Received: 2 September 2023 | Revised: 19 February 2024 | Accepted: 27 February 2024

IET Networks

DOI: 10.1049/ntw2.12128

REVIEW

Meta-analysis and systematic review for anomaly network intrusion detection systems: Detection methods, dataset, validation methodology, and challenges

Ziadoon K. Maseer[1,2] | Qusay Kanaan Kadhim[2] | Baidaa Al-Bander[3] | Robiah Yusof[4] | Abdu Saif[5]

[1]Faculty of Computer Technology Engineering, Bilad Al Rafidain University College, Baquba, Iraq

[2]Department of Computer Science, College of Science, University of Diyala, Baquba, Diyala, Iraq

[3]School of Computing, Keele University, Keele, UK

[4]Faculty of Information and Communication Technology, Universiti Teknikal Malaysia Melaka, Melaka, Malaysia

[5]Faculty of Engineering, Taiz University, Taiz, Yemen

Correspondence

Baidaa Al-Bander.

Email: b.al-bander@keele.ac.uk

Abstract

Intrusion detection systems built on artificial intelligence (AI) are presented as latent mechanisms for actively detecting fresh attacks over a complex network. The authors used a qualitative method for analysing and evaluating the performance of network intrusion detection system (NIDS) in a systematic way. However, their approach has limitations as it only identifies gaps by analysing and summarising data comparisons without considering quantitative measurements of NIDS's performance. The authors provide a detailed discussion of various deep learning (DL) methods and explain data intrusion networks based on an infrastructure of networks and attack types. The authors' main contribution is a systematic review that utilises meta-analysis to provide an in-depth analysis of DL and traditional machine learning (ML) in notable recent works. The authors assess validation methodologies and clarify recent trends related to dataset intrusion, detected attacks, and classification tasks to improve traditional ML and DL in NIDS-based publications. Finally, challenges and future developments are discussed to pose new risks and complexities for network security.

KEYWORDS

computer network security, computer networks

1 | INTRODUCTION

According to Cybersecurity Ventures, the overall volume of data gathered in the cloud comprises public clouds operated by industrial and social media companies (for instance, Twitter, Microsoft, Google, Facebook, Apple, etc.) [1, 2]. Consumers and corporations can utilise government-owned cloud services. Mid-to-large businesses own cloud storage providers and private clouds [3, 4]. By 2025, data will have achieved 100 ZB, and worldwide data storage will have surpassed 200 ZB [5]. Data stored on public and private network infrastructure data centres and private computing devices, for instance, Internet-of-Things (IoT) devices, smartphones, and PCs, are included in the cloud. Given the interchange of vast volumes of sensitive information through resource-constrained devices and across the untrusted Internet utilising communication protocols and heterogeneous technologies, this fast expansion creates significant security issues. Robust security controls and resilience analysis should be performed in the early stages before installation to ensure secure and sustainable cyberspace. To preserve these technologies progressing, the implemented security controls are accountable for deterring, identifying, and resolving attacks [6]. With an increasing number of devices, numerous companies will have limited resources for network defence to preserve their systems from intrusion, and they practice traditional defence methodologies and instruments such as antispam, antivirus, firewalls etc. [7]. However, these methodologies and devices are exposed to attacks [8]. Network intrusion detection systems (NIDS) are thus critical tools for recognising intrusions, tracking malicious actions, and monitoring network traffic. NIDS implements robust protection systems facing numerous threats. Furthermore, intrusion

detection system (IDS) techniques are categorised into signature and anomaly approaches depending on their work processes [9]. In general, there are two main approaches to NIDSs: Anomaly NIDS and Signature NIDS. The Anomaly NIDS approach identifies unusual network activities to distinguish between regular and intrusive behaviour and can also protect against unknown attacks [10]. On the other hand, Signature NIDS models rely on pre-determined attack criteria and a broad range of network traffic, including platforms, social applications, and IoT media [11, 12]. Massive improvements have been done to develop current industries such as data centre, medical care IT [13], and 5G services [14]. The emergence of new threats could lead to significant difficulties in discovering new attacks within the hybrid network by NIDS models. It is unrealistic to rely on recognising pre-defined patterns to detect intrusions. Therefore, modern networks effectively employ anomaly-based IDS models to detect unknown attacks using traditional machine learning (ML) or deep learning (DL) tools [15].

Several traditional ML and DL approaches were proposed in the latest decade to enhance the NIDS's effectiveness in exposing ill-disposed crimes. Evaluating the effectiveness of traditional ML and DL algorithms is a crucial aspect of building NIDS models. Accuracy is often used to evaluate the performance of AI algorithms during testing. However, real-world modelling problems are rarely samples of attack classes. We may encounter imbalanced datasets or multiclass and multilabel classification problems. Additionally, achieving high accuracy may not always be our primary objective. As we tackle more complex ML problems, calculating and using an accuracy become less straightforward and require careful consideration [16]. The authors suggest a confusion matrix to evaluate the performance of an anomaly NIDS models. This matrix provides us with an output that could be used to calculate precision, recall, and *F*-score to describe the model's overall performance. It is essential to understand the limitations of NIDS models based on the results of different ML contexts. Unfortunately, the current systematic review cannot provide a precise answer to the search query as most authors do not use processes to deal with the outcomes' effectiveness (accuracy, precision, recall, and time detection) of anomaly NIDS studies. Thus, there remains an enormous opportunity to review articles regarding its effectiveness to give a clear picture of gaps and recent trends for NIDS.

Quantitative research methodology uses statistics to analyse numerical data gathered by researchers to answer their research questions [17]. Regarding the quantitative testing of DL methods, a meta-analysis technique has been used to summarise and compare the effectiveness (numerical values) and other metrics of the included studies [18, 19]. It is important to note that while meta-analysis can help summarise and compare data from different research papers, it is not enough to provide a comprehensive overview of primary research on a specific topic. To address this, we propose a systematic review that uses the meta-analysis technique to analyse and observe the latest trends in anomaly-based NIDSs. This approach could help researchers get an in-depth understanding of the topic using a quantitative analysis strategy. It could also provide a clear indication of the most widely used algorithms, metrics, and datasets in the field of intrusion detection. Our focus will be on selecting current issues from studies conducted between 2017 and 2022. We believe that this could help identify future trends in the security domain of NIDS. Our research differs from previous studies [19–31] due to the shortcomings identified in the articles we reviewed concerning exploring gaps and recent trends in anomaly NIDS, as shown in Table 1. Some articles only review future trends in anomaly detection without conducting a systematic review [20–24], while others discuss the performance of anomaly NIDS using a qualitative analysis-based systematic approach without assessing the overall effectiveness of these approaches, including DL methods [30, 31].

Therefore, the contributions are summarised as follows:

1) An overview of different IDS techniques, including traditional ML and DL, provides details about the architecture. Additionally, the various datasets are discussed based on the infrastructure hardware network configuration and date of generation.
2) Analysis (summarisation and comparison) validation of methodology using statistical techniques for current works based on four factors (used algorithms, dataset intrusion, time detection and classification task) as mentioned in Table A1.
3) Observe the recent trends of anomaly NIDS models as well as Identify challenges that researchers and developers face when implementing DL models for NIDS models development.

TABLE 1 Comparison of review articles: (✓: yes, ✗: no).

Review article	Year	ML	DL	Systematic study	Meta-analysis	Current issues	Future trends
[20]	2017	✗	✓	✗	✗	✗	✓
[21]	2017	✗	✗	✗	✗	✗	✓
[19]	2020	✗	✗	✗	✗	✗	✓
[22]	2020	✗	✓	✗	✓	✗	✓
[23]	2019	✓	✓	✗	✗	✗	✗
[24]	2019	✓	✗	✗	✗	✓	✗
[25]	2020	✓	✓	✓	✗	✗	✓
[26]	2021	✗	✗	✓	✗	✗	✗
[27]	2021	✗	✗	✗	✗	✗	✓
[28]	2021	✗	✗	✗	✗	✓	✓
[29]	2021	✓	✓	✗	✗	✗	✓
[30]	2022	✓	✗	✓	✗	✗	✗
[31]	2022	✗	✗	✓	✗	✗	✗
This article	2023	✓	✓	✓	✓	✓	✓

Abbreviations: DL, deep learning; ML, machine learning.

All the abbreviations used in this paper are listed in Table 2. The rest of this paper is organised as follows. Section 2 presents the search map used during this work to identify core anomaly NIDS research using systematic approach including four phases according to the proposed search query. A brief background on NIDSs and their approaches are provided in Section 3. Section 4 provides a taxonomy to NIDS detection methods, which are DL and ML approaches, and a comprehensive overview is presented to show the architectures of DL algorithms and traditional ML algorithms. Network intrusion data traffics are presented and explained that used to train and evaluate anomaly NIDS Models with history line generation from 1999 to 2019 as shown in Section 5. Section 6 provides a comparative analysis and a deep discussion of the reviewed proposals or validation methodology based on the meta-analysis technique. Besides that, Sections 7 and 8 provide current problems and obstacles and future aspirations. Finally, Section 9 is the conclusion of our study.

TABLE 2 List of abbreviations.

Abbreviation	Description
AE	Auto encoder
AI	Artificial intelligence
ANIDS	Anomaly network intrusion detection system
ANNs	Artificial neural networks
AWID	Aegean WiFi Intrusion Dataset
BP	Backpropagation
CICIDS2017	Canadian Institute for Cybersecurity Intrusion Detection System-2017
CICIDS2018	Canadian Institute for Cybersecurity Intrusion Detection System-2018
CNN	Convolutional neural network
DBN	Deep belief network
DL	Deep learning
DNN	Deep neural network
DT	Decision tree
EM	Expectation-maximisation
FFNN	Forward neural network
GAN	Generative adversarial network
K-NN	K-nearest-neighbour
KDD CUP 99	Knowledge discovery and data mining
ML	Machine learning
NB	Naive Bayes
NSL-KDD	Network security laboratory-knowledge discovery and data mining
PNN	Probabilistic neural network
RBM	Restricted Boltzmann machine
RNN	Recurrent neural networks
RF	Random forest
SL	Supervised learning
SOM	Self-organising map
SVM	Support vector machine
ToN_IoT	Telemetry data, operating systems' data, and network - internet of things
UNSW-NB15	University of New South Wales-network based 15

2 | ARTICLE SELECTION METHODOLOGY

This section presents a methodology to implement a systematic literature review (SLR) based on a set of steps to identify, examine, and summarise valuable knowledge from state of the art related to genuine research topics. For the published journal papers from 2017 to 2022, various DL- and ML-based NIDS are evaluated and studied. A SLR is developed and organised into four phases, illustrated in Figure 1.

2.1 | Phase 1

Considering its capabilities to investigate all identified databases, Google Scholar was selected as the default search engine. We determined a search query using words recognised for research. The database will be searched using the following keywords related to IDS research: 'anomaly' and 'intrusion detection system' from 2017 to the first quarter of 2022. These keywords are the most proper guidelines in this domain.

2.2 | Phase 2

With the initial results of the search query in Google Scholar based on title research, it was 17,500 papers. We identified journal articles published between 2017 and 2022. There are random works related to anomaly NIDS models from different resources. We identify good resources in computer science and information technology: *Springer*, *Elsevier*, *IEEE and Scholar*. The references are examined and commented by scientists with details related with anomaly IDS in measurements, datasets, and methodology. The essential papers are selected to review and analyse based on relayed information in this study, as depicted in Table 3.

To reduce the selected papers from the initial set and produce this systematic review paper, we defined inclusion and exclusion criteria. Inclusion criteria included papers focusing specifically on anomaly-based NIDS, papers published within a certain time frame (2017–2022) to ensure relevance, papers understanding, papers reporting original research studies, including empirical studies, experiments, simulations, or case

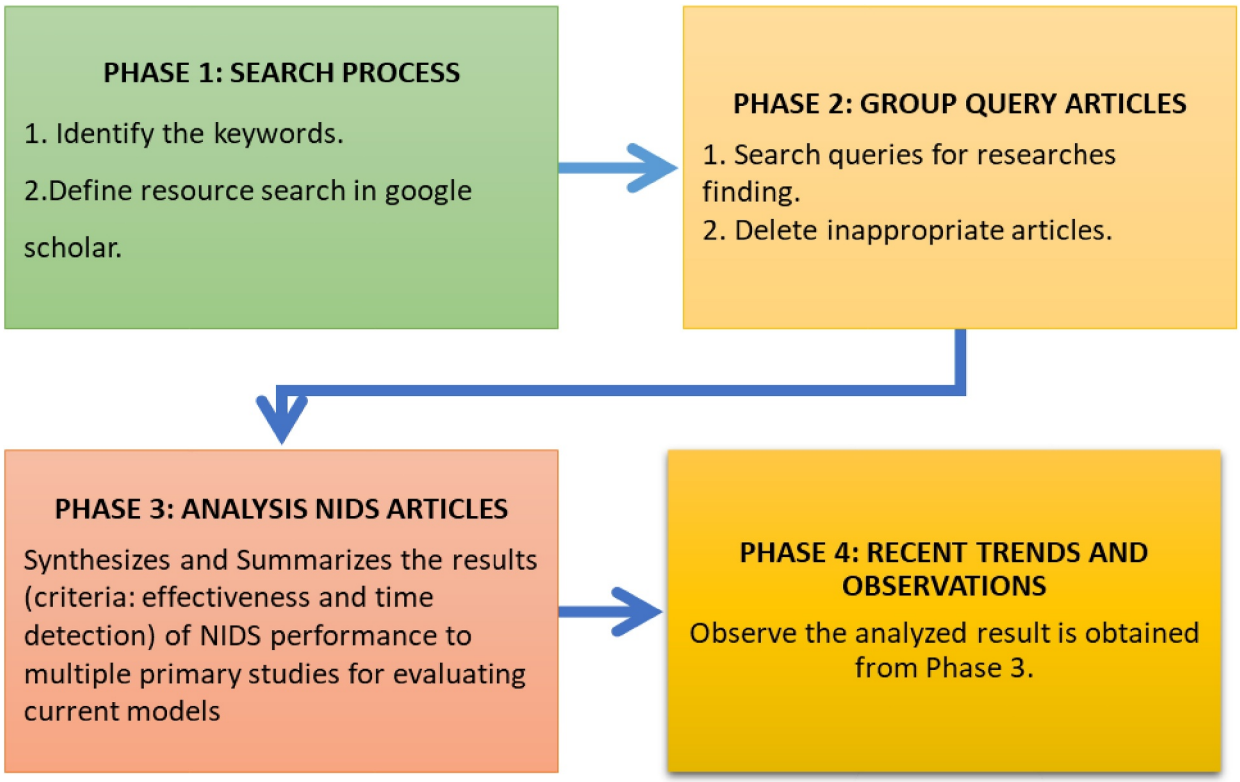

FIGURE 1 Systematic review process.

TABLE 3 Parameters selection for NIDS articles.

Data retrieved	Description
Title	Title of the main study
Journal	Name of the Journal publishing the article
Application area	The area within which the case has been studied
Approach	The proposed approach for IDS as ML/DL
Method	The algorithm implemented to conduct the study

Abbreviations: DL, deep learning; IDS, intrusion detection system; ML, machine learning; NIDS, network intrusion detection system.

studies, papers presenting novel techniques, algorithms, or approaches related to anomaly-based NIDS, papers that evaluate the performance or effectiveness of anomaly-based NIDS using appropriate evaluation metrics, papers providing detailed methodologies and technical descriptions of the anomaly-based NIDS, papers discussing real-world deployment, implementation, or practical considerations of anomaly-based NIDS, papers that provide insights into the challenges, limitations, or future directions of anomaly-based NIDS, and papers from reputable journals, conferences, or academic sources to ensure credibility, whereas exclusion criteria included papers focusing on other types of IDSs rather than anomaly-based or behaviour-based NIDS, papers not directly related to the research question or objective of the systematic review, papers without access to the full text (e.g. abstract-only papers, inaccessible conference proceedings), papers that primarily discuss general network security topics without specific emphasis on anomaly-based NIDS, papers that are duplicates or multiple publications of the same study, papers lacking sufficient details or methodology sections to assess the quality and rigour of the study, papers that are primarily theoretical discussions or opinion-based without empirical evidence or evaluations, papers not published in recognised academic or peer-reviewed sources.

We conduct an initial screening by screening the titles and abstracts of the papers to identify those that potentially meet the inclusion criteria. This initial screening helped to eliminate papers that were obviously irrelevant to the research question. After the initial screening, we carefully evaluated the full texts of the remaining papers. We thoroughly read each paper and compared it against the inclusion and exclusion criteria. Papers that clearly do not meet the criteria were excluded at this stage.

2.3 | Phase 3

As we analyse the full-text papers by extracting a relevant data such as the research objectives, methodology, findings, and conclusions. We created a structured form to record this information. Spreadsheet tools (Microsoft Excel) were used for data extraction during a systematic review. We created customised templates with relevant fields to record information from selected studies. Based on the extracted data, we categorised the papers into different groups or themes (based on the NIDS techniques employed, evaluation methods used, datasets utilised, etc.). Each appropriate research is summarised

and investigated for the suggested DL/ML-based NIDS methodology. The parameters are centered on the most often utilised datasets, testing metrics, and whether the categorisation might be a multi- or binary task to evaluate and analyse the works. The essential criterion concerns the classification task for NIDS because most datasets are imbalanced classes, and IDS software is designed to detect multi-attacks. The software will be more general than the designed software for binary attacks. Therefore, we established the accuracy of the suggested attack detection algorithms. Moreover, the time complexity is an essential factor in measuring the time of algorithms required to complete the task of the NIDS to avoid overhead and packet loss [32]. With those criteria, we provided a significant study of current NIDS based on ML/DL.

2.4 | Phase 4

Lastly, we observe that the analysis data in Table A1 has summarised information findings in reviewed articles for forming the future trends and challenges of this research for AI-based NIDS.

3 | NETWORK IDSs

Anomalies refer to patterns, events, or observations that deviate significantly from the expected or normal behaviour in a given context. In the context of anomaly detection, anomalies are considered as instances that differ significantly from the majority of the data or exhibit behaviours that are unusual or suspicious. Anomaly detection involves identifying and flagging such abnormal instances or patterns within a dataset or system. Anomaly detection, also known as outlier detection, is the process of identifying these unusual or unexpected patterns within a dataset or system [33]. The goal is to distinguish anomalies from the majority of normal instances and potentially identify potential threats, fraud, errors, or unusual behaviour that might require further investigation or action. The use cases for anomaly detection are diverse and span various domains including network intrusion detection [34], fraud detection [35], manufacturing and quality control [36], cybersecurity [37], health monitoring [38], and IoT device monitoring [39].

NIDS are a crucial component of network security infrastructure [40]. They are designed to identify and respond to malicious activities or unauthorised access attempts within computer networks. NIDS play a vital role in detecting and preventing network-based attacks, providing a proactive defence mechanism against cyber threats. The background of NIDSs can be traced back to the increasing prevalence and sophistication of network attacks. As computer networks became more interconnected and vital for organisations, attackers started exploiting vulnerabilities to gain unauthorised access, steal sensitive information, disrupt services, or execute malicious activities [41]. The need for effective intrusion detection mechanisms led to the development of NIDS. NIDS are software or hardware systems that monitor network traffic in real time, analysing it for signs of suspicious or malicious behaviour. They work by inspecting network packets, network protocols, and traffic patterns to identify potential threats and attacks. There are two primary approaches to NIDS: signature-based and anomaly-based detection. Signature-based NIDS rely on a database of known attack patterns or signatures. They compare the network traffic against these signatures to identify matches and raise alarms when a known attack is detected. Signature-based NIDS are effective in detecting known attacks but may struggle with new or evolving attack techniques that do not match existing signatures. On the other hand, anomaly-based NIDS focus on establishing a baseline of normal network behaviour and identifying deviations from this baseline. They use statistical analysis, ML, or rule-based approaches to detect abnormal patterns or behaviours that may indicate an ongoing attack. Anomaly-based NIDS can detect novel attacks but may have a higher false positive rate due to the inherent challenges in distinguishing between legitimate and malicious anomalies [42].

NIDS are typically deployed at strategic points within a network, such as at network gateways, routers, or switches. They monitor network traffic in real time and generate alerts or take automated actions when suspicious activities are detected. NIDS can detect various types of network attacks, including network scanning, port scanning, denial-of-service (DoS) attacks, malware propagation, and unauthorised access attempts. Over time, NIDS have evolved to incorporate advanced techniques, such as deep packet inspection, behavioural analysis, and threat intelligence integration. They have become an integral part of network security architectures, working alongside other security components such as firewalls, intrusion prevention systems, and security information and event management systems. The continuous development and improvement of NIDS are driven by the ever-evolving threat landscape and the need for a robust network security. As attackers employ sophisticated techniques, NIDS must adapt and enhance their detection capabilities to ensure the early identification and mitigation of network-based threats [43].

4 | DETECTION METHODS OF NIDS

In general, learning to train machine and DL ML/DL algorithms can be either supervised or unsupervised. Algorithms belonging to supervised learning (SL) are those trained by classifying cases based on their data labels and then continuing to learn until reaching optimal or maximum value criteria with minimum loss. The SL algorithms in traditional ML are the support vector machine (SVM), random forest (RF), Naive Bayes (NB), K-NN, decision tree (DT), and artificial neural network (ANN). Data instances that are not labelled can be found in unsupervised learning in which clustering dominates the learning approach. Self-organising map (SOM), expectation-maximisation (EM), and *K*-means are unsupervised learning algorithms. Another anomaly-IDS technique is the DL approach, which has robust detection compared to ML by

extracting features by defining attacks such as deep neural network (DNN), deep belief network (DBN), convolutional neural network (CNN), and recurrent neural networks (RNN). These types of algorithms are portrayed in Figure 2.

4.1 | Traditional ML method

The traditional ML method has been widely used in the field of network intrusion detection for many years. These approaches involve the application of various algorithms to classify network traffic as normal or malicious based on historical data. Here are some key aspects of using traditional ML in network intrusion detection.

