Saudi Arabia
[a]aldeweesh.ar@gmail.com, [b]abderhab@ksu.edu.sa, [c]aemam@ksu.edu.sa

Abstract

The massive growth of data that are transmitted through a variety of devices and communication protocols have raised serious security concerns, which have increased the importance of developing advanced intrusion detection systems (IDSs). Deep learning is an advanced branch of machine learning, composed of multiple layers of neurons that represent the learning process. Deep learning can cope with large-scale data and has shown success in different fields. Therefore, researchers have paid more attention to investigating deep learning for intrusion detection. This survey comprehensively reviews and compares the key previous deep learning-focused cybersecurity surveys. Through an extensive review, this survey provides a novel fine-grained taxonomy that categorizes the current state-of-the-art deep learning-based IDSs with respect to different facets, including input data, detection, deployment, and evaluation strategies. Each facet is further classified according to different criteria. This survey also compares and discusses the related experimental solutions proposed as deep learning-based IDSs.

By analysing the experimental studies, this survey discusses the role of deep learning in intrusion detection, the impact of intrusion detection datasets, and the efficiency and effectiveness of the proposed approaches. The findings demonstrate that further effort is required to improve the current state-of-the art. Finally, open research challenges are identified, and future research directions for deep learning-based IDSs are recommended.

***Keywords*:** Intrusion detection, Anomaly detection, Deep learning.

1. Introduction

In recent years, the world has witnessed a significant evolution in the different areas of connected technologies such as smart grids, the Internet of vehicles, long-term evolution, and 5G communication. By 2022, it is expected that the number of IP-connected devices will be three times larger than the global population, producing 4.8 ZB of IP traffic annually, as reported by Cisco [1]. This accelerated growth raises overwhelming security concerns due to the exchange of huge amounts of sensitive information through resource-constrained devices and over the untrusted "Internet" using heterogeneous technologies and communication protocols. To

maintain sustainable and secure cyberspace, advanced security controls and resilience analysis [2] should be applied in the earlier stages before deployment.

The applied security controls are responsible for preventing, detecting, and responding to attacks. For detection purposes, an intrusion detection system (IDS) is a widely used technique for detecting internal and external intrusions that target a system, as well as anomalies that indicate potential intrusions and suspicious activities. An IDS involves a set of tools and mechanisms for monitoring the computer system and the network traffic, in addition to analysing activities with the aim of detecting possible intrusions targeting the system [3]. An IDS can be implemented as signature-based, anomaly-based, or hybrid IDS. In signature-based IDS, intrusions are detected by comparing monitored behaviours with pre-defined intrusion patterns, while anomaly-based IDS focuses on knowing normal behaviour in order to identify any deviation [4]. Different techniques are used to detect anomalies, such as statistical-based, knowledge-based, and machine learning techniques; recently, deep learning methods have been investigated [5].

Deep learning is an advanced branch of machine learning which uses multi-layer networks. The layers are connected through neurons, which represent the mathematical computation of the learning processes [6]. Deep learning has shown success in different fields, such as image and video recognition, audio processing, natural language processing, autonomous systems, and robotics, etc. [7]. Deep networks are classified into three main categories: 1) generative architectures, which apply unsupervised learning to learn automatically from an unlabelled dataset, 2) discriminative architectures, which apply supervised learning mainly to distinguish patterns for prediction tasks, and 3) hybrid architectures, which incorporate both generative and discriminative models [8].

Researchers have put forward many machine learning approaches for anomaly-based intrusion detection. With the emergence of new technologies and increased Internet traffic, large-scale and multi-dimensional data are produced, and attack scenarios are becoming more sophisticated, which makes approaches that rely on shallow machine learning ineffective in dealing with the growing security challenges. Deep learning techniques have shown their effectiveness with respect to dimensionality reduction and classification tasks. In the context of a deep learning-based IDS, deep networks learn from historical traffic data, which is made up of both normal and anomalous traffic. Deep networks can automatically reduce the network traffic complexity to find the correlations among data without human intervention [7]. Furthermore, deep learning is more powerful in detecting zero-day attacks and sophisticated attack patterns by learning from a large number of training samples to build the detection model.

Unlike shallow machine learning algorithms, deep learning approaches can be designed to perform feature extraction and classification tasks together. Furthermore, they have data dependency and hardware dependency [9]. Currently, researchers are investigating deep learning models for intrusion detection and have proposed several intrusion detection approaches using deep learning as dimensionality reduction techniques, classifiers, or both.

In the literature, we can find several intrusion detection surveys [10,11,12,13,14,15], which address the different facets of using deep learning in cybersecurity. This survey extends the previous surveys by focusing on the utilization of deep learning in IDSs. It provides the reader with a comprehensive review, analysis, and taxonomy that covers the main deep learning architectures that have been adopted for use in IDS. The survey reviews and compares the key previous surveys that addressed deep learning methods for cybersecurity.

In addition, this research proposes a fine-grained taxonomy for deep learning-based IDS, which categorizes the described solutions according to different characteristics, including input data strategy, detection strategy, deployment strategy, and evaluation strategy. Each strategy is further classified according to different criteria. The provided taxonomy aims to provide researchers with the required elements for developing, deploying, and evaluating deep learning-based IDS.

Furthermore, this survey comprehensively analyses and discusses the deep learning-based intrusion detection approaches published in different academic venues. It compares solutions in terms of the characteristics of the proposed taxonomy as closely as possible, although some characteristics are not considered by some solutions, as will be discussed later in this survey. The comparison is based on input datasets, dimensionality reduction techniques, detection techniques, classification type, testing methodology, processing component, and effectiveness and efficiency evaluation metrics. The survey concludes by deriving several challenges and insights for future research directions.

The main contributions of the survey are the following:

- We present and compare earlier deep learning-focused cybersecurity surveys, and identify their gaps, and the main differences with respect to our survey.
- We propose a novel fine-grained taxonomy that classifies the current state-of-the-art deep learning-based IDS solutions with respect to different facets, including input data strategy, detection strategy, deployment strategy, and evaluation strategy. The facets are further classified with respect to a set of criteria. This fine-grained taxonomy is intended to help researchers compare the different deep learning-based solutions in detail.
- We propose a comparative and descriptive analysis of the reviewed deep learning-based IDS methods, by providing a side-by-side comparison in a tabular form. The methods are compared in terms of feature learning, classification technique, testing methodology, effectiveness, and efficiency.
- We highlight open research challenges and outline possible future research directions.

The remainder of this paper is organized as follows. Section 2 reviews and compares the previous surveys on deep learning-focused cybersecurity. Section 3 provides an overview of deep learning architectures. Then, an overview of IDS methodologies, techniques, and utilization of deep learning in intrusion detection follows in Section 4. In Section 5, a fine-grained taxonomy for classifying deep learning-based IDSs is provided. In Section 6, a description and comparison of related experimental studies are presented, followed by a discussion of the

compared studies and the obtained findings in Section 7. Challenges and future directions are provided in Section 8. Finally, Section 9 concludes this survey.

2. Related Surveys

In the literature, there are six related surveys that cover different aspects of deep learning in cybersecurity. Surveys [10,11,12] mainly focus on deep learning for intrusion detection. The other deep learning surveys do not consider the intrusion detection domain. On the other hand, [13,14,15] are not completely dedicated to deep learning but also describe shallow machine learning techniques. Table 1 compares different aspects of the previous surveys with respect to 1) the outlines of the survey, 2) the existence of an IDS taxonomy, 3) the security domain focus 4) the covered deep learning architectures, and 5) the reviewed deep learning-based solutions. In addition, we specify the type of the conducted study as either descriptive or comparative.

Aminato and Kim [10] focused on the implementation of a stacked auto-encoder. The same authors extended their work in [11] to cover all architectures, and mainly discussed the use of deep learning for IDS, as either a feature extractor or a classifier. Kown et al. [12] focused on generative deep learning architectures. Xin et al. [14] discussed only DBN, RNN and CNN deep learning architectures. Furthermore, Hodo et al. [13] reviewed deep learning architectures and presented some solutions as examples of each architecture, except for CNN, which had not yet been proposed for IDS at the time of their survey. This survey will review and compare more recent solutions, up to late 2018. Unlike Al-Garadi et al. [15], which considered the use of deep learning for different security aspects of the Internet of Things (IoT), our work focuses on deep learning for intrusion detection security, without restriction to a specific application domain. It also covers both generative and discriminative deep learning architectures, and proposes a taxonomy for analysing and comparing deep learning-based IDS solutions.

Our work differs from the abovementioned surveys in the following points:

- It provides both comparative and descriptive analysis of the reviewed solutions, whereas the abovementioned surveys present either descriptive or comparative analysis.
- It proposes a novel fine-grained taxonomy that classifies the current state-of-the-art deep learning-based IDS solutions, whereas the abovementioned surveys either provide a generic IDS classification or no classification.
- The comparative study covers recent deep learning-based intrusion detection systems published between 2014 and 2018.

Table 1: IDS survey comparison

Article	Outlines of the Survey	IDS Taxonomy	Focused Security Domain	Covered Deep Learning Architectures				Surveyed Deep Learning-based IDS Experiments	
				AE	RBM DBN	RNN	CNN	Years	Study Type and Covered Studies
Xin et al. (2018) [14]	- Similarities and differences between shallow and deep learning. - Machine learning and deep learning methods used in IDS and some related research.	-	Cyber-security	-	✓	✓	✓	2015-2017	Comparative, includes: [16,17,18,19,20, 21,22,23,24,25,26,27 ,28,29,30]
Kim et al. (2017) [10]	- Limitations of shallow machine learning approaches. - Stacked AE implementation for IDS including their previous experiments.	-	IDS	✓	-	-	-	2015-2017	Comparative, includes: [31,31,32,34]
Aminanto et al. (2017) [11]	-The role of deep learning methods in IDS (feature extraction or classification). -Discussed 12 deep learning solutions, which use feature extraction and classification	-	IDS	✓	✓	-	✓	2011-2017	Descriptive, includes: [17,23,35,36,37]
Hodo et al. (2017) [13]	-Generic IDS taxonomy. - Reviewed machine learning algorithms and analysed performance. - Reviewed deep learning methods and some proposed approaches.	Generic (for IDS techniques)	IDS	✓	✓	✓	-	2013-2016	Descriptive, includes: [17,20,23,38,39]
Kwon et al. (2017) [12]	-Generative deep learning architectures. -Reviewed 7 deep learning-based solutions. -Discussed their own experiment using a fully connected network (FCN).	-	IDS	✓	✓	✓	-	2011-2016	Descriptive, includes: [20,23,35,37,38,40, 41]
Al-Garadi et al. (2018) [15]	- Attack surfaces in IoT and potential threats. -Reviewing and comparing shallow and deep learning methods for IoT security. -The applicability of shallow and deep learning for each layer of IoT.	-	IoT Security	✓	✓	✓	✓	2013-2018	Comparative, includes: [31,32,35,38,39,42, 43,44,45]
This Survey	-Reviews previous surveys. -Provides a taxonomy for deep learning-based IDS. -Conducts a comparative and descriptive analysis on the proposed deep learning-based IDSs.	Fine-grained taxonomy for deep learning-based IDSs	IDS	✓	✓	✓	✓	2014-2018	Descriptive and comparative analysis.

3. Background of Deep Learning Architectures

Deep learning is an evolution of machine learning that originated from artificial neural networks (ANN). It is composed of different layers constructing a deep neural network (DNN). Deep learning is a machine learning algorithm with more than two layers of neural networks, useful for modelling complex concepts and relationships [46]. Currently, deep learning is being investigated in different research areas, such as image recognition, speech recognition, natural language processing, social network filtering, etc. Deep learning algorithms are different in their ability to simultaneously accomplish feature learning and classification or clustering tasks in addition to finding correlations among large scale data from different sources [9]. Deep learning architectures are classified into three main categories: generative, discriminative, and hybrid architectures [8].

3.1 Generative Architectures

Generative (or unsupervised) deep learning architectures can learn automatically from unlabelled raw data to accomplish different tasks. The following are the most common architectures in this category.

