Received: 20 May 2020 | Revised: 31 August 2020 | Accepted: 29 September 2020

DOI: 10.1002/ett.4150

SURVEY PAPER

WILEY

Network intrusion detection system: A systematic study of machine learning and deep learning approaches

Zeeshan Ahmad[1,2] | **Adnan Shahid Khan**[1] | **Cheah Wai Shiang**[1] | **Johari Abdullah**[1] | **Farhan Ahmad**[3,4]

[1]Faculty of Computer Science and Information Technology, Universiti Malaysia Sarawak, Sarawak, Malaysia

[2]Department of Electrical Engineering, College of Engineering, King Khalid University, Abha, Kingdom of Saudi Arabia

[3]Cyber Security Research Group, College of Engineering and Technology, University of Derby, Derby, UK

[4]Institute for Future Transport and Cities, Coventry University, Coventry, UK

Correspondence
Adnan Shahid Khan, Faculty of Computer Science and Information Technology, Universiti Malaysia Sarawak, 94300 Kota Samarahan, Malaysia.
Email: skadnan@unimas.my

Abstract

The rapid advances in the internet and communication fields have resulted in a huge increase in the network size and the corresponding data. As a result, many novel attacks are being generated and have posed challenges for network security to accurately detect intrusions. Furthermore, the presence of the intruders with the aim to launch various attacks within the network cannot be ignored. An intrusion detection system (IDS) is one such tool that prevents the network from possible intrusions by inspecting the network traffic, to ensure its confidentiality, integrity, and availability. Despite enormous efforts by the researchers, IDS still faces challenges in improving detection accuracy while reducing false alarm rates and in detecting novel intrusions. Recently, machine learning (ML) and deep learning (DL)-based IDS systems are being deployed as potential solutions to detect intrusions across the network in an efficient manner. This article first clarifies the concept of IDS and then provides the taxonomy based on the notable ML and DL techniques adopted in designing network-based IDS (NIDS) systems. A comprehensive review of the recent NIDS-based articles is provided by discussing the strengths and limitations of the proposed solutions. Then, recent trends and advancements of ML and DL-based NIDS are provided in terms of the proposed methodology, evaluation metrics, and dataset selection. Using the shortcomings of the proposed methods, we highlighted various research challenges and provided the future scope for the research in improving ML and DL-based NIDS.

KEYWORDS

Deep learning, Machine learning, Network anomaly detection, Network intrusion detection system, Network security

1 | INTRODUCTION

With the recent interest and progress in the development of internet and communication technologies over the last decade, network security has emerged as a vital research domain. It employs tools like firewall, antivirus software, and intrusion detection system (IDS) to ensure the security of the network and all its associated assets within a cyberspace.[1] Among these, network-based intrusion detection system (NIDS) is the attack detection mechanism that provides the desired security by constantly monitoring the network traffic for malicious and suspicious behavior.[2,3]

The idea of IDS was first proposed by Jim Anderson in 1980.[4] Since then, many IDS products were developed and matured to satisfy the needs of network security.[5] However, the immense evolution in the technologies over the last decade has resulted in a large expansion in the network size, and the number of applications handled by the network nodes. As a result, a huge amount of important data is being generated and shared across different network nodes. The security of these data and network nodes has become a challenging task due to the generation of a large number of new attacks either through the mutation of an old attack or a novel attack. Almost every node within a network is vulnerable to security threats. For instance, the data node may be very important for an organization. Any compromise to the node's information may cause a huge impact on that organization in terms of its market reputation and financial losses. Existing IDSs have shown inefficiency in detecting various attacks including zero-day attacks and reducing the false alarm rates (FAR).[6] This eventually results in a demand for an efficient, accurate, and cost-effective NIDS to provide strong security to the network.

To fulfill the requirements of an effective IDS, the researchers have explored the possibility of using machine learning (ML) and deep learning (DL) techniques. Both ML and DL come under the big umbrella of artificial intelligence (AI) and aim at learning useful information from the big data.[7] These techniques have gained enormous popularity in the field of network security, over the last decade due to the invention of very powerful graphics processor units (GPUs).[8] Both ML and DL are powerful tools in learning useful features from the network traffic and predicting the normal and abnormal activities based on the learned patterns. The ML-based IDS depends heavily on feature engineering to learn useful information from the network traffic.[9] While DL-based IDS do not rely on feature engineering and are good at automatically learning complex features from the raw data due to its deep structure.[10]

Over the last decade, various ML- and DL-based solutions were proposed by the researchers to make NIDS efficient in detecting malicious attacks. However, the massive increase in the network traffic and the resulting security threats has posed many challenges for the NIDS systems to detect malicious intrusions efficiently. The research on using the DL methods for NIDS is currently in its early stage and there is still an enormous room to explore this technology within NIDS to efficiently detect intruders within the network. The purpose of this research paper is to provide a broad overview of the recent trends and advancements in ML- and DL-based solutions for NIDSs. The key idea is to furnish up-to-date information on recent ML- and DL-based NIDS to provide a baseline for the new researchers who want to start exploring this important domain. The main contributions of this article are 3-fold. (i) We conducted a systematic study to select recent journal articles focusing on various ML- and DL-based NIDS which are published during the last 3 years (2017-April 2020). (ii) We reviewed each article extensively and discussed its various features such as its proposed methodology, strength, weakness, evaluation metrics, and the used datasets. (iii) Based on these observations, we provided the recent trends of using AI methods for NIDS then highlighted various challenges in ML-/DL-based NIDs and we provided different future directions in this important domain.

There are many survey papers in the literature that provide some implementation details on the IDS. Our article is different from the other review articles[11-16] from three aspects: (i) We followed a systematic article selection process to obtain more focused articles on NIDS design considering AI tools. While the other studies reviewed the general IDS system without using the systematic approach. (ii) Our study reviewed the articles published between 2017 and April 2020. So it provides more updated information and the recent trends followed in the design of AI-based NIDS. (iii) In our study, an extensive review of the recent NIDS based on ML and DL approach is provided where they are critically analyzed according to their methods, techniques, datasets, and evaluation metrics. The focus is to provide researchers with more updated knowledge on AI-based NIDS in one place, where they can find the recent trends and potential research areas in the domain to start exploring it. A detailed comparison of this article with other review articles is provided in Table 1.

The rest of the paper is organized as follows: Section 2 describes the research methodology adopted in this study. Section 3 provides the basic IDS concept and classification methods. Section 4 elaborates the DL and ML methodologies adopted. The details about the evaluation metrics and the benchmark public datasets is illustrated in Section 5 and Section 6, respectively. Observations, recent trends in NIDS design, research challenges, and the future research scope are provided in Section 7. Finally, Section 8 concludes this review article. The abbreviations used in this article are summarized in Table 2.

TABLE 1 Comparison with other similar review articles: (✓: Yes, ×: No)

Review Article	Year	Systematic Study	NIDS Focused	AI-based approaches		Future Trends
				ML	DL	
Vasilomanolakis et al[11]	2015	×	×	✓	×	✓
Buczak et al[12]	2015	×	×	✓	×	✓
Thomas et al[13]	2018	×	✓	✓	×	✓
Liu et al[14]	2019	×	✓	✓	✓	✓
Khraisat et al[15]	2019	×	×	✓	×	✓
Da Costa et al[16]	2019	×	×	✓	✓	✓
This Article		✓	✓	✓	✓	✓

Abbreviations: AI, artificial intelligence; NIDS, network-based intrusion detection system.

TABLE 2 Summary of abbreviations

Abbreviation	Definition	Abbreviation	Definition
AE	Autoencoder	IoT	Internet of Things
AI	Artificial Intelligence	KNN	K-Nearest Neighbor
AIDS	Anomaly Intrusion Detection System	LSTM	Long Short Term Memory
ANN	Artificial Neural Network	ML	Machine Learning
CIC	Canadian Institute of Cyber security	NIDS	Network based Intrusion Detection System
CNN	Convolutional Neural Network	PCA	Principle Component Analysis
CSE	Communication Security Establishment	PSO	Particle Swarm Optimization
DBN	Deep Belief Network	R2L	Remote to Local
DL	Deep Learning	RBM	Restricted Boltzmann Machine
DNN	Deep Neural Network	ReLU	Rectified Linear Unit
DoS	Denial of Service	RF	Random Forest
DT	Decision Tree	RNN	Recurrent Neural Network
ELM	Extreme Learning Machine	SCADA	Supervisory Control and Data Acquisition
FAR	False Alarm Rate	SIDS	Signature based Intursion Detection System
FCN	Fully Connected Networks	SMOTE	Synthetic Minority Over-Sampling Technique
FLN	Fast Learning Network	SVM	Support Vector Machine
FN	False Negative	TN	True Negative
FP	False Positive	TP	True Positive
FSL	Few-shot Learning	UAV	Unmanned Aerial Vehicles
GPU	Graphics Processing Unit	U2R	User to Root
GRU	Gated Recurrent Unit	VAE	Variational Autoencoder
HIDS	Host-based Intrusion Detection System	WSN	Wireless Sensor Network
IDS	Intrusion Detection System		

2 | METHODOLOGY

This study conducts a systematic literature review of the different ML- and DL-based NIDS and investigates the published journal articles between *2017* to *first quarter of 2020*. A systematic literature review is a methodology followed to identify, examine, and extract needful information from the literature related to certain research topics.[17] We performed this systematic review in two phases. *Phase-1* identifies the information resource (search engine) and keywords to execute a query to obtain an initial list of articles. *Phase-2* applies certain criteria on the initial list to select the most related and core articles and store them into final list which are reviewed in this article. The main purpose of this review

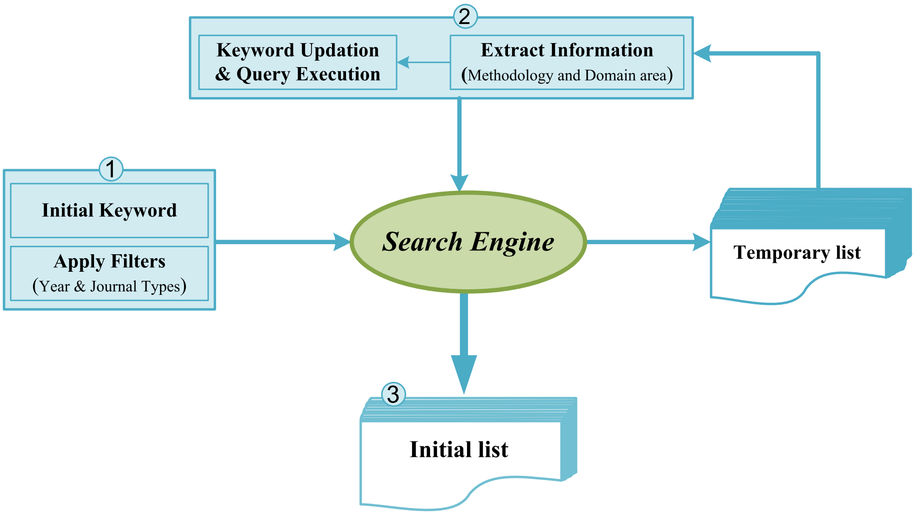

2
Keyword Updation & Query Execution
Extract Information (Methodology and Domain area)
1
Initial Keyword
Apply Filters (Year & Journal Types)
Search Engine
Temporary list
3
Initial list

FIGURE 1 Research methodology: Phase-1

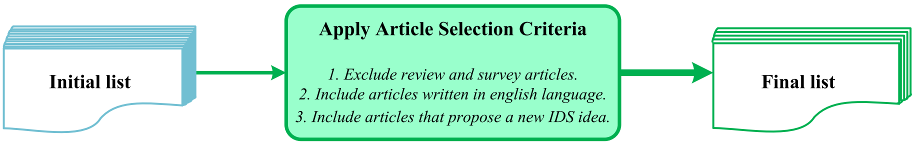

Initial list
Apply Article Selection Criteria
1. Exclude review and survey articles.
2. Include articles written in english language.
3. Include articles that propose a new IDS idea.
Final list

FIGURE 2 Research methodology: Phase-2

article is to answer the following questions: (i) What are the recent trends in the design of AI-based NIDS? (ii) What are the recent ML and DL methodologies adopted for NIDS design? (iii) What are the merits and demerits of each adopted methodology? (iv) Which datasets are recently used for the AI-based NIDS testing purposes? (v) What are the most frequent performance metrics used for evaluation purposes? and (vi) What is the future scope of research in AI-powered NIDS?

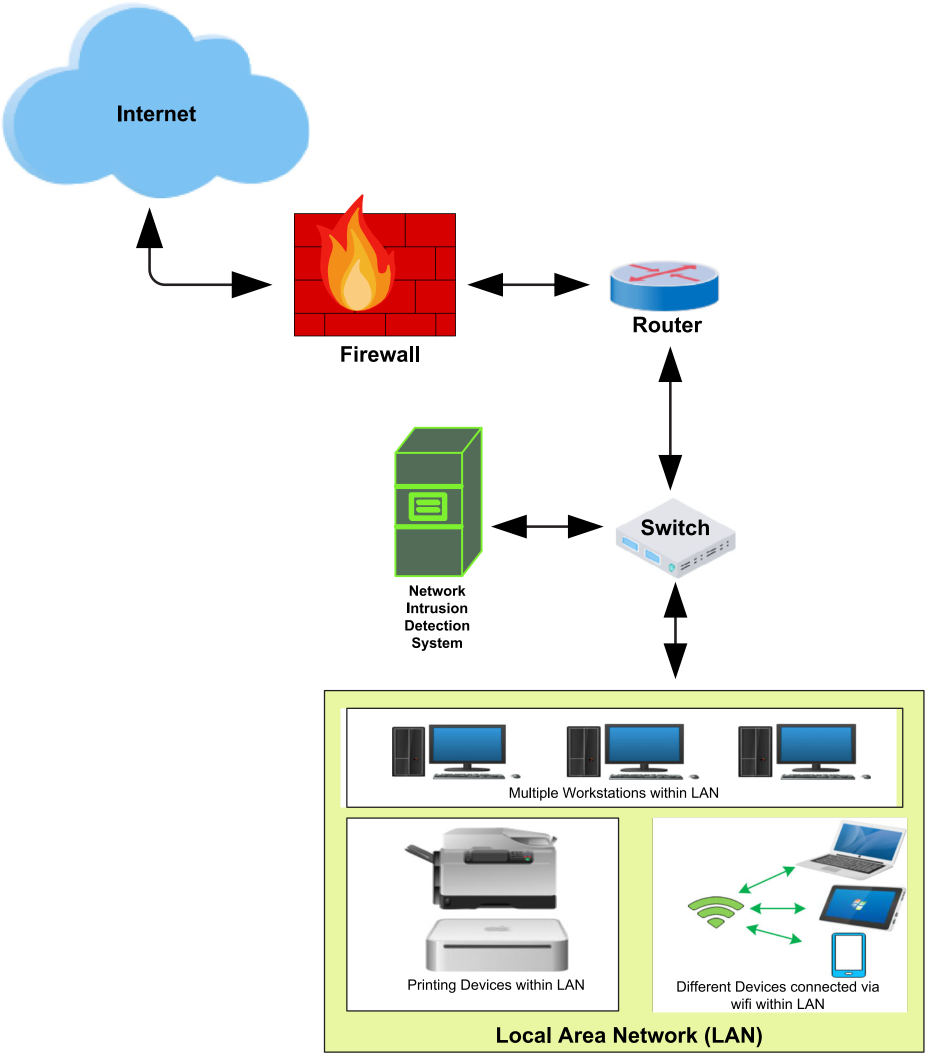
In phase-1, firstly search engines and keywords are identified for article search. A *Scopus document search*[18] is chosen as a potential search engine due to its ability to search from almost all the well-known databases. We executed a search QUERY using an initial keyword *"intrusion detection system"* and adjusted the filter to show journal articles published between 2017 and 2020. The initial search QUERY resulted in the articles that proposed the IDS using different approaches like AI-based, watchdog-based, trust-based and game theoretic-based, etc. for different domain areas including wireless sensor network (WSN), internet of things (IoT), cloud computing, etc. We then redefined our keyword as *intrusion detection system*, *network anomaly detection*, and *signature-based network intrusion detection*with the combination of *machine learning* or *deep learning* to obtain more relevant articles. As a result of phase-1, relevant articles based on the keywords were selected and stored as an initial list. The detailed steps used in phase-1 to obtain an initial list are summarized in Figure 1.