4.1.1 | Supervised learning methods

Traditional ML learning, a function task that translates input to an output premised on sample input–output pairs, is known as SL. In the traditional ML-NIDS, six supervised traditional ML algorithms are evaluated. The following subsections go through the underlying notions of these algorithms in more detail:

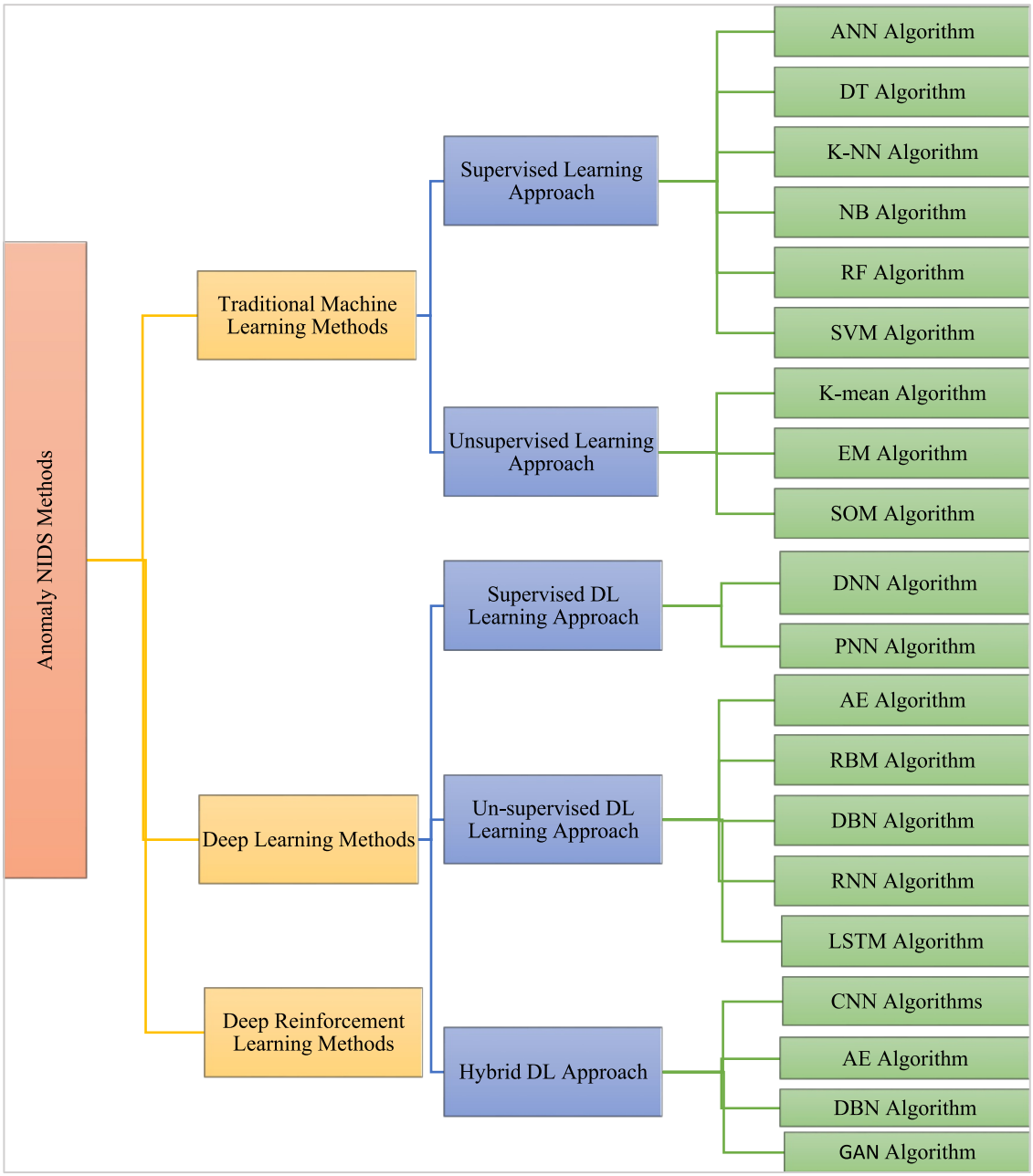

FIGURE 2 Taxonomy of anomaly NIDS methods. NIDS, network intrusion detection system.

Artificial neural networks

ANNs are computing systems that model non-linear problems and foresee output values based on their training values. The ANN comprises three aspects: weights, edges, and nodes or neurons to learn things and make decisions in a human-like manner. An ANN has three layers for constructing the ANN architecture: the output, hidden, and input layers, and each layer comprises a set of nodes. Each neuron's directed connection is affiliated with its weight and connects to other nodes in the next layer via an edge. The output layer represents a machine's or graph's outputs, and the output neurons are equal to labels of data training. The activation functions and weights construct the hidden layers to help discover the network security domain's underlying or structural data features [44]. Figure 3 depicts a general ANN structure (I-H-O) for traditional learning. (I) denotes the input layer nodes, H denotes the hidden layer nodes, and O denotes the output layer nodes. According to the authors in Ref. [45], an ANN algorithm for detecting unknown attacks used KDD CUP 99 datasets to train a multi-layer forward neural network and the mean loss function to reduce the errors. Another approach of an ANN is backpropagation, which uses a backward mechanism from the last layer to update the hidden layers' biases and weights. The significant research works proposed to solve the low detection predictor found in an anomaly NIDS using a neural network algorithm because of the overfitting or underfitting phenomenon [46]. Feature selection was presented to remove irrelevant features for enhancing the training of a neural network algorithm using the NLS-KDD dataset [47]. *K*-fold cross-validation is another solution to handle the overfitting issue by counting the average accuracy within training. Besides this, the meta-heuristic or heuristic algorithms effectively reduce the overfitting issue by selecting optimal parameters such as weight, bias, and the number of hidden neurons [48].

Decision tree

A decision support tool, or DT, is a collection of SL algorithms often deployed to tackle ML categorisation problems. A tree-like model of decision types relates to a circumstance in which the target variable accepts discrete values as input, known as classification trees. DTs comprise leaves or nodes, branches, and one root [49]. While nodes represent the classes of the dataset, branches are the subset of features used to indicate the class labels. A DT may learn for both continuous and discrete data. Given an essential splitter in input determinants, the DT algorithm divides the attributes into two or more analogous sets. The overfitting problem is encountered by a DT, which is overcome via sampling methods, boosting, and bagging [50]. A common structural example of a DT is indicated in Figure 4 [51].

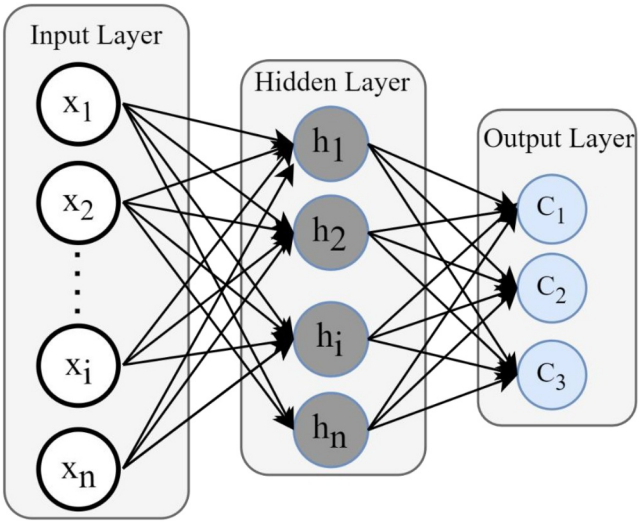

FIGURE 3 Artificial neural network architecture [44].

K-nearest-neighbour (K-NN)

The KNN algorithm is one of the most basic traditional ML algorithms that may be utilised to tackle both regression and classification issues. It is based on the SL technique. According to the authors in Ref. [52], the model can be a regression or classification method. The primary work of the K-NN algorithm uses the distance function to compute the similarity behaviour of points or differences between a pair of points, denoted as D(a, b) in Equation (1) [53]:

$$D(a,b) = \sqrt{\sum_{i=1}^{r} (a_i - b_i)^2} \tag{1}$$

in which a_i denotes the ith-featured element of the instance a, b_i is the ith-featured element of the instance b, while r is the dataset that features the entire quantity. It represents a non-frontier approach with no intuition to publish the underlying data. This algorithm is a simple training of traditional ML models based on the dataset because it requires a small dataset to find the distance between instances or instances and labels. At the same time, a more significant part of the dataset is used for testing this type of model. This is useful as the bulk of the actual dataset is effortlessly derived compared to what is done in a lazy algorithm as the parameter model. It assists the training stage, yet the testing stage is high in speed and memory and less generalised cases for detecting attacks.

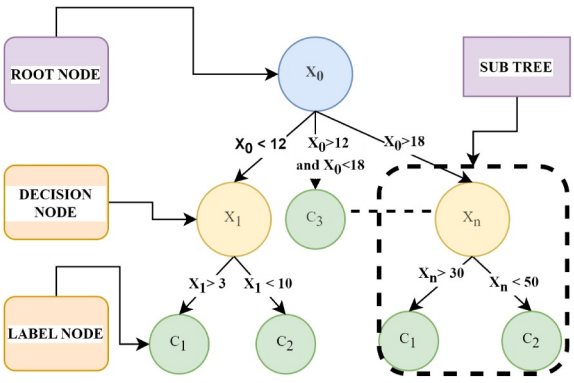

FIGURE 4 Decision tree architecture [51].

Naive Bayes

The NB methods belong to a simple group of probabilistic algorithms established by the Bayes theorem. It addresses naive conjectures of feature independence and dependent features with dataset labels [54]. It is easily trained through the utilisation of a SL structure representing the Bayes equation:

$$P(A|B) = \frac{P(B|A)\,P(A)}{P(B)}. \quad (2)$$

A represents the labels or dependent events, and B represents the values of features. The previous label probability denotes $P(A)$, and the features prior probability denotes $P(B)$, both of which must not be zero. Provided that hypothesis A is true, $P(A|B)$ is the posterior probability of B, and $P(B|A)$ is the probability of the features.

Random forest

An RF is a simple method to build a model using the bagging technique to solve a DT experiencing overfitting and detection issues. An RF addresses this matter by enabling the middle of deep decision trees [55]. An RF is an approach to solving regression and classification problems utilising an ensemble learning approach. Its functions create multiple DTs within the training stage using split data as the bootstrap part. The output considers each feature of the dataset as the classes' mode of a particular DT during the execution of a classification function. Figure 5 shows the RF architecture based on forest trees.

Support vector machine

It belongs to supervised and linear learning that uses a plane that classifies the instances into varying categories. Numerous planes can separate the training instance sets. However, the optimal plan finds support vectors with the highest distance or margins from any class's nearest instances and plane. A hyperplane can be used as a prediction, as presented in Equation (3).

$$g(x) = \left\{ \begin{array}{l} +1 \text{ if } w.x + b \geq 0 \\ -1 \text{ if } w.x + b \leq 0 \end{array} \right\}, \quad (3)$$

$g(x)$ is the predicted class if $g(x)$ is more significant than zero as the normal class, else $g(x)$ is less than zero as the abnormal class [56]. Figure 6 shows an optimal hybrid plane to classify data based on maximised margins for points or support vectors. The disadvantages of linear SVM are that it cannot classify non-linear data, has a low accuracy with noise, and has an overlapping issue. Furthermore, training SVM with a large dataset takes a long time [57]. The genetic algorithm was proposed as a feature selection to select significant features to overcome the disadvantage of the SVM classifier [58]. Other research proposed SVM and K-NN as a preprocessing stage to classify data to improve training the weighted majority algorithm for the generating model [59].

4.1.2 | Unsupervised learning method

Unsupervised learning is a sort of traditional ML that utilises an unlabelled dataset and operates on it without being supervised. There are three selected algorithms as basic concepts of unsupervised ML, and they are further discussed in the following subsections:

K-means clustering

K-means clustering represents an unsupervised learning method for classifying identities into a fixed number (k) of clusters utilising an unlabelled dataset. It is among the common simple unsupervised learning approaches according to the outlook distance of points. It divides the n samples into k groups, and each instance is linked to the cluster that occupies the closest mean with similar behaviour. The main disadvantage of this algorithm is that it needs the prespecification of the number of clusters k. Provided that a set of instances (p_1, p_2,…, p_n) in which each instance

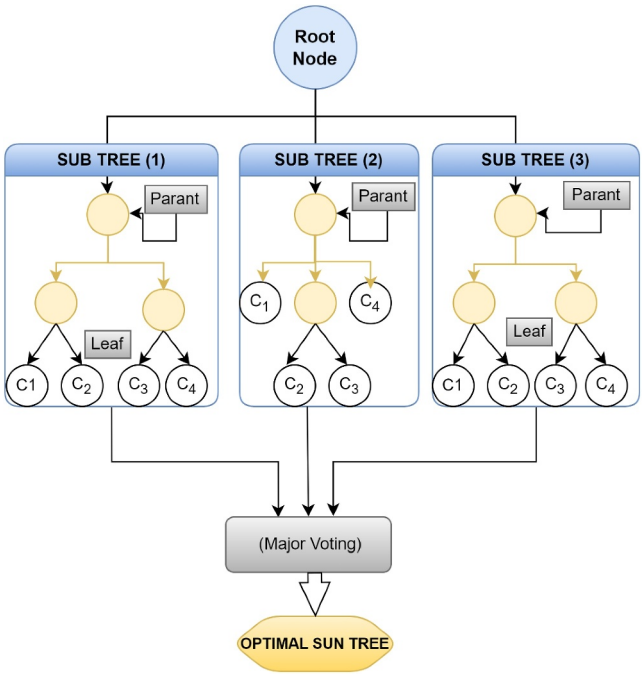

FIGURE 5 Random forest architecture [55].

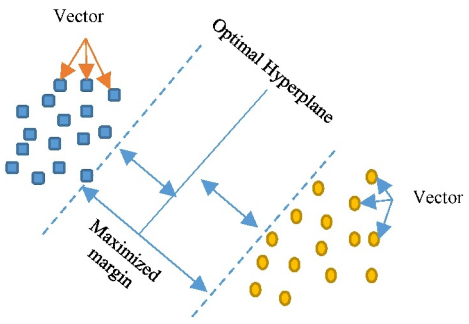

FIGURE 6 Support vector machine architecture [56].

represents a d-dimensional real vector, k signifies clustering that targets to partition p instances into ($k \leq p$) sets $Z = \{Z_1, Z_2 \ldots Z_k\}$ to minimise the variance. Then, k-means is expressed as Equation (4) [60, 61].

$$\text{argmin}_z \sum_{i=1}^{k} \sum_{p \in Z_i}^{\max} ||p - m_i||^2 = a_z \min \sum_{i=0}^{k} |Z_i| \text{Var} Z_i. \quad (4)$$

(argmin) denotes an argument of the class m_i denotes the mean of points in set Z_i.

Expectation-maximisation

EM algorithm performs maximum likelihood estimation with incomplete data, missing and unobserved (hidden) latent variables for EM such as k-means [62]. Thus, the EM algorithm computes cluster membership probabilities based on one or more distributions. Given the final clusters, it aims to maximise the data's overall probability.

Self-organising map

A SOM, shown in Figure 7, is an unsupervised neural network that employs a map to reflect the input distribution's dimensionality. It is predicated on a neural network model known as unsupervised learning. A SOM can cluster data without prior knowledge of input data class groupings [63]. Its research adds a topology mapping from high-dimensional data created by mapping neurons known as units. The training process involves randomly selecting input data from the dataset and computing the similarity using Euclidean distance or cosine similarity between the input data and the weights of each neuron. The neuron whose weight vector is closest to the input data is called the 'winner' neuron. A neighbourhood of neighbouring neurons around the winner neuron is identified, known as neighbourhood selection. This neighbourhood typically shrinks over time during training. The weights of the winner neuron and its neighbours are updated to make them more similar to the input data, which is referred to as Weight Update. The closer a neuron is to the winner neuron, the more its weights are updated. These steps are repeated for a specified number of iterations or until convergence.

4.2 | Deep learning approach

A 'DL approach' refers to the utilisation of deep neural networks to solve complex problems or tasks. Deep learning is a subset of ML that involves the use of neural networks with multiple layers (deep neural networks) to automatically learn hierarchical representations of data. In general, SL, unsupervised learning, and

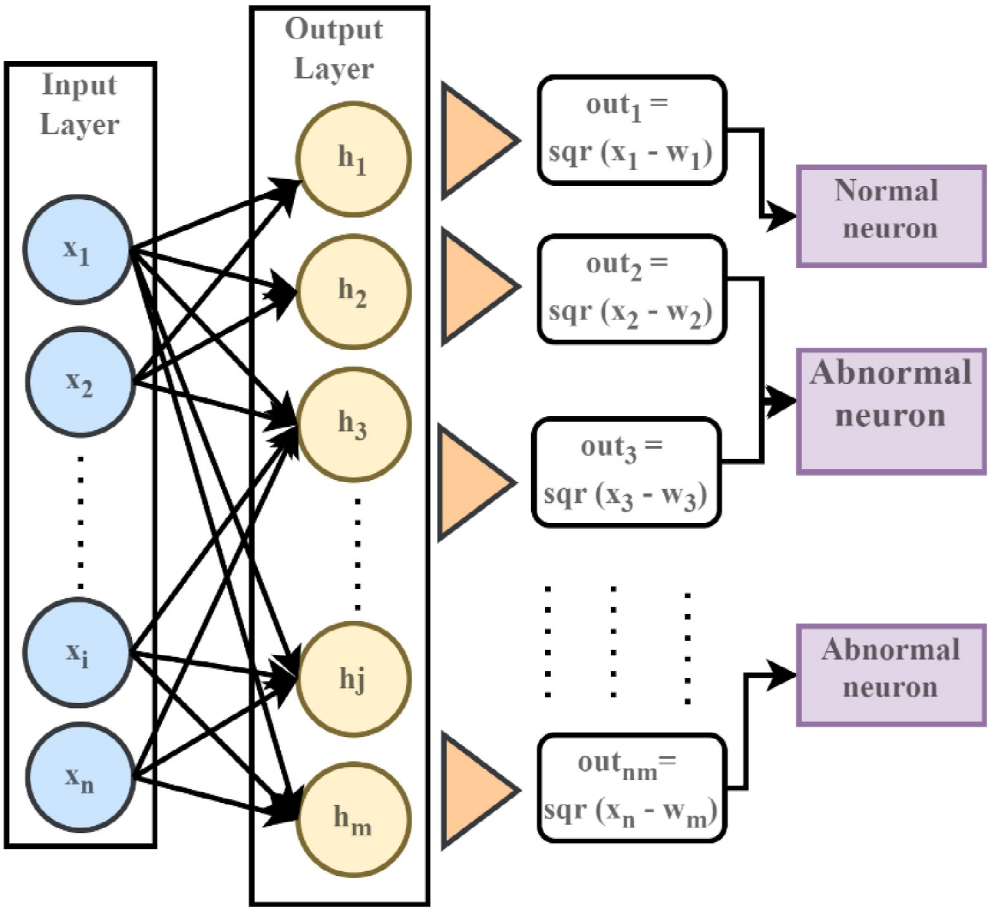

FIGURE 7 Self-organising map architecture [63].

hybrid learning are three main paradigms within the realm of DL, each serving different purposes in the context of ML. Here is a brief overview of each:

4.2.1 | Supervised DL learning approach

In SL, the algorithm is trained on a labelled dataset, where the input data is paired with corresponding output labels. The goal is to learn a mapping from inputs to outputs, enabling the algorithm to make predictions or classifications on new, unseen data.

Deep neural network

An ANN with several layers between the output and input layers is known as a DNN. Deep learning refers to neural networks having more than three layers and more than one hidden layer. Nowadays, the number of network layers utilised in DL ranges from five to over a thousand. As illustrated in Figure 8, DNNs can learn high-level features with greater complexity and abstraction than shallower neural networks.

DNNs are used to process ADFA-WD data to display this point [64]. The data application samples are supplied into the first layer of a DNN in this context. The layer's outputs can indicate various low-level label properties, for instance, attacks and normal. Thus, these traits are integrated to determine the likelihood of higher-level features present at succeeding layers. Moreover, considering all this data, the network probability (comprised of these high-level attributes) is a specific scene or object in the last stage [41]. DNNs can offer improved detection rate performance due to this deep feature hierarchy.

Probabilistic neural network

Probabilistic neural network (PNN) is a feedforward neural network, which is widely used in classification and pattern recognition problems. In the PNN algorithm, the parent probability distribution function (PDF) of each class is approximated by a Parzen window and a non-parametric function. Then, using the PDF of each class, the class probability of a new input data is estimated, and Bayes' rule is then employed to allocate the class with highest posterior probability to new input data. By this method, the probability of misclassification is minimised. This type of ANN was derived from the Bayesian network and a statistical algorithm called Kernel Fisher discriminant analysis. It was introduced by D.F. Specht in 1966. In a PNN, the operations are organised into a multilayered feedforward network with four layers [65, 66].

1. Input layer
2. Pattern layer
3. Summation layer
4. Output layer

4.2.2 | Un-supervised DL learning approach

Unsupervised learning involves training on an unlabelled dataset, where the algorithm explores the inherent structure or patterns within the data without explicit output labels. The primary goal is to uncover hidden patterns, relationships, or representations in the data. The main algorithms are descripted as follows:

Auto encoder

The auto encoder (AE) is another unsupervised approach in DL to learn a compressed representation of raw data. This modification of an ANN has at least three layers: output, input (data), or hidden layers. An encoding function feeds encoded data to the hidden layer from the input layer. To code the compressed form of the input data, hidden layers must have fewer nodes than the input layer. The output data is represented as new features to be input to the next layer as the decoder layer. Which is used to re-construct the features for detecting important features. The difference between the output and input layers is employed to create an error function, and weights are modified to minimise the error. In this context, unsupervised learning learns talent parameters as the weight and basis.

Restricted Boltzmann machine

A restricted Boltzmann machine (RBM) is a two-layer neural network. There are input or visible and hidden layers. Nodes in a visible layer are connected to nodes in a hidden layer, as shown in Figure 9. The nodes inside the hidden layer are also backpropagated to the visible layer in a traditional Boltzmann

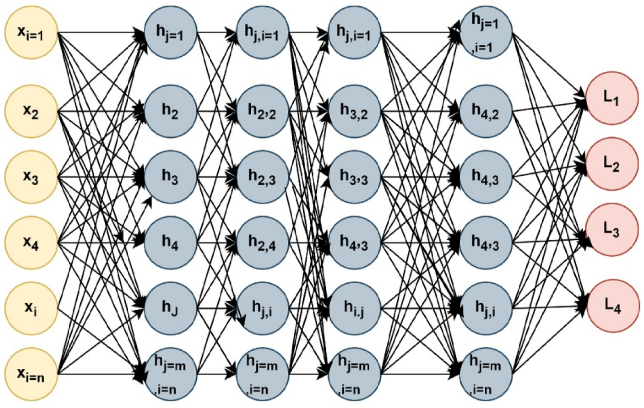

FIGURE 8 Deep neural network architecture [64].

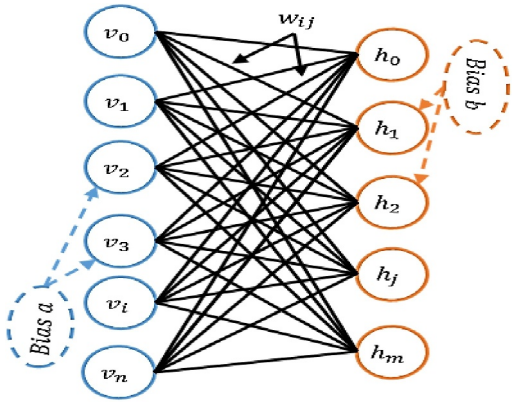

FIGURE 9 Restricted Boltzmann machine architecture [67].

machine (BM). To reduce computational complexity, the neurons of a layer do not connect in an RBM. RBMs use a stochastic technique to calculate the probability distribution of the training dataset throughout the training phase. Once the training commences, each node is given random values. There are also prejudices in the visible and hidden layers of the model. The data is sent to the visible neurons or units that aid in rebuilding the input as it is being transferred to develop the hidden layer's activations. The model's reconstructed input does not match the data input. Generative models are another term for them [67].

Deep belief network

A DBN algorithm is a layer-wise approach for discovering data characteristics from unlabelled features unsupervised that are heavy or greedy [68]. The DBN is made of RBMs; each RBM's output becomes the input to the next RBM. The pre-training of the DBN shows how to enhance the ANN by training each layer separately to solve shallow learning. The DBN layers are separately trained until they reach optimal parameters.