3.1.1 Auto-Encoder (AE)

An AE is a deep neural network introduced by Holden et al. [47], typically used for dimensionality reduction by producing better data representation than the raw data input. An AE consists of input and output layers with an equal number of feature vectors, in addition to a hidden layer with low-dimensional feature representation. An AE combines an encoder and decoder, and trains them together using backpropagation. The encoder extracts the raw features and learns the data representation by converting the input into low-dimensional abstraction. Then, the decoder receives the low-dimensional representations and reconstructs the original features [48]. The conceptual structure of an AE is represented in Fig. 1. There are several AE extensions, including stacked AE (SAE), sparse AE, and de-noising AE.

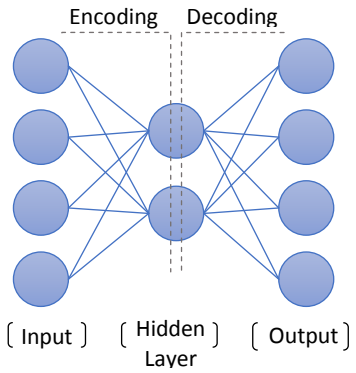


Figure 1: Conceptual structure of AE

1. **SAE:** More than one hidden layer is cascaded to construct a deep network and form the SAE. Input features are learned progressively in depth to construct a new data representation [31].

2. **Sparse AE:** The hidden units in sparse AE have sparsity constraints. The AE remains useful for learning data representations even if there are many hidden units. Sparsity constraints aim to produce low average output by making a large number of neurons inactive most of the time [5].
3. **De-noising AE:** The principle of de-noising is the use of corrupted data as input to produce a refined data representation, where the hidden layers use only robust feature vectors [50].

3.1.2 Restricted Boltzmann Machine (RBM)

The Boltzmann machine (BM) is a probabilistic neural network introduced by Hinton and Sejnowsk [51]. A BM network consists of binary units paired symmetrically, and decides which ones are activated. However, there are many connections among units, which results in very slow learning [52].

RBM is a unidirectional model proposed by Smolensky in 1986 to solve issues arising from the complexity of BM. The principle behind RBM is to eliminate the connections among neurons in the same layer. Fig. 2 shows the difference between BM and RBM architectures. RBM consists of a visible layer for the initial input variables, and a hidden layer holding latent (hidden) variables. Each unit in the visible layer is connected to all units in the hidden layer with associated weights. The hidden units learn the feature distribution from input variables [53]. RBM is practically used as an initial stage of another learning network, either as a feature extractor in preprocessing or for initializing the parameters of the other network. In addition, RBM can be used as a classification model. Larochelle and Bengaio [54] trained the discriminative restricted Boltzmann machine (DRBM) to be a nonlinear, stand-alone classifier. If more than one Boltzmann machine is cascaded, it is called a deep Boltzmann machine (DBM).

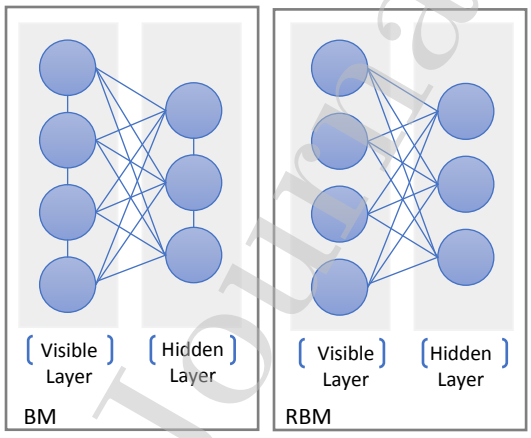


Figure 2: Difference between BM and RBM architectures

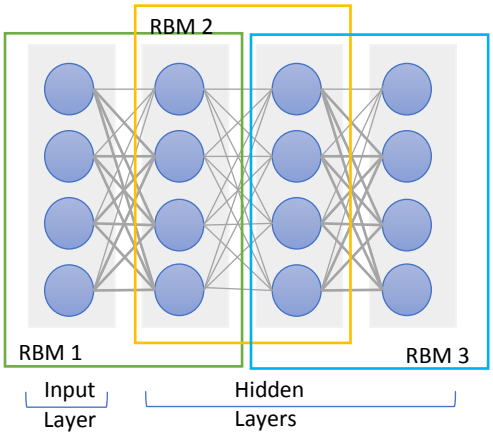


Figure 3: Conceptual architecture of DBN

3.1.3 Deep Belief Network (DBN)

A deep belief network (DBN) is composed of stacked RBMs, which are trained in a greedy layer-wise fashion, as shown in Fig. 3. Each RBM is trained on top of the previous one, where

each hidden layer of an RBM is considered an input to the next RBM. This training mechanism results in an efficient and fast deep learning algorithm [55]. In practical applications, a DBN is applied for dimensionality reduction as well as a stand-alone classifier when an additional discrimination layer is added [56].

3.1.4 Recurrent Neural Network (RNN)

An RNN is a dynamic feed-forward neural network introduced by Hopfield in 1982. It is distinguished by its ability to learn sequential data over timesteps. In conventional feed-forward neural networks, the output of each unit depends on the current input, with no dependency between input and previous output of the same unit. However, some applications rely on sequential data, such as speech recognition or time-series data such as sensor data, in which each sample depends on the analysis of previous samples. Therefore, the conventional feed-forward neural network is not appropriate for these kinds of applications. RNNs handle this issue by modelling data as time series. The output of each hidden unit in RNN is based on the current timestep input and the output of the previous timestep. Each hidden unit has a feedback loop that passes the unit output back to the same unit to be associated with the next timestep. Fig. 4 shows the difference between the hidden unit in feed-forward neural networks and RNNs [57]. RNNs have been extended with different memory unit variants, including long short time memory (LSTM) and gated recurrent unit (GRU).

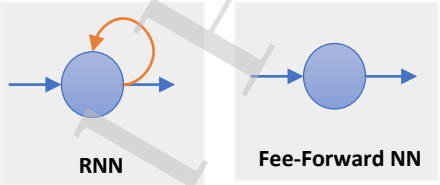


Figure 4: Difference between hidden units in RNN and feed-forward neural networks

1. **Long short time memory (LSTM):** LSTM solves the vanishing gradient problem in vanilla RNN. It has the ability to learn long-term dependencies through the use of the gating mechanism. Each LSTM unit is equipped with a memory cell that holds old states [58].
2. **Gated recurrent unit (GRU):** GRU is a lightweight version of LSTM. It is constructed of simpler architecture, merging the gates and integrating the states [59].

3.2 Discriminative Architectures

Discriminative (or supervised) architectures are mainly applied to labelled data to distinguish patterns for prediction tasks. The following are the most common discriminative deep learning architectures.

3.2.1 Convolutional Neural Network (CNN)

CNNs were introduced to handle intensive connections between DNN layers. CNNs train multiple layers with nonlinear mappings to classify high-dimensional input data into a set of

classes at the output layer. A CNN is composed of convolutional layers and pooling layers, followed by optional fully connected layers. Convolutional layers encompass filters, which represent smaller dimensional slices of the input data. The filters convolve across the entire input to produce feature maps. The pooling layer then operates over the feature maps to perform subsampling, which reduces the dimensionality of the feature maps [46].

The convolutional layers directly receive multi-dimensional inputs, avoiding the data reconstruction in conventional architectures. Therefore, CNNs are well fitted to multi-dimensional data such as images and speech signals. Furthermore, a CNN requires fewer parameters with the same depth of network compared with other deep networks, which reduces complexity and speeds the learning process [60]. Recently, CNNs have also been investigated as feature extractors and classifiers for intrusion detection, due to their ability to deal with complex data.

3.3 Hybrid Architectures

Hybrid architectures incorporate both generative and discriminative models. This takes advantage of generative features in early phases and discriminative features at later stages to distinguish data.

3.3.1 Generative Adversarial Network (GAN):

A GAN is an example of a hybrid deep network introduced by a group of researchers at Google Brain in 2014. A GAN is essentially an inner cycle of two networks: a generative network and a discriminative network. A GAN relies on a minimax game in which one network seeks to maximize the function value and the other tries to minimize it. In each adversarial round, the generator produces random samples from noise. The discriminator then receives the random data along with real samples and attempts to distinguish between them. The generator performs well when it successfully floods the discriminator while the discriminator is trained to be an accurate classifier [61].

4 Intrusion Detection System (IDS)

Intrusions are a series of related malicious actions performed by an internal or external intruder that attempt to compromise the targeted system [62]. Intrusion detection involves monitoring computer systems and network traffic and analysing activities to detect possible intrusions targeting the system. For this purpose, a set of tools and mechanisms known as an intrusion detection system (IDS) are applied [3].

Generally, most IDSs provide common capabilities to maintain network security. An IDS starts by gathering data from observed activities. It performs comprehensive logging for event-related data and correlates events from multiple sources. The core of an IDS is the detection engine, which uses a variety of methodologies and related techniques, depending on the situation. Moreover, prevention capabilities can also be provided. In this case, the system is called an intrusion detection and prevention system (IDPS) [3].

4.1 Intrusion Detection Methodologies

Signature-based detection and anomaly-based detection are the most popular methodologies used for intrusion detection. They are commonly used together, either integrated or separately, to increase detection accuracy.

4.1.1 Signature-Based Detection

A signature is a pre-configured pattern that matches a known intrusion. Signature-based detection is defined in [3] as "the process of comparing signatures against observed events to identify possible incidents". Signature-based detection is also known as misused detection or knowledge-based detection due to the use of knowledge gathered from previous intrusions and vulnerabilities. However, this method is not sufficient to detect unknown intrusions and variants of known ones, since their patterns are unfamiliar. Moreover, keeping the knowledge up-to-date is another problem, since it is a time-consuming and difficult process [4].

4.1.2 Anomaly-Based Detection

An anomaly is any deviation from normal behaviours. Anomaly-based detection, also called behaviour-based detection, is defined as "the process of comparing normal activities against observed events to identify significant deviations" [3]. Anomaly-based detection consists of three general modules: 1) Parameterization: representing the observed behaviour in a profile that consists of different attributes and characteristics of what to investigate, such as network connections, host, and applications [63]. 2) Training: processing the parametrized profiles to build a classification model that distinguishes between normal and abnormal behaviours. 3) Detection: using the constructed classification model to detect new traffic anomalies [64].

To accomplish the abovementioned anomaly detection stages, different techniques can be used:

1. **Statistical-based techniques:** The anomaly is identified by scoring the degree of deviation from a specific behaviour using standard deviations, means, thresholds, and probabilities [65]. The earliest approaches used univariant models. Later approaches adopted multivariant models and time-series models [5].
3. **Knowledge-based techniques:** These rely on the existence of past knowledge of observed parameters under normal and abnormal operations. Knowledge-based techniques can use expert systems, finite state machines, description languages, and data clustering [66].
4. **Machine learning and deep learning techniques:** Learning algorithms enhance the performance of an IDS through learning from past experiences without human intervention. A variety of machine learning algorithms have been investigated for IDS application. The most common algorithms used in the literature include support vector machine (SVM), naïve Bayes, genetic algorithm (GA), k-nearest neighbour (K-NN), decision tree (DT), fuzzy logic, and artificial neural networks (ANN) [13]. Moreover, intrusion datasets with imbalanced class distributions can be addressed with sampling techniques or imbalanced learning algorithms [67]. Recently, researchers have used the unique nature of the deep network for feature learning and classification, as will be discussed later in this survey.

4.2 Intrusion Detection: From Shallow to Deep Learning

Deep learning as defined in [6] is "a particular kind of machine learning that achieves great power and flexibility by learning to represent the world as nested hierarchy of concepts, with each concept defined in relation to simpler concepts, and more abstract representations computed in terms of less abstract ones." This definition shows that deep learning has a huge advantage over machine learning. In shallow machine learning, features are often identified by an expert and then encoded to a data type, which is a time-consuming and difficult task when dealing with large-scale data. The main difference between shallow machine learning and deep learning is the ability of deep architectures to learn features with different levels of abstractions at different processing layers without human intervention. Thus, deep models automatically find complicated correlations and mapping between raw input and output [68].

Moreover, deep learning supports end-to-end problem solving, while shallow machine learning algorithms divide a problem into several parts, solving each of them individually and subsequently combining them again to obtain the results.