In phase-2, we started with the initial list and defined certain criteria to obtain articles that are more focused for review. We selected those articles which were written in the English language and proposed a new AI-based idea. We did not consider the review and survey articles. Based on these criteria, we were able to identify articles for this review, stored them in the final list, and then used them for the analysis purposes. Each selected article is analyzed based on the proposed ML- or DL-based methodology and the advantages and disadvantages of each methodology. We also analyzed the most frequent used datasets and evaluation metrics used for testing and evaluation purposes. Finally, we identify the future scope of research and challenges in the design of an efficient AI-based NIDS. The detailed process used in phase-2 for the selection of articles in the final list for review articles is summarized in Figure 2.

3 | IDS: CONCEPT AND CLASSIFICATION

This section first explains the concept of IDS and then provides the details about the classification of IDS based on its deployment and the detection methodology.

FIGURE 3 Passive deployment of network-based intrusion detection system


Internet
Firewall
Router
Switch
Network Intrusion Detection System
Multiple Workstations within LAN
Printing Devices within LAN
Different Devices connected via wifi within LAN
Local Area Network (LAN)

3.1 | Concept

An IDS is the combination of two words “intrusion” and “detection system.” *Intrusion* refers to an unauthorized access to the information within a computer or network systems to compromise its integrity, confidentiality, or availability.[19,20] While *detection system* is a security mechanism for the detection of such illegal activity. So, *IDS* is a security tool that constantly monitors the host and network traffic to detect any suspicious behavior that violates the security policy and compromises its confidentiality, integrity, and availability.[11,21] The IDS will generate alerts about detected malicious behavior to the host or network administrators. Figure 3 depicts a passive deployment of NIDS, where it is connected to a network switch configured with the port mirroring technology. The task is to mirror all the incoming and outgoing network traffic to NIDS for performing traffic monitoring to detect intrusions. NIDS can also be deployed in between the firewall and the network switch to allow all the traffic to pass through NIDS.[2]

3.2 | Classification of IDS

IDS can be classified with the perspective of its deployment or detection methods. A classification taxonomy is given in Figure 4.

3.2.1 | Deployment method based IDS

From the deployment-based IDS perspective, IDS is further subclassified as host-based-IDS (HIDS) or NIDS.[19,22] HIDS is deployed on the single information host. Its task is to monitor all the activities on this single host and scans for its security

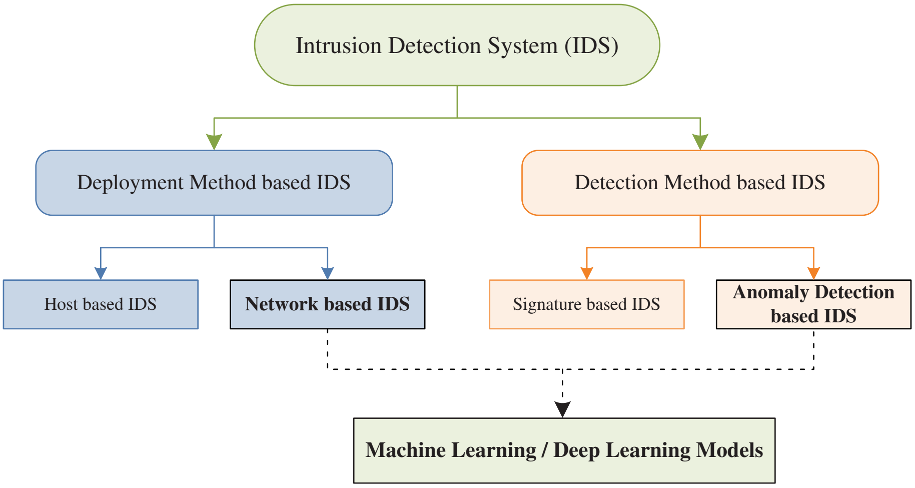

Intrusion Detection System (IDS)
Deployment Method based IDS
Detection Method based IDS
Host based IDS
Network based IDS
Signature based IDS
Anomaly Detection based IDS
Machine Learning / Deep Learning Models

FIGURE 4 Intrusion detection system classification taxonomy

policy violations and suspicious activities. The main drawback is its deployment on all the hosts that require intrusion protection, which results in extra-processing overhead for each node and ultimately degrades the performance of the IDS.[23,24] In contrast, NIDS is deployed on the network with the aim to protect all devices and the entire network from intrusions. The NIDS will constantly monitor the network traffic and scans for potential security breaches and violations. This article focuses on the different methods used in the NIDS.

3.2.2 | Detection method based IDS

From the detection IDS perspective, the IDS is further subdivided into "Signature-based intrusion detection (SIDS)" and "Anomaly detection-based intrusion detection (AIDS)". SIDS, also known as the "misuse intrusion detection" or "knowledge-based intrusion detection," is based on the idea of defining a signature for attack patterns. These signatures are stored in the signature database and the data patterns are matched with these stored signatures for attack detection.[25,26] The advantage includes the high detection efficiency for the known attacks due to the availability of signature for those attacks. On the other hand, this method lacks the ability to detect the novel and new attacks due to the absence of signature patterns.[23] Also, a huge signature database is maintained and compared with the data packets for possible intrusions, which makes it a resource-consuming approach.[27] AIDS, also called the "behavior-based IDS," is based on the idea of clearly defining a profile for normal activity. Any deviation from this normal profile will be considered as an anomaly or abnormal behavior.[28,29] The major advantages of AIDS are its ability to detect unknown and new attacks[30] and the customized nature of the normal activity profile for different networks and applications.[31] However, the main drawback is the high FAR as it is difficult to find the boundary between the normal and abnormal profiles for intrusion detection.[32] The popularity of the IoT paradigm due to network technologies advancement has resulted in exponential growth in the use of IoT devices.[33,34] One of the vital technology used in the development of an IoT network is the WSN, which comprises of a collection of sensor nodes for information collection.[35] A huge amount of critical information is collected by these IoT sensor devices and is shared over the internet.[36] This big data along with the complex structure of WSN comprising of the resource-limited sensor nodes causes security challenges for the IoT network.[37,38] To this end, IDS is considered as one of the effective mechanism for the security of IoT and WSN. Many different IDS approaches are proposed in the literature, which are based on the efficient use of watchdogs, trust models, and game-theoretic concepts.

Watchdogs are the network nodes that are assigned the task of watching and monitoring the neighboring nodes' network traffic. Then a decision is made regarding the misbehaving nodes by using some set of rules. Many solutions are proposed for the anomaly and intrusion detection using watchdogs in the domain of WSN,[39] AdHoc networks,[40] and IoT.[41] Trust models are another tool used to improve the performance of an IDS. An IDS based on the trust model evaluates the trustworthiness of their nodes to identify the malicious nodes by constantly mentoring the network traffic for abnormal behaviors. Different implementations of the IDS using trust models are based on watchdog,[42] Bayesian trust model,[43] and game theory-based trust model.[44,45] In the context of IoT, a trust management scheme can be used in a distributed manner to reduce the computational overheads of resource-constrained sensor nodes.[46,47] Similarly, game theory is widely used for the efficient designing of IDS. It is an applied mathematical concept used to model the strategic interactions among the players by describing a game. Each game includes the set of players and each player has a set of strategies along with an action plan and payoff for each action within a game. The solution of the game is based on an equilibrium state which is based on the player's strategy to maximize the payoff. A game can be cooperative or noncooperative depending upon entities' interaction cooperatively or competitively. From the perspective of IDS for IoT and WSN, a game is modeled between the attackers and the defenders either by their interaction or by using the prediction strategy of an attacker.[45,48-50]

In this article, we have focused on reviewing the AI-based NIDS, which can be deployed for the security of an IoT network by monitoring the network traffic entered through the edge router. The most common AI-based algorithms used in the design of an efficient NIDS over the past three years are briefly explained in Section 4.

4 | AI METHODS FOR NIDS

This section provides a general methodology of the AI-based NIDS along with the details of the most commonly used ML and DL algorithms used to design an efficient NIDS. Both ML and DL are broadly classified as supervised and unsupervised algorithms.[51] In *supervised algorithms*, the useful information is extracted from the labeled data. While *unsupervised algorithms* rely on the unlabeled data to extract useful features and information.[52]

4.1 | A general AI-based NIDS methodology

A NIDS developed using ML and DL methods usually involves following three major steps as depicted in Figure 5, that is, (i) Data preprocessing phase, (ii) Training phase, and (iii) Testing phase. For all the proposed solutions, the dataset is first preprocessed to transform it into the format suitable to be used by the algorithm. This stage typically involves encoding and normalization. Sometimes, the dataset requires cleaning in terms of removing entries with missing data and duplicate entries, which is also performed during this phase. The presprocessed data is then divided randomly into two portions, the *training dataset*, and the *testing dataset*. Typically, the training dataset comprises almost 80% of the original dataset size and the remaining 20% forms testing dataset.[53,54] The ML or DL algorithm is then trained using the training dataset in the training phase. The time taken by the algorithm in learning depends upon the size of the dataset and the complexity of the proposed model. Normally, the training time for the DL models requires more training time due to its deep and complex structure. Once the model is trained, it is tested using the testing dataset and evaluated based on the predictions it made. In the case of NIDS models, the network traffic instance will be predicted to belong to either benign (normal) or attack class.

In the following section, we provide an extensive overview of widely used ML and DL algorithms for NIDS systems. Further, Figure 6 highlights the taxonomy of recent ML- and DL-based techniques used for NIDS.

4.2 | ML algorithms

ML is a subset of AI that includes all the methods and algorithms which enable the machines to learn automatically using mathematical models in order to extract useful information from the large datasets.[13,55] The most common ML

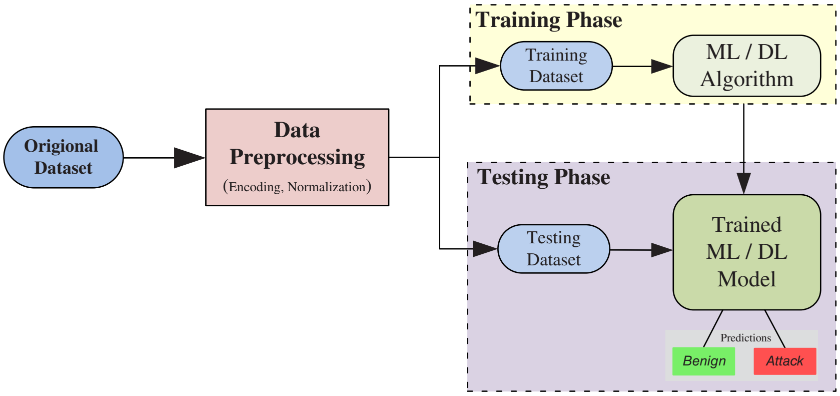


FIGURE 5 Generalized machine learning-/deep learning-based network-based intrusion detection system methodology

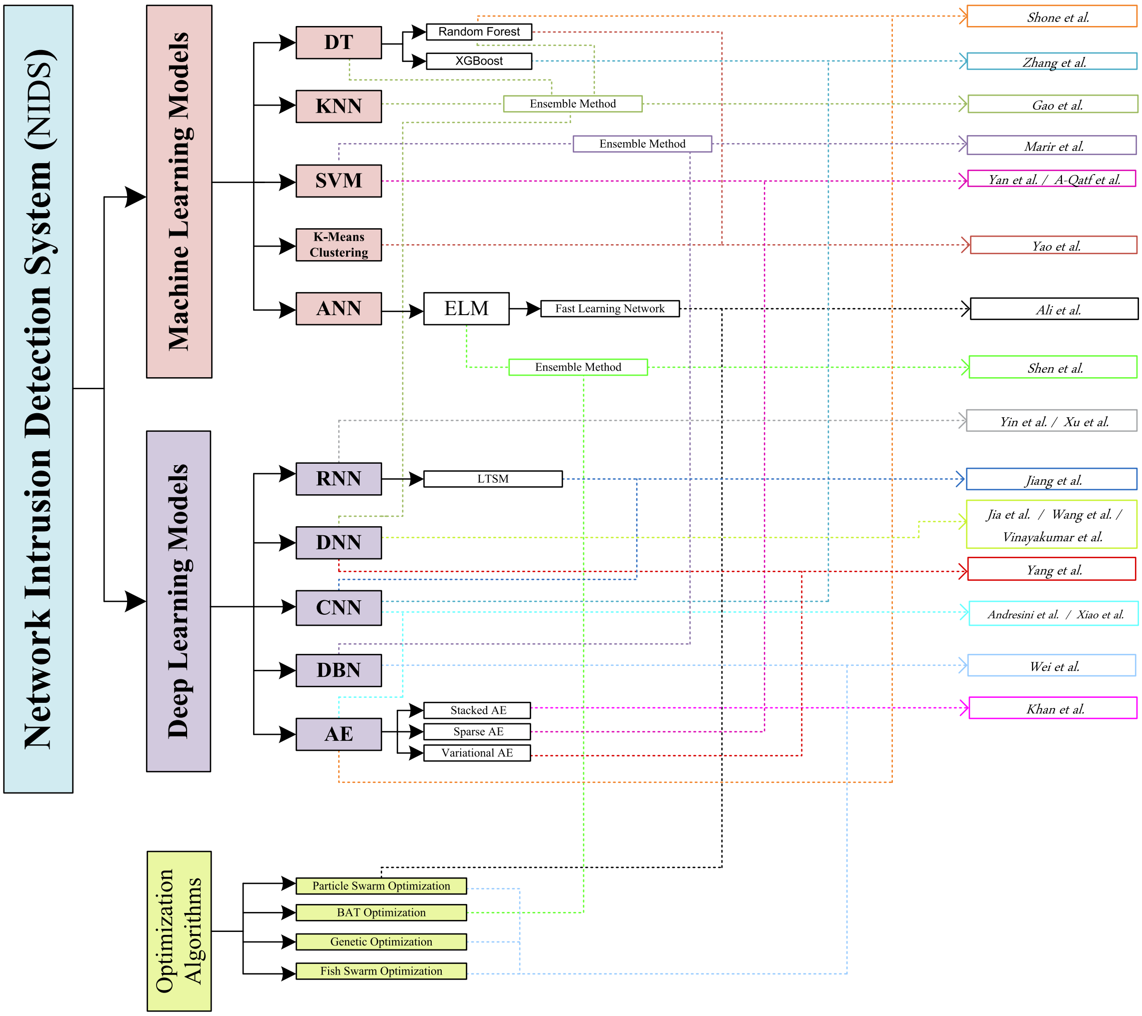


FIGURE 6 Taxonomy For machine learning- and deep learning-based network-based intrusion detection system

(also called Shallow Learning) algorithms used for IDS are Decision Tree, *K*-Nearest Neighbor (KNN), Artificial Neural Network (ANN), Support Vector Machine (SVM), *K*-Mean Clustering, Fast Learning Network, and Ensemble Methods.

