A Review on Recent Approaches of Machine Learning, Deep Learning, and Explainable Artificial Intelligence in Intrusion Detection Systems

Seshu Bhavani Mallampati[1], Hari Seetha[2*]
1- VIT-AP University, School of Computer Science and Engineering, Near Vijayawada, Andhra Pradesh, India.
Email: bhavani.20phd7017@vitap.ac.in
2- VIT-AP University, Center of Excellence, AI and Robotics, Near Vijayawada, Andhra Pradesh, India.
Email: seetha.hari@vitap.ac.in(Corresponding author)

Received: 25 September 2022 Revised: 12 November 2022 Accepted: 20 December 2022

ABSTRACT:
In recent decades, network security has become increasingly crucial, and intrusion detection systems play a critical role in securing it. An intrusion Detection System (IDS) is a mechanism that protects the network from various possible intrusions by analyzing network traffic. It provides confidentiality and ensures the integrity and availability of data. Intrusion detection is a classification task that classifies network data into benign and attack by using various machine learning and deep learning models. It further develops a better potential solution for detecting intrusions across the network and mitigating the false alarm rate efficiently. This paper presents an overview of current machine learning (ML), deep learning (DL), and Explainable Artificial intelligence (XAI) techniques. Our findings provide helpful advice to researchers who are thinking about integrating ML and DL models into network intrusion detection. At the conclusion of this work, we outline various open challenges.

KEYWORDS: Network Security, Intrusion Detection, IPS, Preprocessing, SMOTE, Datasets, Attacks, Feature Selection, XAI.

1. INTRODUCTION

In the present era, the number of devices related to smart homes, transportation, manufacturing, and health care has grown, so the volume of confidential and crucial data traveling across the network has expanded significantly over the last decade. However, with the accelerated growth in technology, there was a momentous change in the network size. As an outcome, a large volume of information was generated and shared among the network nodes [1]. Providing security to such data has become challenging because every node present in the network is endangered due to several zero-day attacks. According to research, ransomware attackers targeted the financial, government, and transportation industries the most. For example, Fig. 1 shows the total number of ransomware attacks worldwide from 2016 to the first half of 2022 [2].

To address security issues, various measures such as firewalls and authentication protocols can be used. They serve as the first layer of defense against various external threats to edge devices. On the other hand, these security mechanisms have limitations and can be easily exploited by attackers. In addition, Anderson Jim proposed an IDS in the year 1980 [3]. Since then, a number of monitoring solutions, including intrusion detection and prevention systems (IDS and IPS), have been suggested and subsequently implemented[4].

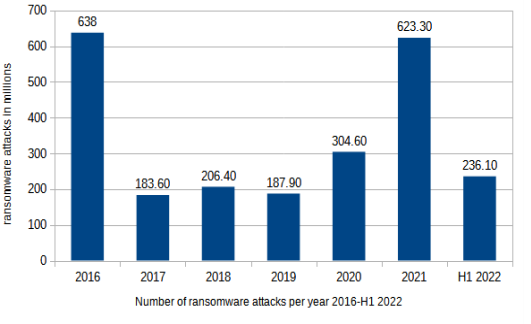

Fig. 1. Number of ransomware attacks.

The IDS is classified based on Deployment and the Detection mechanism. The classification of IDS is illustrated in Fig. 2 [1].

Paper type: Review paper
DOI: 10.30486/mjee.2023.1976657.1046
How to cite this paper S. B. Mallampati, H. Seetha, **"A Review on Recent Approaches of Machine Learning, Deep Learning, and Explainable Artificial Intelligence in Intrusion Detection Systems"**, *Majlesi Journal of Electrical Engineering*, Vol. 17, No. 1, pp. 29-54, 2023.

The deployment-based IDS is further classified as host-based, network-based, and hybrid. Deployment-based IDS depends on how events related to attacks are gathered, processed, and dealt with, and these systems can be distributed, centralized, or hybrid. Each method offers benefits and drawbacks regarding cost, performance, and other factors.

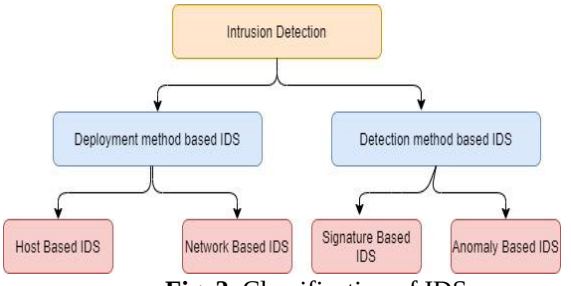

Fig. 2. Classification of IDS.

The detection-based IDS is categorized as signature-based and anomaly-based. While signature-based IDS has been extensively used to detect known attacks accurately with a low false alarm rate, it still has been excoriated for its inability to mitigate unknown attacks [5]. One solution to address this issue is updating the database regularly, which is not feasible and is more expensive [6]. Anomaly-based detection approaches generate profiles for normal user behavior and compare actual user behaviors to the built profiles. If an anomaly is discovered, the IDS raises the alarm, alerting the system about the invasive activity. The ability to detect unknown threats is the main benefit of such systems. However, they give relatively good level of service compared to signature-based IDS [7]. The difference between signature and anomaly-based IDS is depicted in Table 1.

Table 1. Difference between Signature and anomaly-based IDS.

	Signature Based	Anomaly Based
Type of assaults	It identifies known assaults	Identifies Known and unknown attacks
Performance	Low false positives	High false positives
Resources	Requires fewer system resources	It needs more system resources
Behavior	Focus on attack behavior.	Concentrates on normal behavior to detect unknown assaults
Database updates	Requires frequent database updates as attack behavior changes frequently	No need for database updates

The phrases Artificial Intelligence (AI), ML, and DL, are widely used synonymously to refer to the same concepts in application development, as demonstrated in Fig. 3, where ML and DL are subfields of AI.

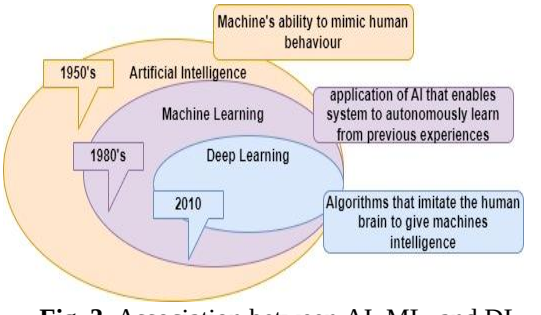

Fig. 3. Association between AI, ML, and DL.