Deep learning methods can be utilized in anomaly detection for both dimensionality reduction and classification tasks. With the rapid increase in transmitted traffic, manual feature engineering fails to cope with multi-dimensional and large-scale data, whereas deep learning models automatically learn complex data. In addition, deep learning models can be used to deal with the dynamic nature of network traffic and continuous changes in attack scenarios [69]. Thus, deep learning models can be trained with large amounts of historical data to build an anomaly detection model. The model classifies the new traffic into either the normal or anomaly class. If a multi-class classification is used, the model can further classify the infected traffic to different

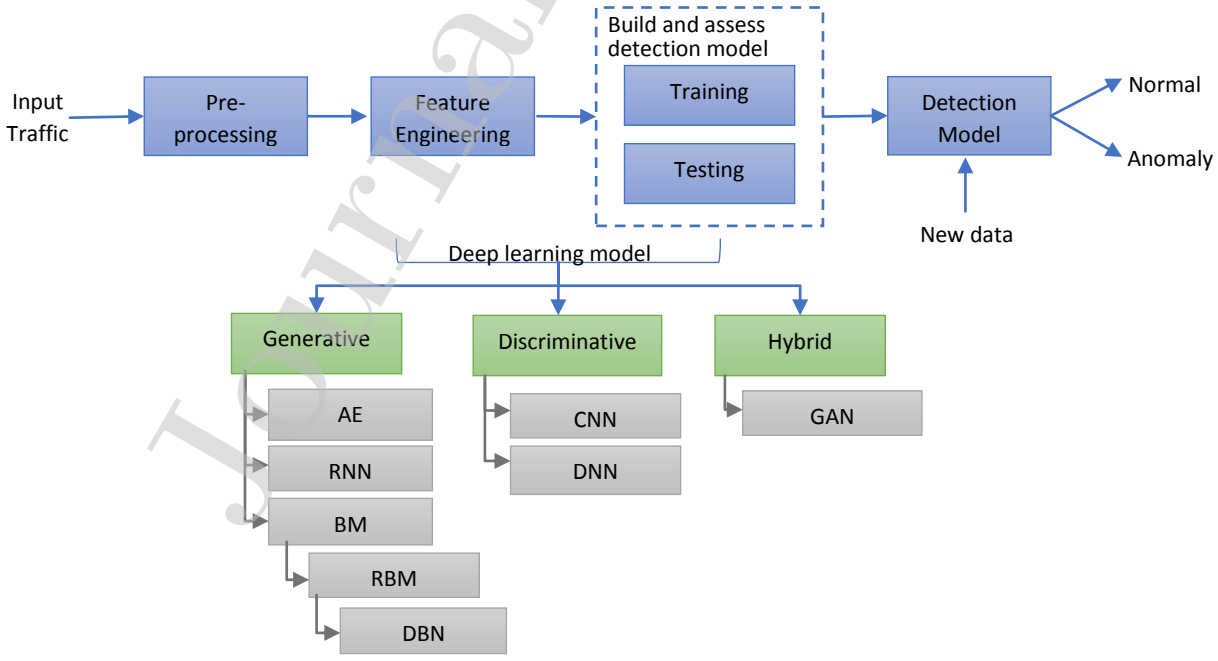


Figure 5: Deep learning-based IDS Architecture

classes and subclasses of attacks. Fig. 5 illustrates the overall architecture of a deep learning-based IDS. In terms of complexity, deep learning approaches involve time-consuming and intensive mathematical operations performed through multiple hidden layers and a large number of parameters during the training phase. However, deep learning algorithms inherently deal with a large number of matrix multiplication operations using advanced processing hardware. With the rapid advancement of processing components, GPUs and AI accelerators are affordable and have recently been integrated with smart phones and IoT devices.

5. Taxonomy of a Deep Learning-Based IDS

Several generic IDS taxonomies have been provided in the literature. Debar et al. [70] and Liao et al. [4] proposed taxonomies that highlighted different IDS characteristics. Hindy et al. [71] provided a global overview of IDS design, while Balasaraswathi et al. [70] focused on feature selection techniques. Butun et al. [66] is another example of specific taxonomy, which concentrated on an IDS designed for a wireless sensor network (WSN). This survey, on the other hand, proposes a fine-grained taxonomy, which classifies the state-of-the-art of deep learning-based IDSs with respect to four different aspects: input strategy, detection strategy, deployment strategy, and performance evaluation strategy. Furthermore, each aspect is classified with respect to a set of criteria, as shown in Fig. 6.

5.1 Input Data Strategy

Data are the core component when evaluating any IDS. Data can be collected from different sources, including host logs, network traffic, and application data. The input dataset, which is used for training by real-time solutions, can be either generated in a simulated or real environment. For instance, Kang [73] implemented IDS in a simulated in-vehicle network while Anyanwu et al. [74] implemented IDS in a real-time application.

However, most of the researchers use the existing benchmark datasets to evaluate their IDS in off-line mode, since these datasets do not support real-time processing [75]. The most adopted benchmark network intrusion dataset is KDD99 [76]. KDD99 was released in 1999 based on the DARPA 1998 dataset, and consists of 4,900,000 labelled records. Each record consists of 41 features, including basic, content, and network features. Based on these features, the records can be classified into 22 attack types and four main categories, namely, denial of service (DoS) attacks, user to root (U2R) attacks, root to local (R2L) attacks, and probing (probe) attacks, in addition to the normal category [77]. The second common dataset is NSL-KDD, an improved version of KDD99 released in the last decade [78]. NSL-KDD has several advantages over KDD99, such as the exclusion of duplicate records in both the training and testing sets, and a reasonable number of records [79].

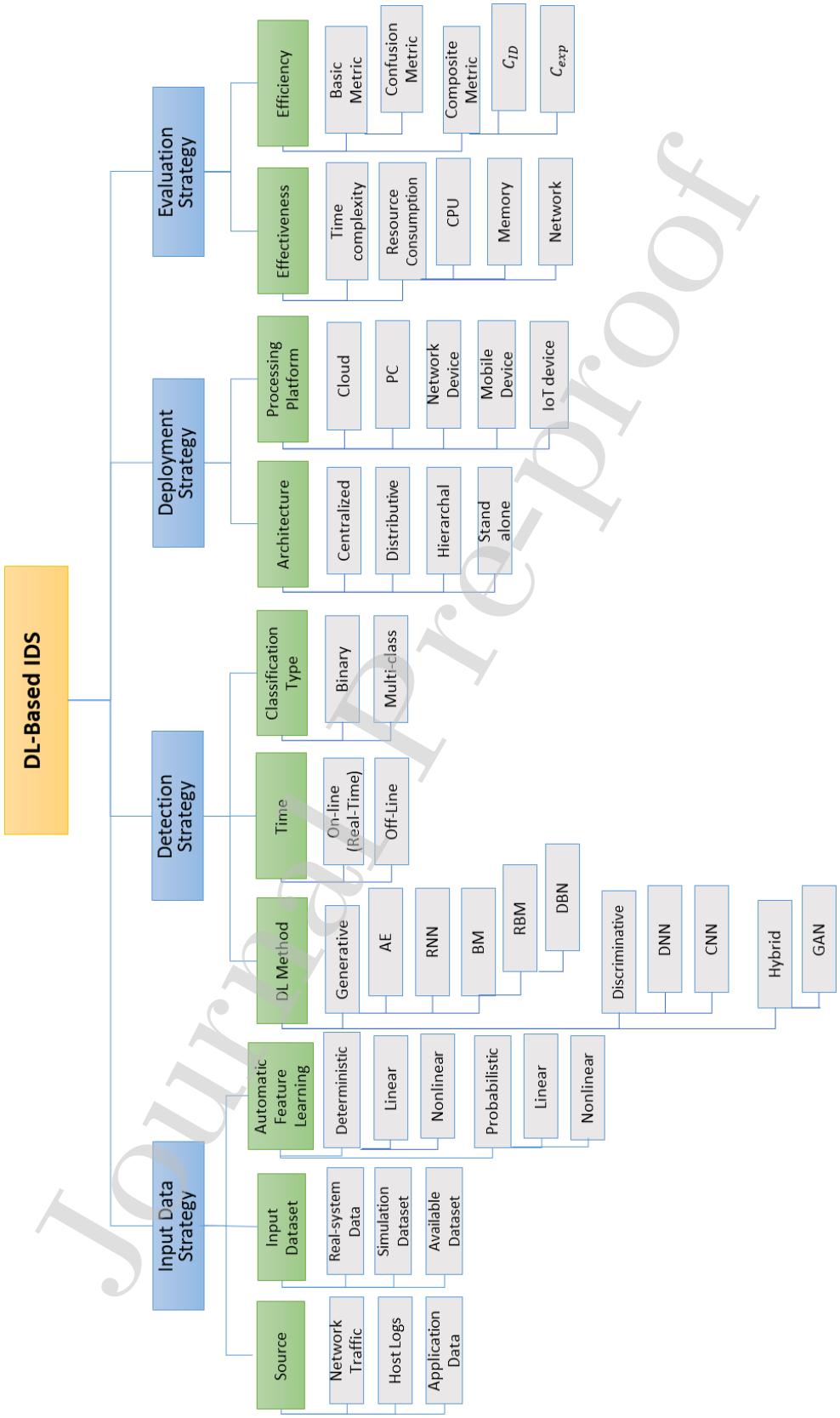


Figure 6: Deep learning-based IDS taxonomy

Different techniques are used to tackle the high dimensionality problem, such as discretization of continuous attributes [80] and feature learning. The latter is a pre-processing phase, which aims to produce a low-dimensional feature representation to be used to train the classifier. Feature learning techniques are broadly classified into probabilistic and deterministic frameworks. Each framework involves linear and non-linear methods. PCA and KernalPCA are examples of linear and non-linear deterministic methods, respectively.

On the other hand, factor analysis (FA) and Gaussian are examples of linear and non-linear probabilistic methods, respectively [38]. Deep learning methods used for dimensionality reduction can fall under probabilistic or deterministic frameworks.

5.2 Detection Strategy

Different architectures can be employed in anomaly-based detection approaches based on deep learning, including generative, discriminative, and hybrid methods, as depicted in Fig. 6. The classification of intrusions could be binary when the goal is to distinguish between normal and abnormal behaviours, or multi-class when the intrusion is attributed to a specific attack category. The multi-class classification could be further decomposed into hierarchal structures, which have multiple levels of subclasses. There are some studies that used hierarchical datasets for anomaly detection, as in [81]. Furthermore, the detection system either operates online, which is suitable for real-time systems, or off-line.

5.3 Deployment Strategy

The deployment architecture determines the IDS component configuration, which can be classified according to the type of architecture: centralized, distributed, or hierarchal. In the centralized architecture, data is obtained from single or multiple sources. Then, all operations are performed in a central location. In the distributed architecture, the IDS components are distributed among different physical locations. In WSN and IoT, the IDS system components can be placed in the hierarchal architecture, in which the processed data is moved up through the layers. Additionally, in WSN and IoT, the IDS can be run independently on each node by using the stand-alone architecture [66]. The IDS operations can be carried out on different processing platforms, including cloud, PC, server, network device, mobile device, and IoT device.

5.4 Evaluation Strategy

The designed IDS should fulfil security and performance requirements. The confusion matrix is the key metric typically considered to measure IDS effectiveness. Many measures can be derived from the confusion matrix, including accuracy, precision, detection rate, recall, f-score, false alarm rate (FAR), receiver operating characteristic (ROC) curve, and area under the curve (AUC) [82]. Additionally, intrusion detection capability (C_{ID}) and expected cost metric (C_{exp}) are the common composite metrics used to evaluate effectiveness, as proposed by Gu et al. [83] and Gaffney and Ulvila [84], respectively.

(C_{ID}) is a comparative measure that is used to evaluate several IDSs under different configurations and is given by Eq. (1):

$$C_{ID} = \frac{H(X)-H(X|Y)}{H(X)} \quad (1)$$

where (X) represents the input stream (intrusion denoted by X=1, benign denoted by X=0) and (Y) represents the output stream (alert denoted by Y=1, no alert denoted by Y=0). Therefore, the entropies H(X) and H(Y) reflect the degree of uncertainty of the input and output, respectively [83].