4.2.1 | Decision tree

DT is one of the basic supervised ML algorithms which is used for both classification and regression of the given dataset by applying the series of decisions (rules). The model has a conventional tree structure with nodes, branches, and leaf.[56] Each node represents an attribute or a feature. The branch represents a decision or a rule while each leaf represents a possible outcome or class label.[57] The DT algorithm automatically selects the best features for building a tree and then perform pruning operation to remove irrelevant branches from the tree to avoid the over-fitting. The most common DT models are CART, C4.5, and ID3.[58] Many advanced learning algorithms like Random Forest (RF)[59] and XGBoost[60] are made from multiple decision trees.

4.2.2 | *K*-Nearest Neighbor

KNN is one of the simplest supervised ML algorithms which utilizes the idea of "feature similarity" to predict the class of a certain data sample. It identifies a sample based on its neighbors by calculating its distance from the neighbors. In the KNN algorithm, the parameter *k* affects the performance of the model. If the value of *k* is very smaller, the model may be susceptible to over-fitting. While, a very large selection of k value may result in misclassification of the sample instance.[61,62] Karatas et al[63] compared the performance of different ML algorithms using an up-to-date benchmark dataset CSE-CIC-IDS2018. They addressed the dataset imbalance problem by reducing the imbalance ratio using Synthetic Minority Oversampling Technique (SMOTE),[64] which resulted in detection rate improvement for minority class attacks.

4.2.3 | Support vector machine

SVM is a supervised ML algorithm based on the idea of max-margin separation hyper-plane in n-dimensional feature space. It is used for the solution of both linear and nonlinear problems. For nonlinear problems, kernel functions are used. The idea is to first map a low dimensional input vector into a high dimensional feature space using the kernel function. Next, an optimal maximum marginal hyper-plane is obtained, which works as a decision boundary using the support vectors.[65,66] For NIDS, the SVM algorithm can be used to enhance its efficiency and accuracy by correctly predicting the normal and malicious classes.[67,68]

4.2.4 | *K*-mean clustering

The clustering is an idea of dividing data into meaningful clusters (or groups), by putting the highly similar data into the same cluster. K-Mean clustering is one of the popular centroid-based iterative ML algorithm that learns in an unsupervised manner. K represents the number of centroids (center of the cluster) within a dataset. For assigning certain data points to a cluster, normally distance is calculated. The main idea is to reduce the sum of the distances between the data points and their respective centroids within a cluster.[69-71]

Yao et al[72] proposed a multilevel intrusion detection model framework named multilevel semi-supervised ML (MSML) to address IDS, The clustering concept is used along with the RF model. The proposed solution was composed of four modules as pure cluster extraction, pattern discovery, fine-grained classification, and model updating. The idea is if an attack is not labeled in one module then it is forwarded to the next one for detection. The proposed methodology was tested using the KDD Cup'99 dataset. Experimental results showed the model superiority for detecting the attacks even with low instances in the dataset.

4.2.5 | Artificial neural network

ANN is also a supervised ML algorithm and is inspired by the working of the nervous system of the human brain. It is made up of the processing elements called the neurons (nodes) and the connection between them. These nodes are organized in an input layer, many hidden layers, and an output layer. The backpropagation algorithm is used as a learning technique for the ANN. The main advantage of using an ANN technique is its ability to perform nonlinear modeling by learning from larger datasets.[73] However, the main issue with training the ANN model is the high time consumption due to its complex nature,[74] which slows down the learning process and to reach a suboptimal solution.[75]

To overcome the limitations of ANN, Huang et al[76] proposed a new ANN called an extreme learning machine (ELM). The ELM is a single hidden layer feed-forward neural network, which randomly uses the input weights and hidden layer bias without tuning and determines the output weights in an analytical way.[77] Based on the idea of ELM, Li et al[78] proposed a Fast Learning Network (FLN). FLN is based on connecting the multilayer feed-forward neural network and a single-layer feed-forward neural network in parallel. FLN showed reasonable performance and stability using a smaller number of hidden nodes and utilizing less time.

Ali et al[79] addressed the IDS problem by proposing a model based on FLN and particle swarm optimization[80] (PSO-FLN) and tested the model using KDD Cup'99 dataset. The model was tested by comparing the FLN with different optimization algorithms. Results showed that PSO-FLN outperforms the other FLN models with different optimization algorithms as Genetic Algorithm, Harmoney Search optimization, and Ameliorated Teaching Learning-based optimization. They also demonstrated that increasing the number of neurons in the hidden layer increases accuracy. The main drawback is the lower detection rate accuracy for lower attack classes.

4.2.6 | Ensemble methods

The key idea behind ensemble methods is to get benefit from the different classifier by learning in an ensemble way. Since each classifier has some strengths and weaknesses. Some may perform well for detecting a specific type of attacks and shows poor performance on other types of attack. The ensemble approach is to combine weak classifiers by training multiple classifiers and then form a stronger classifier by selecting using a voting algorithm.

Shen et al[81] proposed an IDS using an ensemble method by selecting ELM as a base classifier. To optimize the proposed methodology, a BAT optimization algorithm is used during the ensemble pruning phase. The model was tested using KDD Cup'99, NSL-KDD, and Kyoto datasets. Experimental results showed that many ELMs combined in ensemble manner outperform individual ELM in performance.

Gao et al[82] proposed an adaptive ensemble model by using several base classifiers as DT, RF, KNN, Deep Neural Network (DNN), and choosing the best using adaptive voting algorithm. The proposed methodology was verified by performing experiments using the NSL-KDD dataset. Experimental results demonstrated the performance efficiency by comparing other models. The proposed methodology did not have satisfactory results for the weaker attack classes.

4.3 | Deep learning algorithms

DL is the subset of the ML which includes many hidden layers to get the characteristics of the deep network. These techniques are more efficient than the ML due to their deep structure and ability to learn the important features from the dataset on its own and generate an output. This section presents the DL approaches adopted to propose DL-based NIDS solutions in the reviewed articles.

4.3.1 | Recurrent neural networks

Recurrent Neural Networks (RNN) extends the capabilities of the traditional feed-forward neural network and is designed to model the sequence data. RNN is made of input, hidden, and output units, where the hidden units are considered to be the memory elements. To make a decision, each RNN unit relies on its current input and the output of the previous input. RNN is widely used in different fields like speech processing, human activity recognition, handwriting prediction, and semantic understanding, to name a few.[83-86] For an IDS, RNN can be used for the supervised classification and feature extraction. RNN normally can handle limited length sequences and will suffer from short-term memory if the sequence length is long.[87] Different RNN variants like Long short-term memory (LSTM)[88] and gated recurrent unit (GRU)[89,90] are proposed to solve these issues.

RNN-based IDS was proposed by Yin et al[91] in the context of binary and multi class classification of the NSL-KDD dataset. The model was tested using a different number of hidden nodes and learning rates. Results showed that different learning rates and the number of hidden nodes affect the accuracy of the model. Best accuracy was obtained using 80 hidden nodes and a learning rate of 0.1 and 0.5 for binary and multi class scenarios. The proposed model performed well compared to ML algorithms and a reduced-sized RNN model proposed in Reference 92. The main shortcoming of this work is the increase in computational processing which results in high model training time and lower detection rate for the R2L and U2R classes. The article also lacks the performance comparison of the proposed model with different other DL methodologies.

In Reference 93, Xu et al proposed an IDS based on RNN using GRU as the main memory together with the multilayer perceptron and a softmax classifier. The proposed methodology was tested using KDD Cup'99 and NSL-KDD datasets. Experimental results showed good detection rates for comparing other methodologies. The major drawback of their model is lower detection rates for minority attack classes like U2R and R2L.

Naseer et al[94] performed a comparative analysis of IDS based on different DL and ML algorithms and implemented on GPU-based testbed. NSL-KDD is considered as the benchmark dataset and the experimental results showed that LSTM and Deep CNN achieved higher accuracy results comparing other models.

4.3.2 | AutoEncoder

AutoEncoder (AE) is a popular DL technique that belongs to the family of unsupervised neural networks.[95] It works on the idea of matching the output as close to input as possible by learning the best features. It contains input and output layers of the same dimension, while the dimensions of the hidden layers are normally smaller than the input layer. AE is symmetric and works in Encoder-Decoder fashion. Different variants of AE are Stacked AE, Sparse AE, and Variational AE.[96]

Shone et al[97] proposed an IDS based on deep AE and ML technique RF. To make the model efficient in terms of computational and time, only the encoder part of AE is utilized to make it work in a nonsymmetric fashion. Two non-symmetric deep AEs, with three hidden layers each, are arranged in a stacked manner. RF was used for classification. Experiments were performed for multiclass classification scenarios using KDD Cup '99 and NSL-KDD datasets. The proposed method showed their efficiency compare to Deep Belief Network (DBN) used in Reference 98 in terms of detection accuracy and reduced training time. But the model showed inefficiency for detecting R2L and U2R attacks due to lack of data for training the model.

Yan et al[67] proposed an IDS using stacked sparse autoencoder (SSAE) and SVM. The SSAE was used as the feature extraction method and SVM as a classifier. Binary-class and multi-class classification problem is considered for conducting experiments. The results showed the proposed model superiority in performance comparing different feature selection, ML, and DL methods using the NSL-KDD dataset. Although, the model achieves reasonable detection rates for U2R and R2L attacks but it is still less comparing the other classes of the dataset.

A-Qatf et al[99] also proposed a similar idea of self-taught learning based on sparse AE and SVM. To validate their performance, they performed experiments on the proposed model considering the NSL-KDD dataset. The results showed improved overall performance comparing other DL and ML models. But the proposed methodology performance in R2L and U2R class is not discussed.

Papmartizivanous et al[100] proposed an autonomous misuse detection system by combining the advantages of self-taught learning[101] and MAPE-K frameworks.[102] They used sparse AE for the unsupervised learning algorithm to learn useful features while performing the Plan activity within the MAPE-K Framework. Experiments performed using the KDD Cup'99 and NSL-KDD datasets. The main drawback is the lack of detection accuracy for U2R and R2L attack classes.

Khan et al[103] proposed an efficient two-stage model based on deep stacked AE. The initial stage classified the dataset into the attack and normal classes with probability values. These probability scores are then used as an additional feature and are input to the final decision stage for normal and multiclass attack classification. The performance of the proposed model was tested using KDD Cup'99 and UNSWNB15 datasets. To reduce the problems due to class imbalance of the datasets, a different methodology was adopted for both datasets. For KDD Cup'99, the downsampling was performed to remove repeated records. While, to balance the distribution of records in UNSWNB15, upsampling of the dataset was performed using SMOTE. This preprocessing of the dataset dramatically improves the DR efficiency of attack class with lower training instances.

Malaiya et al[104] proposed different IDS models based on fully connected networks, Variational AE, and Sequence-to-Sequence (Seq2Seq) structures, respectively. These models were examined for different datasets NSL-KDD, KyotoHoneypot, UNSW-NB15, IDS2017, and MAWILab traces.[105] Results showed that the Seq2Seq model constructed using two RNNs performed the best comparing other models in terms of detection accuracy across all the datasets.

Yang et al[106] proposed a model for ID based on the supervised adversarial variational AE with regularization and DNN (SAVAER-DNN). The performance of the model was tested using benchmark data NSL-KDD and UNSW-NB15. Experimental results confirm the model's effectiveness in detecting low frequency and new attacks.

Andresini et al[107] incorporated the idea of AE to proposed a multistage model involving the ID convolution layer and two stacked fully connected layers. In the initial unsupervised stage, two AEs were trained separately using Normal and Attack flows to reconstruct the samples again. In the supervised stage, these new reconstructed samples are used to build a new augmented dataset that is used as input to a 1D-CNN. Then the output of this convolution layer is flattened and fed to fully connected layers, and lastly, a softmax layer classifies the dataset. Experiments were performed on the KDD Cup'99, UNSWNB15, and CICIDS2017 datasets and the proposed methodology achieves superior performance comparing different DL models. They have not shown how the minority classes perform using this methodology. The second drawback is that it does not provide any information on the characteristics of the attack.

4.3.3 | Deep neural network

DNN is a basic DL structure that allows the model to learn in multiple layers. It is composed of an input layer, an output layer, and many hidden layers. DNN is used to model complex nonlinear functions. Increased number of hidden layers enhances the abstraction level of the model to increase its capability.[108] Jia et al[109] proposed a network IDS based on DNN with four hidden layers to classify the datasets KDD cup'99 and NSL-KDD. The output layer included one fully connected layer and softmax classifier for classification purposes. For the hidden layer, a rectified linear unit was used as the activation function.[110] Results showed the robustness of the proposed model as it achieved higher detection rates for almost all the attack classes except U2R due to presence of less number of records. According to the authors, increasing the number of nodes and layers leads to a complex structure that increases the computing time and consumes more resources. The solution to these issues is the optimization algorithm and automatic tuning.

Wang et al[111] studied the DNN-based IDS with adversaries and evaluated using the NSL-KDD dataset. They comprehensively studied the roles of individual features in generating adversarial examples. The adversarial samples were produced by FGSM,[112] JSMA,[113] DeepFool,[114] and CW attacks.[115] Results showed that the most commonly used attributes are more vulnerable to DL-based IDS and require more attention to safeguard the network from attacks.

Vinayakumar et al[116] proposed a hybrid scalable DNN framework called as scale-hybrid-IDS-AlertNet, for intrusion identification at both host and network level. Apache Spark cluster computing platform[117] was used for implementing the scalable platform. For NIDS, the proposed model was tested using publically available datasets like KDDCup 99, NSL-KDD, Kyoto, UNSW-NB15, WSN-DS, and CICIDS 2017. Experiment results showed the superiority of the proposed model comparing different ML algorithms.

4.3.4 | Deep belief network

DBN is a DL model constructed by stacking many Restricted Boltzmann Machines (RBM) in layers followed by a softmax classification layer.[118] An RBM is a two-layer (input and hidden layer) model with the data flow in both directions. In DBN, each node within a layer is connected to each of the other nodes in the previous and next layers, but within one layer, the nodes are not connected. DBN is pretrained using the greedy layer-wise learning approach in an unsupervised fashion, followed by a supervised fine-tuning methodology for learning useful features.[119] For IDS, DBN is used for feature extraction and classification tasks.