Several researchers have suggested and introduced several Machine ML models like Support vector machines (SVM), Decision trees (DT), Random Forest (RF), K-Nearest Neighbors (KNN), etc., over the past few years. On the other hand, in recent years DL -based strategies like Autoencoder (AE), Deep Neural Networks (DNN), Convolution Neural Networks (CNN), Long Short Memory (LSTM), and many more have been used to create robust IDS systems.

Despite being widely used, ML and DL models are still primarily black boxes. Most of these approaches show an excellent detection rate and low false positive rate (FPR). However, as models become more sophisticated, users find it increasingly difficult to understand the rationale behind their predictions. Therefore, understanding the reasons behind the prediction is essential to gaining trust. As a result, XAI is the center of current AI research because Interpretability provides neutrality in decision-making by assisting in detecting and correcting bias in the training dataset. It also includes trust by giving valid inferences and reasoning.

The primary goal of this research article is to analyze and explore up-to-date information on existing IDS systems. For new researchers, these technologies are a foundation for designing an efficient and robust IDS system. In this paper, we have reviewed various articles that demonstrate the usage of AI tools and XAI techniques and discuss methodologies proposed in the literature along with their strengths, weaknesses, performance metrics, and analyzed datasets. By analyzing various observations, we provide some of the recent trends to design a better IDS system.

The significant contributions of the paper include the following.

- It presents various ML and DL models in intrusion detection with their strengths, limitations, datasets, and performance metrics for evaluating IDS.

- It provides the importance of XAI to improve trust management in areas that human specialists can comprehend, such as causal inference and underlying data evidence. Further global and local explanations are used to describe the influence of extracted features and, consequently, the class to which each particular instance belongs. These explanations improve a model's understanding and reliability.
- It provides information about various public datasets useful for building IDS models and various open issues and challenges in IDS.

The rest of the work is organized as follows. Section 2 provides information about public datasets for building IDS models. Section 3 presents various evaluation metrics used in IDS models. Section 4 offers a detailed elaboration of ML and DL methodologies. Section 5 presents a detailed analysis of XAI in IDS. Section 6 presents various existing surveys of IDS. Observations, open issues, and research challenges are discussed in Section 7, and Section 8 provides a conclusion.

2. PRELIMINARY DISCUSSION ON DATASETS

Collecting data from the real-time network is a complex task. Therefore, many researchers use IDS datasets that are openly available. Many benchmark datasets are available such as KDDCUP199, KYOTO2006+, NSL-KDD, UNSW-NB15, CIC-IDS 2017, CIC-IDS2018, ADFA, ADFA-LB, CICDDoS2019 dataset, etc. Therefore, appropriately selecting and utilizing the data are essential for any security research. The following sub-section provides various benchmark datasets most widely used by researchers.

2.1. Datasets

2.1.1 UNSW-NB dataset

It was generated by IXIA PerfectStorm. It was conceptualized by the Australian center for cybersecurity. It is used to create and replicate real-world and modern attack models. Tcpdump is a utility that has up to 100 GB of Pcap files that may be used to simulate nine distinct sorts of attacks. DOS, ShellCode, Worms, Fuzzers, Backdoors, Exploits, Analysis, Generic, and Reconnaissance are among the attacks [8]. In addition, the dataset contains 49 features with 2 million records corresponding to the class label. Different types of attacks on the dataset are depicted in Fig. 4 [9].

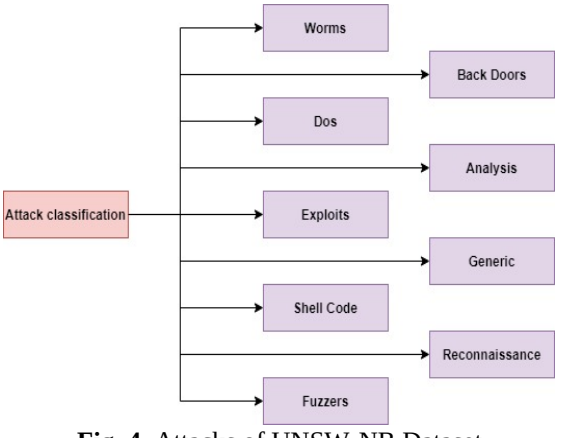

Fig. 4. Attacks of UNSW-NB Dataset.

2.1.2 KDD-CUP99 dataset

It was created in the year of 1999 by the MIT Lincon laboratory. This dataset is a subset of the DARPA-98 dataset. The KDD-99 dataset is multi-variate. It is one of the most popular datasets that is used in IDS. This dataset contains 5 different classes, which are 4 four attack classes and one normal class. It contains 5 million records for training and 2 million records for testing. Each record present in this dataset has 41 different features. The attributes utilized in the dataset have category and numeric features. The data set comprises mainly of four sorts of attacks which are given below.

- DOS stands for Denial of Service; for example-Neptune attacks
- U2R-Unauthorized access to superuser ("root") privileges on a local computer. For instance, Rootkit attacks.
- R2L-unauthorized access from remote workstations for example Multihop attacks.
- Probing-Surveillance and another probing. Example - port_sweep attacks. The classes of this dataset are shown in Fig. 5 [10].

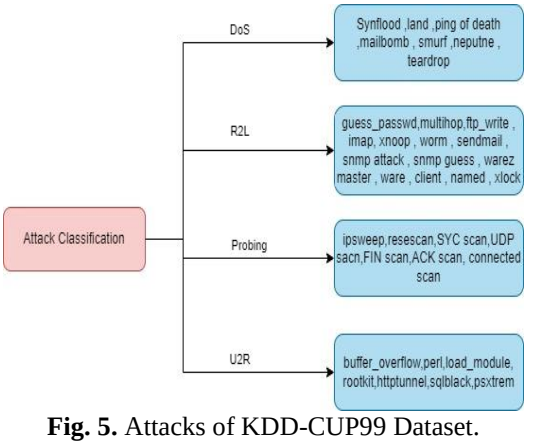

Fig. 5. Attacks of KDD-CUP99 Dataset.