The other metric (C_{exp}) integrates the ROC curve with a cost analysis to calculate the expected cost of a specific IDS operating point and is given by Eq. (2):

$$C_{exp} = Min\big(C\beta B, (1-\alpha)(1-B)\big) + Min(C(1-\beta)B, \alpha(1-B)) \quad (2)$$

The following cost ratio, shown in Eq. (3), is used in the computation of C_{exp}:

$$C = C_{\beta}/C_{\alpha} \quad (3)$$

where α represents false positive, β represents false negative, and C_{β} represents the cost when an intrusion exists but is not detected, C_{α} represents the cost when there is an alert without the occurrence of an intrusion, and B represents the base rate [84].

The other element of performance evaluation is measuring IDS efficiency, including resource consumption and time complexity. Resource consumption, which includes CPU, memory, and network consumption is usually evaluated for IDSs designed for resource-constrained devices [82].

6. Descriptive and Comparative Study of Deep Learning-based IDS Methods

According to Wan in [85], three approaches have been proposed for deep learning: Dropout, DropConnect, and Hybrid Drop for regularizing large, fully connected layers in neural networks. The Dropout approach was presented by Hinton in 2012 as a form of regularization for fully connected neural network layers [86]. The Dropout approach is applied on the output layer, where each element is kept with probability p. Otherwise, it is probabilistically set to 0.

DropConnect is an enhanced Dropout algorithm. Instead of dropping out some of the activations as in Dropout, DropConnect drops some of the weights. The four basic components of the DropConnect model are the feature extractor, the DropConnect layer, the softmax classification layer, and cross-entropy loss.

Hybrid Drop and No-Drop adopt the same concept as Dropout and DropConnect, but without dropping out any activations or weights. The main reason behind the third approach is to prevent overfitting by using a mask vector/matrix.

In general, each layer of the deep network learns independently, bypassing the previous pertaining procedure. It then allows checking a good initial approach to run the backpropagation algorithm [60].

Different deep learning architectures have been employed in the past few years for dimensionality reduction, classification of intrusions, or both. We create a detailed descriptive and comparative analysis of the published deep learning-based intrusion detection solutions, as shown in Table 2. The solutions are classified based on the targeted application, the dataset used for training and validating the model, feature extraction and detection techniques, classification type, testing methodology, effectiveness and efficiency measures, and the applied deep learning architecture.

6.1 AE-based Methods

AE is the most frequent architecture investigated in the literature for both dimensionality reduction and classification phases. Several versions of AE have been investigated, including vanilla AE, stacked AE, sparse AE, and stacked sparse AE.

6.1.1 Vanilla AE-based Methods

Vanilla AE is mostly used as a dimensionality reduction technique. Abolhasanzadeh [38] proposed an approach for dimensionality reduction to reduce the space complexity and the time spent by the detection system. They utilized a neural network bottleneck feature that is used in AEs. They reported higher performance compared with the traditional linear dimensionality techniques such as principal component analysis (PCA), factor analysis, and nonlinear KernalPCA.

6.1.2 SAE-based Methods

Some solutions cascaded several AEs to construct a stacked AE. Studies [87,88] both used stacked AEs. AE is trained using a greedy layer-wise approach for feature learning combined with softmax regression as a classification layer to detect multi-class attacks. Farahnakian and Heikkonen [87] cascaded four stacked AEs, which were trained on 10% of the KDD99 dataset with all the features, while study [88] stacked two AEs, which were trained on all the features of the entire NSL-KDD dataset. Both studies [87,88] provided reasonable accuracy (above 95%), as presented in Table 2, except for the second study [88], which had accuracies of 13% and 39.6% for R2L and U2R attacks, respectively. Vartouni et al. [89] investigated different configurations of stacked AE for feature learning combined with isolation forest to identify anomalies in HTTP traffic to protect web servers. Among the tested configurations, the best performance was with the deeper architectures based on the Sigmoid activation function and Adam optimization.

Aminanto et al. [34,32] focused on detecting impersonation attacks in the Aegean Wi-fi intrusion dataset (AWID). In [34], they used ANN as a first hidden layer for feature selection, and implemented SAE composed of two encoders and a softmax regression layer. Later, in [32], the same authors used SAEs of two hidden layers combined with k-mean clustering. The second approach provides an accuracy of 94.81%, compared to 98.59% in [34]. While [34,32] focused on impersonation attacks, the same authors in [33] used the same dataset and proposed a generalized semi-supervised approach to detect three active attacks: impersonation, injection, and flooding attacks using SAE with a softmax regression layer. In [33], they achieved higher

accuracy in detecting impersonation attacks compared to their previous works. Shone et al. [90] developed a solution based on a stacked non-symmetrical auto encoder (NSAE), in which the hidden layers are non-symmetric, in addition to using random forest (RF) for classification. They utilized the encoder to reduce the time and computation overhead, and significantly reduced the training time compared to DBN.

6.1.3 Sparse AE-based Methods

Other methods utilized sparse AE with constraints that reduce the activation rate, since each neuron is activated only for a specific type of input. Niyaz et al. [41] proposed an approach based on self-taught-learning, which includes two-stage processing: sparse AE for feature extraction combined with softmax regression for classification. The proposed approach is evaluated by using the full set of features in the NSL-KDD dataset, and obtained better results than applying softmax regression alone.

Along the same direction, researchers in [31,91] cascaded layers of sparse AEs to produce stacked sparse AE for feature learning. The authors of [31] proposed a three-layer solution to detect impersonation attacks in Wi-fi networks using the AWID dataset. First, the stacked sparse AE is utilized for feature extraction. Then, the shallow machine learning SVM, DT, and ANN are used for weighted feature selection. Finally, ANN is used for classification. Among the tested algorithms, SVM had the best accuracy, but it incurred the longest training time. In addition, this approach outperforms the previous work [34] that used SAW with ANN in terms of detection rate and FAR. The work in [91] proposed a hybrid classifier composed of the Xgboost algorithm based on stacked sparse AE (SSAE-XGB) for feature learning, in addition to a binary tree and ensemble method for classification. This approach achieved high performance in terms of F1, and SSAE-XGB significantly outperformed linear dimensionality reduction (PCA).

6.2 DBN-based Methods

BM has a large number of connections among units, which results in high computation complexity and long execution time [52]. RBM is lightweight and has less computation complexity, since there are no connections among units in the same layer. When several layers of RBMs are connected, a DBN is constructed. As mentioned previously in Section 4, DBN can be used as a generative architecture, as a feature extractor for dimensionality reduction, and as a discriminator architecture for classifying intrusions.

Zahangir and Taha [48] conducted a series of experiments using AE and RBM for dimensionality reduction combined with an iterative k-mean clustering for clustering intrusions using the KDD99 dataset. Moreover, they investigated unsupervised extreme learning machine (ELM) for intrusion detection. The best accuracy, 92.12%, was obtained from training RBM with the iterative k-mean using nine input features. Another study, [92], achieved higher accuracy than ELM by using a constrained-optimization-based ELM (C-ELM), a modified version of ELM integrated with least square SVM.

Supervised DBN is utilized as a classifier and is considered a discriminator when it is combined with a final discrimination layer and each feature vector is assigned with a class label. However, Kang [73] proposed a real-time anomaly detection approach for in-vehicle networks using simulated vehicular network communication. First, they applied off-line training of packets using DBN tuned with conventional stochastic gradient descent for binary classification, followed by a detection phase using real network traffic. They reported higher performance compared with feed-forward ANN.

Gao et al. [17] developed a hierarchal IDS, which combines DBN for dimensionality reduction with an unsupervised greedy contrastive divergence algorithm. The resultant low-dimensional features are then fine-tuned by a backpropagation layer. The experimental results showed higher accuracy compared with SVM and ANN in classifying intrusions with the KDD99 dataset. Using the same dataset, Alrawashdeh and Purdy [19] utilized one RBM layer for feature learning, then passed the weighted result to another RBM layer, forming a DBN. Finally, a fine-tuning layer with a softmax regression classifier was used for multi-class intrusion classification. The authors compared their accuracy with [17] and the hybrid approach [35], which used the same dataset. Their experimental results showed a higher accuracy of 97.9% compared with [35] and [17], which had accuracies of 93.94% and 92.1%, respectively.

Moreover, Alom et al. [20] developed an IDS that uses DBN as a classifier with a discrimination layer for classifying network intrusions on 40% of the NSL-KDD dataset and using all the features. Through a set of experiments, they concluded that the proposed DBN solution outperforms SVM and DBN-SVM in terms of training time and detection accuracy.

Another study in [93] outlined the problem of an imbalanced dataset. To overcome this problem, the authors proposed two IDSs that combined a de-noising method and deep learning architectures, which are trained using the entire NSL-KDD dataset. First, the stacked AE used for dimensionality reduction is combined with an RF classifier. Second, DBN is used as a classifier and is fine-tuned with PB. The addition of a de-noising layer improved the accuracy rate for stacked AE and DBN by 1.5% and 4.5%, respectively.

Zhang and Chen [94] studied and compared the performance of utilizing DBN for pre-training dimensionality reduction with DBN as a classification technique by developing two models. In the first model, a single RBM layer is used for feature extraction; then, the weighted features are passed to SVM for classification. The second model utilized DBN and was constructed of RBM for feature extraction and a BP layer for classification. Both models were trained on KDD99, and DBN-PB had a higher accuracy of 97.16% compared with 96.31% for RBM-SVM.

Finally, some studies conducted an experimental comparison between the performance of AE and DBN in the context of IDS. Two comparisons showed contradictory conclusions using different versions and configurations of each architecture. Van et al. [95] concluded that AE shows better performance regarding attack classification, but it involves more execution time due to its high computation complexity. However, AE is more lightweight in implementation than

RBM. Conversely, in terms of accuracy, the results of [48] and [93] showed superior results for RBM and DBN, respectively, compared to AE. Furthermore, [90] showed a significant (an average of 97.72% under KDD99 and 78.19% under NSL-KDD) reduction in training time with NSAE compared to DBN.

6.3 RNN-based Methods

The generative architecture RNN and its LSTM and GRU versions are widely used for classification and regression tasks. We found a number of studies that employed RNN in intrusion detection.

6.3.1 Vanilla RNN-based Methods

Al-Zewairi et al. [96] applied RNNs along with stochastic gradient descent, and tested with different activation functions on the UNSW-NB15 dataset. The best accuracy rate was obtained when the ReLU function was used, i.e., 98.99%. Furthermore, Yin et al. [21] also utilized RNNs for binary and multi-class intrusion classification on the NSL-KDD dataset.

As reported by the above solutions, vanilla RNNs outperformed a set of shallow machine learning algorithms in [96,21]. However, they consume extensive computation resources and have a large number of neurons.

6.3.2 LSTM-based Methods

LSTM-RNN has been investigated in the literature [97,98,99,22,23] for generating intrusion detection classifiers. The proposed approaches outperformed several shallow machine learning and deep learning methods. Both [97,23] outperformed generalized regression neural network (GRNN), probabilistic neural network (PNN), radial basis neural network (RBNN), k-nearest neighbours algorithm (KNN), SVM, and Bayesian approaches using the NDL-KDD and KDD99 datasets, respectively. Staudemeyer [22] conducted a series of experiments on the KDD dataset using the entire set of features and a minimal set of features to train the attack classes together within one network and train each attack class individually. The overall results outperformed multi-layer perceptron (MLP) and SVM.

Loukas [98] applied cloud offloading computation to overcome the detection latency produced by the intensive deep learning computation requirement. They obtained higher accuracy in detecting DoS and command injection attacks than shallow machine learning and deep MLP.

Jiang et al. [99] aimed to improve the detection accuracy by introducing an approach based on LSTM-RNN, and used multi-channel processing to create different classifiers. The attack confirmation is then determined by a majority voting algorithm. According to their experiment, the approach outperformed Bayesian and SVM classifiers.

6.3.3 GRU-Based Methods

GRU is a simplification of LSTM. The comparative studies [26,99] showed that GRU is more effective than LSTM-RNN. Congyuan et al. [26] proposed the combination of bi-directional

GRU (BGRU) and MLP to classify intrusions. They conducted a series of experiments to evaluate LSTM, GRU, and BGRU. BGRU architecture scored the highest accuracy rate, exceeding 99% under both the KDD99 and NSL-KDD datasets using BGRU. Moreover, they evaluated adding the MLP module, and discovered that MLP improved the accuracy of both LSTM and GRU.