Marir et al[120] proposed a distributed model based on the BDN and multilayer ensemble SVM for large-scale network ID based on Apache Spark. DBN was used for extracting features, which are then forwarded to the ensemble SVM, and then finally output was predicted using a voting mechanism. The efficiency of the proposed method was tested for KDD CUP'99, NSL-KDD, UNSW-NB15, and CICIDS2017 datasets. The proposed system has shown high performances in the detection of abnormal behavior in a distributed way.

To improve the detection accuracy of IDS, Wei et al[121] proposed a DL-based model DBN, which is optimized by combining the particle swarm, fish swarm, and genetic algorithms. The model was tested using the NSL-KDD dataset. Results showed a large improvement in the detection rate of U2R and R2L class. The main drawback of the proposed model is the increase in the training time of the model due to its complex structure.

4.3.5 | Convolutional neural network

Convolutional neural network (CNN) is a DL structure more suitable for the data stored in arrays. It consists of an input layer, the stack of convolutional and pooling layers for feature extraction, and finally a fully connected layer and a softmax classifier in the classification layer. CNN is widely successful in the computer vision field.[122] For the IDS, they are used for the supervised feature extraction and classification purposes.

Xiao et al[123] proposed an efficient IDS based on the CNN. The main idea is first to perform feature extraction using Principle Component Analysis and AE. Then transform the one-dimensional vector (feature set) into a two-dimensional matrix and input to the convolution Neural Network. Experiments were performed on the KDD Cup'99 dataset. Experiments show its effectiveness in terms of time taken by algorithms in the training and test phase. The main drawback is lower detection rates for the U2R and R2L classes comparing to other attack classes.

Zhang et al[124] proposed a complex multilayer IDS model based on CNN and gcForest. They also proposed a novel P-Zigzag algorithm for converting the raw data into two-dimensional greyscale images. They used an improved CNN model (GoogLeNetNP) in a coarse grain layer for initial detection. Then in the fine-grained layer, gcForest (caXGBoost) is used to further classifies the abnormal classes into *N*-1 subclasses. They used a dataset by combining UNSW-NB15 and CIC-IDS2017 datasets. The experimental results show that the proposed model significantly improves the accuracy and detection rate compared to the single algorithms while reducing the FAR.

Jiang et al[125] proposed an efficient IDS system by combining CNN and bidirectional long short-term memory (BiLSTM) in a deep hierarchy. The class imbalance problem is addressed by using the SMOTE to increase the minority samples, which helps the model to fully learn the features. The CNN was used for extracting spatial features while BiLTSM was used to temporal features. Experiments were performed using NSL-KDD and UNSWNB15 datasets. The proposed methodology achieves higher performance in terms of accuracy and detection rate. The detection rate of minority data classes improved slightly but still its is very low comparing other attack classes. Due to the complex structure, the training time is higher.

Yu et al[126] proposed an IDS model based on novel DL idea of Few-shot Learning (FSL).[127] The idea is to train using a small amount of balanced labeled data from the dataset. DNN and CNN are adopted as embedding functions in the model for extracting the essential feature and reducing the dimension. Experimental results performed on NSL-KDD and UNSW-NB15 datasets showed model efficiency in getting reasonable detection rates for minority attack classes. The proposed model only utilized less than 2% data for training to achieve such a remarkable performance for the considered dataset.

In this section, we identified various ML and DL techniques (as summarized in Table 3), which are developed by researchers recently to detect intruders in the network. Also, the strengths and weaknesses of each methodology is highlighted in Table 4. However, these techniques need to be evaluated based on certain metrics. In the next section, we

TABLE 3 Recent machine learning- (ML) and deep learning (DL)-based network-based intrusion studies and their methods

Study	Algorithm		Methodology
	ML	DL	
Yin et al[91]		✓	Recurrent Neural Network
Shen et al[81]	✓		Ensemble Method with Optimization using BAT algorithm
Shone et al[97]	✓	✓	Non Symmetric Deep Auto Encoder with Random Forest
Ali et al[79]	✓		Fast Learning Network and Particle Swarm Algorithm
Jia et al[109]		✓	Deep Neural Network
Wang et al[111]		✓	Deep Neural Network
Yan et al[67]	✓	✓	Sparse Auto Encoder with SVM
Naseer et al[94]	✓	✓	Comparison between different ML and DL-based IDS Models
Xu et al[93]		✓	Neural Network using gated Recurrent Units as memory with Multiple Layer Perception.
Al-Qatf et al[99]	✓	✓	Self Taught Learning Model based on Sparse Auto Encoder and SVM
Marir et al[120]	✓	✓	Deep Belief Network and SVM
Papamartizivanous et al[100]		✓	Self Taught Learning based on Sparse Auto Encoder
Khan et al[103]		✓	Two-Stage Model using Stacked Auto Encoder
Xiao et al[123]		✓	Convolutional Neural Network with Principal component analysis (PCA) and Auto Encoder for dimension reduction
Yao et al[72]	✓		A Multilevel Model based on *K*-Means Clustering and Random Forest
Vinayakumar et al[116]		✓	Deep Neural Network
Gao et al[82]	✓		Ensemble Machine Learning methods with Voting algorithm
Wei et al[121]		✓	Deep Belief Network along with optimization algorithms Particle Swarm, Fish Swarm and Genetic Algorithms.
Zhang et al[124]		✓	Multi Layer Convolutional Neural Network
Malaiya et al[104]		✓	Model based on Fully Connected Networks (FCNs), Variational AutoEncoder (VAE),and Sequence-to-Sequence (Seq2Seq) structures
Karatas et al[63]	✓		Performance Comparison of different ML algorithm by first reducing the dataset imbalance ratio using (SMOTE)
Jiang et al[125]		✓	Deep Hierarchical Network based on CNN and BiLSTM
Yang et al[106]		✓	Supervised adversarial Variational Auto Encoder with regularization (SAVAER) and Deep Neural Network (DNN)
Yu et al[126]		✓	Few Shot Learning (FSL) using DNN and CNN
Andresini et al[107]		✓	Multistage Auto Encoder and CNN

TABLE 4 Strengths and weaknesses of the proposed methodologies

Study	Strengths	Weaknesses
Yin et al[91]	A NIDS based on RNN model is proposed that achieves higher performance in terms of detection accuracy using ML algorithms	The model has increased complexity and requires more time for training. Also, it uses an older dataset NSL-KDD for model evaluation. Results demonstrate lower detection rates for minority attack classes like R2L and U2R.
Shen et al[81]	An idea of using a BAT optimization algorithm together with the Ensemble method based on ELM as the base classifier. Many ELMs combined in an Ensemble way outperform individual ELM performance.	The use of old datasets KDD Cup'99, NSL-KDD, and Kyoto for model evaluation. Moreover, the model achieves lower detection accuracy for the U2R attack class.
Shone et al[97]	Proposed a NIDS model using a non-symmetric deep AE and RF classifier. The complexity of the model is reduced by using non-symmetric deep AE for efficient feature selection.	The model is evaluated using old datasets KDD Cup'99 and NSL-KDD. The model performance for minority classes R2L and U2R is on the lower side.
Ali et al[79]	Use of FLN and particle swarm optimization algorithm for proposing a NIDS model. The proposed model showed better performance than other models based on FLN with different other optimization algorithms.	The model is tested using a very old dataset KDD Cup'99. Also, the detection rate for the attack class with less training data (R2L and U2R) is low.
Jia et al[109]	Proposed a NIDS based on DNN with four hidden layers that have shown superior performance than IDS based on ML methods.	For validating the model, older datasets KDD cup'99 and NSLKDD are used. Also, lower detection accuracy for the U2R attack class is recorded.
Wang et al[111]	Studied the effect of DNN-based IDS against adversaries and evaluated the state-of-the-art attack algorithms against DNN based intrusion detection on the NSL-KDD dataset.	Use of older dataset NSLKDD for validation purposes.
Yan et al[67]	Efficient use of stacked sparse autoencoder (SSAE) for feature extraction and SVM as a classifier to propose a NIDS.	An older dataset NSL-KDD is used for evaluating the model. The model achieves reasonable detection rates for U2R and R2L attacks but it is still lower comparing the other attack classes of the dataset.
Naseer et al[94]	Comparison of ML- and DL-based NIDS algorithms by implementing on GPU-integrated testbed.	Use of an older dataset NSL-KDD for evaluation.
Xu et al[93]	Use of GRU as the main memory for RNN together with the multilayer perceptron and a softmax classifier to propose a NIDS solution.	The model is evaluated using old datasets KDD Cup'99 and NSL-KDD. Also, lower detection rates for R2L and U2R attack classes are recorded.
Al-Qatf et al[99]	The proposal of a NIDS by an efficient idea of self-taught learning based on Sparse AE and SVM.	The model is tested using an older dataset NSL-KDD. Also, no results are provided for the performance of the model against minority attack classes (e.g. R2L and U2R).
Marir et al[120]	Use of DBN and ensemble SVM to detect abnormal behavior in a distributed way. DBN is used for feature extraction, which is then forwarded to the ensemble SVM and finally, the voting mechanism is used for prediction.	The model is complex and the training time increases slightly for deeper layers.

(Continues)

TABLE 4 (Continued)

Study	Strengths	Weaknesses
Papamartizivanous et al[100]	Propose an autonomous misuse detection system by combining the self-taught learning and MAPE-K frameworks. For learning useful features, a sparse AE is used.	The proposed model is validated using old datasets KDD Cup'99 and NSLKDD. Also, it shows the incapability of detecting R2L and U2R attack classes
Khan et al[103]	Propose an efficient two-stage NIDS model based on deep stacked AE. The result of the first stage along with the obtained probability scores are input the final decision stage as an additional feature to improve for normal and multiclass attack classification. The use of different methodologies (downsampling/SMOTE for different datasets) during the preprocessing of the dataset dramatically improves the detection rate efficiency of minority attack classes.	For evaluation purposes, KDD Cup'99 and UNSW-NB15 datasets are used. The detection accuracy for KDD Cup'99 is 99.996% while for the UNSW-NB15 dataset detection accuracy is 89.134% which suggests that although the model performed well for an older dataset, however for the relatively newer dataset, the detection accuracy is decreased by almost 10%. Further, it is not shown how the model performs in detecting minority attack classes.
Xiao et al[123]	An efficient NIDS is proposed based on the DL algorithm CNN. For dimensionality reduction, feature extraction is performed considering PCA and AE.	The performance of the proposed model is evaluated using an older dataset KDD Cup'99. Moreover, the model achieves lower detection rates for the U2R and R2L classes.
Yao et al[72]	A clustering concept is used along with RF to propose a multilevel intrusion detection model. The model exhibits superior performance in detecting the attacks even with low instances in the dataset.	Use of a very old dataset KDD Cup'99 for model validation purposes.
Vinayakumar et al[116]	A hybrid scalable DNN framework called scale-hybrid-IDS-AlertNet is proposed for effective real-time monitoring of network traffic to detect intrusions. The model is tested using different new and old datasets.	A complex model which gives a lower detection rate performance for minority attack class.
Gao et al[82]	Use of an adaptive ensemble model by using several base classifiers as decision tree, RF, K-Nearest Neighbor, DNN, and choosing the best classifier using an adaptive voting algorithm.	Model is tested using an older dataset NSL-KDD which exhibits unsatisfactory results for weaker attack classes.
Wei et al[121]	Proposed a DL-based model DBN, and is optimized by combining the different optimization algorithms as Particle Swarm, Fish swarm, and Genetic Algorithm.	The proposed model is complex and requires more training time. Furthermore, it is evaluated using an older dataset NSL-KDD.
Zhang et al[124]	A multilayer NIDS model based on CNN and gcForest is proposed. Also, for converting raw data into a two-dimensional greyscale image, a novel P-Zigzag algorithm is proposed. The model is evaluated using by combining new datasets UNSW-NB15 and CIC-IDS2017.	The detection accuracy for the attack classes with less training data is low.
Malaiya et al[104]	Different models for ID detection is proposed using fully connected networks, Variational AE, and Seq2Seq structures, respectively. Both new and older datasets are used for model evaluation purposes.	The Seq2Seq model performs best among other models at the cost of greater training overhead than others. Moreover, it is not shown which model performs best in the detection of the minority attack classes.

(Continues)

TABLE 4 (Continued)

Study	Strengths	Weaknesses
Karatas et al[63]	The analysis of six ML-based NIDSs is presented that addresses dataset imbalance by reducing the imbalance ratio using SMOTE. This eventually improves the detection rate for the minority attack class. Also, an up to date dataset CSECIC-IDS2018 is used for evaluation.	Adaboost algorithm is shown to achieve higher detection accuracy, but it is done at the cost of higher execution time.
Jiang et al[125]	An IDS is proposed using CNN and bi-directional long short-term memory (BiLSTM) in a deep hierarchy. The class imbalance problem is addressed by using SMOTE to increase the minority samples. Evaluation is performed using both older (NSL-KDD) and newer (UNSW-NB15) datasets.	The proposed model is complex and the detection rate of minority data classes improved slightly but still, it is very low comparing other attack classes.
Yang et al[106]	A NIDS is proposed based on the supervised adversarial variational AE with regularization and DNN. Evaluation is performed using both older (NSL-KDD) and newer (UNSW-NB15) datasets.	For the NSL-KDD dataset, the model achieves reasonable detection rates for U2R and R2L attacks but it is still lower comparing the other attack classes of the dataset. For the UNSW-NB15 dataset, the model performance for the detection of minority class attacks is not shown.
Yu et al[126]	An efficient NIDS model is proposed based on the novel DL idea of Few-shot Learning (FSL). DL algorithms as DNN and CNN are adopted as embedding functions in the model for extracting the essential feature and reducing the dimension. The model is evaluated using NSL-KDD and UNSW-NB15 datasets.	For the NSL-KDD dataset, the model achieves reasonable detection rates for U2R and R2L attacks but it is still lower comparing the other attack classes of the dataset. Also, FSL requires labeled data for learning which makes its application limited in scenarios, where the model is trained frequently using the unlabeled network traffic to help it learn more patterns for effective detection.
Andresini et al[107]	Use of AE to proposed a multi-stage model involving the intrusion detection convolution layer and two stacked fully connected layers. Also, the model's performance is tested using both new and older datasets.	The model's effectiveness in detecting the minority attack class is not given. Also, the model is unable to provide details about the structure and characteristics of the attack.

Abbreviations: AE, AutoEncoder; CNN, Convolutional neural network; DL, deep learning; DNN, Deep Neural Network; IDS, intrusion detection system; ML, machine learning; NIDS, network-based intrusion detection system; RF, random forest; SVM, support vector machine.

will discuss the evaluation metrics which were used in the reviewed articles to evaluate the efficiency of the proposed solutions.

5 | EVALUATION METRICS

This section explains the most commonly used evaluation metrics for measuring the performance of ML and DL methods for IDS. All the evaluation metrics are based on the different attributes used in the *Confusion Matrix*, which is a two-dimensional matrix providing information about the Actual and Predicted class[128] and includes;

i. *True Positive* (*TP*): The data instances correctly predicted as an Attack by the classifier.
ii. *False Negative* (*FN*): The data instances wrongly predicted as Normal instances.
iii. *False Positive* (*FP*): The data instances wrongly classified as an Attack.
iv. *True Negative* (*TN*): The instances correctly classified as Normal instances.