Recurrent neural networks

The RNN is a SL approach of a feed-forward neural network that uses the internal state known as internal memory. The RNN architecture is different from a traditional neural network. It is a correlation between units or nodes in the same layer and feedback to preceding layers [69]. RNNs may maintain a recollection of previous inputs attributable to this feedback. As an outcome, the RNN contains a lot of parameters, and training a model takes a long time.

In contrast, an RNN outperforms a traditional network by solving the vanishing gradients [70]. RNNs consist of a deep architecture. A long short-term memory (LSTM) network is what it is popularly known as. There are three gates in an LSTM network: input, forget, and output. The critical differentiator is network feedback, which might come from a combination thereof the output layer or a hidden layer [71]. In the condition of the binary and multiclass classification of the NSL-KDD dataset, RNN-IDS was recommended to investigate a variety of hidden units and learning rates. Therefore, the findings revealed that the algorithm's accuracy affects numerous learning rates and the number of hidden units. The most dependable accuracy for binary and multiclass cases was attained with 80 hidden units and a learning rate between 0.1 and 0.5.

In comparison to traditional ML models, the proposed model performed well. The weaknesses of this research are that it requires high computational processing and takes a long training time with a low detection for skewed attacks such as R2L and U2R attacks. For the NIDS model with the multilayer perceptron (MPL) and SoftMax module, the gated recurrent unit (GRU) technique was devised. The proposed IDS model consists of preprocessing and a classifier. The essential features are extracted and stored by the GRU. The information is transmitted to the MPL output layer, classifying the line connection as abnormal or usual. The result appeared to outperform LSTM in terms of the detection rate and fast training as fast calculation, and it is easy to stack [72].

Long short-term memory

LSTM network is a DL, sequential neural network that allows information to persist. It is a special type of RNN which is capable of handling the vanishing gradient problem faced by RNN. LSTM was designed by Hochreiter and Schmidhuber [73] that resolves the problem caused by traditional ML algorithms. RNNs work similarly to LSTM; they remember the previous information and use it for processing the current input. The shortcoming of RNN is they cannot remember long-term dependencies due to the vanishing gradient. LSTMs are explicitly designed to avoid long-term dependency problems.

Hybrid deep learning approach

A hybrid DL approach typically refers to the integration or combination of different DL architectures, techniques, or models to enhance the overall performance or address specific challenges [74]. This approach aims to leverage the strengths of multiple methods to achieve better results than using any single technique alone. Here are a few examples of hybrid DL methods:

Deep belief network hybrid

The DBN is learned by the unsupervised approach by RBMs layers and likelihood loss. In addition to, a DBN [75] may be utilised to categorise detected features in SL by combining two backpropagated layers to construct the classifier and the final RBMs to classify detected features as illustrated in Figure 10.

Auto encoder hybrid

The architecture AE hybrid is the same AE unsupervised but add two supervised layers. Combine an autoencoder with a classification model. After pre-training the autoencoder on the data, the learned encoding can be used as input features for a separate classifier (e.g. a softmax classifier) for supervised tasks as depicted in Figure 11. Therefore, the traditional unsupervised learning approach is not compared with the outputs against learning the essential features. Furthermore, the versions of an AE are denoising, spare, and contractive, where the

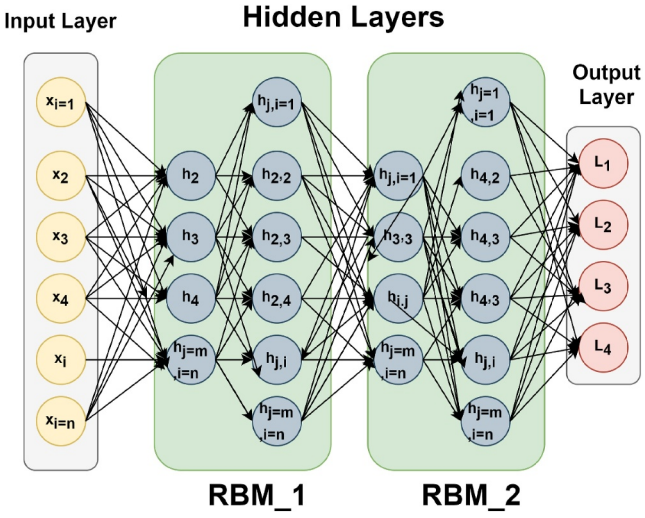

FIGURE 10 Deep belief network architecture [75].

denoising type is commonly used for a NIDS and is significant with datasets in network security [76].

Convolutional neural network

The CNN is a supervised deep-learning algorithm with a three-layer design: flattening, pooling, and convolutional layers. The layers are primarily used to determine the significant features to improve image processing, classification, segmentation, and other auto-correlated data [77, 78]. A typical CNN consists of an input layer that obtains the data input and a convolution layer that generates the feature map by applying a filter matrix to the input data [79]. The pooling layer from the convolution layer determines the feature map's significant values. The flattening layer converts learned multidimensional and learned features to one dimension. This intellectual feature is fed to a fully connected layer that discovers and classifies line connections into normal or abnormal classes. Therefore, the CNN model can be implemented with the sigmoid or SoftMax function to specify a probabilistic value for each category. Remarkable CNN structures are Google Net, AlexNet, ResNe, and VGGNet [80]. A CNN's fundamental architecture for classifying outputs and processing inputs is depicted in Figure 12.

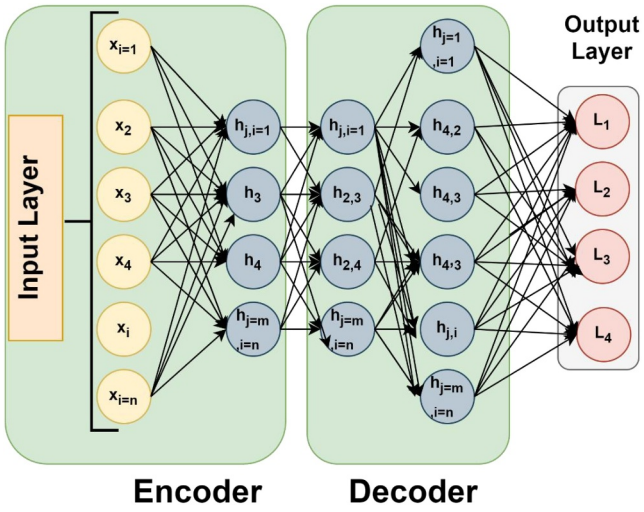

FIGURE 11 Autoencoder architecture [76].

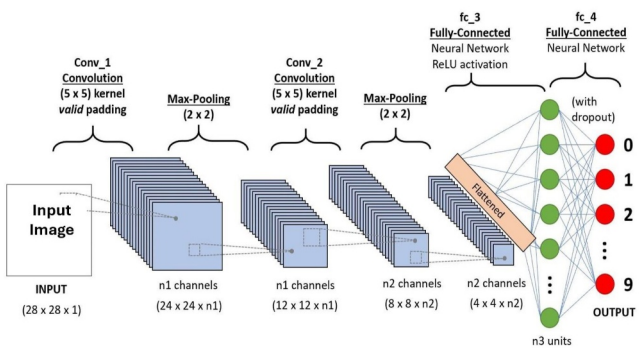

FIGURE 12 Convolutional neural network architecture [80].

Generative adversarial network

Generative adversarial network (GAN) is a type of artificial intelligence (AI) model introduced by Ian Goodfellow and his colleagues in 2014. GANs are designed to generate new data that is similar to a given dataset. The key innovation of GANs lies in their ability to generate realistic data by training two neural networks, a generator, and a discriminator in a competitive and adversarial manner as shown in Figure 13.

High-level overview of the GAN algorithm is described as follows [81]:

1. Generator (G):
   - The generator takes random noise as input and transforms it into data that ideally resembles the real data.
   - The generator is a neural network that learns to map random input vectors (usually sampled from a simple distribution, like a Gaussian distribution) to data points.
2. Discriminator (D):
   - The discriminator is another neural network that evaluates the authenticity of a given piece of data. It distinguishes between real data from the dataset and fake data produced by the generator.
   - The discriminator is trained to classify input data as either real or generated, giving it a binary classification task [81] a GAN was used to process original data samples.

4.3 | Reinforcement learning approach

A reinforcement learning (RL) approach involves designing a system that can learn to make decisions by interacting with an environment. This approach is characterised by the use of an agent, an environment, states, actions, rewards, and a learning algorithm. MCRL can be used to develop IDSs that learn to identify patterns indicative of malicious activities in network traffic. The model can adapt and improve over time as it encounters new attack patterns. In this approach, variety algorithms are used to design the ML [82]. Some of the popular RL algorithms are Q-Learning, Policy Gradient (PG), and Actor-Critic [83]. Several deep RL algorithms have been developed to address various challenges in learning policies for complex

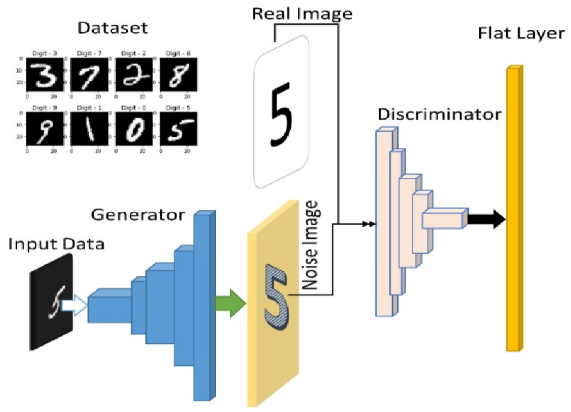

FIGURE 13 Generative adversarial network architecture [81].

tasks. RL is successfully in situations approaching real-world complexity; however, agents are confronted with a difficult task: They must derive efficient representations of the environment from high-dimensional sensory inputs, and use these to generalise past experience to new situations [84]. This makes it possible for machines to mimic some human problem-solving capabilities, even in high-dimensional space, which only a few years ago was difficult to conceive several deep RL algorithms and have been developed to address various challenges in learning policies for complex tasks. The most popular RL algorithms are

- Q-Learning: A model-free algorithm that learns the optimal action-value function iteratively through exploration and exploitation.
- Deep Q-Network: Combines Q-learning with deep neural networks to handle high-dimensional state spaces, using experience replay and target networks for stability.
- PG methods: Directly learn the policy function $\pi(a|s)\pi(a|s)$ by maximising expected rewards, including algorithms like REINFORCE, Actor-Critic, and Proximal Policy Optimisation.
- Deep Deterministic Policy Gradient (DDPG): An actor–critic algorithm suited for continuous action spaces, leveraging deep neural networks to approximate both policy and value functions.
- Twin delayed DDPG (TD3): An extension of DDPG that reduces overestimation bias by using twin Q-networks and delayed policy updates.
- Trust region policy optimisation: Optimises policies by limiting the size of policy updates, ensuring stability and gradual policy improvements.
- Soft actor-critic: Another actor–critic algorithm that incorporates entropy regularisation to balance exploration and exploitation, particularly effective for continuous control tasks.
- Asynchronous advantage actor–critic (A3C): A distributed variant of actor–critic methods that use multiple agents running in parallel to accelerate learning.

5 | DATASET

For the algorithm to learn, a data set (or dataset) is a collection of data utilised to generate by entering a set of training data for which the classes are pre-labelled. Here are the details of the public data utilised by the researchers for experimenting with performing their intended works, namely KDD CUP 99, NSL-KDD, UNSW-NB15, WAID, CICIDS2017, CICIDS2018, and ToN_IoT, refer to Figure 14. Therefore, a report of the dataset and the attacks were discussed in the following subsection.

5.1 | KDD CUP 99

This dataset is designed to assess an IDS constructed in 1998 by MIT Lincoln Labs as a simulation dataset. KDD CUP 99 is

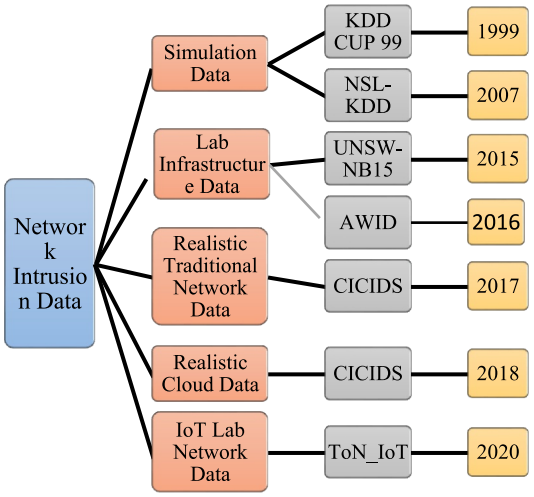

FIGURE 14 Data network.

intensively applied in the AI field as it consists of two parts (training and test data) and consists of 41 features representing characteristics and content traffic [85]. The standard dataset of KDD CUP 99 consists of approximately 5 million raw data. An estimated 80 percent of this dataset has attack data. The dataset is categorised into two main categories: attack and routine.

5.1.1 | Normal

Non-attack or expected behaviour of data.

5.1.2 | Attack types

Denial of service (DOS), user to root (U2R), probing attacks (Probe), and root to local (R2L).

There are 22 attack types, and every single one is part of an attack category classified above. The KDD CUP 99 datasets contain features shown as an actual number and text values about request categories. Furthermore, these datasets also consist of an addend characteristic at the end of the process, indicating the data labelling segregating an intrusion or normal category. The KDD Cup 99 dataset comprises 41 extracted features that capture different aspects of network connections, including basic information, content-related characteristics, and traffic properties. These features serve as inputs for developing and evaluating ML models for network intrusion detection. Basic features provide basic information about network connections. They include attributes such as duration (the length of the connection in seconds), protocol type (TCP, UDP), service (HTTP, FTP), and source/destination IP addresses and port numbers. Content-related features capture the characteristics of the network packets or data content within the connections. They include attributes such as the number of failed login attempts, the number of shell prompts, the number of urgent packets, and the number of bytes transferred. Traffic features describe the statistical properties of network traffic.

They include attributes such as the number of connections to the same host in the past 2 s, the number of connections to the same service in the past 2 s, and various other statistical measures related to time-based properties of network traffic.

5.2 | NSL-KDD

This dataset has been proposed as a solution to some of the difficulties with the KDD'99 datasets. It includes an updated version of the whole KDD CUP 99 dataset. It possesses data features with a different number of classes [86]. Developing the NSL-KDD dataset by reducing the data size by deleting replicated records facilitates ML algorithms' performance. In the NSL-KDD dataset, there are a total of 41 extracted features, similar to the original KDD Cup 99 dataset. However, the NSL-KDD dataset introduces some modifications and improvements to enhance its usability and address certain issues in the original dataset. The NSL-KDD dataset comprises 41 extracted features, including basic features, content-related features, traffic features, and binary features. Binary features were introduced in the NSL-KDD dataset by introducing a new group of binary features that encode specific aspects of network connections. These binary features represent the absence or presence of certain attributes, such as whether a particular type of attack was present or not in a connection. This addition allows for a more precise and balanced representation of different attack categories.

5.3 | UNSW-NB15

This dataset is a network intrusion dataset. The IXIA Perfect Storm tool software generated this dataset. Unlike NSL-KDD, it includes primary variants of different ID cases appearing more frequently nowadays. The number of regular classes is 175,341, and there are 82,332 anomaly classes [87]. The attacks are worms, shellcode, reconnaissance, generic, exploits, DoS, backdoors, analysis, and worms. The UNSW-NB15 dataset consists of 49 extracted features capturing various aspects of network traffic, including basic information, content-related characteristics, statistical properties, connection behaviour, time-related factors, and service-related attributes. These features are utilised to build and assess ML models for network intrusion detection. Basic features provide fundamental information about network connections. They include attributes such as source/destination IP addresses, source/destination port numbers, protocol type (TCP, UDP), and flags (SYN, ACK) associated with the connection. Content-related features capture the characteristics of the network packets or data content within the connections. They include attributes such as the number of bytes transmitted in both directions, the duration of the connection, and the number of packets exchanged. Statistical features describe the statistical properties of network traffic flows. They include attributes such as the average time between packets, the standard deviation of packet length, and the rate of incoming and outgoing packets. Connection-based features focus on the behaviour of network connections. They include attributes such as the number of connections from the same source IP address, the number of connections to the same destination port, and the ratio of incoming to outgoing packets. Time-based features capture the time-related characteristics of network traffic. They include attributes such as the time since the start of the first connection in seconds, the duration of the connection relative to the total duration of the data capture, and the time difference between connections. Service-based features pertain to the specific network services used in the connections. They include attributes such as the service name (HTTP, FTP), the number of connections associated with the service, and the ratio of connections for each service.

5.4 | Aegean WiFi Intrusion Dataset

Aegean WiFi Intrusion Dataset (AWID) is a publicly available dataset used for research in the field of wireless LAN (WLAN) IDSs. The dataset provides network traffic data captured from a wireless network for the purpose of evaluating and developing intrusion detection algorithms. It was collected by the AARNet (Australian Academic and Research Network) and is commonly used in cybersecurity research. The dataset captures network traffic in a WLAN environment, making it suitable for studying security challenges and intrusion detection in WLANs. The dataset includes both normal (benign) traffic and various types of attacks or anomalies, allowing researchers to train and evaluate IDSs on a diverse set of scenarios.

The captured traffic includes information related to the MAC layer, IP layer, and higher-layer protocols [88]. This enables the analysis of network-level and application-level behaviours. Different types of attacks are included in the dataset, such as denial-of-service attacks, de-authentication attacks, and other intrusion attempts. This allows researchers to assess the effectiveness of intrusion detection methods in detecting various security threats. The AWID dataset is sizeable, providing a sufficient amount of data for training and testing intrusion detection algorithms. Large datasets are crucial for developing robust models and avoiding overfitting. Instances in the dataset are typically labelled to indicate whether they represent normal behaviour or instances of attacks. This labelling is crucial for SL approaches in intrusion detection [89].

5.5 | CICIDS2017

CICIDS2017 approximately simulates real-world networks. The CICFlowmeter-V3.0 is intended to create realistic data that includes a set of labels and features. The dataset includes email protocols, SSH, FTP, HTTPS, and HTTP transmitted via an entire network of 25 nodes [90]. The data is collected over a duration of time. The following attacks are recommended in the 2016 McAfee report: DDoS, Botnet, Infiltration, Web Attack, Heartbleed, and DoS. The early dataset [91] did not reveal any harmful attacks. Utilising the B-Profile system, the dataset is

employed to accomplish conceptual feature profiling of users' communication, while the Alpha profile is intended to carry out various attack scenarios. The dataset has characteristics that distinguish it from other data in terms of realism. The phenomenon of realism is covered by 11 criteria, such as completed traffic, protocols, to ensure the quality of the evaluation. The CICIDS2017 dataset consists of 78 extracted features capturing various aspects of network traffic, including basic information, content-related characteristics, statistical properties, connection behaviour, time-related factors, host-based attributes, and traffic-based insights. These features serve as inputs for building and evaluating ML models for network intrusion detection and cybersecurity research. Basic features provide fundamental information about network connections. They include attributes such as source/destination IP addresses, source/destination port numbers, protocol type (TCP, UDP), and flags associated with the connection. Content-related features capture the characteristics of the network packets or data content within the connections. They include attributes such as the number of bytes transmitted in both directions, the duration of the connection, and the number of packets exchanged. Statistical features describe the statistical properties of network traffic flows. They include attributes such as the average and standard deviation of packet length, the average and standard deviation of packet inter-arrival time, and various statistical measures related to payload and flow properties. Connection-based features focus on the behaviour of network connections [92]. They include attributes such as the number of connections from the same source IP address, the number of connections to the same destination port, and various ratios and counts related to connection behaviour. Time-based features capture time-related characteristics of network traffic. They include attributes such as the time since the start of the first connection in seconds, the duration of the connection relative to the total duration of the data capture, and various time-related statistics. Host-based features represent information related to the host or device involved in the network connection. They include attributes such as the number of connections from the same host, the number of different services used by the host, and various ratios and counts related to host behaviour. Traffic-based features provide insights into the overall network traffic characteristics. They include attributes such as the total number of packets, the total number of bytes, and various traffic-related statistics.

5.6 | CICIDS2018

It was a collaborative project between the Communications Security Establishment and the Canadian Institute for Cybersecurity [93]. It provided 10 days of traffic, from Wednesday, 14 February 2018, to Friday, 2 March 2018, focusing on Amazon Web Services. This included seven different attack scenarios deemed similar to CICIDS2017: brute-force (FTP-Patator and SSH-Patator), Denial of Service (slowloris, SlowHTTPTest, Hulk, GoldenEye), Heartbleed, web attacks (Damn Vulnerable Web App, XSS,brute-force), infiltration of the network from inside, botnet, and Distributed Denial of Service with port scanning [94]. However, CSE-CIC-IDS2018 was a more complete dataset than CICIDS2017, with more data and different network topologies. The CICIDS2018 dataset consists of 79 extracted features capturing various aspects of network traffic, including basic information, content-related characteristics, statistical properties, connection behaviour, time-related factors, host-based attributes, and traffic-based insights.

5.7 | IoT dataset

The TON_IoT datasets are new generations of IoT and Industrial IoT (IIoT) datasets for evaluating the fidelity and efficiency of different cybersecurity applications based on Artificial Intelligence (AI). The datasets have been called 'ToN_IoT' as they include heterogeneous data sources collected from Telemetry datasets of IoT and IIoT sensors, Operating systems datasets of Windows 7 and 10, and Ubuntu 14 and 18 TLS and Network traffic datasets. The datasets were collected from a realistic and large-scale network designed at the IoT Lab of the UNSW Canberra Cyber, the School of Engineering and Information technology, and UNSW Canberra at the Australian Defence Force Academy. The datasets were gathered in parallel processing to collect several regular and cyber-attack events from IoT networks [95]. A new testbed was developed at the IoT lab to connect many virtual machines, physical systems, hacking platforms, cloud and fog platforms, IoT and IIoT sensors to mimic the complexity and scalability of industrial IoT and Industry 4.0 networks [96]. For example, IoT/IIoT datasets typically capture various aspects of sensor data, network communication, and device behaviour in connected environments [97]. The extracted features can be categorised into several types, including sensor data features, network communication features, device metadata features, Time-based features, contextual features, and derived features. Sensor data features represent measurements or readings from IoT/IIoT sensors. They may include attributes such as temperature, humidity, pressure, light intensity, vibration, sound level, or any other relevant environmental or physical measurements. Network communication features capture information related to network communication between IoT/IIoT devices. They may include attributes such as source/destination IP addresses, source/destination port numbers, protocol type (e.g. MQTT, CoAP), message payload size, message frequency, or other network-related characteristics. Device metadata features provide descriptive information about the IoT/IIoT devices themselves. They may include attributes such as device ID, device type, firmware version, hardware specifications, location, or any other relevant device metadata. Time-based features capture time-related characteristics of IoT/IIoT data. They may include attributes such as timestamps, time intervals between data points, or any temporal patterns or trends within the dataset. Contextual features capture contextual information associated with the IoT/IIoT data. They may include attributes such as location information, user context, environmental conditions, or any other contextual factors that

can provide additional insights into the dataset. Derived features are calculated or derived from the raw sensor or network data. They may include statistical measures such as mean, standard deviation, maximum, minimum, or more complex-derived features such as frequency domain analysis, waveform characteristics, or pattern recognition. The number of extracted features in IoT/IIoT datasets can vary significantly depending on the specific dataset and the nature of the collected data. It can range from a few dozens to hundreds or more features, depending on the complexity of the IoT/IIoT system and the goals of the dataset.