Anani and Samarabandu [101] conducted a comparative experimental study of four RNN variants in terms of detection time and accuracy by employing the full KDD99 dataset. The best detection accuracy was achieved by vanilla LSTM and GRU, whereas the lowest training time was attributed to GRU. However, they could not train skip-LSTM, even with different parameter settings.

6.4 CNN-based Methods

Recently, researchers have begun investigating discriminative deep learning architectures for intrusion detection. Studies [101,102] showed promising performance results using CNNs with different classification layers. Chowdhury et al. [101] applied a few-shot learning strategy, which handles the situation when a specific class is limited. In their proposed solution, CNNs are used as a feature extractor and are combined with SVM and 1-nearest neighbour ANN for classification. They trained the model on both the KDD99 and NSL-KDD benchmark datasets, and achieved better accuracy with 1-NN: 96.19% and 86.74% for the KDD99 and NSL-KDD datasets, respectively. Using the KDD99 dataset, Lin et al. [102] used five-layer CNNs to extract weighted features, and identified intrusions with a softmax regression layer. They achieved a high accuracy of 97.53%.

On the other hand, other proposed solutions showed worse performance. Kwo et al. [103] evaluated three depths of CNN (shallow, moderate and deep). The experimental results showed that deeper structures do not show any performance improvement. Furthermore, they observed that the evaluated CNN models outperform the AE models, but they are less effective than deep learning solutions based on Seq2Seq-LSTM and FCN. In general, the overall performance was not promising for any of the tested configurations, as they did not exceed 80% accuracy in the best scenario with the NSL-KDD dataset.

Finally, a comparative study of different variants of AE, LSTM-RNN, and CNN for anomaly-based detection, combined with different classifiers, was conducted by Naseer et al. [104]. They reported higher accuracy when using LSTM, followed by CNN and then AE.

6.5 Ensemble and Hybrid Methods

The authors of [35] proposed a hybrid approach, combining AE and DBN in one solution. First, AE was utilized for dimensionality reduction, and then DBN was fine-tuned by backpropagation for attack classification. According to their experimental results, the hybrid approach provided an accuracy of 92.1%, higher than the use of DBN alone.

Len et al. [105] is the only study that presented a hybrid architecture. They proposed a GAN framework that generates samples of adversarial attacks that try to flood the IDS. The framework involves a generator that convert malicious traffic into malicious adversarial traffic. The discrimination layer simulates the detection system in classifying attacks, and their approach achieved promising results when combined with different classification algorithms, including SVM, Naive Bayes, MLP, LR, DT, RF, and KNN.

Table 2: Comparison of deep learning-based IDS

Deep Learning architecture	Article	Application	Dataset	Feature learning	Classification	Testing			Effectiveness			Processing Component	Efficiency
					Technique	Binary	multi-class	Conventional	Cross-validation	Accuracy (%)	F1 (%)	FAR (%)		
AE	[38], 2015	Network Intrusion	NSL-KDD	AE	-	-	✓	-	-	91	-	-	-	-
	[87], 2018	Network Intrusion	KDD99	SAE	Softmax	✓	✓	✓		Binary:94.71 Multi-class: 96.53	-	-	-	-
	[89], 2018	Web-based anomaly	CSIC 2010	SAE	Isolation forest	-	✓	✓		88.32	84.12	-	**CPU**: 2 Intel Core: 2.67 GHz	-
	[91], 2018	Network Intrusion	NSL-KDD	SSAE-XGB	SSAE-XGB	-	✓	✓		-	92.94	-	-	-
	[42], 2016	Network Intrusion	NSL-KDD	Sparse AE	Softmax	✓	✓	-	-	97.0	-	-	-	-
	[88], 2016	Network Intrusion	NSL-KDD	SAE	Softmax	✓	✓	-	✓	Binary: 97			Different **CPUs** used	Time complexity recorded for each CPU.
	[31], 2018	Wi-Fi Imperson ation	AWID	SSAE for feature extraction + (SVM, ANN, and DT) for feature selection	ANN	✓	-	✓		99.91	99.94	0.01	**CPU**: Intel Xeon 3.30 GHz	Time complexity: SVM scored the longest training time (12,073 s) while (Corr) measures scored the lowest (12064 s)
	[33], 2016	Wi-Fi intrusion	AWID	Stacked AE	softmax	-	✓	-	-	97.7	-	-	**CPU**: Intel Xeon 3.30 GHz.	-
	[90], 2017	Network intrusion	10% of KDD99 + Whole NSL-KDD	Stacked NSAE	Stacked NSAEs and RF Classifier	-	✓		✓	KDD99: 97.85 NSL-KDD: 85.42	KDD99: 98.15 NSL-KDD: 87.37	-	-	Compared with DBN, reduced training time by an average of: KDD99: 97.72% NSL-KDD:78.19.
	[34], 2016	Wi-Fi Imperson ation	AWID	ANN	SAE with softmax	✓	-	-	-	98.59	77.16	0.14	**CPU**: Intel Xeon 3.30 GHz.	-
	[32], 2018	Wi-Fi Imperson ation	AWID	3 SAE	K-means clustering	✓	-	-	-	94.81	89.06	4.4	**CPU**: Intel Xeon 3.30 GHz	-
AE +	[95],	Network	KDD99	1-SAE	-	-	✓	-	✓	AE better		-	-	Time complexity:

DBN	2017	intrusion		2-DBN					than DBN.				DBN consumed less training time.
	[48], 2017	Network intrusion	KDD99	1. AE 2. RBM	1, 2. iterative k-mean clustering 3. ELM	✓	-	-	-	AE: 91.68 RBM:92.12	-	-	**CPU**: Intel Core 2 Duo 3.33 GHz	-
	[93], 2017	Network intrusion	NSL-KDD	1.Stacked AE 2.DBN with de-noising	1.RF 2.DBN	✓	-	-	-	1. 85.42 2. 99.96	1.88.60 2.89.24	-	**CPU**: Intel Xeon - 3.60 GHz	
	[73], 2016	In-Vehicle Network	Simulation of in-vehicular network	DBN	Conventional stochastic gradient descent method	-	✓	N/A	N/A	97.8	-	-	-	-
	[20], 2015	Network intrusion	40% of NSL-KDD	DBN	DBN	-	✓	-	-	97.5	-	-	-	Training time for 40% Input: 0.32 s
DBN	[19], 2016	Network intrusion	10% KDD99	DBN	Soft-max regression	-	-	-	✓	97.9	-	-	**CPU**: 2.1 GHz	Training time: 9 m /10 epoch and 2 hidden layers, Testing time: 0.70 s/batch (1000 record)
	[17], 2014	Network intrusion	KDD99	DBN	DBN + BP	✓	✓	-	-	93.94%			**CPU**: Intel PU 1.86 GHz	
	[94], 2017	Network intrusion	KDD99	1.RBM 2. DBN	1.SVM 2.DBN + BP	-	-	-	✓	1.96.31 2.97.16	-	-	-	Testing time ≅ 33.12 ms/record.
	[26], 2018	Network intrusion	NSL-KDD KDD99	-	GRU/BGRU + MLP +softmax	-	✓	-	✓	KDD99: BGRU:99.84 GRU:99.28 NSL-KDD: BGRU: 99.24 GRU:99.1			**CPU**: Intel Core i7 3.4 GHz	
	[97], 2018	Attack detection in social networks	NSL-KDD	LSTM	LSTM	-	✓	-	-	97.5	-	8.75	**CPU**: Intel 2.5 GHz, **GPU**: NVIDIA GeForce 920MX.	-
RNN	[98], 2017	In-Vehicle Network	Real-time data	-	LSTM	-	✓	✓		86.90			**CPU**: Intel Dual-core Atom D525	Latency: 600 neurons:1.163 s, 800 neurons:1.541 s 1000 neurons:1.704 s
	[99], 2018	Network intrusion	NSL-KDD	-	LSTM	-	✓	-	-	99.23			**CPU**: Intel 2.5 GHz, **GPU**: NVIDIA GeForce 920MX	-
	[22],	Network	KDD99	LSTM	LSTM	-	✓	✓	-	93.82	-	9.86	-	-

	2015	intrusion											
	[96], 2017	Network intrusion	UNSW-NB15	Manually	RNN	✓	-	-	✓	98.99	-	0.72	**CPU**: Intel Core i7 quad 3.4 GHz	Lowest Time: Rectifier function. Highest time: Maxout function.
	[23], 2016	Network intrusion	KDD99	LSTM	softmax	-	✓	-	-	96.93	-	10.04	**CPU**: Intel Core 3.60 GHz	-
	[100], 2018	Network intrusion	KDD99	-	RNN: 1.Vanilla LSTM 2.Bi-directional LSTM 3.GRU	-	✓	-	-	1: 98.85 2:84.99 3: 98.68	-	1:0.90 2:25.8 3:0.94	-	Time (s): 1: 2354.93 2: 190 3: 1903.0
	[21], 2017	Network intrusion	NSL-KDD	-	RNN	✓	✓	-	✓	Binary: 68.55 Multi-class: 64.67	-	-	-	-
CNN	[101], 2017	Network intrusion	10% KDD99 + NSL-KDD	CNN	SVM + 1-NN	-	✓	✓		1-NN: 96.19, 86.74 SVM:95.27 , 77.68	-	-	**CPU**: Intel Core i-7 7700 3.60 GHz	-
	[102], 2018	Network intrusion	KDD99	CNN	Softmax	-	✓	-	✓	97.53	-	-	-	-
	[103], 2018	Network anomaly	NSL-KDD + Kyoto Honeypot + MAWILab	-	CNN	✓	-	-	-	Shallow CNN outperforms moderate and deep CNN.	-	-	**CPU**: Intel Core i7-3770 3.4 GHz	-
ensemble	[35], 2016	Network intrusion	10% KDD99	AE	DBN and BP for fine tuning	✓	-	-	✓	92.1	-	-	**CPU**: Intel Core Duo CPU 2.10 GHz	CPU time: DBN: 1.126 s AE+DBN^(5-5): 2.625 s AE+DBN^(10-5):1.147 s AE+DBN^(10-10): 1.243 s
	[106] ,2018	Network intrusion	KDD99	1. None 2.STL: sparse AE	1.DNN 2. STL: softmax 3.LSTM	-	✓	-	-	DNN: 66 STL: 98.9 LSTM:79.2	DNN: 47 STL: 98 LSTM:70	-	-	-
comparative	[104], 2018	Network intrusion	NSL-KDD	LSTM, CNN, AE (vanilla, sparse, denoising, contractive, convolutional)	SVM, K-NN, DT, RF, and Extreme Learning machine.	✓	-	✓	-	AE: 81 LSTM: 98 CNN: 85	-	-	**CPU**: Intel Quad Core **GPU**: NVIDIA GTX 1070	AE took 367 seconds on GPU, CNN took 109, and LSTM took 208 seconds.
Hybrid	[105] ,2018	Network intrusion	NSL-KDD	GAN	GAN	-	✓	-	-	-	-	-	**CPU**:Intel Core i7-2600	-

BP: Backpropagation

7. Discussion and Findings

The investigation of deep learning methods for intrusion detection has continued since the beginning of this decade. This survey considers articles published between 2014 and 2018. Fig. 7. shows the distribution of the discussed deep learning-based IDS over years for each type of architecture. We noticed that the earliest architectures that were investigated include AE, DBN, and RNN, while research works on CNN recently appeared in 2017.

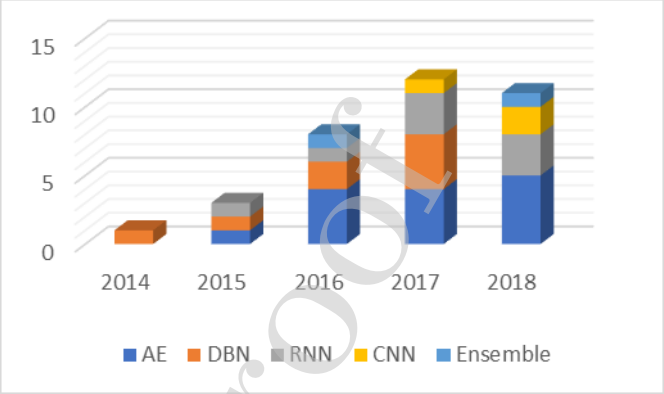


Figure 7: Deep learning-based IDS publication over years

Ensemble and hybrid architectures remain poorly explored and need to be investigated. The following discusses different aspects of the proposed solutions and their findings. We discuss the role of deep learning in the proposed IDSs and the adopted datasets in addition to the effectiveness and efficiency of the solutions that rely on benchmark datasets.