The diagonal of confusion matrix denotes the correct predictions while nondiagonal elements are the wrong predictions of a certain classifier. Table 5 depicts these attributes of confusion matrix. Further, the different evaluation metrics used in the recent studies are,

- **Precision:** It is the ratio of correctly predicted Attacks to all the samples predicted as Attacks.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}. \tag{1}$$

- **Recall:** It is a ratio of all samples correctly classified as Attacks to all the samples that are actually Attacks. It is also called a Detection Rate.

$$\text{Recall} = \text{Detection Rate} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \tag{2}$$

- **False alarm rate:** It is also called the false positive rate and is defined as the ratio of wrongly predicted Attack samples to all the samples that are Normal.

$$\text{False Alarm Rate} = \frac{\text{FP}}{\text{FP} + \text{TN}}. \tag{3}$$

- **True negative rate:** It is defined as the ratio of the number of correctly classified Normal samples to all the samples that are Normal.

$$\text{True Negative Rate} = \frac{\text{TN}}{\text{TN} + \text{FP}}. \tag{4}$$

- **Accuracy:** It is the ratio of correctly classified instances to the total number of instances. It is also called as Detection Accuracy and is a useful performance measure only when a dataset is balanced.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}. \tag{5}$$

TABLE 5 Confusion matrix

		Predicted class	
		Attack	**Normal**
Actual Class	**Attack**	True Positive	False Negative
	Normal	False Positive	True Negative

- **F-Measure:** It is defined as the harmonic mean of the Precision and Recall. In other words, it is a statistical technique for examining the accuracy of a system by considering both precision and recall of the system.

$$\text{F Measure} = 2\left(\frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}\right). \tag{6}$$

These evaluation metrics are calculated by testing the proposed methodologies using benchmark datasets. The next section discusses the popular public dataset used for testing NIDS.

6 | BENCHMARK DATASETS

This section provides detail about the popular datasets used by the researcher for testing the performance of their proposed methodology. A detailed summary of the dataset and the attack classes within each is given in Table 6.

- **KDD Cup'99:** It is one of the most popular and widely used dataset for IDS. It contains approximately five and two million records for training and testing respectively. Each record contains 41 different features or attributes and is labeled as either normal or attack. The attacks are classified into four different types as Denial of Service (DoS), Probe, Remote to Local (R2L), and User to Root (U2R).[129]
- **Kyoto 2006+:** This dataset is created from the network traffic records, obtained by deploying honeypots, darknet sensors, email servers, web crawler, and other network security measures by Kyoto University.[134] The most latest dataset includes the traffic record from 2006 to 2015. Each record has 24 statistical features, 14 of which are derived from KDD Cup'99 dataset while the remaining 10 are additional features.[130]
- **NSL-KDD:** This is the revised and refined version of the KDD Cup'99 dataset by removing several of its integral issues. This dataset is also a 41 feature dataset with the attacks divided into four classes as discussed in KDD Cup'99.[131]
- **UNSW-NB15:** This dataset is created by the Australian Center for Cyber Security.[132] It contains approximately two million records with a total of 49 features, that are extracted using Bro-IDS, Argus tools, and some newly developed algorithms. This dataset contains the types of attacks named as, Worms, Shellcode, Reconnaissance, Port Scans, Generic, Backdoor, DoS, Exploits, and Fuzzers.[135]
- **CIC-IDS2017:** This dataset is created by the Canadian Institute of Cyber Security (CIC) in 2017.[133] It contains the normal flows and updated real-world attacks. The network traffic is analyzed by CICFlowMeter using the information based on timestamps, source, and destination IP addresses, protocols, and attacks.[136] Moreover, CICIDS2017 includes common attack scenarios like Brute Force Attack, HeartBleed Attack, Botnet, Denial of Service (DoS) Attack, Distributed DoS (DDoS) Attack, Web Attack, and Infiltration Attack..[137]
- **CSE-CIC-IDS2018:** This dataset is jointly created by Communications Security Establishment (CSE) and CIC in 2018.[63] The user profiles containing the abstract representation of the different events is created. For the generation of the

TABLE 6 Summary of public benchmark datasets

Dataset	Year	Attack types	Attacks
KDD Cup'99[129]	1998	4	DoS, Probe, R2L, U2R
Kyoto 2006+[130]	2006	2	Known Attacks, Unknown Attacks
NSL-KDD[131]	2009	4	DoS, Probe, R2L, U2R
UNSW-NB15[132]	2015	9	Backdoors, DoS, Exploits, Fuzzers, Generic, Port scans, Reconnaissance, Shellcode, worms
CIC-IDS2017[133]	2017	7	Brute Force, HeartBleed, Botnet, DoS, DDoS, Web , Infiltration
CSE-CIC-IDS2018[133]	2018	7	HeartBleed, DoS, Botnet, DDoS, Brute Force, Infiltration, Web.

TABLE 7 Datasets and performance evaluation metrics

Study	Dataset							Evaluation metrics							
	KC	*NK*	*KY*	*UN*	*C7*	*C8*	*OT*	*ACC*	*PRE*	*REC*	*F-M*	*FAR*	*ROC*	*TPR*	*OTH*
Yin et al[91]		✓						✓				✓		✓	
Shen et al[81]	✓	✓	✓					✓		✓		✓			
Shone et al[97]	✓	✓						✓	✓	✓	✓	✓			
Ali et al[79]	✓							✓							
Jia et al[109]	✓	✓						✓	✓	✓	✓	✓			✓
Wang et al[111]		✓						✓	✓	✓	✓	✓	✓		
Yan et al[67]		✓						✓		✓		✓			
Naseer et al[94]		✓						✓	✓	✓			✓		✓
Xu et al[93]	✓	✓						✓		✓	✓	✓			
Al-Qatf et al[99]		✓						✓	✓	✓	✓				
Marir et al[120]		✓		✓	✓				✓	✓	✓		✓		✓
Papamartizivanous et al[100]	✓	✓						✓	✓	✓	✓				
Khan et al[103]	✓			✓				✓	✓	✓	✓	✓			
Xiao et al[123]	✓							✓		✓		✓			
Yao et al[72]	✓							✓	✓	✓	✓				✓
Vinayakumar et al[116]	✓	✓	✓	✓	✓		✓	✓	✓		✓	✓	✓	✓	
Gao et al[82]		✓						✓	✓	✓	✓				
Wei et al[121]		✓						✓		✓		✓			✓
Zhang et al[124]				✓	✓			✓		✓		✓			
Malaiya et al[104]	✓	✓	✓	✓	✓		✓	✓	✓	✓	✓				✓
Karatas et al[63]						✓		✓	✓	✓	✓				
Jiang et al[125]		✓		✓				✓	✓	✓	✓				
Yang et al[106]		✓		✓				✓	✓	✓	✓	✓			
Yu et al[126]		✓		✓				✓	✓	✓	✓	✓			
Andresini et al[107]	✓			✓	✓			✓			✓				
KC, KDD Cup'99; NK, NSL-KDD; KY, Kyoto 2006+,								ACC, Accuracy; PRE, Precision; REC, Recall; F-M, F-Measure;							
UN, UNSW-NB15; C7, CICIDS2017;								FAR, False Alarm Rate; ROC, Receiver Operating Characteristic							
C8, CSE-CIC-IDS2018; OT, Others								Curve, TPR, True Positive Rate; OTH, Others.							

dataset, all these profiles are combined with a unique set of features. It includes seven different attack scenarios: Brute-force, Heartbleed, Botnet, DoS, DDoS, Web attacks, and infiltration of the network from inside.[133]

Table 7 provides an overview and summary of various studies that utilize the datasets discussed in Section 6 and the evaluation metrics (discussed in Section 5) to evaluate the efficiency of their proposed techniques.

7 | OBSERVATIONS, CHALLENGES, AND FUTURE TRENDS

This section first discusses recent trends in IDS based on proposed methodology, performance criteria, and the dataset adopted. It also highlights the possible research gap and challenges and finally presents the future trends for the researchers to come up with an efficient, robust, and accurate IDS.

7.1 | Recent trends and observations

The efficiency of the AI-powered NIDS highly depends upon training using a suitable dataset. For ML models, the algorithm can be trained using a small dataset to achieve better results. But in case of the larger dataset, then ML is not suitable unless the dataset is labeled in nature. Since labeling is an expensive and time consuming process, DL methods are preferred for larger datasets. These methods will learn and extract useful patterns from raw datasets. To make NIDS efficient in detecting zero-day attacks, it needs to be trained regularly with the new data obtained as a result of monitoring the network traffic. The large dataset and deep nature of the DL algorithms will make the learning process more resource hungry in terms of computational resources and time consumption. The more the NIDS model is trained, the more efficiently it will detect intrusions.

Table 4 highlights the strengths and weaknesses of the reviewed articles. It is observed that DL-based NIDS methodologies are preferred nowadays over the ML methodologies due to their efficiency in learning from large datasets in raw form. Since the DL algorithms require extensive computational resources, the advent of GPUs and cloud-based platforms have eased the way for the implementation of DL-based methods. We observed that in most of the proposed solutions, models are tested using older datasets such as KDD Cup'99 and NSL-KDD. We also observed that in some proposed solutions, the performance of the model showing extremely excellent results for older datasets is decreased for newer and recently proposed datasets, for instance.[103] Another major drawback exhibited by most of the methodologies is their inefficiency in detecting the attack having fewer samples for the training dataset. This class imbalance problem affects the detection rate and accuracy for these minority attack classes which needs further attention. Also, we observed that some of the methodologies are quite complex and it ultimately requires more model training time. We observe a trade-off between model complexity and the deep structure of DL methods. The deeper the algorithm is, the more complex the model will be and hence it will consume more time and computing resources. So the intelligent selection of useful features for the model training will ultimately improve this drawback.

Based on the reviewed article, it is observed that during the past three years, the researchers focused on DL tools for designing the IDS system as shown in Figure 7. It is noticed that 60% of the proposed methods are purely based on the DL approaches, 20% solutions use hybrid approach involving the combination of ML- and DL-based algorithms while only 20% proposed solutions are based on ML methods. As discussed earlier, DL models are complex in nature and require extensive computational resources. This is made possible due to the advent of the GPU and has contributed to a vast increase in the use of DL-based algorithms in the design of IDS.

Moreover, Figure 8 shows the number of times ML- or DL-based algorithms are adopted by the researchers in the context of designing an efficient IDS solution. It is observed that the four most frequent algorithms used are AE, DNN, CNN, and RNN, respectively, which are all DL in nature. Then the ML-based approaches like RF and SVM come in the list and are mostly used in the hybrid design to assist and improve DL-based algorithms. Also, ML-based algorithms like DT, KNN, and FLN are less frequently adopted algorithms during the reviewed period.

From the reviewed articles, it is observed that AE and its different variations are one of the most utilized algorithms adopted for proposing NIDS solutions. Mostly, AE is adopted specifically used for feature extractions and reduction, followed by the ML-based classifier for classification purposes. This feature reduction approach reduces the overall

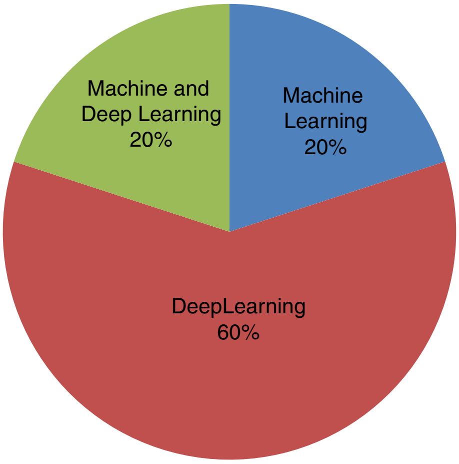


FIGURE 7 Methodology distribution

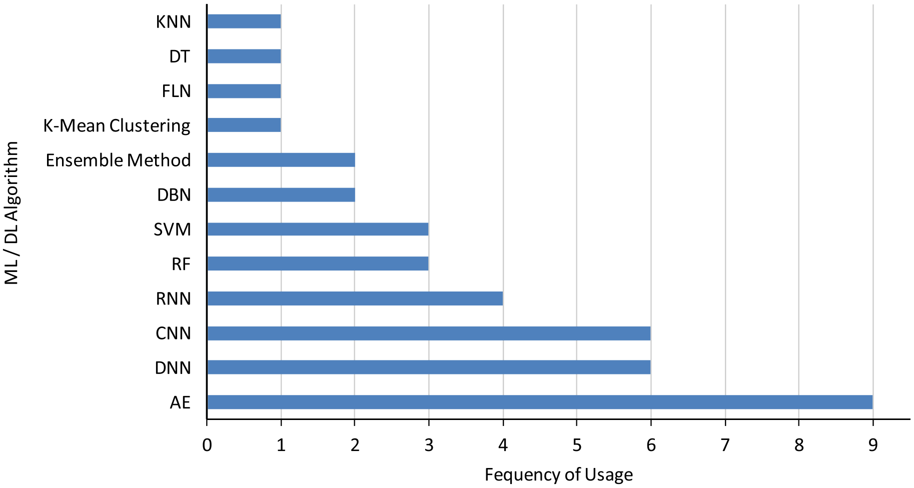


FIGURE 8 Number of time machine learning and deep learning algorithms used

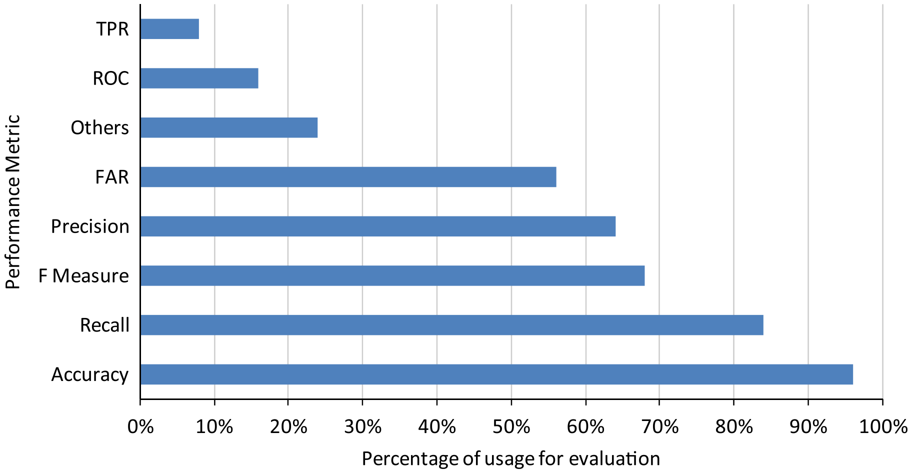


FIGURE 9 Evaluation metrics

complexity of the model and eventually decreases the training time. It is also observed that few proposed methodologies combined two or more DL algorithms in a model. Although these types of methodologies improved the detection accuracy but it is done at the cost of more complexity which demands high computing resources. Similarly, sometimes the data need to be transformed in the format to be used for some specific algorithm. For instance, to use CNN specifically for NIDS, the transformation of data from one-dimensional data vector (feature set) into a two-dimensional matrix is required.