2.1.3 NSL-KDD dataset

The KDDCUP99 has issues like more duplicate

records, which leads the learning algorithms to be biased towards more frequent records, preventing them from learning infrequent records, which are typically more destructive to networks, such as U2R assaults. The NSL-KDD dataset was proposed as a solution to the KDD 99 difficulties by Tavallaee et al. [11]. The NSL-KDD dataset has some of the same features as the KDD cup 99 datasets [12]. In the NSL-KDD dataset, duplicate entries are removed. The dataset contains KDDTrain+ and KDDTest+ with 125,973 and 22,544 records [13].

2.1.4 CIC-IDS 2017

The Canadian Institute of Cyber Security generated the CIC-IDS 2017 dataset (CIC). The majority of typical attacks are represented in this dataset, which closely resemble real-world attacks. This dataset was collected in a small network with regular simulated traffic. A separate network launches six different types of current attacks. Netflow with 80 features and raw packet capture is also offered. It uses a CIC flowmeter to examine network traffic results, including the time stamp, source IP, destination IP, ports, protocols, and seven assaults. Brute ForceFTP, Brute Force SSH, DoS, Heartbleed, Web Attacks, Infiltration, Botnet, and DDoS are the malware threats covered in the CIC-IDS 2017 dataset. Various attack profiles of 6 days are shown in Fig. 6 below [14].

Brute Force Attack: This is one of the most prevalent attacks that may be used to locate hidden pages and material within an online application and crack passwords.

Heartbleed Attack: It is commonly reported as adverse by sending a fraudulent request to the server with a short payload and long duration field to elicit the victim's response.

Botnet: A botnet is a group of Web devices used by the botnet's owner to achieve various goals. It can steal information, send spam, and give the attacker access to the device and its network.

DoS Attack: It is commonly accomplished by overwhelming a device or resource with unnecessary requests to overwhelm systems and prevent some or all valid requests from being fulfilled.

DDoS Attack: It usually occurs when a large number of systems overburdens a victim's bandwidth or resources. A denial-of-service attack occurs when multiple infected systems flood the targeted system with huge quantities of network traffic.

Web Attack: As companies and individuals eventually take security seriously, new attack types arise every day.

Infiltration Attack: Insecure software such as Adobe Acrobat Reader is commonly used to enter a network from the inside. A backdoor will be installed on the victim's workstation after successful deployment, allowing the attacker to perform numerous attacks against the victim's network.

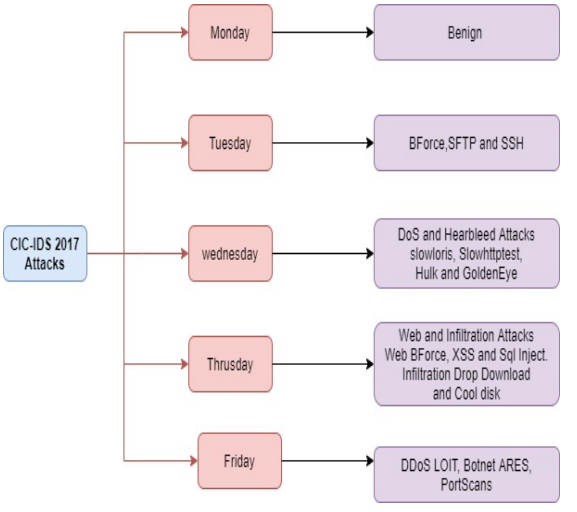

Fig. 6. Six days attack information of CIC-IDS 2017 dataset.

2.1.5 CICDDoS2019

CICDDoS2019 was created by the Canadian Institute of Cyber Security, which contains various DDoS attacks. Data was captured in 2 days, shown in Fig. 7. It has twelve DDoS attacks, including NTP, DNS, LDAP, MSSQL, NetBIOS, SNMP, SSDP, UDP, UDP-Lag, WebDDoS, SYN, and TFTP which were captured on Day 1. Seven attacks, including PortScan, NetBIOS, LDAP, MSSQL, UDP, UDPLag, and SYN were captured on Day 2. In addition, it contains 80 features that were extracted using CICFlowMeter [15].

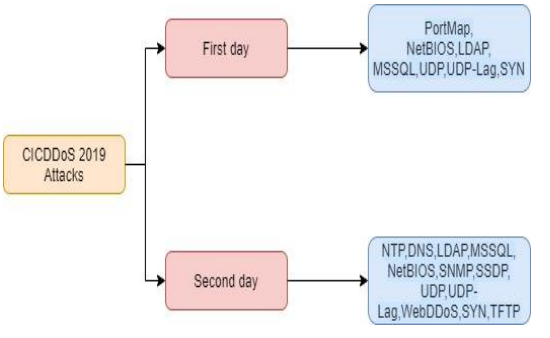

Fig. 7. Two days attack information of CICDDoS2019.

2.1.6 CIRA-CIC-DoHBrw-2020

DoHBrw2020 was developed by the Canadian Institute for Cybersecurity (CIC) project funded by the Canadian Internet Registration Authority in 2020. It contains DNS over HTTPS(DoH) traffic categorized into benign and malicious DoH traffic and non-DoH traffic of the top 10k Alexa websites, browsers, and tunneling tools. It has two layers. In the 1st layer, the

collected traffic is categorized as DoH and non-DoH by using a statistical features classifier. A time-series classifier classified DoH traffic as benign or malicious at the second layer. DoH meter extracts 28 statistical and time-series features from PCAP files [16].

2.2 Data imbalance problem in datasets

Class imbalance in data remains a challenge that impedes the effectiveness of most IDS. The IDS datasets like NSL-KDD, CIC-IDS 2017, CICIDS2018, etc., have a large amount of data associated with different classes. The class with more instances is considered the majority, whereas the class with the fewest instances is regarded as the minority class. When the ratio of data occupied by each class is not evenly distributed, the model may bias towards the majority class, which causes a high FAR. Therefore, data balancing is crucial in improving the model's performance.

In the literature, to handle an imbalance in the data, researchers used various techniques like Synthetic Minority Over-sampling Technique (SMOTE) [17], K-means SMOTE [17], KNN-SMOTE [18], SVM-SMOTE [19], and SMOTE-ENN [20], Borderline SMOTE [21], Generative Adversarial Nets (GAN) [22], reservoir sampling [23], etc. to generate minority samples. Fig. 8 shows the working mechanism of SMOTE.