1. Deep Learning Role in IDS

Deep learning methods have proven their effectiveness in the discovery of sophisticated relationships within raw data with multiple levels of abstraction without human intervention. Deep learning methods have been used for both feature learning and classification tasks in IDS.

Feature learning is the main task of deep networks, which reduces the complexity of the raw features of the dataset. One group of discussed solutions applied deep learning as a pre-training phase for feature learning combined with another classifier, as illustrated in Table 2. The other group of solutions utilized the deep learning method as a classifier by combining the deep network with a fine-tuning layer. Fig. 8 gives an overview of the role of deep learning methods in the discussed solutions. Figure 8 also shows that the AE generative model has been mostly used for feature learning, while RNN is mostly used as a classifier.

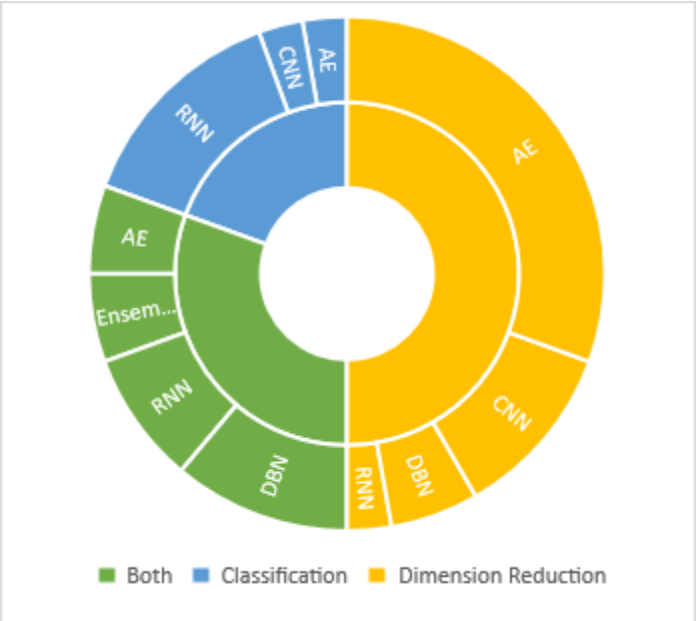


Figure 8: Deep Learning Role in IDS

2. Dataset

The evaluated dataset is an important factor that affects the efficiency and effectiveness of IDS. The majority of proposed IDS used the benchmark datasets KDD99 and its improved version, NSL-KDD. As shown in Fig. 9, 68% of the proposed solutions relied on the benchmark datasets, while only 5% used real-time data either from simulated or real environments. Therefore, the current IDS solutions do not provide enough reliability and applicability in real operation due to the datasets' limitations. First, the benchmark datasets were proposed two decades ago, and do not reflect current traffic behaviour and intrusion scenarios. Furthermore, the benchmark datasets do not have the properties of real-time datasets. However, KDD99 and NSL-KDD together still account for more than 70% of the datasets used by the research community due to their availability and the difficulties in obtaining real system traffic or creating simulated environments.

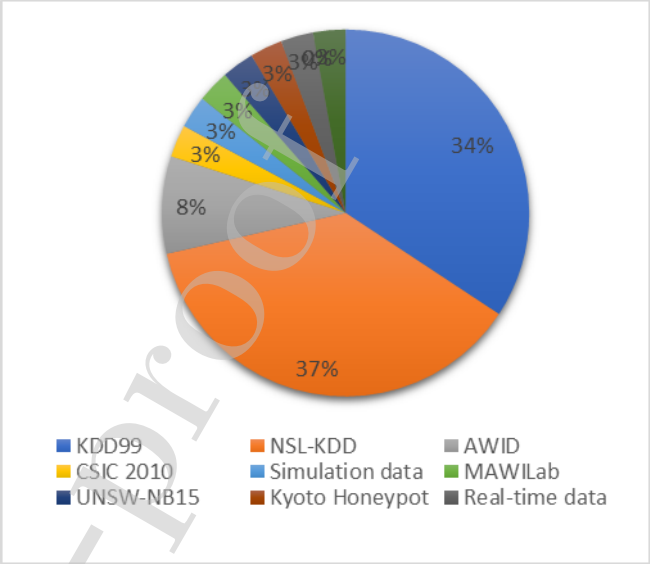


Figure 9: Dataset Distribution

7.3 Efficiency and Effectiveness

Deep learning-based anomaly detection achieved high accuracy with the use of different deep learning architectures. We compared solutions that relied on benchmark datasets, KDD99 and NSL-KDD, to give an overview of the effectiveness of deep learning methods for anomaly detection separately for each dataset, as demonstrated in Fig. 10 and Fig. 11. However, we emphasize that it is not an accurate comparison since we address isolated studies with diverse experimental aspects.

Furthermore, deep learning approaches showed a significant improvement over shallow machine learning, as proven in comparative studies. Studies [17,19]

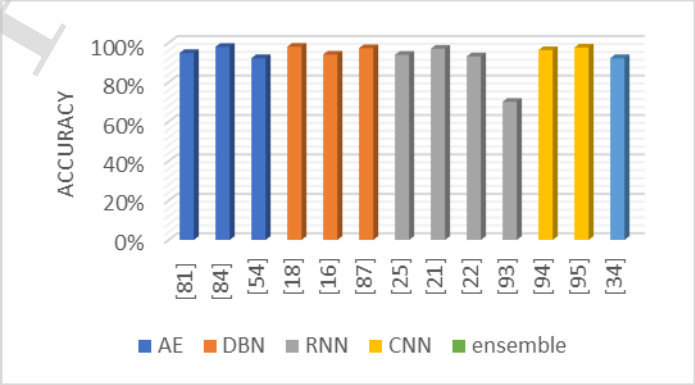


Figure 10: Accuracy of solutions under KDD99 dataset

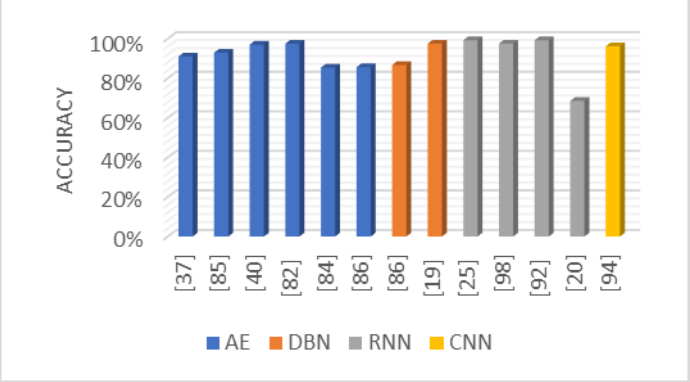


Figure 11: Accuracy of solutions under NSL-KDD dataset

showed DBN outperformed SVM, naïve Bayes, and ANN. DNN outperformed shallow ANN and SVM in [41], and in [98,99,100,22,23,106], LSTM-RNN outperformed shallow ANN, SVM, k-mean, and RF.

When the deep network goes deeper with a large number of layers and neurons, the computation complexity increases. Additionally, as a consequence, a latency problem will arise. As shown in Table 2, there is a lack of reporting on the efficiency aspects in the discussed articles. However, computation complexity and latency were reported with AE in [11], DBN [91], and vanilla RNN in [21]. Resource consumption is an important factor in designing IDS for constrained devices such as IoT devices. This should be taken into consideration when designing lightweight solutions. Additionally, a cloud-based solution is another option to deal with resource consumption problem.

We noticed that many studies used cross-validation methodology for splitting the training and testing subsets. Cross-validation is mainly used in shallow machine learning to overcome the overfitting problem. When a large dataset is used with deep learning, cross-validation increases training cost. Different approaches are used to counter the overfitting problem with deep learning models, such as regularization methods [46].

8. Challenges and Future Directions

Based on the findings obtained in Section 7, the following presents some of the lessons learned, and the most important directions for future research.

- The use of a proper dataset is a significant issue in the development of deep learning-based IDS. As discussed in Section 7, the current proposed deep learning-based IDS do not provide reliable performance results, since they rely on the KDD99 or NSL-KDD benchmark datasets, which contain old traffic, do not represent recent attack scenarios and traffic behaviours, and do not have real-time properties. Therefore, obtaining traffic from simulated environments can overcome this issue by testing more recent datasets, such as the CICIDS2017 IDS intrusion prevention system (IPS) dataset [107], and the N-BaIoT IoT dataset [108]. Datasets could also be generated, and published datasets are available for different domains, such as industrial control systems (ICS) [109].
- The comparison among different deep learning-methods, which are conducted in isolation, do not provide a fair comparison in terms of effectiveness and efficiency, as shown in Fig. 10 and Fig. 11. This is due to diversity in: (1) the used dataset, (2) the portion of the dataset that is adopted, (3) pre-processing, (4) deep network configuration, and (5) hardware platforms. Therefore, there is a need for more comparative experimental studies that use a unified computing platform and common affecting factors for different deep learning architectures in order to obtain a fair comparison result.
- Deep learning approaches do not yet cover intrusion detection in several domains. It is thus necessary revisit the IDS problem in different domains, such as SCADA, smart grid, 5G, and

numerous IoT platforms, which have already been investigated through shallow machine learning and other anomaly detection approaches. Extensibility to different domains requires a dataset that truly reflects the targeted environment and achieves better results.

- Several deep learning-based IDSs rely on CPUs and GPUs for intensive off-line training computations. In response to rapid evolution, chip vendors have produced advanced AI accelerators; the AI chip market is expected to reach $66.3 billion by 2025 [1]. The most common chips are the neural network processing unit (NNPU), the application-specific integrated circuit (ASIC), and the field programmable gate array (FPGA), in addition to the edge TPU, a tiny AI accelerator announced in 2018 by Google for IoT devices. Today's smart phones and IoT devices are equipped with these advanced chips. Therefore, leveraging this advancement to conduct research would produce real-time prototypes, rather than relying on offline datasets. In addition, it would allow the development of more advanced IDS for the constrained devices.
- Further investigation of hybrid deep learning architectures, such as GAN models, is necessary. To the best of our knowledge, this has only been explored by Len et al. [105] in late 2018. Moreover, ensemble approaches have been less studied but show promising results.
- It would be worthwhile to leverage deep learning in order to move from collaborative IDS to collaborative deep learning IDS.

9. Conclusion

Deep learning has drawn the attention of researchers in different fields. Deep models can handle complex data and find correlation among input features without human intervention. With the emergence of new technologies and the rapid growth in transmitted traffic, researchers have been investigating deep learning for intrusion detection. This survey reviewed and compared the key surveys considering deep learning for intrusion detection, and built the current survey upon the previous ones. The study provided a novel fine-grained taxonomy considering different design aspects, including input data, detection, deployment, and evaluation strategies. Accordingly, this survey provided a thorough review of the related experimental studies in deep learning-based IDS.

Through detailed review, we have uncovered different findings and lessons. Deep learning is mostly used for feature learning in intrusion detection approaches, even though some studies used deep learning models as classifiers. However, we observed that most proposed approaches rely on the legacy benchmark datasets. Furthermore, less attention has been paid to reporting the effectiveness of the proposed approaches. The current findings demonstrate that further efforts are required to improve the current state-of-the-art, in view of these findings. This survey also lays out several research challenges and future directions. Since the benchmark datasets do not tackle the current advanced status of different types of networks, there is an urgent need to use and generate more recent datasets and real-time prototypes based on current hardware advances.

Moreover, several domains should be revisited with deep learning approaches instead of shallow machine learning, in addition to conducting further comparative studies and investigating hybrid and ensemble architectures.

Acknowledgements

The authors would like to thank the scientific research deanship for funding and supporting this research through the DSR Graduate Students Research Support (GSR) initiative.

References

[1] Cisco, Cisco visual networking index: forecast and methodology 2016-2021, https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/complete-white-paper-c11-481360.html, 2017 (Accessed 18 December 2018).