The analysis of the performance metrics used by the researchers for the evaluation of the methodology is shown in Figure 9. It is noted that the most widely used performance metrics are Detection Accuracy and Recall (Detection rate). It is obvious that for efficient network security, the IDS demands a higher Accuracy and Detection rate. So, to investigate the efficiency and effectiveness of the proposed methodology, these two performance metrics should be considered. For a typical IDS designed using ML / DL tools, the Accuracy, Recall, and F-measure should be the compulsory performance metric besides others to show its ability in detecting intrusions.

Benchmark datasets are an important ingredient used to test the performance of the proposed methodology. The analysis for the use of the public datasets is shown in Figure 10. It is illustrated that 60% times NSL-KDD and KDD Cup'99 were used for testing and validating purposes. Both of these are quite old datasets but are still very popular to use among researchers due to the availability of extensive results in the literature. Modern network architecture is quite different from the one 20 years ago. In this era of IoT, security, and privacy of the big data and sensor nodes are a prime concern due to the rapid generation of novel attacks. If a new IDS methodology proposed for the modern networks is tested using an old dataset, there are more chances that the proposed method will not perform well when deployed in the real-world environment. It is quite obvious that a model trained and verified using the latest dataset will perform comparatively better than the model trained and verified using an old dataset in the real world.

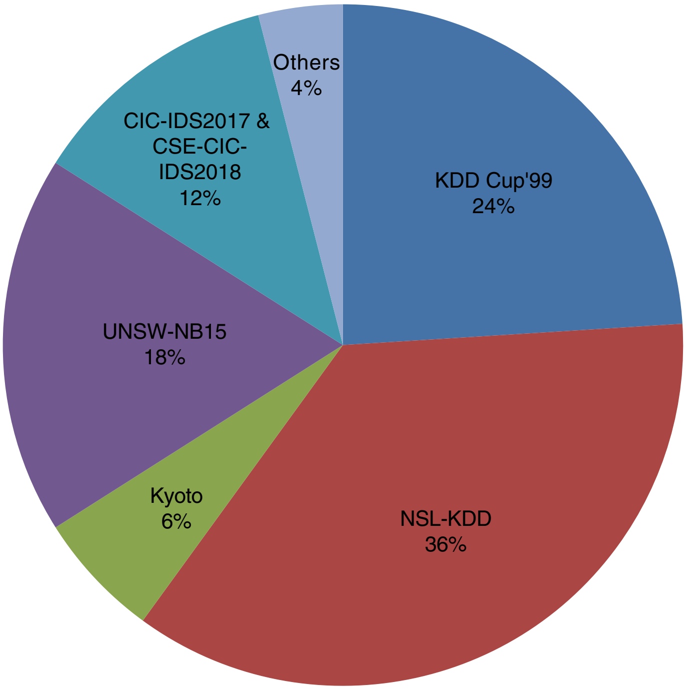


FIGURE 10 Datasets distribution

7.2 | Research challenges

This subsection highlights the research challenges in the field of IDS.

1. **Unavailability of a systematic dataset:** The current study highlighted the unavailability of an up to date dataset that reflects the new attacks for modern networks. Most of the proposed methodologies were not able to detect zero-day attacks because these models were not trained with enough attack types and patterns. To come up with an efficient IDS model, it needs to be tested and verified using the dataset having older and newer attacks. By including the maximum number of attacks definition in a dataset will enable the ML/DL model to learn more patterns and eventually will provide protection against maximum intrusions of different types. But dataset construction is an expensive process that demands a lot of resources and high experts' knowledge. Hence, one of the research challenges for IDS is the systematic construction of an up-to-date dataset with enough instances of almost all the attack types. The dataset should be updated frequently to include the latest intrusion instances and should be made public to help the research community.
2. **Lower detection accuracy due to imbalance dataset:** It is also noted from the current study that most of the proposed IDS methodologies exhibit lower detection accuracy for certain attack types than the overall detection accuracy of the model. This problem is caused by the imbalance nature of the dataset. The detection accuracy for low frequent attacks class is lower than the attacks with more instances. There can be two solutions to this problem. The first is to come up with an up-to-date and balanced dataset. The second solution is to come up with efficient techniques that can increase the number of minority attack instances to balance the dataset. Recently, researchers used certain techniques like SMOTE, RandomOverSampler, and adaptive synthetic sampling approach (ADASYN Algorithm), etc. for reducing the dataset imbalance ratio for improved performance. But there is still room for improvement and demands more research in this direction.
3. **Low performance in real-world environment:** Another research challenge for IDS is their performance in the real-world environment. Since most of the proposed methodologies are tested and verified within a lab using the public datasets. None of the proposed methodologies is tested in a real-world environment. So, it is still not clear how they will perform in real-world scenarios. As stated, most of them still rely on testing using old datasets. So the biggest challenge for the proposed methodology is to be as efficient as demonstrated in the lab tests. The proposed method once tested in the lab should also be tested in a real-time environment to verify its effectiveness for modern networks.
4. **Resources consumed by complex models:** Most IDS methodologies proposed by the researcher are based on very complex models that require a lot of time in processing and computing resources (almost 80% DL-based methods

or DL-ML based methods). This may result in extra overhead for the processing unit and ultimately affects the performance of IDS. The use of a multi-core high-performance GPU can speed up the computation process and reduce time, but it will cost a huge amount of money. Similarly, So to reduce the computational and processing overhead, an efficient feature selection algorithm is needed to intelligently selects the most important features for faster processing. Although many optimization algorithms are being explored by the researchers for feature selection, there is still scope of improvement and research can be carried out in this direction to come up with an efficient feature selection optimization algorithm.

5. **Lightweight IDS for IoT:** An IDS can also be used to provide security to the IoT network and its associated sensor nodes. In an IoT environment, sensor nodes collect a huge amount of critical data that is shared through the internet. Sensor nodes are resource-constrained with limited computational power, storage capacity, and battery life. IDS can either be deployed at the points where network traffic enters into the IoT network from the internet, or it can be deployed in a distributed manner over the sensor nodes. In the first scenario, the NIDS needs to be efficient in detecting malicious attacks and poses the same challenges as discussed earlier. In the second scenario, for the resource-limited sensor nodes, a lightweight IDS model is needed. So the design of a lightweight IDS model which is efficient in terms of computational power, training time, and with higher intrusion detection rate is one of the biggest challenges.

7.3 | Future trends

This subsection highlights the future scope in ML-/ DL-based IDS research.

1. **Efficient NIDS framework:** NIDS is one of the important defense mechanism to a network against intrusions. Recent studies show its limitation in detecting zero-day attacks with a high false alarm. To this end, the performance of IDS can be improved by using an up-to-date, systematic, and balanced dataset. There are very few attempts by the researchers to propose an efficient and complete NIDS framework for a network (more specifically for modern networks like an IoT[138]). Research can be carried in this direction to propose an efficient NIDS framework that can provide complete security against intrusions. The IDS framework should include a mechanism to frequently update the attack definitions in a dataset and keep on training the model with the updated definitions to make the model learn new features. This will eventually improve the IDS model in detecting zero-day attacks and decrease false alarms. The training phase of an ML- or DL-based IDS model normally takes a long time and can be performed offline. The key to attack detection and accuracy of a model lies in the constant process of dataset updating and training for AI-based IDS systems.
2. **Solution to complex models:** According to recent studies, DL-based IDSs have gained enormous popularity due to deep feature learning ability to produce excellent results in detecting malicious attacks. The models which are based on DL algorithms are quite complex and require high resources in terms of computational power, storage capacity, and time. These complex structures put extra challenges for IDS to be implemented in real-time environments. To address these problems, one solution is to use high-performance GPUs for the processing of big datasets quickly and efficiently. But these GPUs are normally expensive. So there is a tradeoff between performance and cost. To reduce cost, cloud-based GPU platforms or services can be explored for model training purposes. Another solution to this problem is to try reducing the complexity of DL algorithms by doing efficient and intelligent feature engineering. By selecting only the key features will result in almost the same detection accuracy as obtained using the complete set of features. This will eventually decrease the complexity of the model and will utilize fewer computing resources in a real-time environment.
3. **Use of DL algorithms:** Recent studies suggested the use of DL-based algorithms for an IDS design. Many DL algorithms (as discussed in this Section 4) are explored and used efficiently in proposing effective solutions. But there are still some DL algorithms that require more attention like deep reinforcement learning, Hidden Markov Models, etc. for proposing IDS solution for IoT network. The research to use DL for IDS is still in the early stages. Researchers can also explore the hybrid idea of using DL for feature extraction and ML for classification. This will reduce the complexity of the proposed model.
4. **Efficient NIDS for cyber-physical systems:** Recently, a massive interest is shown in Cyber-Physical Systems such as Supervisory Control and Data Acquisition (SCADA) networks and Unmanned Aerial Vehicles (UAV)-enabled networks. SCADA networks have various applications, such as smart grids[139] and manufacturing industries, etc. However, SCADA networks are becoming more and more complex as state-of-the-art information and communication technologies (ICT) are embedded within the network, thus providing an opportunity for the attackers to be part of the

network. NIDS plays an important role in such networks, where these can detect intruders by inspecting the network traffic. Further, the use of ML and DL will increase the efficiency of NIDS, as it can provide an extra dimension to detect cyber attacks within the SCADA networks. However, the research in this domain is still in early-stage and more investigation is required to identify and design efficient DL-based NIDS for SCADA networks.

Similarly, UAV-enabled networks also offer a wide range of applications, including traffic monitoring, asset inspection, and securing critical infrastructure, etc.[140] Due to the nature of communication involved over the wireless channel, these networks are accessible not only to legitimate users but also susceptible to the network intruders, which can not only monitor the communication but can also launch various attacks. Therefore, an efficient and intelligent NIDS is required which can detect the intruders within UAV-enabled networks. Furthermore, the use of AI within the NIDS for UAV-enabled networks can be an interesting research direction, which requires more exploration and investigation.[141,142]

8 | CONCLUSIONS

This paper provides an extensive review of the network intrusion detection mechanisms based on the ML and DL methods to provide the new researchers with the updated knowledge, recent trends, and progress of the field. A systematic approach is adopted for the selection of the relevant articles in the field of AI-based NIDS. Firstly, the concept of IDS and its different classification schemes is elaborated extensively based on the reviewed articles. Then the methodology of each article is discussed and the strengths and weaknesses of each are highlighted in terms of the intrusion detection capability and complexity of the model. Based on this study, the recent trend reveals the usage of DL-based methodologies to improve the performance and effectiveness of NIDS in terms of detection accuracy and reduction in FAR. About 80% of the proposed solutions were based on the DL approaches with AE and DNN are the most frequently used algorithms. Although DL schemes have much superior performance than the ML-based methods in terms of their ability to learn features by itself and stronger model fitting abilities. But these schemes are quite complex and require extensive computing resources in terms of processing power and storage capabilities. These challenges need to be addressed to fulfill real-time requirements for NIDS and hence improves NIDS performance.

The study also shows that 60% of the proposed methodologies were tested using KDD Cup'99 and NSL-KDD datasets mainly because of the availability of extensive results using these datasets. But these datasets are quite old to address modern network attacks, and hence limits the performance of the proposed methodologies in real-time environments. For AI-based NIDS methods, the model should be tested with the latest updated dataset like CSE-CIC-IDS2018 for better performance in terms of detection accuracy for intrusions. This article also highlights the research gaps in improving the model performance for low-frequency attacks in a real-world environment and to find efficient solutions to reduce complexity for the proposed models. Proposing an efficient NIDS framework using less complex DL algorithms and have an effective detection mechanism is a potential future scope of research in this area. For future research, we will use this knowledge to design a novel, lightweight, and efficient DL-based NIDS which will effectively detect the intruders within the network.

ACKNOWLEDGMENTS

This research is fully funded by Research, Innovation and Enterprise Centre (RIEC), Universiti Malaysia Sarawak under the grant number F08/PGRG/1908/2019.

DATA AVAILABILITY

Analyzed data for this survey article is available upon request from the corresponding author.

ORCID

Zeeshan Ahmad https://orcid.org/0000-0002-8530-864X

REFERENCES

1. Tarter A. Importance of cyber security. *Community Policing-A European Perspective: Strategies, Best Practices and Guidelines*. New York, NY: Springer; 2017:213-230.
2. Li J, Qu Y, Chao F, Shum HP, Ho ES, Yang L. Machine learning algorithms for network intrusion detection. *AI in Cybersecurity*. New York, NY: Springer; 2019:151-179.