6 | VALIDATION METHODOLOGY AND RECENT TRENDS

In order to develop a reliable and efficient NIDS, it is critical to utilise advanced AI algorithms and a current dataset to construct an anomaly-based NIDS that can identify zero-day attacks. The classification task employed is also an important aspect when creating an anomaly-based NIDS. There are two primary classification tasks in ML: binary and multi-class classification. The effectiveness of the NIDS model in detecting attacks is crucial. Therefore, when examining validation methodologies in current works, four factors should be taken into consideration: algorithm, popular dataset, classification task, and effectiveness. Table A1 provides detailed information on these factors. Based on our analysis of validation methodologies, we have observed the following according to these hypotheses.

6.1 | Benchmarking evaluation

There is a need for a reproducible and benchmarking evaluation environment to investigate and compare different approaches to anomaly detection. The assessment process is highly dependent on the specific application and requires an evaluation that considers the specific characteristics of the used case. According to our survey, a huge anomaly detection works had been done in the last decade. Different contributions had been done by researchers, and they used variant measurements to evaluate their result of effectiveness of ML/DL models. For instance, some researchers used an accuracy rate to measure the effectiveness [98–101]. Other authors used only the precision rate to show precis AI models for detecting attacks [62, 102–104]. The recall and precision rates were used to measure the fraction of true attacks and true normal case that are detected by the anomaly NIDS model based on the KDD CUP 99 and NSL-KDD dataset [99, 100, 105, 106]. Furthermore, other trend was a time detection to measure researcher's contributions for reducing the structure complexity of DL-NIDS models [107–110]. As seen in Table A1, most researchers had used a few criteria to evaluate the methodologies of anomaly NIDS models but not all units of measurement. Some measurements had used time detection or accurate rate to check the detection of models for connection lines to be true positive and true negative. All information units of network security measurements have been summarised from Table A1 as mentioned in Figure 15 and also shows the number times that have been used to measure the ability and complexity of anomaly NIDS models over our searching query.

A higher precision score has been employed (52) times and indicated false positive predictions in NIDS models. The accuracy rate has been used and pointed to an accurate classification in AI models for negative or positive classes with (46) time. Also, recall and *F*-score criteria have been used in (46) and (38) studies, respectively, to evaluate an anomaly NIDS for detecting the attacks correctly. Unfortunately, few full criteria had been used around (4) studies to measure the efficiency of NIDS models for detecting attacks in both cases (positive and negative classes).

The nature of the dataset is a highly imbalanced instance to be realistic and simulate real traffic data of the Internet, where instances are high normal class and rare attacks. Consequently, realistic assessment should use multiple criteria to evaluate and show the real performance of anomaly NIDS in both detection cases.

It is clear that few benchmarking evaluations have been considered and a real performance appeared to detect attacks by DL/ML models in terms of validation methodology.

6.2 | Used algorithms

According to the studies in Table A1, writers have proposed the IDS system utilising the DL technique in the recent 3 years, as demonstrated in Figure 16. It can be noticed that 50% of DL approaches were available to construct the IDS model. On the other hand, only 20% of suggested models employ a hybrid approach that combines DL and traditional ML techniques, whereas only 30% of recommended solutions depend on ML approaches. As previously said, DL algorithms are complex and demand a lot of computing power. In addition, Figure 16 reveals how frequently writers use DL algorithms to create successful IDS models. The four standard multiple algorithms employed for NIDS are RNN, CNN, DNN, and AE, all of which are DL techniques. Subsequently, ML techniques, for instance, SVM and RF, were presented as a hybrid strategy to enhance DL algorithms. Different metrics are typically used to measure anomaly detection in NIDS models. The effectiveness

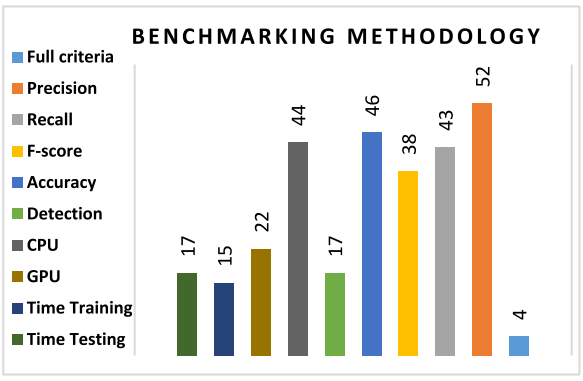

FIGURE 15 Used criterias over 2019–2022.

of ML/DL reflects performance based on accuracy, confusion matrix, precision, recall, and *F*-score.

Figure 17 shows the reviewed articles in the group distribution for DL approaches based on the summarisation of DL works in Table A1. It can be observed that the AE algorithm has used more than supervised or sim-supervised learning algorithms to build NIDS models.

The AE algorithm had most used in a NIDS model with 16 times such as [72, 111], while DNN algorithm had used to detect attacks by some authors such as [112] with 9 times. With unsupervised and hybrid learning types, DBN and AE algorithms and their different versions had the most widely used algorithms for proposing NIDS solutions. With the majority of the time, AE is expressly embraced and utilised for feature reduction and extraction. Meanwhile, for classification purposes, it is accompanied by the ML-based classifier. The SVM, RF, DT, and *K*-means build NIDS models and are used more often than other traditional ML, as shown in Figure 18.

6.3 | Network intrusion dataset

Most datasets have been designed by different institutes of cybersecurity based on the various infrastructure network such as cloud, Software Defined Network, or web database server.

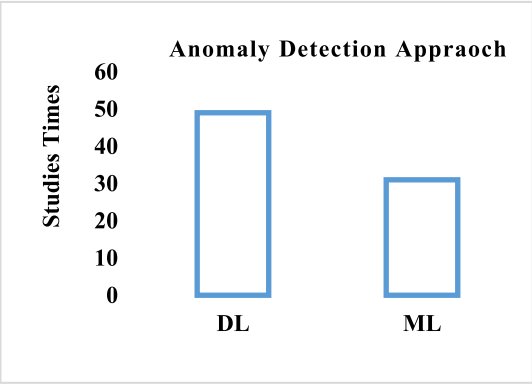

FIGURE 16 Frequency of ML and DL algorithms used in the surveyed materials. DL, deep learning; ML, machine learning.

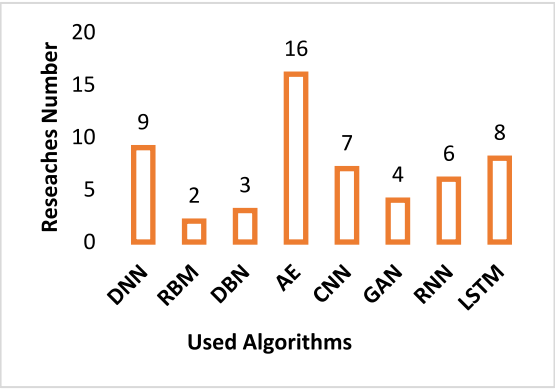

FIGURE 17 Frequency of deep learning techniques used in the surveyed materials.

The researchers have utilised labelled or unlabelled data traffic to build an anomaly NIDS-based learning methods. Labelled data had been used to train the supervised tool-based data classes, while an un-labelled data network had been used to train the AI system by defining the status of the connection lines. According to the analysis of Table A1, we observe that the distribution of data traffic is either to be synthetic or realistic and uses unequal for training and testing in the anomaly NIDS models as shown in Figure 19.

The (54%) studies have used the KDD CUP 99 and NSL-KDD dataset for building and validating DL/ML models using MIT simulation, which includes an environment to acquire 9 weeks of raw TCP dump data for a LAN, simulating a typical U.S. Air Force LAN.

The UNSW_Lab data network was generated by the local simulation network using a specific configuration with the three virtual servers UNSW library. Servers 1 and 3 were configured for the normal spread of the traffic while server 2 formed the abnormal/malicious activities in the network traffic. The adoption rate of UNSW_Lab data is formed (15%)

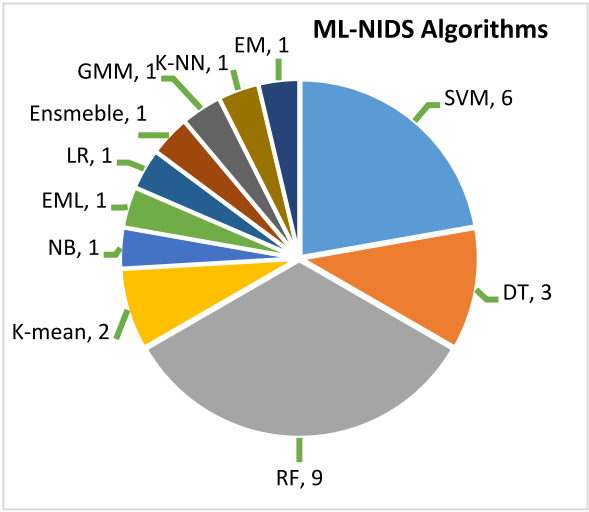

FIGURE 18 Traditional ML algorithms for NIDS. ML, machine learning; NIDS, network intrusion detection system.

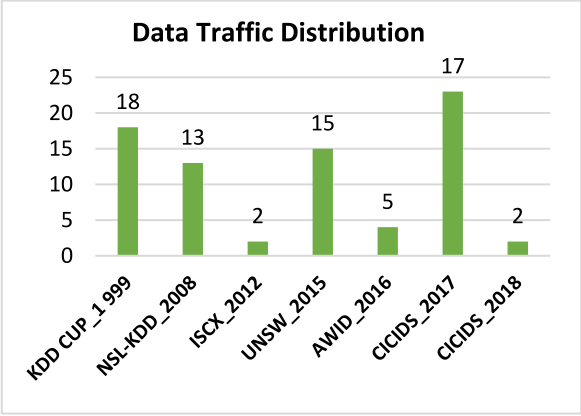

FIGURE 19 Number of the proposed dataset.

of the anomaly detection studies and used for building the anomaly NIDS models by a few researchers.

The modern network is an evolution design that is not the same as an infrastructure network in the last age, and these datasets are ancient. It becomes more dangerous to build the production tool for the advanced networks, and there might be more possibilities that the suggested methods will not play well when placed in real-world situations. Advanced datasets, for instance, BoT-IoT 2018 and CICIDS 2017, are utilised for building a strong AI-IDS model-based configuration setting with different devices such as Modem, Firewall, Switches, Routers, and nodes with different operating systems (Microsoft Windows (like Windows 10, Windows 8, Windows 7, and Windows XP), Apple's macOS iOS, and open-source operating system Linux). This will appear to be more significant in the actual world than models that utilise an outdated dataset. When using these datasets, it is essential to be mindful of the data's characteristics, potential biases, and the specific types of attacks included. Unfortunately, few types of research, as 31% of studies propose a realistic dataset for training anomaly NIDS, are shown in Figure 20.

The highest values to KDD Cup 1999 are 18 studies that were used to define the rules of abnormality in NIDS models for detecting attacks such as DoS, U2R, R2L, and Probe [105, 107, 113–115]. NSL-KDD is an improved version of the original KDD Cup 1999 dataset. It addresses some of the limitations and biases present in the KDD Cup 1999 dataset, but the NSL-KDD dataset is still a little used from KDD Cup 1999 data with frequency in 13 articles only such as in Refs [103, 116–118].

ISCX_2012 data was captured from the real-world network and represented real scenarios for different attacks. However, ISCX_2012 data stands for achieving a benchmark performance of intrusion detection models [32, 119]. Furthermore, the AWID dataset has been utilised to protect the wireless network against three types of attacks such as Normal, Flooding, and Impersonation [120–122]. ISCX and AWID have suggested only two and five frequencies over 2019–2022 for building reliable anomaly NIDS models based on real scenarios for different attacks.

Researchers have used a CICIDS2017 dataset in 17 times of studies in network anomaly detection, which is utilised for detecting both benign behaviours and also new malware attacks such as Brute Force FTP, Brute Force SSH, DoS, Heartbleed, Web Attack, Infiltration, Botnet and DDoS [123–126].

6.4 | Classification task

The classification is an essential task concerning the traditional ML or DL approach. Two parts to training the anomaly models to be an IDS classifier are binary and multi-task. A binary task is a simple operation to classify connection lines into normal and abnormal classes. At the same time, multi-task is a complex operation to classify data into multi classes as usual or attacks. Table 4 shows a statistical classification task for anomaly NIDS detection, and the classification task has been proposed to build anomaly NIDS around (47 rates) in both binary and multi-classification.

The AI algorithms can efficiently train and fit parameters for building anomaly-IDS models in binary classification tasks. At the same time, the multi-classification ML/DL models have many parameters that need to be improved or optimised for concluding optimal non-linear equations for anomaly-IDS models.

6.5 | Binary classification

Figures 21 and 22 show the data comparison to the overall and per-class measurement for anomaly NIDS models in binary classification. Overall measurement shows the anomaly detection system's ability to detect attacks in a binary form while ignoring imbalance or realistic features. The accuracy and precision are higher and are used by recent researchers than other metrics (Recall and *F*-score). The specific measurement is used to show the performance of NIDS to detect per attack or class regarding the imbalance dataset. The precision, recall,

TABLE 4 Statistical classification task in Table A1.

Classification task type	Statistical studies
Muli- classification	47
Binary-classification	47

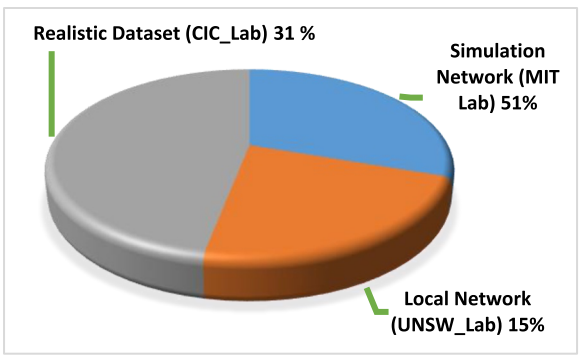

FIGURE 20 Percentage of the network intrusion dataset.

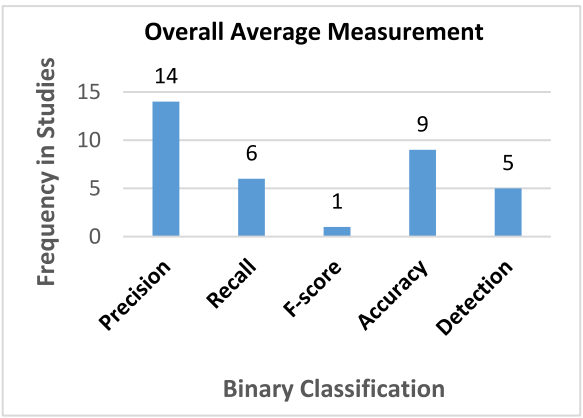

FIGURE 21 Overall measurement in the binary task.

F-score, and accuracy were used with different percentages as research metrics. The recall and *F*-score were used to check the models' fitting than other metrics (precision and *F*-score).

6.6 | Multi-classification

Most cybersecurity datasets are multi-classes and imbalance labels. Building the anomaly detection system based on binary tasks and costly detection with only one attack and a regular connection is unreasonable. Multi-classification is a problematic task via anomaly network detection, and the researchers are required to find an optimal non-linear question for building the AI models for detecting many attacks simultaneously.

Overall and per class are used to evaluate the system, and the researchers used average or overall measurement for all samples or line connection using precision, recall, *F*-score, accuracy, and detection to show researchers' contributions to the anomaly IDS.

As seen in Figure 23, the overall frequency is more accurate than the rest of the measurement, followed by the precision rate. To have a more specific evaluation of anomaly NIDS, the authors proposed metrics to show the detection of per class or attack. In Figure 24, the per-class measurements are different levels that have been used to evaluate the detection of each attack in many types of research, and the accuracy is used more than other metrics to measure the accuracy the anomaly detection per class.

FIGURE 22 Per class measurement in the binary task.

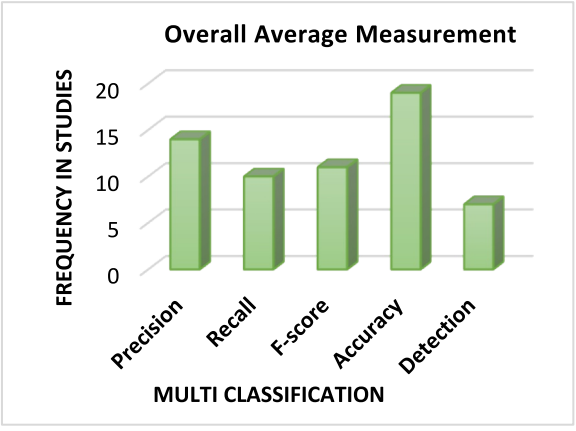

FIGURE 23 Overall measurement in multi-class task.

6.7 | Anomaly NIDS effectiveness

The NIDS effectiveness is measured through confusion metrics of cybersecurity, which are accuracy, precision, and recall. Other authors could propose time detection or prediction to measure the ability of efficiency as shown in Table A1. This article contained a lot of papers to anomaly detection. The best conditions for the anomaly NIDS effectiveness that is pointed to optimal detection of attacks and the benchmarked works indicate actual effectiveness that had been selected from Table A1. We discuss those conditions to give a clear image of the actual performance of anomaly NIDS based on high values for each accuracy, precision, recall, and *F*-score.

Most authors had proposed couples or a few metrics to measure the effectiveness of AI algorithms such as accuracy. The accuracy had only been used to measure the effectiveness of DL-NIDS models, and the results of the accuracy rate could range between 0.83 and 1.00 with (KDD CUP 99, UNSW-NB15, CICIDS2017, and CSE-CIC-IDS2018) datasets, respectively [127, 128]. Similar to the accuracy measurement metric, the precision rate had been used to measure the effectiveness of anomaly detection but examination of the connection lines to be true and false actives by different AI algorithms. The RF algorithm was adopted for multi-detection or classification attacks using KDD CUP 99 data traffic and the precision rate indicated to the acceptable level at 0.99, 0.92, 0.17, 0.66, and 0.55 for normal and four types of attacks such as DoS, U2R, and R2L, respectively. The authors used a simple way to find the overall precision rate by calculating the average true positive and negative for five classes with a 0.99 rate by the AE algorithm [129]. The CNN algorithm had been proposed to improve the detection of multi-classes (regular and four types of attack) based on the NSL-KDD dataset with a good precision percentage

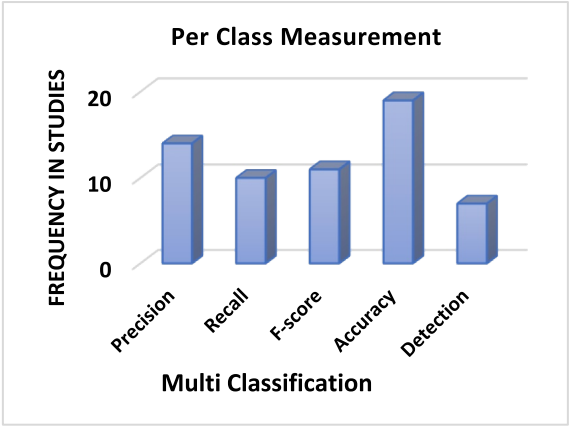

FIGURE 24 Per class measurement in a multi-class task.

of 1.00 (regular), 1.00 (DoS), 0.70 (U2R), 0.70 (R2L) and 1.00 (Probe) [130]. The precision used to measure the overall average effect of an anomaly NIDS by different algorithms such as AE, RF, RBM, and DNN, the authors used to train and evaluate the anomaly NIDS based different datasets is NSL-KDD, AWID, UNSW-NB15, CIDDS-001, and Realistic data [131, 132]. The excellent precision was performed by the RF algorithm with around a 1.00 rate for normal lines and other types of attacks of the NSL-KDD dataset [133].

To effectively address both false positives and false negatives in cybersecurity problem, it is essential to use comprehensive metrics such as precision, recall, *F*-score, and accuracy. These metrics ensure that the evaluation results are accurate and reliable. One of the studies reported in Table A1 [134], observed improvement on the effectiveness of anomaly NIDS by solving false positive and false negative issues in the imbalanced dataset (rare attacks and many samples of a regular action). Indeed, the results reported in the aforementioned study are the closest to the realistic as well as benchmarking due to including full or some of metrics including, false negative, recall, false positive, and *F* score to balance between precision and recall.

In KDD CUP 99, normal actives and attacks (DoS and Probe) detected by a *k*-mean algorithm revealed acceptable detection levels for minor samples of U2R and R2L attack [135]. The effectiveness had enhanced by using the AE algorithm to detect connection lines whether normal or attacks (DoS, U2R, and Probe) in different types of data traffic (NSL-KDD), but anomaly detection could be less effective for detecting R2L attacks [136] as shown in Figure 25. The effectiveness of anomaly detection was measured by three metrics: precision, recall, and *F*-score using LAN data traffic (UNSW-NB15) as shown in Figure 26. Low effectiveness for detecting DoS, Analysis, Backdoor, and Shellcode [41].

These metrics are roughly starting from 0.10 to 0.40 rate. Opposite+ with other connection lines, the effectiveness is increased detection for Normal, Generic, Exploits, Fuzzers, Recon and Worms with precision, recall and *F*-score by starting from 0.70% to 1.00% [137]. In addition, the effectiveness of the AE algorithm had evaluated using a realistic network (CICIDS2017) which contains modern attacks as shown in Figure 27. The result had shown three metrics (precision, recall, and *F*-score), and the effectiveness of performance is high detection for whole connection lines except of low detection in filtration and botnet attacks [138].

7 | CURRENT CHALLENGES OF RESEARCH

The following summarises the current challenges and highlights the high-security issues occurring today as scenarios in NIDS articles for researchers to view directly.

7.1 | False-positive issue

Most suggested IDS approaches exhibit false negatives for low-frequency attacks according to the existing findings. Studies find this drawback with multi-task classification methods, which will be inefficient for detecting attacks with fewer samples during the training dataset [139]. These issues are caused by an imbalance class problem, which biases the model's training for the majority samples of the class compared to minority attack classes.

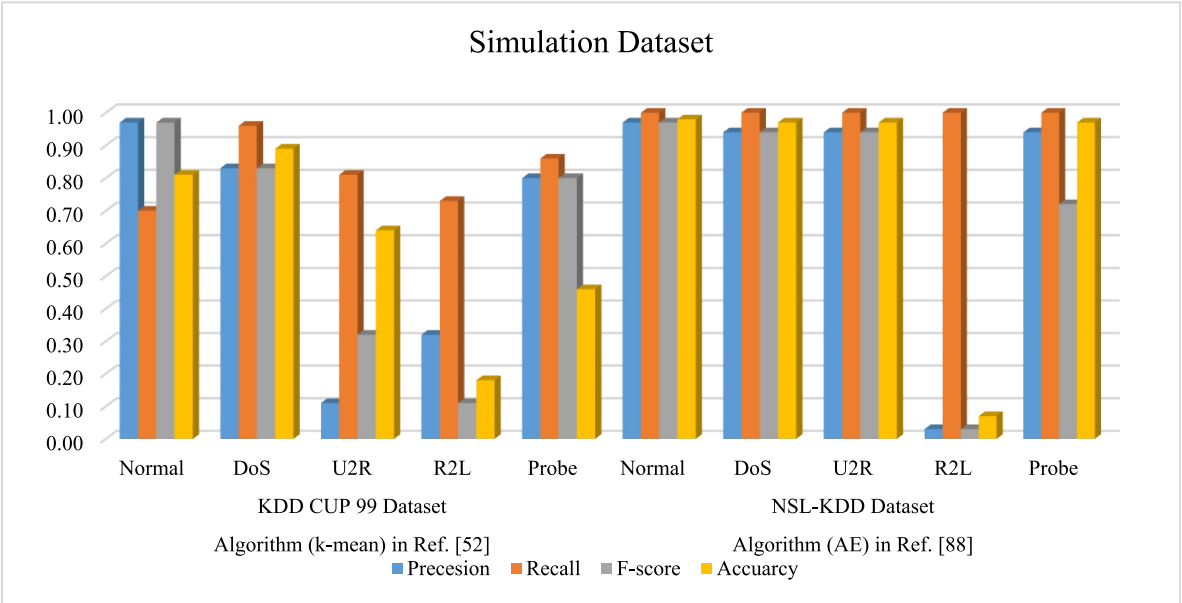

FIGURE 25 Effectiveness of anomaly NIDS-based (KDD CUP 99 and NSL-KDD). KDD CUP 99, knowledge discovery and data mining; NIDS, network intrusion detection system; NSL-KDD, network security laboratory-knowledge discovery and data mining.