[2] H. Fujita, A. Gaeta, V. Loia and F. Orciuoli, Resilience analysis of critical infrastructures: a cognitive approach based on granular computing, IEEE Trans. on Cybernetics, 49(5) (2019) 1835-1848. doi: 10.1109/TCYB.2018.2815178

[3] K. Scarfone, P. Mell, P. Mell, Guide to intrusion detection and prevention systems (IDPS), NIST special publication, (2007).

[4] H.-J. Liao, C.-H. Richard Lin, Y.-C. Lin, K.-Y. Tung, Intrusion detection system: a comprehensive review, J. Netw. Comput. Appl. 36 (2013) 16–24. doi:10.1016/j.jnca.2012.09.004.

[5] P.García-Teodoroa, J.Díaz Verdejoa, G.Maciá-Fernándeza, E.Vázquez, Anomaly-based network intrusion detection: Techniques , systems and challenges, Computers & Security, 28 (2009) 18–28. doi:10.1016/j.cose.2008.08.003.

[6] Y. Bengio, I. Goodfellow, A. Courville, Deep learning, MIT Press (2016). http://www.deeplearningbook.org

[7] W.G. Hatcher, W.E.I. Yu, A Survey of deep learning: platforms , applications and emerging research trends, IEEE Access, 6 (2018) 24411–24432. doi: 10.1109/ACCESS.2018.283066.

[8] L. Deng, A tutorial survey of architectures, algorithms, and applications for deep learning, APSIPA Transactions on Signal and Information Processing, 3 (2018) 1–29. doi:10.1017/ATSIP.2013.99.

[9] B. Dong, X. Wang, Comparison deep learning method to traditional methods using for network intrusion detection, 8th IEEE Int. Conf. Commun. Softw. Networks. (2016) 581–585. doi:10.1109/ICCSN.2016.7586590.

[10] K. Kim, M.E. Aminanto, Deep learning in intrusion detection perspective: overview and further challenges, Int. Workshop on Big Data and inform. Security, (2017) 5–10. doi: 10.1109/IWBIS.2017.8275095

[11] M. Erza, K. Kim, Deep learning in intrusion detection system: an overview, Int. Research Conf. on Engineering and Technology (2016) 1–12.

[12] D. Kwon, H. Kim, I. Kim, K.J. Kim, J. Kim, S.C. Suh, A survey of deep learning-based

network anomaly detection, Cluster Comput. (2017) 1-13. doi:10.1007/s10586-017-1117-8.

[13] E. Hodo, X. Bellekens, A. Hamilton, C. Tachtatzis, Shallow and deep networks intrusion detection system: a taxonomy and survey, arXiv preprint arXiv:1701.02145.

[14] Y. Xin, L. Kong, Z.H.I. Liu, Y. Chen, Y. Li, H. Zhu, M. Gao, H. Hou, C. Wang, Machine learning and deep learning methods for cybersecurity, IEEE Access 6 (2018) 35365–35381. doi: 10.1109/ACCESS.2018.2836950

[15] M.A. Al-garadi, A. Mohamed, A. Al-ali, X. Du, M. Guizani, A survey of machine and deep learning methods for internet of things (iot) security, (2018) arXiv preprint arXiv:1807.11023.

[16] M. Nadeem, O. Marshall, Semi-supervised deep neural network for network intrusion detection, (2016).

[17] N. Gao, L. Gao, Q. Gao, H. Wang, An intrusion detection model based on deep belief networks, IEEE Second Int. Conf. on Advanced Cloud and Big Data (2014) 247-252. doi:10.1109/CBD.2014.41.

[18] G. Zhao, C. Zhang, L. Zheng, Intrusion detection using deep belief network and probabilistic neural network, IEEE Int. Cong. on Computational sci. and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC), Guangzhou, 1 (2017) 639–642. doi:10.1109/CSE-EUC.2017.119.

[19] K. Alrawashdeh, C. Purdy, Toward an online anomaly intrusion detection system based on deep learning, 15th IEEE Int. Conf. on Machine Learning and Applications (ICMLA) (2016) 195-200. doi:10.1109/ICMLA.2016.167.

[20] Z. Alom, V. Bontupalli, T.M. Taha, Intrusion detection using deep belief networks, National Aerospace and Electronics Conf. (NAECON) (2015) 339–344. doi: 10.1109/NAECON.2015.7443094

[21] C. Yin, Y. Zhu, J. Fei, X. He, A deep learning approach for intrusion detection using recurrent neural networks, IEEE Access, 5 (2017) 21954-21961. doi:10.1109/ACCESS.2017.2762418.

[22] R.C. Staudemeyer, Applying long short-term memory recurrent neural networks to intrusion detection, South African Computer Journal, 56 (2015) 136–154.

[23] J. Kim, J. Kim, H. Le, T. Thu, H. Kim, Long Short term memory recurrent neural network classifier for intrusion detection, Int. Conf. on Platform Technology and Service (PlatCon) (2016) 1-5. doi: 10.1109/PlatCon.2016.7456805.

[24] E. For, LSTM-based system-call language modeling and robust ensemble method for designing host-based intrusion detection systems, (2016) arXiv preprint arXiv:1611.01726.

[25] V.L. Cao, J. Mcdermott, Collective anomaly detection based on long short term memory recurrent neural network, Int. Conf. on Future Data and Security Engineering. Springe 23 (2016) 141-152. doi:10.1007/978-3-319-48057-2_9.

[26] C. Xu, S. Member, J. Shen, X.I.N. Du, F.A.N. Zhang, An intrusion detection system using a deep neural network with gated recurrent units, IEEE Access. 6 (2018) 1-1.

doi:10.1109/ACCESS.2018.2867564.

[27] A.F.M. Agarap, A neural network architecture combining gated recurrent unit (GRU) and support vector machine (SVM) for intrusion detection in network traffic data, 10th Int. Conf. on Machine Learning and Computing. ACM, (2018) 26-30. doi:10.1145/3195106.3195117

[28] T. Ergen, S.S. Kozat, S. Member, Efficient online learning algorithms based on lstm neural networks, IEEE Trans. Neural Networks Learn. Syst. 29 (2018) 3772–3783. doi:10.1109/TNNLS.2017.2741598.

[29] Y. Yu, J. Long, Z. Cai, Network intrusion detection through stacking dilated convolutional autoencoders, Security and Communication Networks (2017). doi: 10.1155/2017/4184196.

[30] W. Wang, M. Zhu, End-to-end encrypted traffic classification with one-dimensional convolution neural networks, IEEE Int. Conf. on Intelligence and Security Informatics (ISI) (2017) 43–48. doi: 10.1109/ISI.2017.8004872.

[31] M.E. Aminanto, R. Choi, H.C. Tanuwidjaja, P.D. Yoo, S. Member, K. Kim, Deep abstraction and weighted feature selection for wi-fi impersonation detection, IEEE Trans. on Inform. Forensics and Security, 13 (2018) 621–636. doi:10.1109/TIFS.2017.2762828.

[32] M.E. Aminanto, K. Kim, Improving detection of wi-fi impersonation by fully unsupervised deep learning, Int. Workshop on Inform. Security Applications. Springer (2017) 212-22300. doi: 10.1007/978-3-319-93563-8_18.

[33] S.D., M.E. Aminanto, K. Kim, Detecting active attacks in wifi network by network by semi-supervised deep learning, In Conf. on Information Security and Cryptography (2017) 1–4.

[34] M.E. Aminanto, K. Kim, Detecting impersonation attack in wifi networks using deep learning approach, 10144 (2017) 136-147. doi: 10.1007/978-3-319-56549-1_12.

[35] Li, Yuancheng, Rong Ma, and Runhai Jiao, Hybrid malicious code detection model based on deep learning, International Journal of Security and Its Applications , 9 (2015) 205–216.

[36] W. Jung, S. Kim, Deep learning for zero-day flash malware detection, 36th IEEE symposium on security and privacy. (2015) 2–3.

[37] M.A. Salama, H.F. Eid, R.A. Ramadan, A. Darwish, Hybrid intelligent intrusion detection scheme, Soft computing in industrial applications. Springer, 96 (2011) 1–11. doi: 10.1007/978-3-642-20505-7_26

[38] B. Abolhasanzadeh, Nonlinear dimensionality reduction for intrusion detection using auto-encoder bottleneck features, 7th Conf. on Inform. and Knowledge Technology (IKT) (2015) 1-5. doi: 10.1109/IKT.2015.7288799.

[39] U. Fiore, F. Palmieri, A. Castiglione, A. De Santis, Network anomaly detection with the restricted Boltzmann machine, Neurocomputing. 122 (2013) 13–23. doi:10.1016/j.neucom.2012.11.050.

[40] I. Paper, Deep learning approach for network intrusion detection in software defined networking, Int. Conf. on Wireless Networks and Mobile Communications (WINCOM),

(2016) 258-263. doi:10.1109/WINCOM.2016.7777224

[41] Q. Niyaz, W. Sun, A.Y. Javaid, M. Alam, A Deep learning approach for network intrusion detection system, 9th EAI Int. Conf. on Bio-inspired Information and Communications Technologies (formerly BIONETICS) (2016) 21-26. doi: 10.4108/eai.3-12-2015.2262516

[42] A. Abebe, and N. Chilamkurti, Deep learning: the frontier for distributed attack detection in fog-to-things computing, IEEE Communications Magazine, 56 (2018) 169–175. doi: 10.1109/MCOM.2018.1700332.

[43] M. Yousefi-azar, V. Varadharajan, L. Hamey, U. Tupakula, Autoencoder-based feature learning for cyber security applications, Int. Joint Conf. on Neural Networks (IJCNN) (2017) 3854–3861. doi: 10.1109/IJCNN.2017.7966342.

[44] Yavuz, Furkan Yusuf, Deep learning in cyber security for internet of things, Doctoral Dissertation (2018).

[45] A.A. Diro, N. Chilamkurti, Distributed attack detection scheme using deep learning approach for internet of things, Futur. Gener. Comput. Syst. 82 (2017) 761-768. doi:10.1016/j.future.2017.08.043.

[46] Adam Gibson, Josh Patterson, Deep Learning practitioner's approach, O'REILLY (2017).

[47] Hinton, Geoffrey E., and Ruslan R. Salakhutdino, Reducing the dimensionality of data with neural network, science, 313 (2006) 504–507. doi: 0.1126/science.1127647.

[48] Z. Alom, T.M. Taha, Network intrusion detection for cyber security using unsupervised deep learning approaches, IEEE National Aerospace and Electronics Conference (NAECON) (2017) 63–69. doi: 10.1109/NAECON.2017.8268746.

[49] Andrew. Ng, CS294 Lecture notes Sparse autoencoder, Standford University (2011) 42.

[50] P. Vincent, Stacked denoising autoencoders : learning useful representations in a deep network with a local denoising criterion, Journal of machine learning research, 11 (2010) 3371–3408.

[51] D.H. Ackley, G.E. Hinton, J. Sejnowski, A Learning algorithm for boltzmann machines, cognitive science, 169 (1985) 147–169. doi: 10.1016/S0364-0213(85)80012-4.

[52] G.E. Hinton, Boltzmann Machines, Scholarpedia, (2007) 1–7.

[53] G. Hinton, G. Hinton, A Practical guide to training restricted boltzmann machines, neural networks: tricks of the trade. Springer, (2012) 599-619. doi: 10.1007/978-3-642-35289-8_32.

[54] H. Larochelle, Classification using discriminative restricted boltzmann machines, The 25th int. conf. on Machine learning ACM (2008) 536-543.doi: 10.1145/1390156.1390224.

[55] Y. Bengio, Learning deep architectures for ai, Foundations and trends® in Machine Learning 2.1 (2009) 1-27. doi: 10.1561/2200000006.

[56] M. Mohammadi, G.S. Member, A. Al-fuqaha, S. Member, Deep learning for iot big data and streaming analytics: a survey, IEEE Communications Surveys & Tutorials. 20 (201) 2923-2960. doi: 10.1109/COMST.2018.2844341.

[57] Z.C. Lipton, J. Berkowitz, C. Elkan, A critical review of recurrent neural networks for sequence learning, (2015) arXiv preprint arXiv:1506.00019.

[58] L. Busk Linnebjerg, R. Wetke, Long short term memory, Hear. Balanc. Commun. 12 (1997) 36–40. doi:10.3109/21695717.2013.794593.