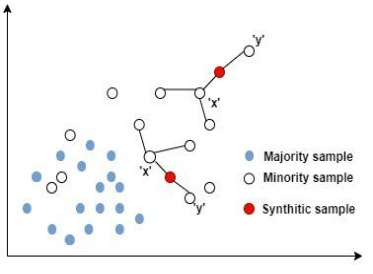

Fig. 8. Working mechanism of SMOTE.

SMOTE is an oversampling approach in which new synthetic samples are generated using existing minority-class samples

- It initially chooses a minority data point, ‘x,’ and identifies ‘k’ nearest neighbors
- Then it generates a synthetic data point by randomly choosing one of the closest k neighbors, "y," by joining "x" and "y" to create a line segment.

The following sub-section presents various IDS that use balancing techniques to improve the model's performance.

IDS applications with class balancing

Bedi et al. [24] dealt with the class imbalance problem in the NSL-KDD dataset using a novel IDS termed Siam-IDS built utilizing Siamese Neural Network. This model uses a similarity score between input pairs to distinguish samples from different classes. The NSL-KDD training dataset was used to train Siam-IDS with similar and dissimilar input pairings. Siam-IDS employed a contrastive loss function to maximize the similarity between similar input pairs while reducing the similarity between dissimilar input pairs.

To handle an imbalance problem in the NSL-KDD, CIDDS-001, and CIC-IDS 2017 datasets, Gupta et al. [25] suggested LIO-IDS. They used Borderline-SMOTE, SVM-SMOTE, and Random Oversampling approaches to balance the data. Further, in stage one, they used LSTM to classify where the data was normal or attack. Finally, they used random forest and Bagging ensembles in stage two to identify the attack type.

Hongpo et al. [26] proposed a model termed SGM-CNN. Initially, for balancing the data, they used SMOTE for oversampling and clustering-based Gaussian Mixture Model for under-sampling. Further, the resampled data was trained with CNN. Finally, when SGM-CNN is compared with other balancing techniques like Random oversampling, K-means + SMOTE, SMOTE, ADASYN and random under sampling+ SMOTE. SGM-CNN detects attacks accurately on UNSW-NB15 and CIC-IDS 2017 datasets with DR of 99.74% and 96.54%.

To increase attack detection, a new hybrid oversampling model based on GAN was presented by Li et al.[27]. The model is divided into three phases. The optimal features are extracted in phase one by Information Gain and Principal Component analysis (PCA). In phase two, DBSCAN is used for data clustering, and in phase three, synthetic data is generated by Wasserstein GAN Divergence. The model is tested on datasets like NSL-KDD, Kyoto2006, and UNSW-NB15 by 6 methods such as XGBM, SVM, Logistic Regression, RF, KNN, and DT. Their experimental results show that their model achieved a high F1-score with XGBoost classifier in comparison with SMOTE, a traditional oversampling method.

Ranjit et al. [23] proposed a dual-stage intrusion detection framework that will remain stable even with high-class imbalanced data using reservoir sampling for generating synthetic samples and subsequent classification was performed using the J48Consolidated algorithm. It is observed that the model is in the conceptual stage. The class imbalance in IDS datasets will lead to poor attack detection. Generally, malicious samples are far fewer than normal samples, resulting in a substantial bias in favor of the normal class. So, handling class imbalance is essential in IDS.

3. EVALUATION METRICS

The frequently utilized evaluation metrics for calculating the performance of ML and DL algorithms

for IDS are presented in this section which is shown in Table 3. The attributes employed in the confusion matrix are the basis for these evaluation measures. They are depicted in Table 2.

Table 2. Confusion Matrix.

	Predicted as Positive	**Predicted as Negative**
Labeled as Positive	True Positive (TP)	False Negative (FN)
Labeled as Negative	False Positive (FP)	True Negative (TN)

TP: The attack data is correctly predicted as an attack.
TN: Normal data is correctly predicted as normal
FP: Predicts normal as an attack.
FN: Predicts attack as normal.

Table 3. Performance metrics used to evaluate IDS.

Performance Metric	Description	Formulae
Accuracy (Acc)	It is the ratio of correctly predicted samples to total samples.	$\frac{TP+TN}{TP+FN+TN+FP}$
Precision (Per)	It is the ratio of correctly predicted attacks to total attacks.	$\frac{TP}{FP+TP}$
Recall (Re)	It is also referred as True Positive Rate (TPR) or Detection Rate (DR). It can be defined as the ratio of detected attacks to actual attacks.	$\frac{TP}{FN+TP}$
False Negative Rate (FNR)	it is the likelihood that the test will fail to detect a true positive.	$\frac{FN}{FP+TN}$
False Positive Rate (FPR) / False Alarm Rate (FAR)	The proportion of samples misclassified to the total number of non-attack samples.	$\frac{FP}{FP+TN}$
TRUE NEGATIVE RATE (TNR)/ Specificity	It is the proportion of the number of correctly classified negative samples to the number of negative samples.	$\frac{TN}{TN+FP}$
F1-Score	It is defined as the harmonic mean of precision and recall for intrusion detection.	$\frac{2*Precision*Reacll}{\Pr ecision+\mathrm{Re}call}$

AUC-ROC curve: It is one of the most significant evaluation measures for assessing the performance of any classification model. A probability curve displays the TPR against the FPR at different threshold levels [28]. The Area under the Curve (AUC) is a summary of the Receiver Operating Characteristic Curve (ROC) curve that measures a classifier's ability to distinguish between classes.

4. INTRUSION DETECTION BASED ON ML AND DL METHODS

Several applications that try to identify constantly evolving threats and assaults have undergone significant modification and evolution in approaches and algorithms to create strong IDS. Researchers initially used ML algorithms for classification. Further, DL models were used to enhance performance and produce exceptional accuracy with low FAR. ML models depend on how the data is trained, but DL models rely on connections between layers of networks to train data without much human interaction. Table 4 shows other differences between ML and DL models.

Table 4. Differences between ML and DL models

	Machine Learning	Deep Learning
Human involvement	It requires more human involvement	Requires less involvement
Structure	It has a simple Structure	It has a complex Structure
Data	They can train with less data	Requires more data to train.
Computation time	Requires less computational time than DL methods	Computational time is more when compared with ML methods
Accuracy	Provides less when	Provides more

	compared with DL models	when compared with ML models
Hyperparameter tuning	It has limited ways to tune the parameters	Parameters can be tuned in several ways
Hardware	They can be processed with CPUs	Mostly they require high-performance computing devices
Feature Selection	Features to be selected manually	Features are automatically extracted
Data interpretation	Few models can be easily interpreted, like RF and DT. But some models are not easy to understand, like XGBM. and SVM	It is not easy to understand
Layers	It can work effectively with the network having input, output, and hidden layers.	It requires a minimum of three layers
Output	It provides numerical output like classification or score	It gives numerical, text, sound, images and etc.,

The following subsections detail the most frequently utilized ML and DL methods to design an efficient IDS model.