3. Lunt TF. A survey of intrusion detection techniques. *Comput Sec*. 1993;12(4):405-418. https://doi.org/10.1016/0167-4048(93)90029-5.
4. Anderson JP. *Computer Security Threat Monitoring and Surveillance*. Fort Washington, PA: James P Anderson Co; 1980.
5. Debar H, Dacier M, Wespi A. Towards a taxonomy of intrusion-detection systems. *Comput Netw*. 1999;31(8):805-822. https://doi.org/10.1016/S1389-1286(98)00017-6.
6. Hoque MS, Mukit M, Bikas M, Naser A, An implementation of intrusion detection system using genetic algorithm; 2012. arXiv preprint arXiv:1204.1336.
7. Prasad R, Rohokale V. Artificial intelligence and machine learning in cyber security. *Cyber Security: The Lifeline of Information and Communication Technology*. New York, NY: Springer; 2020:231-247.
8. Lew J, Shah DA, Pati S, et al. Analyzing machine learning workloads using a detailed GPU simulator. Paper presented at: Proceedings of the IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS). Madison, WI, USA: IEEE; 2019:151-152.
9. Najafabadi MM, Villanustre F, Khoshgoftaar TM, Seliya N, Wald R, Muharemagic E. Deep learning applications and challenges in big data analytics. *J Big Data*. 2015;2(1):1. https://doi.org/10.1186/s40537-014-0007-7.
10. Dong B, Wang X. Comparison deep learning method to traditional methods using for network intrusion detection. Paper presented at: Proceedings of the 8th IEEE International Conference on Communication Software and Networks (ICCSN). Beijing, China: IEEE; 2016:581-585.
11. Vasilomanolakis E, Karuppayah S, Mühlhäuser M, Fischer M. Taxonomy and survey of collaborative intrusion detection. *ACM Comput Surv*. 2015;47(4):1-33. https://doi.org/10.1145/2716260.
12. Buczak AL, Guven E. A survey of data mining and machine learning methods for cyber security intrusion detection. *IEEE Commun Surv Tutor*. 2015;18(2):1153-1176. https://doi.org/10.1109/COMST.2015.2494502.
13. Thomas R, Pavithran D. A survey of intrusion detection models based on NSL-KDD data set. Paper presented at: Proceedings of the 5th HCT Information Technology Trends (ITT). Dubai, United Arab Emirates: IEEE; 2018:286-291.
14. Liu H, Lang B. Machine learning and deep learning methods for intrusion detection systems: a survey. *Appl Sci*. 2019;9(20):4396. https://doi.org/10.3390/app9204396.
15. Khraisat A, Gondal I, Vamplew P, Kamruzzaman J. Survey of intrusion detection systems: techniques, datasets and challenges. *Cybersecurity*. 2019;2(1):20. https://doi.org/10.1186/s42400-019-0038-7.
16. DKA C, Papa JP, Lisboa CO, Munoz R, DVHC A. Internet of Things: a survey on machine learning-based intrusion detection approaches. *Comput Netw*. 2019;151:147-157. https://doi.org/10.1016/j.comnet.2019.01.023.
17. Keele S, Guidelines for Performing Systematic Literature Reviews in Software Engineering. Technical Report, Technical Report, Ver. 2.3, EBSE Technical Report. vol. 5, EBSE; 2007.
18. Scopus Preview Welcome to Scopus Preview; 2020. https://www.scopus.com/. Accessed June 25, 2020.
19. Mukkamala S, Janoski G, Sung A. Intrusion detection using neural networks and support vector machines. Paper presented at: Proceedings of the 2002 International Joint Conference on Neural Networks. IJCNN'02 (Cat. No. 02CH37290). Honolulu, HI, USA: IEEE; vol. 2, 2002:1702-1707.
20. Garcia-Teodoro P, Diaz-Verdejo J, Maciá-Fernández G, Vázquez E. Anomaly-based network intrusion detection: techniques systems and challenges. *Comput Secur*. 2009;28(1-2):18-28. https://doi.org/10.1016/j.cose.2008.08.003.
21. Denning DE. An intrusion-detection model. *IEEE Trans Softw Eng*. 1987;2:222-232. https://doi.org/10.1109/TSE.1987.232894.
22. Verwoerd T, Hunt R. Intrusion detection techniques and approaches. *Comput Commun*. 2002;25(15):1356-1365. https://doi.org/10.1016/S0140-3664(02)00037-3.
23. Kabiri P, Ghorbani AA. Research on intrusion detection and response: a survey. *Int J Netw Secur*. 2005;1(2):84-102. https://doi.org/10.6633/IJNS.200509.1(2).05.
24. Zhang Y, Lee W, Huang YA. Intrusion detection techniques for mobile wireless networks. *Wirel Netw*. 2003;9(5):545-556. https://doi.org/10.1023/A:1024600519144.
25. Axelsson S. *Intrusion Detection Systems: A Survey and Taxonomy. Technical Report 99-15*. Department of Computer Engineering, Chalmers University; 2000.
26. Ahmim A, Derdour M, Ferrag MA. An intrusion detection system based on combining probability predictions of a tree of classifiers. *Int J Commun Syst*. 2018;31(9):e3547. https://doi.org/10.1002/dac.3547.
27. Uddin M, Rahman AA, Uddin N, Memon J, Alsaqour RA, Kazi S. Signature-based multi-layer distributed intrusion detection system using mobile agents. *Int J Netw Secur*. 2013;15(2):97-105. https://doi.org/10.6633/IJNS.201303.15(2).03.
28. Neri F. Comparing local search with respect to genetic evolution to detect intrusions in computer networks. Paper presented at: Proceedings of the Proceedings of the 2000 Congress on Evolutionary Computation. CEC00 (Cat. No. 00TH8512). La Jolla, CA, USA: IEEE; vol. 1, 2000:238-243.
29. Ma W. Analysis of anomaly detection method for Internet of things based on deep learning. *Trans Emerg Telecommun Technol*. 2020;e3893. https://doi.org/10.1002/ett.3893.
30. Zhang Z, Shen H, Sang Y. An observation-centric analysis on the modeling of anomaly-based intrusion detection. *Int J Netw Secur*. 2007;4(3):292-305. https://doi.org/10.6633/IJNS.200705.4(3).08.
31. Guo C, Ping Y, Liu N, Luo SS. A two-level hybrid approach for intrusion detection. *Neurocomputing*. 2016;214:391-400. https://doi.org/10.1016/j.neucom.2016.06.021.
32. Chandola V, Banerjee A, Kumar V. Anomaly detection: a survey. *ACM Comput Surv*. 2009;41(3):1-58. https://doi.org/10.1145/1541880.1541882.

33. Mehmood Y, Ahmad F, Yaqoob I, Adnane A, Imran M, Guizani S. Internet-of-Things-based smart cities: recent advances and challenges. *IEEE Commun Mag*. 2017;55(9):16-24. https://doi.org/10.1109/MCOM.2017.1600514.
34. Ahmad F, Ahmad Z, Kerrache CA, Kurugollu F, Adnane A, Barka E. Blockchain in Internet-of-Things: architecture, applications and research directions. Paper presented at: Proceedings of the IEEE International Conference on Computer and Information Sciences (ICCIS). Sakaka, Saudi Arabia: IEEE; 2019:1-6.
35. Meng W. Intrusion detection in the era of IoT: building trust via traffic filtering and sampling. *Computer*. 2018;51(7):36-43. https://doi.org/10.1109/MC.2018.3011034.
36. Shah SA, Seker DZ, Hameed S, Draheim D. The rising role of big data analytics and IoT in disaster management: recent advances taxonomy and prospects. *IEEE Access*. 2019;7:54595-54614. https://doi.org/10.1109/ACCESS.2019.2913340.
37. Lazarescu MT. Wireless sensor networks for the Internet of Things: barriers and synergies. *Components and Services for IoT Platforms*. New York, NY: Springer; 2017:155-186.
38. Haseeb K, Almogren A, Islam N, Ud Din I, Jan Z. An energy-efficient and secure routing protocol for intrusion avoidance in IoT-based WSN. *Energies*. 2019;12(21):4174. https://doi.org/10.3390/en12214174.
39. Roman R, Zhou J, Lopez J. Applying intrusion detection systems to wireless sensor networks. Paper presented at: Proceedings of the IEEE Consumer Communications & Networking Conference (CCNC 2006). Las Vegas (USA); 2006.
40. Hortelano J, Ruiz JC, Manzoni P. Evaluating the usefulness of watchdogs for intrusion detection in VANETs. Paper presented at: Proceedings of the IEEE International Conference on Communications Workshops. Capetown, South Africa: IEEE; 2010:1-5.
41. Krzysztoń M, Marks M. Simulation of watchdog placement for cooperative anomaly detection in bluetooth mesh intrusion detection system. *Simul Model Pract Theory*. 2020;101:102041. https://doi.org/10.1016/j.simpat.2019.102041.
42. Chen H, Wu H, Hu J, Gao C. Event-based trust framework model in wireless sensor networks. Paper presented at: Proceedings of the International Conference on Networking, Architecture, and Storage. Chongqing, China: IEEE; 2008:359-364.
43. Meng Y, Li W. Evaluation of detecting malicious nodes using Bayesian model in wireless intrusion detection. Paper presented at: Proceedings of the International Conference on Network and System Security; 2013:40-53; Springer, New York, NY.
44. Shen S, Yue G, Cao Q, Yu F. A survey of game theory in wireless sensor networks security. *J Netw*. 2011;6(3):521. https://doi.org/10.4304/jnw.6.3.521-532.
45. Abdalzaher MS, Muta O. A game-theoretic approach for enhancing security and data trustworthiness in IoT applications. *IEEE IoT J*. 2020. https://doi.org/10.1109/JIOT.2020.2996671.
46. Khan ZA, Herrmann P. A trust based distributed intrusion detection mechanism for internet of things. Paper presented at: Proceedings of the IEEE 31st International Conference on Advanced Information Networking and Applications (AINA). Taipei, Taiwan: IEEE; 2017:1169-1176.
47. Ahmad F, Kurugollu F, Adnane A, Hussain R, Hussain F. MARINE: man-in-the-middle attack resistant trust model in connected vehicles. *IEEE IoT J*. 2020;7(4):3310-3322. https://doi.org/10.1109/JIOT.2020.2967568.
48. Abdalzaher MS, Muta O. Employing game theory and TDMA protocol to enhance security and manage power consumption in wsns-based cognitive radio. *IEEE Access*. 2019;7:132923-132936. https://doi.org/10.1109/ACCESS.2019.2940699.
49. Abdalzaher MS, Seddik K, Muta O. An effective stackelberg game for high-assurance of data trustworthiness in wsns. Paper presented at: Proceedings of the IEEE Symposium on Computers and Communications (ISCC). Heraklion, Greece: IEEE; 2017:1257-1262.
50. Abdalzaher MS, Seddik K, Muta O. Using repeated game for maximizing high priority data trustworthiness in wireless sensor networks. Paper presented at: Proceedings of the IEEE Symposium on Computers and Communications (ISCC). Heraklion, Greece: IEEE; 2017:552-557.
51. Berry MW, Mohamed A, Yap BW. *Supervised and Unsupervised Learning for Data Science*. New York, NY: Springer; 2019.
52. Zanero S, Serazzi G. Unsupervised learning algorithms for intrusion detection. Paper presented at: Proceedings of the IEEE Network Operations and Management Symposium. Salvador, Bahia, Brazil: IEEE; 2008:1043-1048.
53. Imamverdiyev Y, Abdullayeva F. Deep learning method for denial of service attack detection based on restricted Boltzmann machine. *Big Data*. 2018;6(2):159-169. https://doi.org/10.1089/big.2018.0023.
54. Alsughayyir B, Qamar AM, Khan R. Developing a network attack detection system using deep learning. Paper presented at: Proceedings of the International Conference on Computer and Information Sciences (ICCIS). Sakaka, Saudi Arabia: IEEE; 2019:1-5.
55. Xin Y, Kong L, Liu Z, et al. Machine learning and deep learning methods for cybersecurity. *IEEE Access*. 2018;6:35365-35381. https://doi.org/10.1109/ACCESS.2018.2836950.
56. Chary S, Rama B. A survey on comparative analysis of decision tree algorithms in data mining, International Journal of Advanced Scientific Technologies, Engineering and Management Sciences; vol. 3, 2017:91-95.
57. Sahani R, Rout C, Badajena JC, Jena AK, Das H. Classification of intrusion detection using data mining techniques. *Progress in Computing, Analytics and Networking*. New York, NY: Springer; 2018:753-764.
58. Rai K, Devi MS, Guleria A. Decision tree based algorithm for intrusion detection. *Int J Adv Netw Appl*. 2016;7(4):2828.
59. Farnaaz N, Jabbar M. Random forest modeling for network intrusion detection system. *Proc Comput Sci*. 2016;89(1):213-217. https://doi.org/10.1016/j.procs.2016.06.047.
60. Dhaliwal SS, Nahid AA, Abbas R. Effective intrusion detection system using XGBoost. *Information*. 2018;9(7):149. https://doi.org/10.3390/info9070149.
61. Ma Z, Kaban A. K-Nearest-Neighbours with a novel similarity measure for intrusion detection. Paper presented at: Proceedings of the 13th UK Workshop on Computational Intelligence (UKCI). Guildford, UK: IEEE; 2013:266-271.

62. Zhang Y, Cao G, Wang B, Li X. A novel ensemble method for k-nearest neighbor. *Pattern Recogn*. 2019;85:13-25. https://doi.org/10.1016/j.patcog.2018.08.003.
63. Karatas G, Demir O, Sahingoz OK. Increasing the performance of machine learning-based IDSs on an imbalanced and up-to-date dataset. *IEEE Access*. 2020;8:32150-32162. https://doi.org/10.1109/ACCESS.2020.2973219.
64. Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP. SMOTE synthetic minority over-sampling technique. *J Artif Intell Res*. 2002;16:321-357. https://doi.org/10.1613/jair.953.
65. Chen WH, Hsu SH, Shen HP. Application of SVM and ANN for intrusion detection. *Comput Oper Res*. 2005;32(10):2617-2634. https://doi.org/10.1016/j.cor.2004.03.019.
66. Roopa Devi E, Suganthe R. Enhanced transductive support vector machine classification with grey wolf optimizer cuckoo search optimization for intrusion detection system. *Concurr Comput Pract Exp*. 2020;32(4):e4999. https://doi.org/10.1002/cpe.4999.
67. Yan B, Han G. Effective feature extraction via stacked sparse autoencoder to improve intrusion detection system. *IEEE Access*. 2018;6:41238-41248. https://doi.org/10.1109/ACCESS.2018.2858277.
68. Ghanem K, Aparicio-Navarro FJ, Kyriakopoulos KG, Lambotharan S, Chambers JA. Support vector machine for network intrusion and cyber-attack detection. Paper presented at: Proceedings of the Sensor Signal Processing for Defence Conference (SSPD). London, UK: IEEE; 2017:1-5. doi:https://doi.org/10.1109/SSPD.2017.8233268..
69. Kumari R, Singh M, Jha R, Singh N. Anomaly detection in network traffic using K-mean clustering. Paper presented at: Proceedings of the 3rd International Conference on Recent Advances in Information Technology (RAIT). Dhanbad, India; 2016:387-393.
70. Li Z, Li Y, Xu L. Anomaly intrusion detection method based on k-means clustering algorithm with particle swarm optimization. Paper presented at: Proceedings of the International Conference of Information Technology, Computer Engineering and Management Sciences. Nanjing, Jiangsu, China: IEEE; vol. 2; 2011:157-161.
71. Munther A, Razif R, AbuAlhaj M, Anbar M, Nizam S. A preliminary performance evaluation of K-means, KNN and EM unsupervised machine learning methods for network flow classification. *Int J Electr Comput Eng*. 2016;6(2):778-784. https://doi.org/10.11591/ijece.v6i2.8909.
72. Yao H, Fu D, Zhang P, Li M, Liu Y. MSML: a novel multilevel semi-supervised machine learning framework for intrusion detection system. *IEEE IoT J*. 2018;6(2):1949-1959. https://doi.org/10.1109/JIOT.2018.2873125.
73. Saritas MM, Yasar A. Performance analysis of ANN and Naive Bayes classification algorithm for data classification. *Int J Intell Syst Appl Eng*. 2019;7(2):88-91. https://doi.org/10.18201//ijisae.2019252786.
74. Anderson JA. *An Introduction to Neural Networks*. Cambridge, MA: MIT Press; 1995.
75. Bangyal WH, Ahmad J, Rauf HT, Shakir R. Evolving artificial neural networks using opposition based particle swarm optimization neural network for data classification. Paper presented at: Proceedings of the International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT). Sakhier, Bahrain; 2018:1-6.
76. Huang GB, Zhu QY, Siew CK. Extreme learning machine: theory and applications. *Neurocomputing*. 2006;70(1-3):489-501. https://doi.org/10.1016/j.neucom.2005.12.126.
77. Li G, Niu P. An enhanced extreme learning machine based on ridge regression for regression. *Neural Comput Appl*. 2013;22(3-4):803-810. https://doi.org/10.1007/s00521-011-0771-7.
78. Li G, Niu P, Duan X, Zhang X. Fast learning network: a novel artificial neural network with a fast learning speed. *Neural Comput Appl*. 2014;24(7-8):1683-1695. https://doi.org/10.1007/s00521-013-1398-7.
79. Ali MH, Al Mohammed BAD, Ismail A, Zolkipli MF. A new intrusion detection system based on fast learning network and particle swarm optimization. *IEEE Access*. 2018;6:20255-20261. https://doi.org/10.1109/ACCESS.2018.2820092.
80. Bai Q. Analysis of particle swarm optimization algorithm. *Comput Inf Sci*. 2010;3(1):180.
81. Shen Y, Zheng K, Wu C, Zhang M, Niu X, Yang Y. An ensemble method based on selection using bat algorithm for intrusion detection. *Comput J*. 2018;61(4):526-538. https://doi.org/10.1093/comjnl/bxx101.
82. Gao X, Shan C, Hu C, Niu Z, Liu Z. An adaptive ensemble machine learning model for intrusion detection. *IEEE Access*. 2019;7:82512-82521. https://doi.org/10.1109/ACCESS.2019.2923640.
83. Graves A, Mohamed A, Hinton G. Speech recognition with deep recurrent neural networks. Paper presented at: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing. Vancouver, BC, Canada: IEEE; 2013:6645-6649.
84. Singh D, Merdivan E, Psychoula I, et al. Human activity recognition using recurrent neural networks. Paper presented at: Proceedings of the International Cross-Domain Conference for Machine Learning and Knowledge Extraction; 2017:267-274; Springer, New York, NY.
85. Nishide S, Okuno HG, Ogata T, Tani J. Handwriting prediction based character recognition using recurrent neural network. Paper presented at: Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics. Anchorage, AK, USA: IEEE; 2011:2549-2554.
86. Mesnil G, Dauphin Y, Yao K, et al. Using recurrent neural networks for slot filling in spoken language understanding. *IEEE/ACM Trans Audio Speech Lang Process*. 2014;23(3):530-539. https://doi.org/10.1109/TASLP.2014.2383614.
87. Liu X, Gherbi A, Li W, Cheriet M. Multi features and multi-time steps LSTM based methodology for bike sharing availability prediction. *Proc Comput Sci*. 2019;155:394-401. https://doi.org/10.1016/j.procs.2019.08.055.
88. Hochreiter S, Schmidhuber J. Long short-term memory. *Neural Comput*. 1997;9(8):1735-1780. https://doi.org/10.1162/neco.1997.9.8.1735.
89. Chung J, Gulcehre C, Cho K, Bengio Y. Empirical evaluation of gated recurrent neural networks on sequence modeling; 2014. arXiv preprint arXiv:1412.3555.