[59] J. Chung, Empirical evaluation of gated recurrent neural networks on sequence modeling, (2014) arXiv preprint arXiv:1412.3555.

[60] A. Krizhevsky, I. Sutskever , G. Hinton, ImageNet classification with deep convolutional neural networks, Adv. Neural Inf. Process Syst, 25 (2012) 1097–1105.

[61] S. Haddadi, D.S. Kim, H. Jasmine, F. van der Meer, M. Czub, M.F. Abdul-Careem, Generative adversarial nets, Advances in neural inform. processing syst.. 155 (2014) 270–275. doi:10.1016/j.vetimm.2013.08.005.

[62] Kruegel, Christopher, Valeur, Fredrik, Vigna, Giovann, Computer secuirty and intrusion detection, in: intrusion detection and correlation, Springer 1 (2005) 8-29. doi: 10.1007/b101493.

[63] A. Patcha, J. Park, An overview of anomaly detection techniques : Existing solutions and latest technological trends, Computer networks, 51 (2007) 3448–3470. doi:10.1016/j.comnet.2007.02.001.

[64] J.M. Estevez-tapiador, P. Garcia-teodoro, J.E. Diaz-verdejo, Anomaly detection methods in wired networks: a survey and taxonomy, Computer Communications, 27 (2004) 1569–1584. doi:10.1016/j.comcom.2004.07.002.

[65] S. Agrawal, J. Agrawal, Survey on anomaly detection using data mining techniques, Procedia Comput. Sci. 60 (2015) 708–713. doi:10.1016/j.procs.2015.08.220.

[66] I. Butun, S.D. Morgera, R. Sankar, A survey of intrusion detection systems in wireless sensor networks, IEEE Communications Surveys & Tutorials, 16 (2014) 266–282. doi: 10.1109/SURV.2013.050113.00191.

[67] Bi, Jingjun, and C. Zhang, An empirical comparison on state-of-the-art multi-class imbalance learning algorithms and a new diversified ensemble learning scheme, Knowledge-Based Systems 158 (2018) 81-93. doi: 10.1016/j.knosys.2018.05.037.

[68] L. Deng, D. Yu, Deep learnign methods and applications, foundations and trends, in Signal Processing7.3–4 (2014): 197-387. doi: 10.1561/2000000039.

[69] G. Marín, P. Casas, RawPower : Deep learning based anomaly detection from raw network traffic measurements, ACM SIGCOMM 2018 Conference on Posters and Demo 7 (2018) 75–77. doi: 10.1145/3234200.3234238.

[70] Debar, Hervé, M. Dacier, and A. Wespi. Towards a taxonomy of intrusion-detection systems, Computer Networks 31.8 (1999) 805-822.

[71] H. Hindy, D. Brosset, E. Bayne, A. Seeam, C. Tachtatzis, R. Atkinson, X. Bellekens, A Taxonomy and survey of intrusion detection system design techniques, network threats and datasets, (2018).

[72] V.R. Balasaraswathi, M. Sugumaran, Y. Hamid, Feature selection techniques for intrusion detection using non-bio-inspired and bio-inspired optimization algorithms, J. Commun. Inf. Networks. 2 (2017) 107–119. doi:10.1007/s41650-017-0033-7.

[73] M. Kang, J. Kang, Intrusion detection system using deep neural network for in-vehicel

network security, PloS one 11.6 (2016) 1–17. doi:10.1371/journal.pone.0155781.

[74] L.O. Anyanwu, D. Ed, Scalable intrusion detection with recurrent neural Networks, Seventh Int. Conf. on Inform. Technology: New Generations (2010) 919-923. doi:10.1109/ITNG.2010.45.

[75] I. Sharafaldin, A. Gharib, A.H. Lashkari, A.A. Ghorbani, Towards a reliable intrusion detection benchmark dataset, Software Networking (2017) 177–200. doi:10.13052/jsn2445-9739.2017.009.

[Dataset] [76] Network-based intrusion detection (KDD99), University of California, Irvine (UCI). http://archive.ics.uci.edu/ml/datasets/kdd+cup+1999+data

[77] M. Tavallaee, E. Bagheri, W. Lu, A.A. Ghorbani, A detailed analysis of the kdd cup 99 dataset, IEEE Symposium on Computational Intelligence for Security and Defense Applications (2009) 1–6. doi: 10.1109/CISDA.2009.5356528

[Dataset] [78] Network-based intrusion detection (NSL-KDD), https://www.unb.ca/cic/datasets/nsl.html.

[79] S. Revathi, A. Malathi, A Detailed analysis on NSL-KDD dataset using various machine learning techniques for intrusion detection, Int. Journal of Engineering Research and Technology. ESRSA Publications, 2 (2013) 1848–1854.

[80] Jiang, Feng, and Y. Sui, A novel approach for discretization of continuous attributes in rough set theory, Knowledge-Based Systems 73 (2015) 324-334. doi: 10.1016/j.knosys.2014.10.014.

[81] R. Yahalom, A. Steren, Y. Nameri et al., Improving the effectiveness of intrusion detection systems for hierarchical data, Knowledge-Based Systems (2019), doi10.1016/j.knosys.2019.01.002.

[82] A. Milenkoski, M. Vieira, S. Kounev, A. Avritzer, B.D. Payne, Evaluating computer intrusion detection systems: a survey of common practices, ACM Computing Surveys (CSUR), 48 (2015). Doi: 10.1145/2808691.

[83] G. Gu, P. Fogla, D. Dagon, B. Skori, Measuring intrusion detection capability: an information-theoretic approach equivalent terms from ids literature, ACM Symposium on Information, computer and communications security (2006) 90-101. doi: 10.1145/1128817.1128834.

[84] J.E. Gaffney, J.W. Ulvila, Evaluation of intrusion detectors: a decision theory approach, IEEE Symposium on Security and Privacy. S&P (2001) 50–61. doi:10.1109/SECPRI.2001.924287.

[85] L. Wan, M. Zeiler, S. Zhang, Y. LeCun, and F. Fergus, Regularization of neural networks using Dropconnect, 30th Int. Conf. on Machine Learning, 28 (2013) 9.

[86] G. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, Improving neural networks by preventing co-adaptation of feature detectors, CoRR, 1 (2012)18.

[87] F. Farahnakian, J. Heikkonen, A deep auto-encoder based approach for intrusion detection system, Int. Conf. on Advanced Communication Technology (ICACT) (2018) 178-183. doi: 10.23919/ICACT.2018.8323688.

[88] S. Potluri, C. Diedrich, Accelerated deep neural networks for enhanced intrusion detection system, 21st Int. Conf. on Emerging Technologies and Factory Automation (ETFA)

(2016) 1-8. doi: 10.1109/ETFA.2016.7733515.

[89] M. Vartouni, An anomaly detection method to detect web attacks using stacked auto-encoder, Iranian Joint Congress on Fuzzy and Intelligent Syst. (CFIS) (2018) 131–134. doi: 10.1109/CFIS.2018.8336654.

[90] N. Shone, T.N. Ngoc, V.D. Phai, Q. Shi, A Deep learning approach to network intrusion detection, IEEE Trans. Emerg. Top. Comput. Intell. 1 (2017) 41–50. doi:10.1109/TETCI.2017.2772792.

[91] B. Zhang, Y. Yu, J. Li, Network intrusion detection based on stacked sparse autoencoder and binary tree ensemble method, IEEE Int. Conf. on Communications (2018) 1-6. doi: 10.1109/ICCW.2018.8403759.

[92] Cheng-Ru Wang, Rong-Fang Xu, Shie-Jue Lee, Chie-Hong Lee, Net-work intrusion detection using equality constrained-optimization-based extreme learning machines, Knowledge-Based Systems (2018), doi: 10.1016/j.knosys.2018.02.015.

[93] P.V. Dinh, T.N. Ngoc, N. Shone, Á. MacDermott, Q. Shi, Deep learning combined with de-noising data for network intrusion detection, Proc. - 2017 21st Asia Pacific Symp. Intell. Evol. Syst. IES (2017) 55–60. doi:10.1109/IESYS.2017.8233561.

[94] X. Zhang, J. Chen, Deep learning based intelligent intrusion detection, Int. conf. on Communication Software and Networks (ICCSN) (2017) 1133-1137. doi: 10.1109/ICCSN.2017.8230287.

[95] N.T. Van, T.N. Thinh, L.T. Sach, An anomaly-based network intrusion detection system using Deep learning, Int. Conf. Syst. Sci. Eng. ICSSE. (2017) 210–214. doi:10.1109/ICSSE.2017.8030867.

[96] M. Al-zewairi, S. Almajali, A. Awajan, Experimental evaluation of a multi-layer feed-forward artificial neural network classifier for network intrusion detection Syst., IFIP Int. Workshop on Information Security Theory and Practice, 7322 (2017). doi:10.1109/ICTCS.2017.29.

[97] Y. Fu, An intelligent network attack detection method based on rnn, IEEE Third Int. Conf. Data Sci. Cybersp. (2018) 483–489. doi:10.1109/DSC.2018.00078.

[98] G. Loukas, T. Vuong, R. Heartfield, G. Sakellari, Y. Yoon, D. Gan, Cloud-based cyber-physical intrusion detection for vehicles using Deep Learning, IEEE Access, 6 (2018) 3491-350. doi:10.1109/ACCESS.2017.2782159.

[99] F. Jiang, Y. Fu, B.B.G.Y. Liang, S. Rho, F. Lou, F. Meng, Deep learning based multi - channel intelligent attack detection for data security, IEEE Trans. on Sustainable Computing. (2018). doi:10.1109/TSUSC.2018.2793284.

[100] A., Wafaa, and J. Samarabandu, Comparison of recurrent nural networks algorithms for intrusoin detection based on predecting packet sequences, IEEE Canadian Conf. on Electrical & Computer Engineering (CCECE) (2018) 1-4. doi: 10.1109/CCECE.2018.8447793.

[101] M.U. Chowdhury, F. Hammond, G. Konowicz, C. Xin, H. Wu, J. Li, A Few-shot deep learning approach for improved intrusion detection, IEEE 8th Annual Ubiquitous Computing, Electronics and Mobile Communication Conf. (UEMCON) (2017) 456–462. doi: 10.1109/UEMCON.2017.8249084.

[102] W. Lin, H. Lin, P. Wang, B. Wu, J. Tsai, Using convolutional neural networks to network intrusion detection for cyber threats, IEEE Int. Conf. on Applied Syst. Invention (ICASI) (2018) 1107–1110. doi: 10.1109/ICASI.2018.8394474.

[103] D. Kwon, K. Natarajan, S.C. Suh, H. Kim, J. Kim, An empirical study on network anomaly detection using convolutional neural networks, IEEE 38th Int. Conf. on Distributed Computing Syst. (ICDCS) (2018) 1595–1598. doi:10.1109/ICDCS.2018.00178.

[104] S. Naseer, Y. Saleem, S. Khalid, M.K. Bashir, J. Han, M.M. Iqbal, K. Han, Enhanced network anomaly detection based on deep neural networks, IEEE Access. PP (2018) 1. doi:10.1109/ACCESS.2018.2863036.

[105] Z. Lin, Y. Shi, Z. Xue, IDSGAN: Generative adversarial networks for attack generation against intrusion detection, (2017) arXiv preprint arXiv:1809.02077.

[106] B. Lee, C. Green, Comparative study of deep learning models for entwork intrusion detection, SMU Data Science Review ,1 (2018).

[Dataset][107] Intrusion detection evaluation dataset (CICIDS2017), Canadian Institute for Cybersecurity (CIC), 2017. https://www.unb.ca/cic/datasets/ids-2017.html

[Dataset][108] Y. Meidan, M. Bohadana, Y. Mathov, Y. Mirsky, D. Breitenbacher, A. Shabtai, and Y. Elovici, N-BaIoT: IoT botnet attacks dataset, 2018. https://archive.ics.uci.edu/ml/datasets/detection_of_IoT_botnet_attacks_N_BaIoT

[Dataset][109] U. Adhikari, S. Pan, T. Morris, R. Borges, and J. Beave, Industrial control system (ICS) cyber-attack datasets, 2014. https://sites.google.com/a/ uah.edu/tommy-morris-uah/ics-data-sets