4.1. ML Techniques

Alan Turing [29] stated AI is used in ML, where a computer or machine learns from its prior experiences (input data) and predicts the future. Such a system's performance ought to be at least human-level. ML was used to analyze assaults and security events, including spam mail, social media analytics, user identification, and attack detection [30]. ML models are classified as supervised, unsupervised, and reinforcement learning, as shown in Fig. 9.

Supervised learning uses labeled data in the training phase to detect attacks. It is primarily used in classification problems. The biggest impediment of it is the lack of sufficient labeled data. However, manually labeling data is expensive and time -consuming.

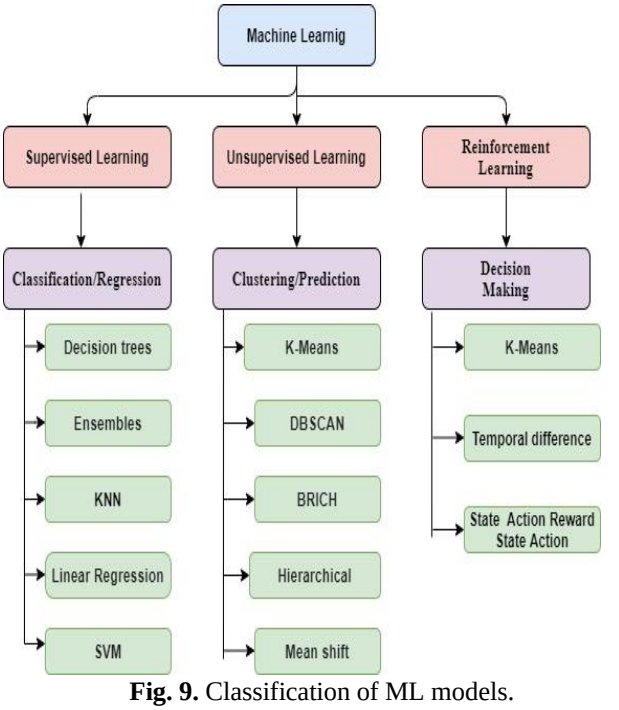

Fig. 9. Classification of ML models.

Unsupervised learning deals with learning tasks with unlabeled or uncompressed data. Clustering is the most widely used unsupervised technique. However, the algorithms are self-employed in detecting and interpreting the data's unique structure. Whereas reinforcement learning is based on a trial-and-error process in which a learning system collects data and takes action. If the action produces a favorable result, a reward is recorded.

On the other hand, if the activity has an unfavorable outcome, the system will learn that similar actions in the future are unlikely to be successful. Model-based and model-free algorithms are the two types of Reinforcement Learning models [31]. Planning is a vital feature of the model-based approach, whereas learning is the central aspect of model-free methods.

The following subsection presents various intrusion detection systems using ML models.

4.1.1. Applications of ML in IDS

Batchu et al. [20] suggested a combinational feature selection technique by combining spearmen correlation, a filter method, and RF, an embedded method. They extracted nine features out of 88 from CICDDoS2019. Then, nine features are passed to various ML classifiers like logistic regression, SVM, KNN, Gradient Boost (GB), and DT. Finally, they proved that GB performs better with an accuracy of 99.97 %.

To reduce the computational time and improve scalability, Borkar et al.[32] proposed a two-stage adaptive SVM classification model to detect known and unknown attacks on wireless sensor networks. In this

model, the dataset is clustered based on the weights of the nodes, and then adaptive chicken swarm optimization (ACS) is used to perform sampling. The benefit of this adaptability is that it primarily tries to reduce the amount of time spent selecting the appropriate cluster head. SVM is used to classify the classes like U2R, probe, R2L, DoS, and unknown attacks.

Raniyah et al. [33] stated a supervised and semi-supervised model by KNN with 5-fold cross-validation to mitigate FAR and increase the DR on the NSL-KDD. In addition, they utilized PCA for dimensionality reduction. Further, the hyperparameters of KNN were tuned to improve the performance by data normalization, identifying the best combinations of nearest neighbors, distance function, and distance weight. Their findings showed that KNN performs better by an accuracy of 98.49 %, Precision 98.71 %, Recall 98.15%, and F1-Score 98.43%. But the suggested models fail when the dataset record size changes, and it is not suitable for identifying real-time attacks.

From the literature, it is evident that SVM has an effective detection rate. However, when dealing with high dimensions, SVM requires more training time than other ML algorithms. Therefore, researchers optimized data to improve SVM training time and detection rate.

Sibi et al. [34] introduce a fusion-based IDS for wireless sensor networks. They presented an optimal Support Vector Machine(O-SVM). The suggested model uses a meta-heuristic whale optimization algorithm (WOA) for efficient kernel selection in the SVM model to reduce the feature space and effectively detect intrusion. Their findings proved model works well with a DR of 95.02%.

Ansam et al. [35] suggested a hybrid model that combines a One-Class SVM and C5 decision tree classifier to identify zero-day attacks on the NSL-KDD dataset. Initially, for detecting known attacks, the C5 decision tree is used. Then, One-Class SVM with an RBF kernel is used for identifying zero-day attacks in the next stage. Finally, the experimental results show the suggested model attains a good detection rate.

To identify unknown, known, and zero-day attacks, Pu et al. [36] proposed an unsupervised anomaly detection approach by combining Sub-Space Clustering and One-Class SVM (SSC-OCSVM). The SSC-OCSVM provides a sorted dissimilarity vector. The samples are classified as potential anomalies if the dissimilarity values exceed the threshold values. Further, to improve the effectiveness of the model, they used F-test to identify relevant features. It is observed that the suggested model requires more computational time because the clusters are processed sequentially. However, it can be processed parallelly to reduce the computation time. Table 5 shows the summary of various ML models in IDS.