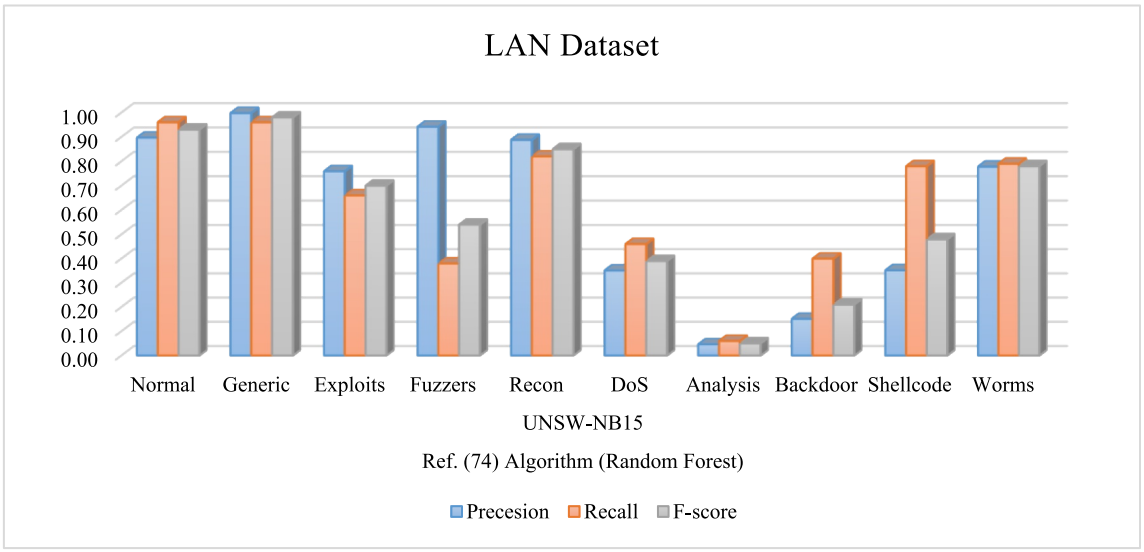

FIGURE 26 Effectiveness of anomaly NIDS-based (UNSW-NB15). NIDS, network intrusion detection system; UNSW-NB15, University of New South Wales-network based 15.

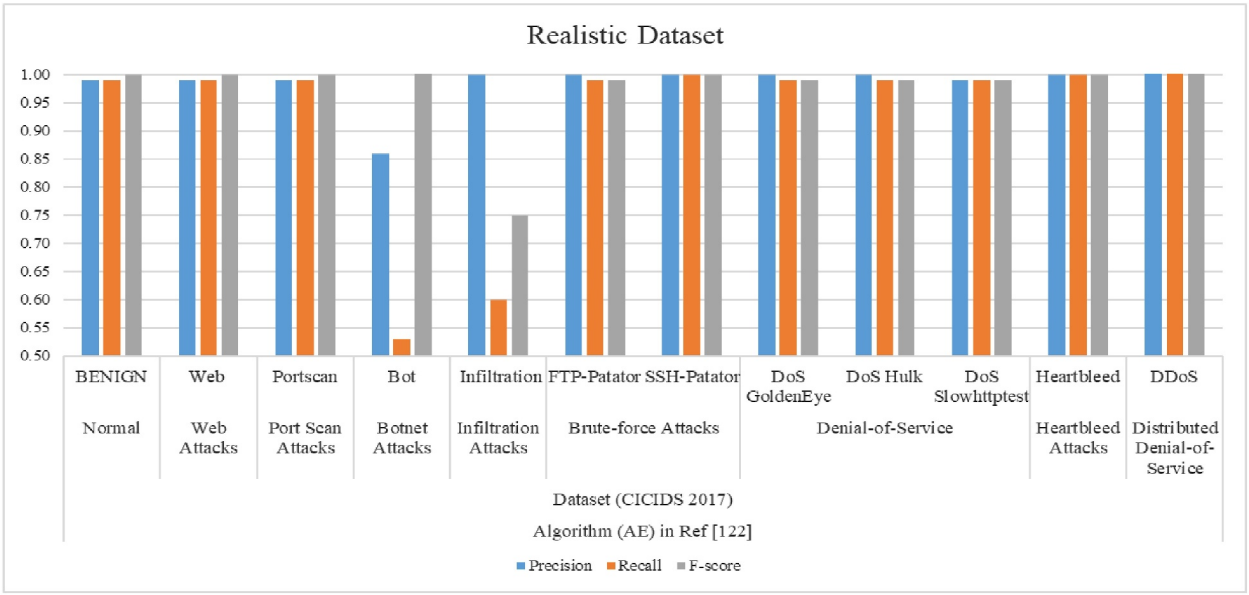

FIGURE 27 Effectiveness of anomaly NIDS-based (CICIDS2017). CICIDS2017, Canadian Institute for Cybersecurity Intrusion Detection System-2017; NIDS, network intrusion detection system.

7.2 | Low-quality dataset

The utilisation of a suitable dataset is an important goal in the development of a DL-based IDS. The existing suggested DL-based IDSs do not deliver dependable overall performance, as detailed in Section 4. They utilise the KDD CUP 99 or NSL-KDD benchmark datasets, which feature ancient traffic, do not reflect existing attack scenarios or traffic habits, and have real-time features. Assessing more existing datasets, for instance, the CICIDS2017 IDS dataset [140] and the Bot-IoT dataset, can address this issue by acquiring traffic from simulated settings. Datasets can also be created and published datasets for many areas, for instance, industrial control systems (ICS) and IoT, are accessible as well as Software-Defined Networking [141].

7.3 | Real time detection issue

An IDS's performance in a real-world environment is another research problem. Neither approach is built in a real-time setting since most NIDS models are trained and evaluated utilising a public dataset-based control environment network. Therefore, the NIDS generates a lot of false alarms, and it is yet unclear how real-world scenarios with noise values would function. As previously indicated, the majority of them continue to depend

on obsolete datasets for testing. Furthermore, the real-time performance of NIDS is not evaluated utilising a destructive test. Thus, the suggested methodology's most significant difficulty is to be effective in real-time detection [142].

7.4 | Poor evaluation practices

The quantitative test depends on metrics to measure the performance of AI systems. The researchers have proposed traditional metrics to estimate the ability of a NIDS for detecting attacks as accuracy and detection rate only for the multi-classification tasks. Those metrics do not have the benchmarking criteria for evaluating and non-reflecting the experimental validation of the anomaly NIDS [143]. Thus, it can lead to misleading results and inaccurate conclusions about the effectiveness of a model.

7.5 | High computation parameters

Most troubleshooting systems are complex parameters in models that need to optimise time to process data traffic. Few DL-based methodologies are implemented based on GPU technology, while multi-core or Duo core CPUs are more used to implement DL-NIDS methodologies. The long execution time using DL methods causes to overload in the CPU (computation time issue) because a limited memory and low speed of CPU for the large data processing. This trend is not proportional to the DL and continuously increases in the volume of data.

8 | FUTURE TRENDS

Upcoming research in this field has a lot of potential, especially in anomaly and intrusion detection utilising DL and traditional ML methods, which are addressed below:

8.1 | Optimising effectiveness

As noted, the effectiveness of a NIDS still has false positives, and there can be solutions to reduce false alarms. An unbalanced class problem can be solved with an up-to-date and balanced dataset. To balance the dataset, efficient strategies can boost the number of minority attack cases, or cost-sensitive learning can enhance the training model for unusual attacks [144, 145]. Also, an anomaly NIDS should have a mechanism to keep the model rules redefined using the dataset for the scenario of attacks to reduce false alarms. Therefore, a few studies are focused on the nature of data and proposed algorithms to implement their methodologies. Most traffic is a sequential data type that needs further attention to suggest a proper algorithm concerning this type, such as an RNN algorithm or a hybrid DL model using feature learning and traditional ML classifier. These solutions can improve the effectiveness of an anomaly NIDS with few false alarms.

8.2 | Low complexity architecture

In 2025, the number of phones linked to the Internet will exceed 18.25 billion, with a traffic size exceeding 79.4 ZB. The architectures of DL insert extra difficulties for a NIDS to be achieved with acceleration in a real-time system. Therefore, high-performance GPUs are used as one solution for quickly and efficiently processing big datasets to handle such difficulty. These GPUs are generally costly. Furthermore, technological and economic constraints impose tight limitations on data's rapidly increasing volume and complexity. Consequently, we have to trade off the effectiveness and cost, and low-cost GPU platforms could be available to train a DL-NIDS. Another approach to this problem is to use intelligent feature engineering and meta-heuristic techniques to lower the temporal complexity of the DL method. The exemplary architecture of the DL model and some of the hidden layers, their neurons, and optimal weights will yield almost the same great accuracy as if the entire set of features was utilised. Simultaneously, the model's complexity will be reduced and use fewer computing resources in a real-time setting.

8.3 | New threat detection

According to the latest industrial report, most anomaly NIDS models were trained and tested based on commonly used datasets and were proposed by different institutions. According to the Panda report, there are new attacks to threaten the network. Most of the studies are not focused on the detection by the traditional ML or DL NIDS model of new threats and are described as follows:

- Ransomware is malware infecting different networks infrastructure parts, such as media outlets, health care, academia, and industry experts. The last report from the Apex Laboratory [146] shows that ransomware attacks damaged the most significant power utility company, mortgage loan servicing company, and real estate agency in Columbia. It indicates that ransomware attacks injected different domains in network security in many countries. Cybercrime is expected to cost $6 trillion this year (up from $3 trillion in 2015). We anticipate an upsurge in ransomware attacks, with novel versions becoming more disruptive and sophisticated.
- Data-poisoning attack, it could be a novel attack against ML techniques by poisoning the training dataset with a few samples of another type of dataset class. It will lead to building a bad AI predictor besides the prediction class as a false diagnosis.
- Zero-day attacks, remote access controls, and insider attacks in cyber-physical attacks target ICS, including industrial factories' infrastructure. Therefore, ICS might destroy or stop the services of factories as has happened with water distribution and power stations. Nevertheless, research in this stage is still in its infancy, and additional study is needed to discover and build reliable DL-based NIDS for SCADA networks.

8.4 | Improving studies to RL

This domain could explore various avenues to enhance the effectiveness and efficiency of RL-based intrusion detection. We recommend investigating methods to make RL-based IDSs more robust against adversarial attacks. Beside that, we recommend developing new methods to explore hybrid models that combine RL with other ML techniques or rule-based systems. Combining RL with traditional signature-based methods or anomaly detection approaches could lead to more comprehensive and accurate IDSs. There is a need to implement the DL NIDS in dynamic network environments by designing RL models capable of adapting to dynamic changes in network environments. This includes scenarios where the network architecture evolves, new services are added, or the nature of attacks changes over time.

8.5 | Chat GPT-NIDS

There is a lack of NIDS studies focused on addressing security issues in Chat GPT applications to avoid novel attacks. With the release of GPT-4, security experts warned that GPT-4 is as useful for malware as its predecessor. GPT-4's better reasoning and language comprehension abilities, as well as its longform text generation capability, can lead to an increase in sophisticated security threats. Cybercriminals can use the generative AI chatbot, ChatGPT, to generate malicious code, such as data stealing malware. Although OpenAI has improved safety metrics, there is still a risk that GPT-4 could be manipulated by cybercriminals to generate a harmful code. An Israel-based cybersecurity firm warned that GPT-4's capabilities, such as writing code for malware that can collect confidential PDFs and transfer them to remote servers, using the programming language C++, can pose a significant risk.

8.6 | Dark web detection

Darknet provides a user with anonymity, but a service was introduced which allowed someone to host a website on the darknet and remain anonymous. This attracted people who do illegal stuff to sells things without getting caught. Our survey indicates that few researchers have implemented network intrusion detection systems for detecting abnormal activities of users on web darknet. Therefore, there is a need to monitor the activity of users by analysing the flow network packets using anomaly detection techniques.

8.7 | Federated learning approach

6G networks will be able to use higher frequencies than 5G networks. With increasing data volume and IoT devices, the researcher's trend to 6 g or 5 g cellular technology. Federated DL is vital part to train and build network IDS using deep algorithms for helegenous and varity data traffic. Most new data applications are store in data centre based on the 6G network. It is necessary to generate DL models based on different rules detection for protecting centralised data traffic from novel attacks in high speed (real time).

9 | CONCLUSION

This paper gives a thorough analysis of a network IDS relying on DL and traditional ML methodologies, intending to allow fresh scholars to update their knowledge regarding prior cyberattacks utilising a NIDS. The collection of relevant publications in the AI-NIDS study realm is conducted using a systematic methodology. The IDS background theory, as well as its approaches, is first examined. The efficiency of intrusion detection and the complexity of ML/DL techniques are then assessed for each study methodology. According to this study, recent patterns reveal how DL techniques may be utilised to improve the performance of a NIDS with regards to the confusion matrix and detection accuracy. Based on the statistical survey, DL techniques were employed in roughly 52% of the suggested solutions, with DNN and AE accounting for 27% of the total, while RNN, DBN, and CNN were employed less often.

Meanwhile, traditional ML is used in 48% of solutions to propose algorithms for an anomaly NIDS model, with a high of 7% for SVM and less for each DT, RF, and *K*-mean. Some studies use hybrid anomaly IDS methodologies, combining DL for feature extraction with traditional ML for anomaly detection. Most of the studies that used the DL approach have the highest effectiveness compared to the traditional ML methods for anomaly NIDS via the capability to learn essential features by themselves. Furthermore, the study found that 59% of the suggested procedures were verified utilising out-of-date datasets, which are NSL-KDD and KDD CUP 99. In contrast, in 41% of the techniques, a realistic dataset was utilised for training and testing the anomaly model. A high percentage uses an old dataset. This issue must be addressed to fulfil the realistic data with real-time environments to improve the NIDS performance. Moreover, there is a lack of studies focused on the complexity of the DL methodology of the NIDS and advanced methods, such as multi-GPU in cloud computing, which are used to reduce training time [98]. These are effective ways to decrease the complexity of anomaly NIDS models. In the future, we advocate developing a unique, lightweight, and practical DL-NIDS that can identify novel network attacks, for instance, cyber-physical systems, CAN, real time, and ransomware attacks. Furthermore, we aim to use a statistical measure that quantifies the strength of a relationship or difference between variables in a study. The effect size typically provides a standardised way of assessing the practical or substantive significance of the observed effect. It provides a meaningful and interpretable measure of the size or impact of an effect beyond statistical significance. It helps understand the practical importance or relevance of the findings, irrespective of the sample size or statistical significance.

AUTHOR CONTRIBUTIONS

Ziadoon K. Maseer: Conceptualisation; data curation; formal analysis; investigation; resources; validation; writing – original draft; writing – review & editing. **Qusay Kanaan Kadhim**: Data curation; investigation; validation; writing – original draft; writing – review & editing. **Baidaa Al-Bander**: Formal analysis; investigation; validation; writing – original draft; writing – review & editing. **Robiah Yusof**: Investigation; validation; writing – original draft; writing – review & editing. **Abdu Saif**: Investigation; writing – review & editing.

CONFLICT OF INTEREST STATEMENT

All the authors have no competing interests.

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are openly available in references described throughout the paper.

ORCID

Baidaa Al-Bander https://orcid.org/0000-0002-2518-7364

REFERENCES

1. Morgan, S.: 2019 official annual cybercrime report. *Cybersecurity Ventur*, pp. 1–12 (2019)
2. Morgan, S.: The 2020 data attack of data by 2025 Oussama El-Hilali. *arcserve*, pp. 1–5 (2020)
3. Ahmed, S.T., Khadhim, B.J., Kadhim, Q.K.: Cloud services and cloud perspectives: a review. IOP Conf. Ser. Mater. Sci. Eng. 012078 (2021). https://doi.org/10.1088/1757-899X/1090/1/012078
4. Kushida, K.E., Murray, J., Zysman, J.: Diffusing the cloud: cloud computing and implications for public policy. J. Ind. Compet. Trade 11(3), 209–237 (2011). https://doi.org/10.1007/s10842-011-0106-5
5. Berisha, B., Mëziu, E., Shabani, I.: Big data analytics in cloud computing: an overview. J. Cloud Comput. 11(1), 24 (2022). https://doi.org/10.1186/s13677-022-00301-w
6. Tahaei, H., et al.: The rise of traffic classification in IoT networks: a survey. J. Netw. Comput. Appl. 154(December 2019), 102538 (2020). https://doi.org/10.1016/j.jnca.2020.102538
7. Gupta, A., et al.: Prevailing and emerging cyber threats and security practices in IoT-enabled smart grids: a survey. J. Netw. Comput. Appl. 132(2019), 118–148 (2019). https://doi.org/10.1016/j.jnca.2019.01.012
8. Thakkar, A., Lohiya, R.: A Review on Machine Learning and Deep Learning Perspectives of IDS for IoT: Recent Updates, Security Issues, and Challenges, no. 0123456789. Springer Netherlands (2020). https://doi.org/10.1007/s11831-020-09496-0
9. Liang, C., et al.: Intrusion detection system for the internet of things based on blockchain and multi-agent systems. Electronics 9(7), 1–27 (2020). https://doi.org/10.3390/electronics9071120
10. Moustafa, N., Hu, J., Slay, J.: A holistic review of network anomaly detection systems: a comprehensive survey. J. Netw. Comput. Appl. 128(December), 33–55 (2018). https://doi.org/10.1016/j.jnca.2018.12.006
11. Jiang, M., et al.: Text classification based on deep belief network and softmax regression. Neural Comput. Appl. 29(1), 61–70 (2018). https://doi.org/10.1007/s00521-016-2401-x
12. D'Alconzo, A., et al.: A survey on big data for network traffic monitoring and analysis. IEEE Trans. Netw. Serv. Manag. 16(3), 800–813 (2019). https://doi.org/10.1109/TNSM.2019.2933358
13. Hasson, M.A., Hasan, T.M., Maseer, Z.K.: GTA 3D-DLD: greedy training approach for 3D deep learning diagnosis based COVID-19 CT scan. Int. J. Intell. Eng. Syst. 16(1), 173–186 (2023). https://doi.org/10.22266/ijies2023.0228.16
14. Rosário, A.T., Dias, J.C.: Industry 4.0 and marketing: towards an integrated future research agenda. J. Sens. Actuator Netw. 11(3), 30 (2022). https://doi.org/10.3390/jsan11030030
15. Sadikin, F., Van Deursen, T., Kumar, S.: A hybrid Zigbee IoT intrusion detection system using secure and efficient data collection. Internet Things 12, 100306 (2020). https://doi.org/10.1016/j.iot.2020.100306
16. Thakkar, A., Lohiya, R.: Attack classification of imbalanced intrusion data for IoT network using ensemble-learning-based deep neural network. IEEE Internet Things J. 10(13), 11888–11895 (2023). https://doi.org/10.1109/JIOT.2023.3244810
17. Verma, J., Bhandari, A., Singh, G.: A meta-analysis of role of network intrusion detection systems in confronting network attacks. In: Proceedings of 2021 8th International Conference on Computing for Sustainable Global Development INDIACom 2021, February, pp. 506–511 (2021). https://doi.org/10.1109/INDIACom51348.2021.00090
18. Chattopadhyay, M., Sen, R., Gupta, S.: A comprehensive review and meta-analysis on applications of machine learning techniques in intrusion detection. Australas. J. Inf. Syst. 22(1995), 1–27 (2018). https://doi.org/10.3127/ajis.v22i0.1667
19. Arshad, J., et al.: A review of performance, energy and privacy of intrusion detection systems for IoT. Electronics 9(4), 1–24 (2020). https://doi.org/10.3390/electronics9040629
20. Kim, K., Aminanto, M.E.: Deep learning in intrusion detection perspective: overview and further challenges. In: Proceedings of the International Research Conference on Engineering and Technology, pp. 1–12. IEEE, Jakarta (2017). [Online]. http://koasas.kaist.ac.kr/bitstream/10203/214353/1/IRCET16_AM.pdf
21. Niksefat, S., Kaghazgaran, P., Sadeghiyan, B.: Privacy issues in intrusion detection systems: a taxonomy, survey and future directions. Comput. Sci. Rev. 25, 69–78 (2017). https://doi.org/10.1016/j.cosrev.2017.07.001
22. Aldweesh, A., Derhab, A., Emam, A.Z.: Deep learning approaches for anomaly-based intrusion detection systems: a survey, taxonomy, and open issues. Knowl. Base Syst. 189, 105124 (2020). https://doi.org/10.1016/j.knosys.2019.105124
23. Gumusbas, D., et al.: A comprehensive survey of databases and deep learning methods for cybersecurity and intrusion detection systems. IEEE Syst. J. 15(2), 1–15 (2020). https://doi.org/10.1109/jsyst.2020.2992966
24. Chaabouni, N., et al.: Network intrusion detection for IoT security based on learning techniques. IEEE Commun. Surv. Tutorials 21(3), 2671–2701 (2019). https://doi.org/10.1109/COMST.2019.2896380
25. Ahmad, Z., et al.: Network intrusion detection system: a systematic study of machine learning and deep learning approaches. Trans. Emerg. Telecommun. Technol. 32(1), 1–29 (2021). https://doi.org/10.1002/ett.4150
26. Rohit, M., et al.: Intrusion detection techniques in network environment: a systematic review. Wireless Network 27(2), 1269–1285 (2021). https://doi.org/10.1007/s11276-020-02529-3
27. Chou, D., Jiang, M.: A survey on data-driven network intrusion detection. ACM Comput. Surv. 54(9), 182–218 (2021). https://doi.org/10.1145/3472753
28. Hajj, S., et al.: Anomaly-based intrusion detection systems: the requirements, methods, measurements, and datasets. Trans. Emerg. Telecommun. Technol. 32(4) (2021). https://doi.org/10.1002/ett.4240
29. Geeta Singh, N.K., Khare, N.: A survey of intrusion detection from the perspective of intrusion datasets and machine learning techniques. Int. J. Comput. Appl. 44(7), 659–669 (2021). https://doi.org/10.1080/1206212X.2021.1885150
30. Maseno, E.M., Wang, Z., Xing, H.: A systematic review on hybrid intrusion detection system. Secur. Commun. Network. 2022, 23 (2022). https://doi.org/10.1155/2022/9663052
31. Sana, L., et al.: Anomaly detection for cyber internet of things attacks: a systematic review anomaly detection for cyber internet of things attacks. Appl. Artif. Intell. 36(1) (2022). https://doi.org/10.1080/08839514.2022.2137639
32. Bul'ajoul, W., James, A., Pannu, M.: Improving network intrusion detection system performance through quality of service configuration