90. Mittal M, Iwendi C, Khan S, Rehman JA. Analysis of security and energy efficiency for shortest route discovery in low-energy adaptive clustering hierarchy protocol using Levenberg-Marquardt neural network and gated recurrent unit for intrusion detection system. *Trans Emerg Telecommun Technol*. 2020;e3997. https://doi.org/10.1002/ett.3997.
91. Yin C, Zhu Y, Fei J, He X. A deep learning approach for intrusion detection using recurrent neural networks. *IEEE Access*. 2017;5:21954-21961. https://doi.org/10.1109/ACCESS.2017.2762418.
92. Sheikhan M, Jadidi Z, Farrokhi A. Intrusion detection using reduced-size RNN based on feature grouping. *Neural Comput Appl*. 2012;21(6):1185-1190. https://doi.org/10.1007/s00521-010-0487-0.
93. Xu C, Shen J, Du X, Zhang F. An intrusion detection system using a deep neural network with gated recurrent units. *IEEE Access*. 2018;6:48697-48707. https://doi.org/10.1109/ACCESS.2018.2867564.
94. Naseer S, Saleem Y, Khalid S, et al. Enhanced network anomaly detection based on deep neural networks. *IEEE Access*. 2018;6:48231-48246. https://doi.org/10.1109/ACCESS.2018.2863036.
95. Farahnakian F, Heikkonen J. A deep auto-encoder based approach for intrusion detection system. Paper presented at: Proceedings of the 20th International Conference on Advanced Communication Technology (ICACT). Chuncheon-si Gangwon-do, Korea (South): IEEE; 2018:178-183.
96. Goodfellow I, Bengio Y, Courville A. Deep Learning. MIT Press. 2016. http://www.deeplearningbook.org.
97. Shone N, Ngoc TN, Phai VD, Shi Q. A deep learning approach to network intrusion detection. *IEEE Trans Emerg Top Comput Intell*. 2018;2(1):41-50. https://doi.org/10.1109/TETCI.2017.2772792.
98. Alrawashdeh K, Purdy C. Toward an online anomaly intrusion detection system based on deep learning. Paper presented at: Proceedings of the 15th IEEE International Conference on Machine Learning and Applications (ICMLA). Anaheim, CA, USA: IEEE; 2016:195-200.
99. Al-Qatf M, Lasheng Y, Al-Habib M, Al-Sabahi K. Deep learning approach combining sparse autoencoder with SVM for network intrusion detection. *IEEE Access*. 2018;6:52843-52856. https://doi.org/10.1109/ACCESS.2018.2869577.
100. Papamartzivanos D, Mármol FG, Kambourakis G. Introducing deep learning self-adaptive misuse network intrusion detection systems. *IEEE Access*. 2019;7:13546-13560. https://doi.org/10.1109/ACCESS.2019.2893871.
101. Raina R, Battle A, Lee H, Packer B, Ng AY. Self-taught learning: transfer learning from unlabeled data. Paper presented at: Proceedings of the 24th International Conference on Machine Learning. Corvalis Oregon USA; 2007:759-766.
102. Kephart JO, Chess DM. The vision of autonomic computing. *Computer*. 2003;36(1):41-50. https://doi.org/10.1109/MC.2003.1160055.
103. Khan FA, Gumaei A, Derhab A, Hussain A. A novel two-stage deep learning model for efficient network intrusion detection. *IEEE Access*. 2019;7:30373-30385. https://doi.org/10.1109/ACCESS.2019.2899721.
104. Malaiya RK, Kwon D, Suh SC, Kim H, Kim I, Kim J. An empirical evaluation of deep learning for network anomaly detection. *IEEE Access*. 2019;7:140806-140817. https://doi.org/10.1109/ACCESS.2019.2943249.
105. Fontugne R, Borgnat P, Abry P, Fukuda K. Mawilab: combining diverse anomaly detectors for automated anomaly labeling and performance benchmarking. Paper presented at: Proceedings of the 6th International Conference on Emerging Networking Experiments And Technologies (Co-Next); 2010:1-12; ACM, New York, NY.
106. Yang Y, Zheng K, Wu B, Yang Y, Wang X. Network intrusion detection based on supervised adversarial variational auto-encoder with regularization. *IEEE Access*. 2020;8:42169-42184. https://doi.org/10.1109/ACCESS.2020.2977007.
107. Andresini G, Appice A, Di Mauro N, Loglisci C, Malerba D. Multi-channel deep feature learning for intrusion detection. *IEEE Access*. 2020;8:53346-53359. https://doi.org/10.1109/ACCESS.2020.2980937.
108. Gu S, Rigazio L. Towards deep neural network architectures robust to adversarial examples; 2014. arXiv preprint arXiv:1412.5068.
109. Jia Y, Wang M, Wang Y. Network intrusion detection algorithm based on deep neural network. *IET Inf Secur*. 2018;13(1):48-53. https://doi.org/10.1049/iet-ifs.2018.5258.
110. Dahl GE, Sainath TN, Hinton GE. Improving deep neural networks for LVCSR using rectified linear units and dropout. Paper presented at: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing. Vancouver, BC, Canada: IEEE; 2013:8609-8613.
111. Wang Z. Deep learning-based intrusion detection with adversaries. *IEEE Access*. 2018;6:38367-38384. https://doi.org/10.1109/ACCESS.2018.2854599.
112. Goodfellow IJ, Shlens J, Szegedy C. Explaining and harnessing adversarial examples; 2014. arXiv preprint arXiv:1412.6572.
113. Papernot N, McDaniel P, Jha S, Fredrikson M, Celik ZB, Swami A. The limitations of deep learning in adversarial settings. Paper presented at: Proceedings of the IEEE European Symposium on Security and Privacy (EuroS&P). Saarbrucken, Germany: IEEE; 2016:372-387.
114. Moosavi-Dezfooli SM, Fawzi A, Frossard P. Deepfool: a simple and accurate method to fool deep neural networks. Paper presented at: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Las Vegas, NV, USA; 2016:2574-2582.
115. Carlini N, Wagner D. Towards evaluating the robustness of neural networks. Paper presented at: Proceedings of the IEEE Symposium on Security and Privacy (sp). San Jose, CA, USA: IEEE; 2017:39-57.
116. Vinayakumar R, Alazab M, Soman K, Poornachandran P, Al-Nemrat A, Venkatraman S. Deep learning approach for intelligent intrusion detection system. *IEEE Access*. 2019;7:41525-41550. https://doi.org/10.1109/ACCESS.2019.2895334.
117. Vinayakumar R, Poornachandran P, Soman K. Scalable framework for cyber threat situational awareness based on domain name systems data analysis. *Big Data in Engineering Applications*. New York, NY: Springer; 2018:113-142.
118. Hinton GE. A practical guide to training restricted Boltzmann machines. *Neural Networks: Tricks of the Trade*. New York, NY: Springer; 2012:599-619.
119. Hinton GE, Osindero S, Teh YW. A fast learning algorithm for deep belief nets. *Neural Comput*. 2006;18(7):1527-1554. https://doi.org/10.1162/neco.2006.18.7.1527.

120. Marir N, Wang H, Feng G, Li B, Jia M. Distributed abnormal behavior detection approach based on deep belief network and ensemble svm using spark. *IEEE Access*. 2018;6:59657-59671. https://doi.org/10.1109/ACCESS.2018.2875045.
121. Wei P, Li Y, Zhang Z, Hu T, Li Z, Liu D. An optimization method for intrusion detection classification model based on deep belief network. *IEEE Access*. 2019;7:87593-87605. https://doi.org/10.1109/ACCESS.2019.2925828.
122. Lawrence S, Giles CL, Tsoi AC, Back AD. Face recognition: a convolutional neural-network approach. *IEEE Trans Neural Netw*. 1997;8(1):98-113. https://doi.org/10.1109/72.554195.
123. Xiao Y, Xing C, Zhang T, Zhao Z. An intrusion detection model based on feature reduction and convolutional neural networks. *IEEE Access*. 2019;7:42210-42219. https://doi.org/10.1109/ACCESS.2019.2904620.
124. Zhang X, Chen J, Zhou Y, Han L, Lin J. A multiple-layer representation learning model for network-based attack detection. *IEEE Access*. 2019;7:91992-92008. https://doi.org/10.1109/ACCESS.2019.2927465.
125. Jiang K, Wang W, Wang A, Network Intrusion WH. Detection combined hybrid sampling with deep hierarchical network. *IEEE Access*. 2020;8:32464-32476. https://doi.org/10.1109/ACCESS.2020.2973730.
126. Yu Y, Bian N. An intrusion detection method using few-shot learning. *IEEE Access*. 2020;8:49730-49740. https://doi.org/10.1109/ACCESS.2020.2980136.
127. Wang Y, Yao Q, Kwok J, Ni LM. Generalizing from a few examples: a survey on few-shot learning; 2019. arXiv: 1904.05046.
128. Deng X, Liu Q, Deng Y, Mahadevan S. An improved method to construct basic probability assignment based on the confusion matrix for classification problem. *Inf Sci*. 2016;340:250-261. https://doi.org/10.1016/j.ins.2016.01.033.
129. Bay S. *The UCI KDD Archive [http://kdd. ics. uci. edu]*. Irvine, CA: University of California, Department of Computer Science; 1999.
130. Song J, Takakura H, Okabe Y, Eto M, Inoue D, Nakao K. Statistical analysis of honeypot data and building of Kyoto 2006+ dataset for NIDS evaluation. Paper presented at: Proceedings of the 1st Workshop on Building Analysis Datasets and Gathering Experience Returns for Security. Salzburg Austria; 2011:29-36.
131. Tavallaee M, Bagheri E, Lu W, Ghorbani AA. A detailed analysis of the KDD CUP 99 data set. Paper presented at: Proceedings of the IEEE Symposium on Computational Intelligence for Security and Defense Applications. Ottawa, ON, Canada: IEEE; 2009:1-6.
132. Moustafa N, Slay J. UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). Paper presented at: Proceedings of the Military Communications and Information Systems Conference (MilCIS). Canberra, ACT, Australia: IEEE; 2015:1-6.
133. Sharafaldin I, Lashkari AH, Ghorbani AA. Toward generating a new intrusion detection dataset and intrusion traffic characterization. Paper presented at: Proceedings of the 4th International Conference on Information Systems Security and Privacy (ICISSP). Madeira, Portugal; 2018:108-116.
134. Singh R, Kumar H, Singla R. An intrusion detection system using network traffic profiling and online sequential extreme learning machine. *Expert Syst Appl*. 2015;42(22):8609-8624. https://doi.org/10.1016/j.eswa.2015.07.015.
135. Moustafa N, Slay J. The evaluation of network anomaly detection systems: statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set. *Inf Sec J A Global Perspect*. 2016;25(1-3):18-31. https://doi.org/10.1080/19393555.2015.1125974.
136. Lashkari AH, Draper-Gil G, Mamun MSI, Ghorbani AA. Characterization of tor traffic using time based features. Paper presented at: Proceedings of the 3rd International Conference on Information Systems Security and Privacy(ICISSP). Porto, Portugal; 2017:253-262.
137. Abdulhammed R, Musafer H, Alessa A, Faezipour M, Abuzneid A. Features dimensionality reduction approaches for machine learning based network intrusion detection. *Electronics*. 2019;8(3):322. https://doi.org/10.3390/electronics8030322.
138. Otoum Y, Liu D, Nayak A. DL-IDS: a deep learning–based intrusion detection framework for securing IoT. *Trans Emerg Telecomm Technol*. 2019;e3803. https://doi.org/10.1002/ett.3803.
139. Yang Y, Xu HQ, Gao L, Yuan YB, McLaughlin K, Sezer S. Multidimensional intrusion detection system for IEC 61850-based SCADA networks. *IEEE Trans Power Deliv*. 2016;32(2):1068-1078. https://doi.org/10.1109/TPWRD.2016.2603339.
140. Barka E, Kerrache CA, Benkraouda H, Shuaib K, Ahmad F, Kurugollu F. Towards a trusted unmanned aerial system using blockchain for the protection of critical infrastructure. *Trans Emerg Telecommun Technol*. 2019;e3706. https://doi.org/10.1002/ett.3706.
141. Alipour-Fanid A, Dabaghchian M, Wang N, Wang P, Zhao L, Zeng K. Machine learning-based delay-aware UAV detection and operation mode identification over encrypted Wi-Fi traffic. *IEEE Trans Inf Forens Secur*. 2019;15:2346-2360. https://doi.org/10.1109/TIFS.2019.2959899.
142. Sciancalepore S, Ibrahim OA, Oligeri G, Di Pietro R. PiNcH: an effective, efficient, and robust solution to drone detection via network traffic analysis. *Comput Netw*. 2020;168:107044. https://doi.org/10.1016/j.comnet.2019.107044.

How to cite this article: Ahmad Z, Shahid Khan A, Wai Shiang C, Abdullah J, Ahmad F. Network intrusion detection system: A systematic study of machine learning and deep learning approaches. *Trans Emerging Tel Tech*. 2021;32:e4150. https://doi.org/10.1002/ett.4150