4.2. Deep Learning in IDS

Traditional ML approaches struggle to be deployed in large environments because they mainly rely on manually extracted features and lack labeled training datasets. Furthermore, shallow learning cannot analyze high-dimensional datasets in-depth. DL models are typically neural network models with multiple hidden layers. These models may learn very sophisticated non-linear functions, and the models can handle high-dimensional data and extract relevant feature representations in a refined and improved manner [37]. Therefore, it performs better than other traditional machine learning methods. As a result, DL architectures have received more attention nowadays than traditional ML methods. They are widely used in image classification, audio recognition, and anomaly detection [38]. The taxonomy of DL models is shown in Fig. 10.

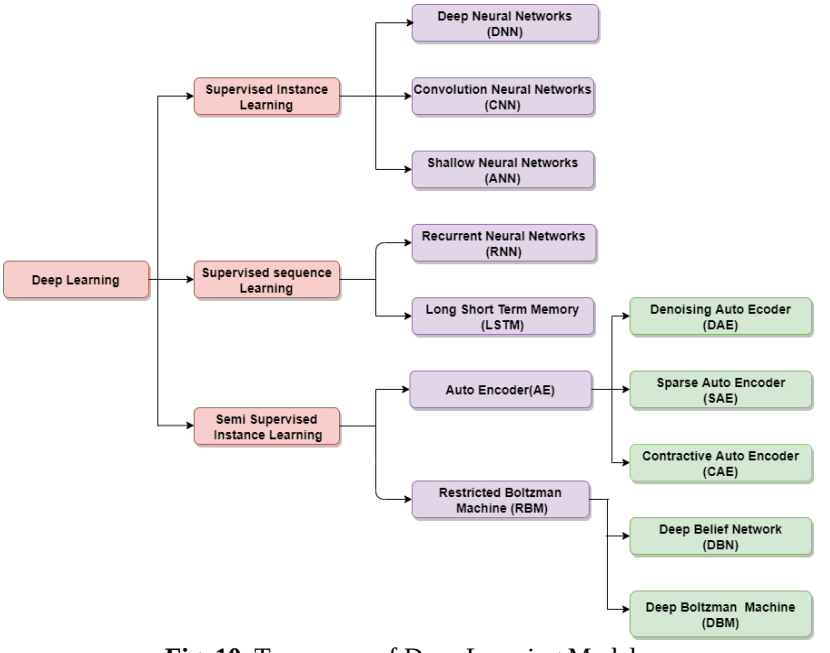

Fig. 10. Taxonomy of Deep Learning Models.

Table 5. Summary of various ML-based IDS with strengths and limitations.

Author /Year	Feature selection method	Classification method	Data set	Type of Attack	Performance Metrics	Strengths	Limitations
[39] 2019	forward feature selection	NB, RF, DT, MLP , and KNN	NSL-KDD	DDoS	Acc, Pre, Re, F1-Score, TPR, FPR	The suggested model works better with RF when compared with other ML models	The FFS guarantees some degree of optimality for smaller feature subsets but does not ensure that the best is found for bigger ones.
[40] 2019	artificial bee colony	AdaBoost	NSL-KDD and ISCXIDS2012 datasets	DoS, Probe, U2R, R2L, Botnet, Infiltration, Bruteforce	Acc, DR, FPR.	To improve the performance of imbalanced data, the AdaBoost meta-algorithm has been utilized in accordance with the correct design. This claim is supported by the proposed method for exact classification of various attacks.	Choosing variables like the number of generations and population size is difficult, which may reduce the performance.
[41] 2020	Info Gain	DT	KDD-99	DoS, U2R and Probe	Acc, Pre, Re, F1-score	The model provides high accuracy.	The dataset was outdated and contained duplicate data, which may mislead the evaluation.

[42] 2020	Sequential backward selection	MLP	NSL-KDD	DoS	Acc, Pre, DR, FAR	Created a feedback mechanism to understand detection errors based on the most recent detection outcomes.	The feedback model can obtain false positive or negative results.
[43] 2020	NB	SVM	UNSWNB15, NSL-KDD, CICIDS2017, and Kyoto 2006+	Binary	Acc, DR, FAR	The high-quality data provided by NB improves the performance of the models in terms of accuracy, DR with less training time	The proposed model did not focus on class balancing, and the interpretability of the model was partially discussed.
[44] 2021	Recursive Feature Elimination (RFE)	DT, SVM, and RF	KDD CUP99 dataset	DoS, U2R, R2L, and Probe	Acc, Pre Re, F1-score	With ten optimal features retrieved by REF. SVM performs better when compared with DT and RF	The data set contains duplicate records, which may affect the model's performance.
[45] 2021	NA	LightGBM	UNSW-NB15, NSL-KDD, and CICIDS2017 data	NA	Acc, FAR	The suggested model performs better when the data is balanced by ADASYN balancing technique.	Feature selection is not utilized in the suggested model, which causes an increase in training and testing time.
[46] 2022	Maximum correlation-based mutual information	Kernel Extreme Learning Machine	KDD-99, NSL-KDD	DoS, U2R, R2L, and Probe	Acc, FPR, Sensitivity, Specificity	The model provides outstanding performance with 18 features on NSL-KDD and KDD-99	The overall Detection rate of the Probe and U2R of KDD-99 was not efficient.
[47] 2022	Wrapper approach	Random Forest, Extra Tree, and Deep Neural Network	IoTID20, Bot-IoT, CICIDS2018, and NSL-KDD.	DoS, Probe, U2R, MITM R2L, Mirai Data theft, Reconnaissance, DoS,DDoS, Botnet, Infiltration, Bruteforce and web attacks	Acc, Pre Re, F1-score	The model has 2 steps. First, in step 1 extra tree is used to detect the data as an attack or normal. Then, in step 2 ensemble of extra trees, RF and MLP are used to detect the type of attacks.	Low detection accuracy for U2R attacks.

4.2.1 Applications of DNN in IDS

It is supervised instance learning which depends on Multi-Layer Perceptron. It is an ANN that has hidden layers between the input and output layers. Each neuron is connected with neurons in successive layers. An activation function acts on the output after each layer of the network, enhancing the effect of network learning. The Back Propagation procedure is being used for data training. It also shrinks the gap between desired and actual values [48].

Maithem et al. [49] recommended an IDS by utilizing DNN for binary and multi-class classification, as shown in Fig. 11. It contains one input layer with 125 neurons, 3 hidden layers of 50,30,2 nodes, respectively,

with Relu as an activation function, and an output layer has 4 neurons with SoftMax as an activation function. Further, they used cross entropy to reduce the loss and Adam optimizer to optimize the data. Finally, they proved that the suggested model performed better.