and parallel technology. J. Comput. Syst. Sci. 81(6), 981–999 (2015). https://doi.org/10.1016/j.jcss.2014.12.012
33. Ahmed, S.T., et al.: Applying the MCMSI for online educational systems using the two-factor authentication. Int. J. Interact. Mob. Technol. 15(13), 162 (2021). https://doi.org/10.3991/ijim.v15i13.23227
34. Aldwairi, T., Perera, D., Novotny, M.A.: An evaluation of the performance of restricted Boltzmann machines as a model for anomaly network intrusion detection. Comput. Network. 144, 111–119 (2018). https://doi.org/10.1016/j.comnet.2018.07.025
35. Pourhabibi, T., et al.: Fraud detection: a systematic literature review of graph-based anomaly detection approaches. Decis. Support Syst. 133(August 2019), 113303 (2020). https://doi.org/10.1016/j.dss.2020.113303
36. Susto, G.A., Terzi, M., Beghi, A.: Anomaly detection approaches for semiconductor manufacturing. Procedia Manuf. 11(June 2017), 2018–2024 (2017). https://doi.org/10.1016/j.promfg.2017.07.353
37. Ten, C.W., Hong, J., Liu, C.C.: Anomaly detection for cybersecurity of the substations. IEEE Trans. Smart Grid 2(4), 865–873 (2011). https://doi.org/10.1109/TSG.2011.2159406
38. Tang, Z., et al.: Convolutional neural network-based data anomaly detection method using multiple information for structural health monitoring. Struct. Control Health Monit. 26(1), e2296 (2019). https://doi.org/10.1002/stc.2296
39. Chatterjee, A., Ahmed, B.S.: IoT anomaly detection methods and applications: a survey. Internet Things 19(July), 100568 (2022). https://doi.org/10.1016/j.iot.2022.100568
40. Kanaan, Q., Mahdi, H.S., Ail, H.K.: Storage architecture for network security in cloud computing. Diyala J. Pure Sci. 14(1), 1–17 (2018). https://doi.org/10.24237/djps.1401.205c
41. Kadhim, Q.K., et al.: The effectiveness of random early detection in data center transmission control protocol - based cloud computing networks. Int. J. Commun. Antenna Propag. 7(October), 1–7 (2017). https://doi.org/10.15866/irecap.v7i5.10104
42. Ortega-Fernandez, I., et al.: Network intrusion detection system for DDoS attacks in ICS using deep autoencoders. Wireless Network 3 (2023). https://doi.org/10.1007/s11276-022-03214-3
43. Thakkar, A., Lohiya, R.: A survey on intrusion detection system: feature selection, model, performance measures, application perspective, challenges, and future research directions. Artif. Intell. Rev. 55(1), 453–563 (2022). https://doi.org/10.1007/s10462-021-10037-9
44. Abdulla, S.M., Al-Dabagh, N.B., Zakaria, O.: Identify features and parameters to devise an accurate intrusion detection system using artificial neural network. World Acad. Sci. Eng. Technol. 46(10), 626–630 (2010)
45. Akashdeep, I.M., Kumar, N.: A feature reduced intrusion detection system using ANN classifier. Expert Syst. Appl. 88, 249–257 (2017). https://doi.org/10.1016/j.eswa.2017.07.005
46. Bindra, N., Sood, M.: Detecting DDoS attacks using machine learning techniques and contemporary intrusion detection dataset. Automat. Control Comput. Sci. 53(5), 419–428 (2019). https://doi.org/10.3103/S0146411619050043
47. Alzahrani, A.O., Alenazi, M.J.F.: Designing a network intrusion detection system based on machine learning for software defined networks. Future Internet 13(5), 111 (2021). https://doi.org/10.3390/fi13050111
48. Dias, L.P., et al.: Using artificial neural network in intrusion detection systems to computer networks. In: 2017 9th Computer Science and Electronic Engineering Conference CEEC 2017 - Proceeding, pp. 145–150 (2017). https://doi.org/10.1109/CEEC.2017.8101615
49. Kadhim, Q.K., et al.: COVID-19 disease diagnosis using artificial intelligence based on gene expression: a review. Sumer J. Pure Sci. 2(2), 88–102 (2023)
50. Yap, B.W., et al.: An application of oversampling, undersampling, bagging and boosting in handling imbalanced datasets. Lect. Notes Electr. Eng. 285, 13–22 (2014). https://doi.org/10.1007/978-981-4585-18-7_2
51. Bhargava, N., Sharma, G.: Decision tree analysis on J48 algorithm for data mining. Int. J. Adv. Res. Comput. Sci. Software Eng. 3(6), 1114–1119 (2013)
52. Cover, T., Hart, P.: Nearest neighbor pattern classification. IEEE Trans. Inf. Theory. 13(1), 21–27 (1967)
53. Kaur, D.: A comparative study of various distance measures for software fault prediction. Int. J. Comput. Trends Technol. 17(3), 117–120 (2014). https://doi.org/10.14445/22312803/ijctt-v17p122
54. Rajbahadur, G.K., et al.: The impact of feature importance methods on the interpretation of defect classifiers. IEEE Trans. Software Eng. 48(7), 2245–2261 (2022). https://doi.org/10.1109/TSE.2021.3056941
55. Farnaaz, N., Jabbar, M.A.: Random forest modeling for network intrusion detection system. In: Procedia Computer Science, pp. 213–217. Elsevier (2016). https://doi.org/10.1016/j.procs.2016.06.047
56. Xue, H., Chen, S., Yang, Q.: Structural support vector machine. In: Advances in Neural Networks, Lecture Notes in Computer Science, pp. 501–511. Springer-Verlag Berlin Heidelberg (2008). https://doi.org/10.1007/978-3-540-87732-5_56
57. Nazeeh, I., et al.: Optimizing blockchain technology using a data sharing model. Indones. J. Electr. Eng. Comput. Sci. 29(1), 431 (2023). https://doi.org/10.11591/ijeecs.v29.i1.pp431-440
58. Hameed, E.M., et al.: Liver disease detection and prediction using SVM techniques. In: 2022 3rd Information Technology to Enhance E-Learning and Other Application (IT-ELA), pp. 61–66. IEEE (2022). https://doi.org/10.1109/IT-ELA57378.2022.10107961
59. Aburomman, A.A., Bin Ibne Reaz, M.: A novel SVM-kNN-PSO ensemble method for intrusion detection system. Appl. Soft Comput. J. 38, 360–372 (2016). https://doi.org/10.1016/j.asoc.2015.10.011
60. Praveen, P., Rama, B.: A k-means clustering algorithm on numeric data. Int. J. Pure Appl. Math. 117(7), 157–164 (2017)
61. hussian Hassan, A.A., et al.: Evaluate the performance of K-means and the fuzzy C-means algorithms to formation balanced clusters in wireless sensor networks. Int. J. Electr. Comput. Eng. 10(2), 1515–1523 (2020). https://doi.org/10.11591/ijece.v10i2.pp1515-1523
62. Moon, T.K.: The expectation maximization algorithm. IEEE Signal Process. Mag. 13(6), 47–60 (1996). https://doi.org/10.1109/79.543975
63. Kohonen, T.: The self-organizing map. Neurocomputing 21(May), 1–6 (1998). https://doi.org/10.1016/s0925-2312(98)00030-7
64. Vinayakumar, R., et al.: Deep learning approach for intelligent intrusion detection system. IEEE Access 7(c), 41525–41550 (2019). https://doi.org/10.1109/ACCESS.2019.2895334
65. Zeinali, Y., Story, B.A.: Competitive probabilistic neural network. Integrated Comput. Aided Eng. 24(2), 105–118 (2017). https://doi.org/10.3233/ICA-170540
66. Mohebali, B., et al.: Probabilistic neural networks: a brief overview of theory, implementation, and application. Handb. Probabilistic Model 4(3), 347–367 (2019). https://doi.org/10.1016/B978-0-12-816514-0.00014-X
67. Mohammed, H.A.A., et al.: Anomaly detection in human disease: a hybrid approach using GWO-SVM for gene selection. Rev. Intelligence Artif. 37(4), 913–919 (2023). https://doi.org/10.18280/ria.370411
68. Tao, J., Liu, Y., Yang, D.: Bearing fault diagnosis based on deep belief network and multisensor information fusion. Shock Vib. 2016, 1–9 (2016). https://doi.org/10.1155/2016/9306205
69. Khadhim, B.J., et al.: Diagnose COVID-19 by using hybrid CNN-RNN for chest X-ray. Indones. J. Electr. Eng. Comput. Sci. 29(2), 852–860 (2023). https://doi.org/10.11591/ijeecs.v29.i2.pp852-860
70. Sherstinsky, A.: Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network. Phys. D Nonlinear Phenom. 404(March), 1–43 (2020). https://doi.org/10.1016/j.physd.2019.132306
71. Yin, C., et al.: A deep learning approach for intrusion detection using recurrent neural networks. IEEE Access 5, 21954–21961 (2017). https://doi.org/10.1109/ACCESS.2017.2762418
72. Xu, C., et al.: An intrusion detection system using a deep neural network with gated recurrent units. IEEE Access 6, 48697–48707 (2018). https://doi.org/10.1109/ACCESS.2018.2867564
73. Manickam, R., Ramu, K.: Understanding long short-term memory LSTM models in IBM SPSS statistics. J. Innov. Teach. Learn. 2(March), 19–28 (2023). https://doi.org/10.46632/jitl/2/1/3
74. Khadhim, B.J., et al.: Virtualization in mobile cloud computing for augmented reality challenges. In: Proceedings of 2021 2nd Information Technology to Enhance e-Learning and Other Application Conference

IT-ELA 2021, June 2022, pp. 113–118 (2021). https://doi.org/10.1109/IT-ELA52201.2021.9773680
75. Alom, Z., Bontupalli, V., Taha, T.M.: Intrusion detection using deep belief networks. In: 2015 National Aerospace and Electronics Conference (NAECON), pp. 339–344. IEEE, Dayton (2015)
76. Al-Qatf, M., et al.: Deep learning approach combining sparse autoencoder with SVM for network intrusion detection. IEEE Access 6(c), 52843–52856 (2018). https://doi.org/10.1109/ACCESS.2018.2869577
77. SuperDataScience Team: The ultimate guide to convolutional neural networks (CNN). Super Data Science. [Online]. https://www.superdatascience.com/blogs/the-ultimate-guide-to-convolutional-neural-networks-cnn
78. Alsultani, H.S.M., et al.: The use of spatial relationships and object identification in image understanding. Int. J. Civ. Eng. Technol. 9(5), 487–496 (2018)
79. Kadhim, Q.K., Al-nedawe, B.M., Hameed, E.M.: Encryption and decryption of images using GGH algorithm: proposed encryption and decryption of images using GGH algorithm: proposed. In: IOP Conference Series: Materials Science and Engineering, p. 012063 (2021). https://doi.org/10.1088/1757-899X/1090/1/012063
80. Das, S.: CNN architectures: LeNet, AlexNet, VGG, GoogLeNet, ResNet and more…. Analytics Vidhya. [Online]. https://medium.com/analytics-vidhya/cnns-architectures-lenet-alexnet-vgg-googlenet-resnet-and-more-666091488df5
81. Dong, S., Xia, Y., Peng, T.: Traffic identification model based on generative adversarial deep convolutional network. Ann. Telecommun. 77(9–10), 573–587 (2022). https://doi.org/10.1007/s12243-021-00876-6
82. Dong, S., Xia, Y., Peng, T.: Network abnormal traffic detection model based on semi-supervised deep reinforcement learning. IEEE Trans. Netw. Serv. Manag. 18(4), 4197–4212 (2021). https://doi.org/10.1109/TNSM.2021.3120804
83. Kadhim, Q.K., Altameemi, A., Jasim, S.: Artificial intelligence techniques for colon cancer detection: a review. J. Yarmouk 21(2), 11–18 (2023)
84. Xia, Y., et al.: Wireless network abnormal traffic detection method based on deep transfer reinforcement learning. In: 2021 17th International Conference on Mobility, Sensing and Networking (MSN), pp. 528–535. IEEE (2021). https://doi.org/10.1109/MSN53354.2021.00083
85. Eldos, T., Siddiqui, M., Kanan, A.: On the KDD'99 dataset: statistical analysis for feature selection. J. Data Min. Knowl. Discov. 3(3), 88–90 (2012). [Online]. http://www.bioinfo.in/uploadfiles/13470975543_3_1_JDMKD.pdf
86. Kadhim, Q.K.: Classification of human skin diseases using data mining. Int. J. Adv. Eng. Res. Sci. 4(1), 159–163 (2017). https://doi.org/10.22161/ijaers.4.1.25
87. Oleiwi, H.W., Mhawi, D.N., Al-Raweshidy, H.: MLTs-ADCNs: machine learning techniques for anomaly detection in communication networks. IEEE Access 10, 91006–91017 (2022). https://doi.org/10.1109/ACCESS.2022.3201869
88. Ali, H.A., Kanaan, Q., Sadeq, H.: Evaluation of routing protocols on ad hoc network modelling from medical data using OpNet simulation. Diyala J. Pure Sci. 13(2), 33–48 (2017). https://doi.org/10.24237/djps.1302.204c
89. Gowdhaman, V., Dhanapal, R.: An intrusion detection system for wireless sensor networks using deep neural network an intrusion detection system for wireless sensor networks using deep neural network. Soft Comput. 7(December), 1–55 (2021). https://doi.org/10.1007/s00500-021-06473-y
90. Sharafaldin, I., Habibi Lashkari, A., Ghorbani, A.A.: A Detailed Analysis of the CICIDS2017 Data Set, no. Cic. Springer International Publishing (2019). https://doi.org/10.1007/978-3-030-25109-3_9
91. and Biomechanics, A.B.: Retracted: adaptive anomaly detection framework model objects in cyberspace. Appl. Bionics Biomech. 2023, 1–14 (2023). https://doi.org/10.1155/2023/9819236
92. Alsultani, H.S.M., Kanaan, Q., Khudhair, I.Y.: Empirical investigation of TCP incast congestion in wireless cloud computing networks. J. Comput. Sci. 14(5), 663–672 (2018). https://doi.org/10.3844/jcssp.2018.663.672
93. Iman Sharafaldin, A.A.G., Habibi Lashkari, A.: CSE-CIC-IDS2018 on AWS. Canadian Institute for Cybersecurity. [Online]. https://www.unb.ca/cic/datasets/ids-2018.html
94. Ferrag, M.A., et al.: Deep learning for cyber security intrusion detection: approaches, datasets, and comparative study. J. Inf. Secur. Appl. 50, 102419 (2020). https://doi.org/10.1016/j.jisa.2019.102419
95. Kadhim, Q.K., Yusof, R., Selamat, S.R.: The cloud computing control in the government services. J. Adv Res. Dyn. Control Syst. 10(04), 1136–1147 (2018)
96. Moustafa, N., et al.: Federated TON_IoT windows datasets for evaluating AI-based security applications. In: Proceedings of - 2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications Trust. 2020, pp. 848–855 (2020). https://doi.org/10.1109/TrustCom50675.2020.00114
97. Ahmed, H.A.S., et al.: A review of challenges and security risks of cloud computing. J. Telecommun. Electron. Comput. Eng. 9(1–2), 87–91 (2017)
98. Kanaan Kadhim, Q., et al.: A review study on cloud computing issues. In: Journal of Physics: Conference Series, Institute of Physics Publishing (2018). https://doi.org/10.1088/1742-6596/1018/1/012006
99. Hussain, J., Lalmuanawma, S., Chhakchhuak, L.: A two-stage hybrid classification technique for network intrusion detection system. Int. J. Comput. Intell. Syst. 9(5), 863–875 (2016). https://doi.org/10.1080/18756891.2016.1237186
100. Mohammadi, S., Mirvaziri, H., Ghazizadeh-ahsaee, M.: Multivariate correlation coefficient and mutual information-based feature selection in intrusion detection. Inf. Secur. J. 00(00), 1–11 (2017). https://doi.org/10.1080/19393555.2017.1358779
101. Yao, H., et al.: MSML: a novel multi-level semi-supervised machine learning framework for intrusion detection system. IEEE Internet Things J. 6(2), 1–10 (2018)
102. Thamilarasu, G., Chawla, S.: Towards deep-learning-driven intrusion detection for the internet of things. Sensors 19(9) (2019). https://doi.org/10.3390/s19091977
103. Otoum, S., Kantarci, B., Mouftah, H.T.: On the feasibility of deep learning in sensor network intrusion detection. IEEE Netw. Lett. PP(1), 1 (2019). https://doi.org/10.1109/LNET.2019.2901792
104. Reddy, D.K.K., et al.: Ensemble bagging approach for IoT sensor based anomaly detection. Lect. Notes Electr. Eng. 702, 647–665 (2021). https://doi.org/10.1007/978-981-15-8439-8_52
105. Subba, B., Biswas, S., Karmakar, S.: Enhancing performance of anomaly based intrusion detection systems through dimensionality reduction using principal component analysis. In: 2016 IEEE International Conference on Advanced Networks and Telecommunications Systems, pp. 1–6 (2016). https://doi.org/10.1109/ANTS.2016.7947776
106. Javaid, A., et al.: A deep learning approach for network intrusion detection system. In: Proceedings of 9th EAI International Conference on Bio-inspired Information and Communications Technologies (Formerly BIONETICS), December 2015, pp. 21–26 (2016). https://doi.org/10.4108/eai.3-12-2015.2262516
107. Peng, K., et al.: Intrusion detection system based on decision tree over big data in fog environment. Wireless Commun. Mobile Comput. 2018, 4680867 (2018)
108. Chakrabarty, B., Chanda, O., Saiful, M.: Anomaly based intrusion detection system using genetic algorithm and K-centroid clustering. Int. J. Comput. Appl. 163(11), 13–17 (2017). https://doi.org/10.5120/ijca2017913762
109. Ashfaq, R.A.R., et al.: Fuzziness based semi-supervised learning approach for intrusion detection system. Inf. Sci. 378, 484–497 (2017). https://doi.org/10.1016/j.ins.2016.04.019
110. Belouch, M., El, S., Idhammad, M.: A two-stage classifier approach using RepTree algorithm for network intrusion detection. Int. J. Adv. Comput. Sci. Appl. 8(6), 389–394 (2017). https://doi.org/10.14569/ijacsa.2017.080651
111. Laghrissi, F.E., et al.: Intrusion detection systems using long short-term memory (LSTM). J. Big Data 8(1), 1–19 (2021). https://doi.org/10.1186/s40537-021-00448-4

112. Hussien, Z.K., Dhannoon, B.N.: Anomaly detection approach based on deep neural network and dropout. Baghdad Sci. J. 17(2), 701–709 (2020). https://doi.org/10.21123/bsj.2020.17.2(SI).0701
113. Chahar, V., et al.: Significance of hybrid feature selection technique for intrusion detection systems. Indian J. Sci. Technol. 9(48), 1–7 (2016). https://doi.org/10.17485/ijst/2016/v9i48/105827
114. Hijazi, A., El Safadi, E.A., Flaus, J.M.: A deep learning approach for intrusion detection system in industry network. CEUR Workshop Proc. 2343(December), 55–62 (2018)
115. Abusitta, A., et al.: A deep learning approach for proactive multi-cloud cooperative intrusion detection system. Future Generat. Comput. Syst. 98, 308–318 (2019). https://doi.org/10.1016/j.future.2019.03.043
116. Hassan, M.M., et al.: A hybrid deep learning model for efficient intrusion detection in big data environment. Inf. Sci. 513, 386–396 (2020). https://doi.org/10.1016/j.ins.2019.10.069
117. Zhao, G., Zhang, C., Zheng, L.: Intrusion detection using deep belief network and probabilistic neural network. In: 2017 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC), pp. 639–642. IEEE, Guangzhou, China (2017). https://doi.org/10.1109/CSE-EUC.2017.119
118. Zhang, H., et al.: Multi-dimensional feature fusion and stacking ensemble mechanism for network intrusion detection. Future Generat. Comput. Syst. 122, 130–143 (2021). https://doi.org/10.1016/j.future.2021.03.024
119. Atefi, K., Hashim, H., Khodadadi, T.: A hybrid anomaly classification with deep learning (DL) and binary algorithms (BA) as optimizer in the intrusion detection system (IDS). In: Proceedings - 2020 16th IEEE International Colloquium on Signal Processing and its Applications, CSPA 2020, pp. 29–34. IEEE, Langkawi, Malaysia A (2020). https://doi.org/10.1109/CSPA48992.2020.9068725
120. Dao, T.N., Lee, H.J.: Stacked autoencoder-based probabilistic feature extraction for on-device network intrusion detection. IEEE Internet Things J. 9(16), 14438–14451 (2021). https://doi.org/10.1109/JIOT.2021.3078292
121. Chatzoglou, E., et al.: Pick quality over quantity: expert feature selection and data preprocessing for 802.11 intrusion detection systems. IEEE Access 10, 64761–64784 (2022). https://doi.org/10.1109/ACCESS.2022.3183597
122. Sethi, K., et al.: A context-aware robust intrusion detection system: a reinforcement learning-based approach. Int. J. Inf. Secur. (1) (2019). https://doi.org/10.1007/s10207-019-00482-7
123. Mayuranathan, M., Murugan, M., Dhanakoti, V.: Best features based intrusion detection system by RBM model for detecting DDoS in cloud environment. J. Ambient Intell. Hum. Comput. 12(2021), 3609–3619 (2019). https://doi.org/10.1007/s12652-019-01611-9
124. Haider, S., et al.: A deep CNN ensemble framework for efficient DDoS attack detection in software defined networks. IEEE Access 8, 53972–53983 (2020). https://doi.org/10.1109/ACCESS.2020.2976908
125. He, H., et al.: A novel multimodal-sequential approach based on multi-view features for network intrusion detection. IEEE Access 7, 183207–183221 (2019). https://doi.org/10.1109/ACCESS.2019.2959131
126. Ma, C., Du, X., Cao, L.: Analysis of multi-types of flow features based on hybrid neural network for improving network anomaly detection. IEEE Access 7, 148363–148380 (2019). https://doi.org/10.1109/ACCESS.2019.2946708
127. Das and Pramod, A.: Exploratory analysis on anomaly-based IDS data using DASK and ensemble learning: a data parallelization approach. Int. J. Eng. Trends Technol. 70(12), 370–391 (2022). https://doi.org/10.14445/22315381/IJETT-V70I12P236
128. Imanbayev, A., et al.: Research of machine learning algorithms for the development of intrusion detection systems in 5G mobile networks and beyond. Sensors 22(24), 1–29 (2022). https://doi.org/10.3390/s22249957
129. Rahma, A.M., Abbas, A.: A modified matrices approach in advanced encryption standard algorithm. Eng. Technol. J. 37(3B), 86–91 (2019). https://doi.org/10.30684/etj.37.3b.4
130. Mohammed, B., Gbashi, E.: Intrusion detection system for NSL-KDD dataset based on deep learning and recursive feature elimination. Eng. Technol. J. 39(7), 1069–1079 (2021). https://doi.org/10.30684/etj.v39i7.1695
131. Guezzaz, A., et al.: A lightweight hybrid intrusion detection framework using machine learning for edge-based IIoT security. Int. Arab J. Inf. Technol. 19(5), 822–830 (2022). https://doi.org/10.34028/iajit/19/5/14
132. Almomani, A., et al.: Ensemble-based approach for efficient intrusion detection in network traffic. Intell. Autom. Soft Comput. 37(2), 2499–2517 (2023). https://doi.org/10.32604/iasc.2023.039687
133. Ring, M., et al.: A survey of network-based intrusion detection data sets. Comput. Secur. 86, 147–167 (2019). https://doi.org/10.1016/j.cose.2019.06.005
134. Ma, W., Gou, C., Hou, Y.: Research on adaptive 1DCNN network intrusion detection technology based on BSGM mixed sampling. Sensors 23(13) (2023). https://doi.org/10.3390/s23136206
135. Hassan, D.: Cost-sensitive access control for detecting remote to local (R2L) and user to root (U2R) attacks. Int. J. Comput. Trends Technol. 43(2), 124–129 (2017). https://doi.org/10.14445/22312803/ijctt-v43p118
136. Saeed, M.M., et al.: Anomaly detection in 6G networks using machine learning methods. Electronics 12(15) (2023). https://doi.org/10.3390/electronics12153300
137. Dhanya, K.A., et al.: Detection of network attacks using machine learning and deep learning models. Procedia Comput. Sci. 218, 57–66 (2023). https://doi.org/10.1016/j.procs.2022.12.401
138. Rihan, S.D.A., Anbar, M., Alabsi, B.A.: Approach for detecting attacks on IoT networks based on ensemble feature selection and deep learning models. Sensors 23(17), 7342 (2023). https://doi.org/10.3390/s23177342
139. Vanin, P., et al.: A study of network intrusion detection systems using artificial intelligence/machine learning. Appl. Sci. 12(22) (2022). https://doi.org/10.3390/app122211752
140. Rodríguez, M., et al.: Evaluation of machine learning techniques for traffic flow-based intrusion detection. Sensors 22(23) (2022). https://doi.org/10.3390/s22239326
141. Bahalul Haque, A.K.M., et al.: Attacks and countermeasures in IoT based smart healthcare applications, pp. 67–90 (2022). https://doi.org/10.1007/978-3-030-90119-6_6
142. Naeem, H., Ahmad, J., Tayyab, M.: Real-time object detection and tracking. In: 2013 16th International Multi-Topic Conference INMIC 2013, January 2021, pp. 148–153 (2013). https://doi.org/10.1109/INMIC.2013.6731341
143. Pham-Quoc, C., Bao, T.H.Q., Thinh, T.N.: FPGA/AI-powered architecture for anomaly network intrusion detection systems. Electronics 12(3), 668 (2023). https://doi.org/10.3390/electronics12030668
144. Abdelkhalek, A., Mashaly, M.: Addressing the class imbalance problem in network intrusion detection systems using data resampling and deep learning. J. Supercomput. 79(10), 10611–10644 (2023). https://doi.org/10.1007/s11227-023-05073-x
145. Bedi, P., Gupta, N., Jindal, V.: Siam-IDS: handling class imbalance problem in intrusion detection systems using siamese neural network. Procedia Comput. Sci. 171, 780–789 (2020). https://doi.org/10.1016/j.procs.2020.04.085
146. Malik, N., et al.: Recent advances in cyber security laws and practices in India, pp. 220–241 (2023). https://doi.org/10.4018/978-1-6684-8133-2.ch012
147. Masarat, S., Sharifian, S., Taheri, H.: Modified parallel random forest for intrusion detection systems. J. Supercomput. 72(6), 2235–2258 (2016). https://doi.org/10.1007/s11227-016-1727-6
148. Hasan, M.A.M., et al.: Feature selection for intrusion detection using random forest. J. Inf. Secur. 07(03), 129–140 (2016). https://doi.org/10.4236/jis.2016.73009
149. Sedghi, S., Mirnia, M.: Integration bat algorithm with k-means for intrusion detection system. Int. J. Comput. Sci. Netw. Secur. 17(7), 315–319 (2017)

150. Natesan, P., et al.: Hadoop based parallel binary bat algorithm for network intrusion detection. Int. J. Parallel Program. 45(5), 1194–1213 (2017). https://doi.org/10.1007/s10766-016-0456-z
151. Ikram, S.T., Cherukuri, A.K.: Improving accuracy of intrusion detection model using PCA and optimized SVM. J. Comput. Inf. Technol. 24(2), 133–148 (2016). https://doi.org/10.20532/cit.2016.1002701
152. Gurung, S., Kanti Ghose, M., Subedi, A.: Deep learning approach on network intrusion detection system using NSL-KDD dataset. Int. J. Comput. Netw. Inf. Secur. 11(3), 8–14 (2019). https://doi.org/10.5815/ijcnis.2019.03.02
153. Wang, C.R., et al.: Network intrusion detection using equality constrained-optimization-based extreme learning machines. Knowl. Base Syst. 147, 68–80 (2018). https://doi.org/10.1016/j.knosys.2018.02.015
154. Moustafa, N., Slay, J., Creech, G.: Novel geometric area analysis technique for anomaly detection using trapezoidal area estimation on large-scale networks. IEEE Trans. Big Data 7790(c), 1 (2017). https://doi.org/10.1109/tbdata.2017.2715166
155. Khammassi, C., Krichen, S.: A GA-LR wrapper approach for feature selection in network intrusion detection ☆. Comput. Secur. (October) (2017). https://doi.org/10.1016/j.cose.2017.06.005
156. Verma, A., Ranga, V.: On evaluation of network intrusion detection systems: statistical analysis of CIDDS-001 dataset using machine learning techniques. Pertanika J. Sci. Technol. 26(3), 1307–1332 (2018)
157. Carrasco, R.S.M., Sicilia, M.A.: Unsupervised intrusion detection through skip-gram models of network behavior. Comput. Secur. 78, 187–197 (2018). https://doi.org/10.1016/j.cose.2018.07.003
158. Idhammad, M., Afdel, K., Belouch, M.: Semi-supervised machine learning approach for DDoS detection. Appl. Intell. 48(10), 3193–3208 (2018). https://doi.org/10.1007/s10489-018-1141-2
159. Gauthama Raman, M.R., et al.: An Efficient Intrusion Detection Technique Based on Support Vector Machine and Improved Binary Gravitational Search Algorithm, no. 0123456789. Springer Netherlands (2019). https://doi.org/10.1007/s10462-019-09762-z
160. He, W., Li, H., Li, J.: Ensemble feature selection for improving intrusion detection classification accuracy. In: ACM International Conference Proceedings Series, pp. 28–33 (2019). https://doi.org/10.1145/3349341.3349364
161. Papamartzivanos, D., Gómez Mármol, F., Kambourakis, G.: Dendron: genetic trees driven rule induction for network intrusion detection systems. Future Generat. Comput. Syst. 79, 558–574 (2018). https://doi.org/10.1016/j.future.2017.09.056
162. Kolte, P.M.: Performance analysis of intrusion detection system utilizing deep learning techniques. J. Gujarat Res. Soc. 21(10), 1358–1366 (2019)
163. Tama, B.A., Rhee, K.H.: An in-depth experimental study of anomaly detection using gradient boosted machine. Neural Comput. Appl. 31(4), 955–965 (2019). https://doi.org/10.1007/s00521-017-3128-z
164. Ren, J., et al.: Building an effective intrusion detection system by using hybrid data optimization based on machine learning algorithms. Secur. Commun. Network. 2019 (2019). https://doi.org/10.1155/2019/7130868
165. Al-Zewairi, M., Almajali, S., Awajan, A.: Experimental evaluation of a multi-layer feed-forward artificial neural network classifier for network intrusion detection system. In: Proceedings - 2017 International Conference on New Trends in Computing Sciences, ICTCS 2017, pp. 167–172. IEEE, Amman (2017). https://doi.org/10.1109/ICTCS.2017.29
166. Cepheli, Ö., Büyükçorak, S., Kurt, G.G.K.: Hybrid intrusion detection system for DDoS attacks. J. Electr. Comput. Eng. 2016, 8 (2016). [Online]. https://doi.org/10.1155/2016/1075648
167. Wang, G., Chen, J., Yang, L.T.: Effectiveness of Machine Learning Based Intrusion Detection Systems, vol. 1. Springer Nature Switzerland (2018). https://doi.org/10.1007/978-3-030-24907-6
168. Lee, J.H., Park, K.H.: GAN-based imbalanced data intrusion detection system. Personal Ubiquitous Comput. 25(12), 121–128 (2019). https://doi.org/10.1007/s00779-019-01332-y
169. Hosseini, S., Seilani, H.: Anomaly process detection using negative selection algorithm and classification techniques. Evol. Syst. (0123456789) (2019). https://doi.org/10.1007/s12530-019-09317-1
170. Lee, J., et al.: Cyber threat detection based on artificial neural networks using event profiles. IEEE Access 7, 165607–165626 (2019)
171. Ferrag, M.A., Maglaras, L.: DeepCoin: a novel deep learning and blockchain-based energy exchange framework for smart grids. IEEE Trans. Eng. Manag. 67(4), 1285–1297 (2019). https://doi.org/10.1109/TEM.2019.2922936
172. Maseer, Z.K., et al.: Benchmarking of machine learning for anomaly based intrusion detection systems in the CICIDS2017 dataset. IEEE Access 9, 22351–22370 (2021). https://doi.org/10.1109/access.2021.3056614
173. Fernández, G.C., Xu, S.: A case study on using deep learning for network intrusion detection. In: MILCOM 2019-2019 IEEE Military Communications Conference (MILCOM), pp. 1–6. IEEE, Norfolk (2019). https://doi.org/10.1109/MILCOM47813.2019.9020824
174. Riyaz, B., Ganapathy, S.: A deep learning approach for effective intrusion detection in wireless networks using CNN. Soft Comput. 24(22), 17265–17278 (2020). https://doi.org/10.1007/s00500-020-05017-0
175. Roy, B., Cheung, H.: A deep learning approach for intrusion detection in internet of things using bi-directional long short-term memory recurrent neural network. In: 2018 28th International Telecommunication Networks and Applications Conference ITNAC 2018, pp. 1–6 (2019). https://doi.org/10.1109/ATNAC.2018.8615294
176. Zhang, C., et al.: A deep learning approach for network intrusion detection based on NSL-KDD dataset. In: 2019 IEEE 13th International Conference on Anti-counterfeiting, Security, and Identification (ASID), pp. 41–45. IEEE, Xiamen, China (2019). https://doi.org/10.1109/ICASID.2019.8925239
177. Shone, N., et al.: A deep learning approach to network intrusion detection. IEEE Trans. Emerg. Top. Comput. Intell. 2(1), 41–50 (2018). https://doi.org/10.1109/tetci.2017.2772792
178. Kasongo, S.M., Sun, Y.: A deep learning method with filter based feature engineering for wireless intrusion detection system. IEEE Access 7, 38597–38607 (2019). https://doi.org/10.1109/ACCESS.2019.2905633
179. Kasongo, S.M., Sun, Y.: A deep learning method with wrapper based feature extraction for wireless intrusion detection system. Comput. Secur. 92 (2020). https://doi.org/10.1016/j.cose.2020.101752
180. Louati, F., Ktata, F.B.: A deep learning-based multi-agent system for intrusion detection. SN Appl. Sci. 2(4) (2020). https://doi.org/10.1007/s42452-020-2414-z
181. Hsu, Y.F., Matsuoka, M.: A deep reinforcement learning approach for anomaly network intrusion detection system. In: 2020 IEEE 9th International Conference on Cloud Networking (CloudNet), pp. 1–6. IEEE, Piscataway, NJ, USA (2020). https://doi.org/10.1109/CloudNet51028.2020.9335796
182. Mighan, S.N., Kahani, M.: A novel scalable intrusion detection system based on deep learning. Int. J. Inf. Secur. 20, 387–403 (2020). https://doi.org/10.1007/s10207-020-00508-5
183. Yang, K., et al.: Adversarial examples against the deep learning based network intrusion detection systems. In: Proceedings - IEEE Military Communications Conference MILCOM, vol. 2019-Octob, pp. 559–564 (2019). https://doi.org/10.1109/MILCOM.2018.8599759
184. Kim, A., Park, M., Lee, D.H.: AI-IDS: application of deep learning to real-time web intrusion detection. IEEE Access 8, 70245–70261 (2020). https://doi.org/10.1109/ACCESS.2020.2986882
185. Zhang, H., et al.: An effective deep learning based scheme for network intrusion detection. Proc. - Int. Conf. Pattern Recognit. 2018-Augus, 682–687 (2018). https://doi.org/10.1109/ICPR.2018.8546162
186. Kumar, P., et al.: Analysis of intrusion detection in cyber attacks using DEEP learning neural networks. Peer Peer Netw. Appl. (2020). https://doi.org/10.1007/s12083-020-00999-y
187. Ibitoye, O., Shafiq, O., Matrawy, A.: Analyzing adversarial attacks against deep learning for intrusion detection in IoT networks. arXiv (2019)
188. Lopez-Martin, M., Carro, B., Sanchez-Esguevillas, A.: Application of deep reinforcement learning to intrusion detection for supervised problems. Expert Syst. Appl. 141, 112963 (2020). https://doi.org/10.1016/j.eswa.2019.112963
189. Loukas, G., et al.: Cloud-based cyber-physical intrusion detection for vehicles using deep learning. IEEE Access 6, 3491–3508 (2017). https://doi.org/10.1109/ACCESS.2017.2782159
190. Shu, J., et al.: Collaborative intrusion detection for VANETs: a deep learning-based distributed SDN approach. IEEE Trans. Intell. Transport. Syst., 1–12 (2020). https://doi.org/10.1109/tits.2020.3027390
191. Yang, H., Qin, G., Ye, L.: Combined wireless network intrusion detection model based on deep learning. IEEE Access 7, 82624–82632 (2019). https://doi.org/10.1109/ACCESS.2019.2923814

192. Abdulhammed, R., et al.: Deep and machine learning approaches for anomaly-based intrusion detection of imbalanced network traffic. IEEE Sensors Lett. 3(1), 2018–2021 (2019). https://doi.org/10.1109/LSENS.2018.2879990
193. Alom, M.Z., Taha, T.M.: Network intrusion detection for cyber security using unsupervised deep learning approaches. In: Proceedings of the International Joint Conference on Neural Networks, pp. 3830–3837. IEEE, Dayton (2017). https://doi.org/10.1109/IJCNN.2017.7966339
194. Ieracitano, C., et al.: A novel statistical analysis and autoencoder driven intelligent intrusion detection approach. Neurocomputing 387, 51–62 (2020). https://doi.org/10.1016/j.neucom.2019.11.016
195. Zong, W., Chow, Y.W., Susilo, W.: Interactive three-dimensional visualization of network intrusion detection data for machine learning. Future Generat. Comput. Syst. 102, 292–306 (2020). https://doi.org/10.1016/j.future.2019.07.045
196. Xiaolan, W., et al.: Evolving anomaly detection for network streaming data. Inf. Sci. 608, 757–777 (2022). https://doi.org/10.1016/j.ins.2022.06.064
197. Kim, C., Park, J.S.: Designing online network intrusion detection using deep auto-encoder Q-learning. Comput. Electr. Eng. 79 (2019). https://doi.org/10.1016/j.compeleceng.2019.106460
198. Qazi, E.H., et al.: An intelligent and efficient network intrusion detection system using deep learning. Comput. Electr. Eng. 99(February), 107764 (2022). https://doi.org/10.1016/j.compeleceng.2022.107764
199. Ravi, V., Chaganti, R., Alazab, M.: Recurrent deep learning-based feature fusion ensemble meta-classifier approach for intelligent network intrusion detection system. Comput. Electr. Eng. 102(February), 108156 (2022). https://doi.org/10.1016/j.compeleceng.2022.108156
200. Jia, H., et al.: Network intrusion detection based on IE-DBN model. Comput. Commun. 178(January), 131–140 (2021). https://doi.org/10.1016/j.comcom.2021.07.016
201. Lopes, I.O., et al.: Effective network intrusion detection via representation learning: a Denoising AutoEncoder approach. Comput. Commun. 194(February), 55–65 (2022). https://doi.org/10.1016/j.comcom.2022.07.027
202. Chiba, Z., et al.: A novel architecture combined with optimal parameters for back propagation neural networks applied to anomaly network intrusion detection. Comput. Secur. 75, 36–58 (2018). https://doi.org/10.1016/j.cose.2018.01.023
203. Nazir, A., Khan, R.A.: A novel combinatorial optimization based feature selection method for network intrusion detection. Comput. Secur. 102, 102164 (2021). https://doi.org/10.1016/j.cose.2020.102164
204. Liu, J., Gao, Y., Hu, F.: A fast network intrusion detection system using adaptive synthetic oversampling and LightGBM. Comput. Secur. 106, 102289 (2021). https://doi.org/10.1016/j.cose.2021.102289
205. Hammad, M., Hewahi, N., Elmedany, W.: MMM-RF: a novel high accuracy multinomial mixture model for network intrusion detection systems. Comput. Secur. 120, 102777 (2022). https://doi.org/10.1016/j.cose.2022.102777
206. Moizuddin, M.D., Jose, M.V.: A bio-inspired hybrid deep learning model for network intrusion detection. Knowl. Base Syst. 238, 107894 (2022). https://doi.org/10.1016/j.knosys.2021.107894
207. Shahraki, A., Abbasi, M., Haugen, Ø.: Boosting algorithms for network intrusion detection: a comparative evaluation of Real AdaBoost, Gentle AdaBoost and Modest AdaBoost. Eng. Appl. Artif. Intell. 94(February), 103770 (2020). https://doi.org/10.1016/j.engappai.2020.103770
208. Hayatu, I., et al.: An improved binary manta ray foraging optimization algorithm based feature selection and random forest classifier for network intrusion detection. Intell. Syst. Appl. 16(June), 200114 (2022). https://doi.org/10.1016/j.iswa.2022.200114
209. Malaiya, R.K., et al.: An empirical evaluation of deep learning for network anomaly detection. IEEE Access 7, 140806–140817 (2019). https://doi.org/10.1109/ACCESS.2019.2943249
210. Wang, W., et al.: CNN-based hybrid optimization for anomaly detection of rudder system. IEEE Access 9, 121845–121858 (2021). https://doi.org/10.1109/ACCESS.2021.3109630
211. Reyes, A.A., et al.: A machine learning based two-stage Wi-Fi network intrusion detection system. Electronics 9(10), 1689 (2020). https://doi.org/10.3390/electronics9101689
212. Muhammad, G., et al.: Stacked autoencoder-based intrusion detection system to combat financial fraudulent. IEEE Internet Things J. 10(3), 2071–2078 (2023). https://doi.org/10.1109/JIOT.2020.3041184
213. Yang, L., et al.: Real-time intrusion detection in wireless network: a deep learning-based intelligent mechanism. IEEE Access 8, 170128–170139 (2020). https://doi.org/10.1109/access.2020.3019973
214. Kim, T., Pak, W.: Early detection of network intrusions using a GAN-based one-class classifier. IEEE Access 10(October), 119357–119367 (2022). https://doi.org/10.1109/ACCESS.2022.3221400
215. Park, C., et al.: An enhanced AI-based network intrusion detection system using generative adversarial networks. IEEE Internet Things J. 10(3), 2330–2345 (2023). https://doi.org/10.1109/JIOT.2022.3211346

How to cite this article: Maseer, Z.K., et al.: Meta-analysis and systematic review for anomaly network intrusion detection systems: detection methods, dataset, validation methodology, and challenges. IET Netw. 1–38 (2024). https://doi.org/10.1049/ntw2.12128

APPENDIX A

TABLE A1 Summarised information of reviewed articles. The symbol * refers no values/data reported by authors.

	Validation methodology														
					Effectiveness					Classification task		Hardware implementation		Time-based on methodology	
					Classification report										
Refs	Proposed ML/DL for NIDS		Dataset	Detected attacks	Precision	Recall	*F*-score	Accuracy	Detection	Binary	Multiclass	CPU	GPU	Time training	Time testing
[99]	SVM	DNN	NSL-KDD	5-classes	*	*	*	1.00	*	✗	✓	Core i5	✗	✗	✗
[147]	DT	✗	KDD CUP 99	5-classes	0.99	0.98	0.79	1.00	*	✗	✓	Core duo	✗	✗	✗
[148]	RF	✗	KDD CUP 99	Normal	0.98	*	*	*	*	✗	✓	Core i5	✗	✗	7.98
				DoS	0.91	*	*	*	*						
				U2R	0.17	*	*	*	*						
				R2L	0.66	*	*	*	*						
				Probe	0.56	*	*	*	*						
[149]	*k*-means	✗	KDD CUP 99	5-classes	*	*	*	*	*	✗	✓	✗	✗	✗	✗
[100]	✗	LS-SVM	KDD CUP 99	2-classes	*	*	*	0.96	0.95	✓		✗		✗	✗
[101]	K-means	✗	KDD CUP 99	Normal	0.96	0.85	0.92	*	0.86	✗	✓	✗	✗	✗	✗
				DoS	0.97	0.97	0.97	*	1.00						
				U2R	0.75	0.73	0.75	*	0.98						
				R2L	0.65	0.92	0.73	*	0.91						
				Probe	0.85	0.95	0.92	*	0.73						
[107]	DT	✗	KDD CUP 99	22-classes	*	*	*	0.20	*	✗	✓	Core i7	✗	✗	✗
[150]	NB	✗	KDD CUP 99	Normal	*	*	*	*	0.98	✗	✓	Core i5	✗	1467 s	792 s
				DoS	*	*	*	*	0.97						
				U2R	*	*	*	*	0.75						
				R2L	*	*	*	*	0.75						
				Probe	*	*	*	*	0.94						
[108]	K-centroid	✗	KDD CUP 99	Normal	*	*	*	0.86	*	✗	✓	✗	✗	✗	✗
				DoS	*	*	*	0.92	*						
				U2R	*	*	*	0.96	*						
				R2L	*	*	*	0.77	*						
				Probe	*	*	*	0.32	*						
[151]	SVM	✗	NSL-KDD	5-classes	0.94	0.98	0.96	*	*	✗	✓	✗	✗	2100 s	1171 s
[113]	SVM	✗	KDD CUP 99	5-classes	*	*	*	0.90	*	✗	✓	✗	✗	✗	✗

TABLE A1 (Continued)