Cil et al.[50] suggested a DL-based DNN method for detecting Distributed DoS (DDoS) attacks effectively on the CICDDoS-2019 data set. They build their DL-IDS with an input layer of 69 features as nodes and sigmoid as an activation function. Based on the recall values, they have selected 3 hidden layers with 50 neurons each and have one output layer of 2 nodes with SoftMax activation. It is observed that the suggested model detects DDoS accurately, but it has not produced better results when classifying the DDoS attack types

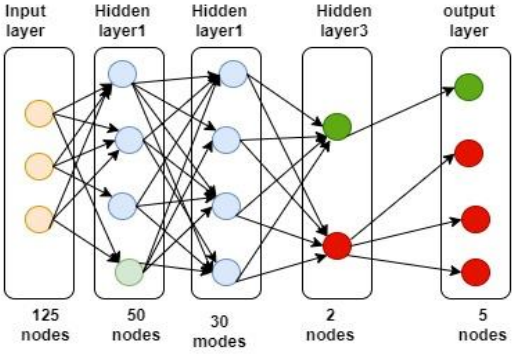

Fig. 11. DNN topology used by Maithem et al. [49]

To handle an imbalance in data and scarcity, Folino et al. [51] suggested an ensemble DNN model with dropout capabilities, skip connections, and a cost-sensitive loss function to learn deep base classifiers from minimal training sets. They did not assess their model in an online IDS context despite their experimental results deal with recurrent and changing behaviors. Further, the model is not suitable for detecting zero-day attacks.

4.2.2 Applications of CNN in IDS

CNN is a supervised instance learning that acquires adequate functions for incoming data. It contains a convolution layer that extracts information, and a fully connected layer determines to which class the input belongs. The convolution layer retrieves unique features, and adding a pooling layer shrinks the volume of the characteristic data. Various CNN models, such as GoogleNet by Szegedy et al.[52], VGG Net by Simonyan et al. [53] and ResNet by He et al. [54].

Several researchers use CNN to detect attacks. For instance, to detect DDoS attacks in SDN networks de Assis et al.[55] suggested a DL- model by using CNN. The suggested architecture is shown in Fig. 12. The model has a detection module and a mitigation module. Initially, they considered network traffic. It contains quantitative and qualitative features. The qualitative features are converted to quantitative by using Shannon Entropy, and these features are fed into CNN for detecting the attacks. Further, the mitigation module used the game theory approach to mitigate DDoS attacks. The suggested model was tested on simulated data and CIC-DDoS 2019 dataset.

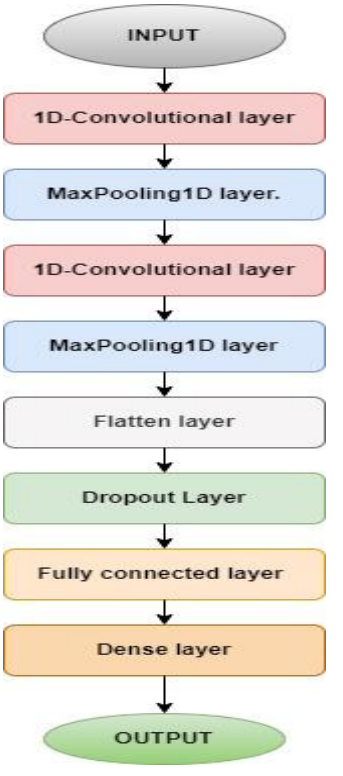

Fig. 12. Architecture of CNN used by de Assis et al.[55]

To detect DDoS attacks, Chen et al.[56] recommended (MC-CNN) a multi-channel CNN. Initially, the features of KDD-99 and CIC-IDS 2017 are partitioned based on a traffic level, packet level, and host level. Further, they are trained by using MC-CNN to detect attacks. To improve training performance, they use incremental learning. The experimental findings show that the model works well with the KDD-99 than CIC-IDS 2017 dataset.

Said et al.[57] suggested a hybrid DL model, a combination of CNN and a novel SD-regularization. To combat overfitting, a standard deviation of the weight matrix was used on the InSDN dataset. Further, CNN is used to select optimal features and to improve the performance, CNN parameters are tuned. They trained classifiers such as RF, SVM, KNN, CNN-RF, CNN-KNN, etc., with original features and nine optimal features separately. Their findings demonstrate that CNN-RF performed admirably across all evaluation metrics

Instead of using publicly available datasets, Yu et al.[58] proposed a hierarchical CNN based on packet bytes. Abstract characteristics of the packet from raw PAC files are extracted in the first level. The representation is built in the second stage from packets in a stream or session. It uses multiple pooling layers as filters, 1-layer TextCNN to get traffic flow, and three fully connected layers to classify the attack patterns. The author conducted tests on the CIC-IDS2017 and CSE-CIC-IDS2018 datasets.

DL uses fully connected neural networks to classify the data. But the limitation of a fully connected network is parameter optimization, the loss of neighborhood information, and it is not translation invariant. To address the issue, Agalit et al.[59] instead of a fully connected neural network, use CNN for feature extraction and a decision tree for classification. Initially, they preprocessed the data using a min-max scaler. Then they utilize one hot encoding to convert numerical data to a greyscale pixel to form an image. Further, these images are passed to their proposed model. To avoid overfitting, they used three pooling layers and three convolution layers to select optimal features. They used an average pooling layer to preserve the features of input data. Finally, the optimal features are trained by using a decision tree to detect assaults. The proposed method was tested on the NSL-KDD dataset.

4.2.3 Application of Autoencoder in IDS

In order to learn how to encode unlabeled data, create new data models, or reduce dimensionality, an AE uses an unsupervised learning approach. It condenses the input into a compact representation that can be used to recreate the information. Generally, AEs are used as the basic units of classifiers for feature variation and anomaly detection. It contains an encoder, decoder, and bottleneck layer, as shown in Fig. 13.

Raj et al.[18] recommended a hybrid detection technique to detect DDoS attacks. To reduce the dimensionality of the network traffic a sparse deep Autoencoder (SDA) is developed, as depicted in Fig. 13. The SDA contains 2 hidden layers in the encoder and decoder, by varying the various activation function in the output layer and hidden layer. The performance was improved when ELU was used in the hidden layer and swish in the output layer with ADAM optimizer and elastic net regularization to extract the optimal features. Further, the optimal set was trained by using tuned LGBM to detect DDoS attacks on the CIC-DDoS 2019 and CIC-IDS 2017 datasets.