	Validation methodology														
					Effectiveness					Classification task		Hardware implementation		Time-based on methodology	
					Classification report										
Refs	Proposed	ML/DL for NIDS	Dataset	Detected attacks	Precision	Recall	*F*-score	Accuracy	Detection	Binary	Multiclass	CPU	GPU	Time training	Time testing
[105]	SVM	✗	NSL-KDD	Normal	*	*	*	0.95	*	✗	✓	✗	✗	✗	✗
				DoS	*	*	*	0.99	*						
				U2R	*	*	*	0.99	*						
				R2L	*	*	*	0.70	*						
				Probe	*	*	*	1.00	*						
[106]	✗	AE	NSL-KDD	5-classes	0.86	0.96	0.76	*	*	✗	✓	✗	✗	✗	✗
[152]	✗	SAE	NSL-KDD	2-classes	0.85	0.93	*	0.87	*	✓	✗	✗	✗	✗	✗
[109]	NN	✗	KDD CUP 99	2-classes	*	*	*	0.85	*	✓	✗	Core i7	✗	✗	✗
[110]	RepTree	✗	UNSW-NB15	2-classes	*	*	*	0.90	*	✓	✗	Core i5		2.69 s	0.37 s
[153]	EML	✗	UNSW-NB15	2-classes	*	*	*	0.83	*	✓	✗	CPU	✗	1.2 min	✗
[154]	✗	GAN	UNSW-NB15	10-classes	*	*	*	0.99	*	*	✓	Core i5	✗	✗	✗
[155]	LR	✗	KDD CUP 99	5-classes	*	*	*	*	0.99	*	✓	Intel core duo	✗	✗	✗
[156]	RF	✗	CIDDS-001	2-classes	*	*	*	1.00	1.00	✓	*	Core i5	✗	✗	✗
[157]	Skip-gram	✗	UNSW-NB15	2-classes	0.99	0.82	*	*	0.91	✓	*	✗	✗	✗	✗
[158]	Extra-tree	*	UNSW-NB15	Normal	*	*	*	*	0.94	✓	*	✗	✗	✗	✗
				DDoS	*	*	*	*							
[159]	SVM	*	UNSW-NB15	2-classes	*	*	*	*	0.96	✓	*	✗	✗	✗	✗
[160]	Ensemble	*	CIDDS-001	2-classes	*	*	*	*	0.94	✓	*	✗	✗	✗	✗
[161]	DT	*	UNSW-NB15	Normal	0.93	0.93	*	0.47	*	*	✓	✗	✗	✗	✗
				Generic	0.98	0.82	*								
				Exploits	76.22	0.77	*								
				Fuzzers	0.74	0.65	*								
[162]	*	LSTM	UNSW-NB15	5-classes	0.91	0.92	0.92	0.90	*	*	✓	✗	✗	✗	✗
[163]	*	GB	UNSW-NB15	2-classes	*	*	*	*	1.00	✓	*	✗	✗	✗	✗
[164]	RF	*	UNSW-NB15	Normal	0.897	0.96	0.93	*	*	*	✓	Core i5	✗	✗	✗
				Generic	0.998	0.96	0.98								
				Exploits	0.759	0.66	0.70								
				Fuzzers	0.942	0.38	0.54								
				Recon.	0.888	0.82	0.85								
				DoS	0.351	0.46	0.39								
				Analysis	0.046	0.06	0.05								

(Continues)

TABLE A1 (Continued)

	Validation methodology															
					Effectiveness					Classification task		Hardware implementation		Time-based on methodology	
					Classification report										
Refs	Proposed ML/DL for NIDS		Dataset	Detected attacks	Precision	Recall	*F*-score	Accuracy	Detection	Binary	Multiclass	CPU	GPU	Time training	Time testing
				Backdoor	0.151	0.40	0.21								
				Shellcode	0.352	0.78	0.48								
				Worms	0.778	0.79	0.78								
[165]	ANN	*	UNSW-NB15	Normal	0.989	0.99	0.99	*	*	✓	✗	✗	✗	✗	✗
				Abnormal											
[166]	GMM	*	CICIDS2017	BENIGN	*	*	*	1.00	*	✓	✗	✗	✗	✗	✗
				DDoS	*	*	*								
[167]	K-NN	*	CICIDS2017	All types of attacks	0.99	0.99	0.99	*	0.99	*	✗	✗	✗	✗	✗
[168]	*	GAN	CICIDS2017	BENIGN	0.99	0.99	1.00	*	*	*	✓	✗	✗	✗	✗
				Brute force											
				SQL injection											
				XSS											
				BENIGN	0.99	99.9	1.00	*	*	✓	*				
				PortScan											
				BENIGN	0.86	0.53	65.77	*	*	✓	*				
				Bot											
				BENIGN	100	0.60	0.75	*	*	✓	*				
				Infiltration											
				BENIGN	*	*	*	*	*	*	✓				
				FTP-Patator	1.00	0.99	0.99	*	*						
				SSH-Patator	1.00	1.00	1.00	*	*						
				BENIGN	*	*	*	*	*	*	✓				
				DoS GoldenEye	1.00	0.99	0.99	*	*						
				DoS hulk	1.00	0.99	0.99	*	*						
				DoS SlowHTTPTest	0.99	0.99	0.99	*	*						
				Heartbleed	1.00	1.00	1.00	*	*						
				Benign	99.90	99.9	99.9	*	*	✓	*				
				DDoS											
[169]	RF	*	CICIDS 2017	All types of attacks	*	*	*	0.99	1.00	*	✓	✗	✗	✗	✗
[46]	RF	*	CICIDS 2017	DDoS	*	*	*	*	0.96	*	✓	Core 2 duo	✗	27.36 s	✗
[170]	*	RNN	CICIDS 2017	*	*	*	*	*	0.98	*	✓	Intel Xeon	GPUs	14 min	1.3 s

TABLE A1 (Continued)

	Validation methodology														
					Effectiveness					Classification task		Hardware implementation		Time-based on methodology	
					Classification report										
Refs	Proposed	ML/DL for NIDS	Dataset	Detected attacks	Precision	Recall	*F*-score	Accuracy	Detection	Binary	Multiclass	CPU	GPU	Time training	Time testing
[171]	*	RNN	CICIDS 2017	BENIGN	*	*	*	*	*	*	*	CPU	GPU	20.7 s	7.21 s
				Brute force	*	*	*	*	0.82						
				SQL injection	*	*	*	*	0.82						
				XSS	*	*	*	*	0.92						
				BENIGN	*	*	*	*	*	✓	*				
				PortScan	*	*	*	*	0.90						
				BENIGN	*	*	*	*	*	✓	*				
				BoT	*	*	*	*	0.98						
				BENIGN	*	*	*	*	*	✓	*				
				Infiltration	*	*	*	*	1.00						
				BENIGN	*	*	*	*	*	*	✓				
				FTP-Pastor	*	*	*	*	1.00						
				SSH-Pastor	*	*	*	*	1.00						
				BENIGN	*	*	*	*	0.98	*	✓				
				DoS goldeneye	*	*	*	*	0.77						
				DoS hulk	*	*	*	*	0.98						
				DoS SlowHTTPTest	*	*	*	*	0.94						
				Heartbleed	*	*	*	*	1.00						
				BENIGN	*	*	*	*	1.00	✓	*				
				DDoS	*	*	*	*							
[172]	*	RF	CICIDS2017	Normal	1.00	1.00	1.00	1.00		✗	✓	Core i3	GPU	9 s	8 s
				Brute-force	1.00	0.78	0.77	0.76							
				XSS	0.56	0.47	0.49	0.49							
				SQL-injection	0.22	1.00	0.22	0.36							
[173]	*	AE	KDD CUP 99	Normal, abnormal	0.99	✗	✗	✗	✗	✓	✗	✗	✗	✗	✗
[174]	*	CNN, AE	NSL-KDD	Normal	1.00	✗	✗	✗	✗	✗	✓	Core i5	✗	✗	✗
				DoS	1.00	✗	✗	✗	✗						
				U2R	0.70	✗	✗	✗	✗						
				R2L	0.70	✗	✗	✗	✗						
				Probe	1.00	✗	✗	✗	✗						
[175]	*	Bi-RNN	UNSW-NB15	Normal, attack	0.95	1.00	✗	✗	✗	✓	✗	Core i7	*	3228 s	54 s

(Continues)

TABLE A1 (Continued)

	Validation methodology															
					Effectiveness					Classification task		Hardware implementation		Time-based on methodology	
					Classification report										
Refs	Proposed	ML/DL for NIDS	Dataset	Detected attacks	Precision	Recall	*F*-score	Accuracy	Detection	Binary	Multiclass	CPU	GPU	Time training	Time testing
[114]	*	DNN	Realistic data	Normal, abnormal	0.99	✗	✗	✗	✗	✓	✗	Core i7	✗	✗	✗
[176]	*	AE	NSL-KDD	Normal	0.97	0.70	0.97	0.81	✗	✗	✓	Intel Xeon	✗	✗	✗
				DoS	0.83	0.96	0.83	0.89	✗						
				U2R	0.11	0.81	0.32	0.64	✗						
				R2L	0.32	0.73	0.11	0.18	✗						
				Probe	0.80	0.86	0.80	0.46	✗						
[115]	*	SAE	KDD CUP 99	Normal, attack	0.95	✗	✗	✗	✗	✓	✗	✗	GPU	✗	✗
[177]	*	SAE	NSL-KDD	Normal	0.97	1.00	0.97	0.98	✗	✗	✓	✗	GPU	✗	✗
				DoS	0.94	1.00	0.94	0.97	✗						
				U2R	0.94	1.00	0.94	0.97	✗						
				R2L	0.03	1.00	0.03	0.07	✗						
				Probe	0.94	1.00	0.72	0.97	✗						
[178]	*	DNN	NSL-KDD	Normal, DoS, U2R, R2L, probe	0.86	✗	✗	✗	✗	✗	✓	Core i3	✗	✗	✗
[179]	*	DNN	AWID	Normal, flooding, impersonation, and injection	0.99	✗	✗	✗	✗	✗	✓	Core i3	✗	✗	✗
[180]	DNN	AE	KDD CUP 99	Normal, DoS, U2R, R2L, probe	0.99	0.99	0.99	✗	✗	✗	✓	Core i3	✗	✗	✗
[181]	*	DNN	UNSW-NB15	Normal, attack	0.92	✗	✗	✗	✗	✗	✓	Core i5	✗	✗	✗
[119]	*	DNN	CICIDS2017	Normal, attack	0.99	0.99	0.98	✗	✗	✓	✗	Core i3	✗	✗	✗
[116]	*	WDLSTM, CNN	UNSW-NB15	Normal	0.98	1.00	1.00	1.00	*	✗	✓	Core i7	✗	✗	✗
				DoS		0.64	0.80	0.71	*						
				Exploits		0.32	0.27	0.29	*						
				Backdoor		0.50	0.07	0.12	*						
				Analysis		0.44	0.09	0.15	*						
				Fuzzers		0.71	0.61	0.66	*						
				Generic		1.00	0.99	0.99	*						
				Reconnaissance		0.93	0.77	0.84	*						
				Shellcode		0.82	0.79	0.81	*						
				Worms		0.50	0.09	0.15	*						
[182]	SVM	AE	ISCX 2012	Normal, attack	0.994	0.99	0.994	0.99	✗	✓	✗	Core i7 quad-core	✗	✗	✗
[183]	*	GAN	NSL-KDD	Normal, attack	0.89	0.89	0.90	0.10	✗	✓	✗	✗	GPU	✗	✗

TABLE A1 (Continued)

	Validation methodology															
					Effectiveness					Classification task		Hardware implementation		Time-based on methodology	
					Classification report										
Refs	Proposed	ML/DL for NIDS	Dataset	Detected attacks	Precision	Recall	*F*-score	Accuracy	Detection	Binary	Multiclass	CPU	GPU	Time training	Time testing
[184]	*	CNN-RNN-LSTM	CIDDS-001	Normal, malicious	0.91	0.98	0.68	0.80	✗	✓	✗	Core i3	GPU	✗	✗
[185]	*	DAE-MLP	UNSW-NB15	Normal	*	0.99	0.99	0.99	*	✓	✗	✗	✗	✗	✗
				Attack	*	0.95	0.94	0.95	*						
[186]	*	CNN	NSL-KDD	Normal, DoS, U2R, R2L, probe	✗	✗	✗	✗	✗	✗	✓	✗	✗	✗	✗
[187]	*	RNN	UNSWNB15	DDoS DoS	✗	0.93	✗	✗	✗	✗	✓	✗	✗	✗	✗
				Reconnaisance											
				Normal theft											
[188]	*	GAN	AWID	Flooding	0.993	0.74	0.921	0.618	✗	✗	✓	✗	✗	✗	✗
				Impersonation	0.965	0.00	0.00	0.00							
				Injection	0.998	0.96	0.934	0.999							
				Normal	0.957	0.97	0.95	0.99							
[189]	*	LSTM	Real time	Normal, abnormal	✗	✗	✗	✗		✓	✗	✗	GPU	✗	✗
[190]	*	DNN	CIDS	Normal, malicious	0.96	0.91	0.91	0.91	✗	✓	✗	Core i5	✗	✗	✗
[191]	SVM	DBN	NSL-KDD	Normal, attack	0.974	0.97	0.977	0.97	*	✓	*	Core i7	*	*	*
[192]	RF	*	CIDDS-001	Normal, malicious	0.99	✗	✗	✗	✗	✗	✗	✗	GPU	✗	✗
[64]	*	DNN	CICIDS2017	Normal	0.56	✗	✗	✗	✗	✗	✓	✗	GPU	✗	✗
				SSH-Patater	0.95	✗	✗	✗	✗						
				FTP-Patater	0.92	✗	✗	✗	✗						
				Web	0.98	✗	✗	✗	✗						
				Bot	0.95	✗	✗	✗	✗						
				DDoS	0.85	✗	✗	✗	✗						
				Portscan	0.85	✗	✗	✗	✗						
[102]	*	DNN	Realistic data	Blackhole	✗	0.97		0.97	✗	✗	✓	✗	✗	✗	✗
				Opportunistic service	✗	0.95	0.98	0.97	✗						
				000DDoS	✗	0.96	0.98	0.96	✗						
				Sinkhole	✗	0.99	0.99	0.99	✗						
				Wormhole	✗	0.96	0.97	0.98	✗						
[103]	*	RBM	Realistic data	Normal, malicious	0.99	✗	✗	✗	✗	✓	✗	✗	✗	✗	✗
[193]	*k*-mean	RBM	NSL-KDD	Normal, malicious	0.92	✗	✗	✗	✗	✗	✗	Core due	✗	✗	✗

(Continues)

TABLE A1 (Continued)

	Validation methodology														
					Effectiveness					Classification task		Hardware implementation		Time-based on methodology	
					Classification report										
Refs	Proposed	ML/DL for NIDS	Dataset	Detected attacks	Precision	Recall	*F*-score	Accuracy	Detection	Binary	Multiclass	CPU	GPU	Time training	Time testing
[104]	RF	*	NSL-KDD	Normal	1.00	✗	✗	✗	✗	✗	✓	Core i7	GPU	✗	✗
				DDOS	1.00	✗	✗	✗							
				DOS	0.98	✗	✗	✗							
				Reconnaissance	1.00	✗	✗	✗							
				Theft	0.96	✗	✗	✗							
[117]	✗	DBN, PNN	KDD CUP 99	Normal, abnormal	0.99	✗	✗	✗	✗	✓	✗	✗	GPU	✗	✗
[34]	✗	RBM	ISCX 2012	Normal, abnormal	0.88	✗	✗	✗	✗	✓	✗	✗	✗	✗	✗
[123]	✗	RBM	KDD CUP 99	Normal, DDoS	0.99	✗	✗	0.99	✗	✓	✗	✗	✗	✗	✗
[194]	✗	AE	NSL-KDD	Normal	✗	0.85	0.96	0.90	✗	✗	✓	Core i7	GPU	✗	✗
				DoS	✗	0.95	0.98	0.97	✗						
				Probe	✗	0.69	0.94	0.80	✗						
				R2L	✗	0.99	0.39	0.56	✗						
				U2R	✗	✗	✗	✗	✗						
[118]	RF	✗	CIC-IDS2017	BENIGN	1.00	1.00	1.00	✗	✗	*	✓	Core i5	8 GB	89.9	1.22
				DoS GoldenEye	1.00	1.00	1.00	✗	✗						
				DoS hulk	1.00	1.00	1.00	✗	✗						
				DoS SlowHTTPTest	1.00	1.00	1.00	✗	✗						
				Heartbleed	1.00	1.00	1.00	✗	✗						
[195]	SVM.	✗	UNSW-NB15	Normal	✗	✗	0.60	✗	✗	*	✓	Core(TM) i5	4 GB	✗	✗
				DoS	✗	✗	0.0	✗	✗						
				Exploits	✗	✗	0.80	✗	✗						
				Backdoor	✗	✗	0.00	✗	✗						
				Analysis	✗	✗	0.00	✗	✗						
				Fuzzers	✗	✗	0.64	✗	✗						
				Generic	✗	✗	0.96	✗	✗						
				Reconnai	✗	✗	0.36	✗	✗						
				Shellcode	✗	✗	0.81	✗	✗						
				Worms	✗	✗	0.00	✗	✗						
[196]	✗	AE	NSL-KDD	All classes	✗	✗	✗	0.93	0.99	✗	✓	Core i7	8 GB	✗	✗
[197]	✗	AE	Privet	Normal	✗	✗	✗	✗	✗	✓	✗	✗	✗	✗	✗
				Abnormal											

TABLE A1 (Continued)

	Validation methodology														
					Effectiveness					Classification task		Hardware implementation		Time-based on methodology	
					Classification report										
Refs	Proposed	ML/DL for NIDS	Dataset	Detected attacks	Precision	Recall	*F*-score	Accuracy	Detection	Binary	Multiclass	CPU	GPU	Time training	Time testing
[198]	✗	AE	KDD CUP'99	All classes	1.00	1.00	1.00	1.00	✗	✗	✓	Core i7	32 GB	✗	✗
[199]	✗	RNN	CICIDS-2017.	All classes	0.99	0.99	0.99	0.99	0.97	✗	✓	✗	✗	✗	✗
[200]	✗	DBN	KDD CUP 99	All classes	✗	✗	✗	0.90	✗	✗	✓	Core i7	16 GB		51.07
[201]	✗	AE	CICIDS2018	Benign	99.7	99.9	99.8	✗	✗	✗	✓	Core i9	128 GB	✗	3885
				Bot	1.00	0.99	1.00	✗	✗						
				Web	0.85	0.94	0.90	✗	✗						
				XSS	1.00	0.74	0.85	✗	✗						
				HOIC	1.00	1.00	1.00	✗	✗						
				LOIC-UDP	0.96	0.99	0.98	✗	✗						
				LOIC-UDP	0.96	0.99	0.98	✗	✗						
				LOIC-HTTP	1.00	1.00	1.00	✗	✗						
				GoldenEye	1.00	1.00	1.00	✗	✗						
				Hulk	1.00	1.00	1.00	✗	✗						
				SlowHTTPTest	1.00	0.97	98.4	✗	✗						
				Slowloris	1.00	99.9	1.00	✗	✗						
				BruteForce	98.4	100	1.00	✗	✗						
				Bot-Iot	99.1	0.98	0.98	✗	✗						
				SQL injection	1.00	0.60	0.74	✗	✗						
				Bruteforce	1.00	1.00	1.00	✗	✗						
[202]	*	PNN	✗	All classes	0.99		0.99	0.99	✗	✗	✓	✗	✗	✗	✗
[203]	RF	✗	UNSW-NB15	All classes	✗	✗	✗	0.83	✗	✗	✓	✗	✗	✗	✗
[204]	xgboost	✗	CICIDS2017	All classes	✗	✗	✗	0.99	✗	✗	✓	Core i5	8 GB	✗	✗
[205]	EM	✗	CSE-CIC-IDS2018	All classes	✗	✗	✗	0.900	✗	✗	✓	✗	✗	✗	✗
[206]	✗	AE	BoT-IoT	Normal	✗	✗	✗	1.00	✗	✗	✓	✗	✗	2012.6	✗
				Information gathering	✗	✗	✗	1.00	✗						
				DoS	✗	✗	✗	1.00	✗						
				DDoS	✗	✗	✗	1.00	✗						
				Information theft	✗	✗	✗	1.00	✗						
[207]	AdaBoost	✗	CICIDS2017	Binary-class	✗	✗	✗	✗	✗	✓	✗	✗	✗	✗	131.9
[208]	RF	✗	CIC-IDS2017	Multi-class	0.99	0.94	0.96	0.99	✗	✗	✓	Intel pentium, CPU B960	4 GB	✗	455.317
[124]	✗	CNN	CIC-IDS2017	Binary-class	0.99	0.99	0.99	0.99	✗	✓	✗	Core i5	8 GB	39.52	0.061

(Continues)

TABLE A1 (Continued)

	Validation methodology														
					Effectiveness					Classification task		Hardware implementation		Time-based on methodology	
					Classification report										
Refs	Proposed	ML/DL for NIDS	Dataset	Detected attacks	Precision	Recall	*F*-score	Accuracy	Detection	Binary	Multiclass	CPU	GPU	Time training	Time testing
[125]	✗	AE	CIC-IDS2017	Multi-class	0.99	0.99	0.99	0.99	✗	✗	✓	✗	✗	✗	0.09
[209]	✗	LSTM	CIC-IDS2017	Multi-class	1.00	1.00	1.00	1.00	✗	✗	✓	✗	✗	9.20	✗
[126]	✗	CNN	CIC-IDS2017	Multi-class	✗	✗	✗	✗	0.90	✗	✓	Core i7	16 GB	120	180
[210]	✗	CNN	CIC-IDS2017	Multi-class	✗	✗	✗	0.99	✗	✗	✓	✗	✗	✗	✗
[120]	✗	AE	CIC-IDS2017	Normal	0.99	0.99	0.97	✗	✗	✗	✓	Core i7	16 GB	1851.02	0.43
				SSH	0.99	0.50	0.67	✗	✗						
				FTP-Patater	0.98	0.93	0.96	✗	✗						
				Web	0.98	✗	✗	✗	✗						
				DoS GoldenEye	0.99	0.88	0.93	✗	✗						
				DoS hulk	0.72	0.73	0.81								
				Bot	0.0	0.00	0.00	✗	✗						
				DDoS	0.99	0.92	0.95	✗	✗						
				Portscan	0.85	✗	✗	✗	✗						
				Infiltration	0.00	0.00	0.00	✗	✗						
[121]	DT	✗	AWID	Normal, flooding, and impersonation	0.95	0.93	0.92	0.90	✗	✗	✗	✗	✗	✗	✗
[122]	✗	AE, deep Q-network	AWID	Normal, flooding, and impersonation	0.86	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
[211]	RF	✗	AWID	Normal, flooding	0.99	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
[212]	✗	AE, DNN	AWID	Normal	0.99	✗	✗	✗	✗	✗	✓		✗	✗	✗
				Flooding	0.92	✗	✗	✗	✗	✗					
				Impersonation	0.99	✗	✗	✗	✗	✗					
[213]	✗	DBN	AWID	Normal	0.96	0.97	0.97	0.97	0.91	✗	✓	✗	✗	✗	✗
				Flooding											
				Injection											
[214]	✗	GAN	CSE2018	All types of attacks	✗	0.99	0.86	0.81	0.83	✗	✓	✗	✗	✗	✗
[215]	✗	GAN	UNSW-NB15	Normal	0.90	0.97	0.85	0.90	✗	✗	✗	✗	✗	✗	✗
				Abnormal	0.83	0.96	0.96	0.86	✗	✗					

Abbreviations: AE, auto encoder; ANN, artificial neural network; CNN, convolutional neural network; DAE-MPL, denoising autoencoder - multi-layer perceptron; DBN, deep belief network; DL, deep learning; DNN, deep neural network; DT, decision tree; EML, ensemble machine learning; GAN, generative adversarial network; GMM, gaussian mixture model; K-NN, K-nearest-neighbour; ML, machine learning; NB, Naive Bayes; NIDS, network intrusion detection system; PNN, probabilistic neural network; RBM, restricted Boltzmann machine; RF, random forest; RNN, recurrent neural network; SAE, stacked autoencoder; SVM, support vector machine; UNSW-NB15, University of New South Wales-network based 15.