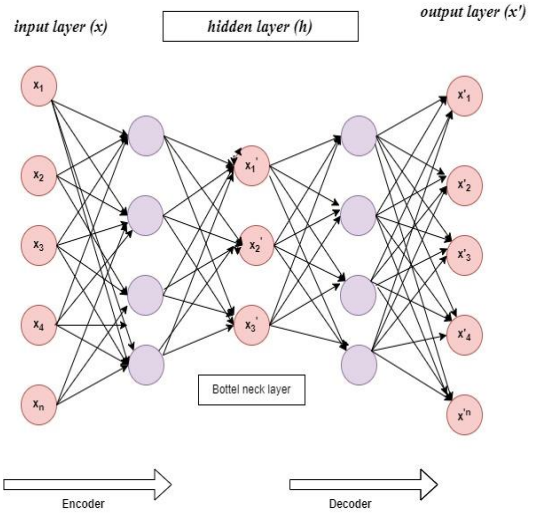

Fig. 13. Architecture of SDA used by Raj et al.[18]

Shone et al.[60] suggested an IDS detect intrusions effectively on KDD Cup -99 and NSL-KDD datasets. Initially, they used a non-symmetric deep autoencoder for dimensionality reduction (NADE). The reduced feature set was trained using a stacked non-symmetric deep autoencoder with a RF classifier to reduce the computational time. The experimental results show that stacked NADE detects attacks accurately.

Kim et al. [61] proposed reinforcement learning that utilizes a deep autoencoder in Q-network to monitor real-time traffic. To achieve the highest predictive performance in online learning systems that take continuous behavior patterns as input and train with significant weights to detect intrusion in a network.

Li et al. [62] proposed an autoencoder technique to detect intrusions on real-time data. A RF is used to obtain the ranks of the features. Further, the affinity propagation clustering model is used to form groups based on the similarity of features. Finally, they are trained by auto encoder to classify the data with a high detection rate.

To handle redundant network data and imbalance ratio, Yao et al.[63] suggested a feature engineering-based IDS for detecting attacks in the Smart Distribution Network. The authors use Borderline- SMOTE to make the classes evenly distributed. Further, the hyperparameters of AE are tuned to retrieve optimal features. Finally, the optimal set is trained by using LGBM to detect attacks in the network. The experimental results show that the AE-LGBM performs well when compared with CNN, LSTM, and LGBM, with an accuracy of 99.90% for KDD-99 and 99.70% for the NSL-KDD dataset.

4.2.4 Applications of Long short-term memory

(LSTM) in IDS

Recurrent neural networks, like LSTMs, are suitable candidates for the network intrusion detection problem because network traffic is sequential. LSTM can remember the output of previous layers in the course of the update process via the internal complex gate structure, which has residual memory. The importance of LSTM is to capture valuable memory and ignore unused memory [64].

To detect intrusion effectively, Althubiti et al.[65] used LSTM. They tweaked the hyperparameters to increase the model performance. It contains an input layer, one hidden layer, an output layer with a learning rate of 0.01, and a Rmsprop optimizer. They calculate loss using categorical cross-entropy. The experiments proved that the suggested technique performs well when compared with Naive Bayes, SVM, and MLP.

Pooja et al.[66] proposed a novel method using Bi-directional LSTM to classify the attacks accurately on UNSW-NB15 and KDDCUP-99 datasets with an accuracy of 99%. They used activation functions like SoftMax and Rectified Linear Activation Function (Relu). To avoid overfitting, they used a dropout layer. Further, they calculated loss using binary cross-entropy and Nesterov-accelerated adaptive moment estimation optimizer. It is observed that the suggested study does not focus on online network attack testing.

Imrana et al. [67] suggested a unique model with a high detection rate for detecting intrusions on the NSL-KDD dataset utilizing bi-directional Long-Short-Term Memory. However, the model was built using an Adam optimizer with a learning rate of 0.01. Furthermore, for calculating loss of binary classification, they used binary cross entropy; for multi-class, they used categorical cross-entropy. Overfitting of the model is addressed by using a dropout layer. Their experimental findings depict that their model performed better when compared with SVM, J48, Random Forest, and Random tree. It is observed that the training time of the suggested technique is more than a normal LSTM.

Zhiqiang et al.[68] proposed an IDS to retrieve optimal features and detect attacks accurately. Their suggested model integrates PCA and empirical model decomposition to retrieve optimal features. They are trained by using LSTM to detect an attack in NSL-KDD, CICIDS2017, KDD-99, and UNSWNB-15. The summary of Applications of DL-based IDS is outlined in Table 6.

Table 6. Summary of various DL-based IDS with strengths and limitations.

Paper/ year	Method	Dataset	Performance metrics	Attack types	Model details	Limitations
[49]	DNN	KDD-99	Acc, Pre, Re, F1-score, AUC, specificity	DoS, R2L, U2R, Probe	The input layer has 125 nodes. It has 3 hidden layers with 50,30,2 nodes and an output layer with 5. Further, it uses RELU activation in the hidden layer and sigmoid activation at the output layer. Finally, the models perform better when using cross-entropy loss and Adam optimizer	The model does not perform better with R2L attacks.
[50] 2021	DNN	CICDDoS 2019	Acc, Pre, Re, F1-Score	DDoS	DNN has an input layer with sigmoid activation. Based on the recall values, it has 3 hidden layers of 50 neurons each with sigmoid activation and. Finally, the output layer has 2 nodes with SoftMax activation	The model performs better with less data, but with more data, performance of the model was reduced.
[55]	CNN	CICDDoS - 2019	Acc, Pre, Re, F1-score	DDoS	The model contains two 1D convolution layers, two 1D Max pool layers, one flatten layer, one dropout, and one fully connected layer	We observed that the model does not provide good sensitivity from the experimental results.
[57] 2021	CNN	InSDN, CIC-IDS2018 and UNSW-NB15	Acc, Pre, Re, F1-score, ROC	Botnet, DDoS,DoS, Probe,U2R, Web attacks and Password-Guessing	It has two 3*3 convolutional layers, one 2*2 max pool layer, one fully connected layer, one dropout, and one flatten layer.	1) It is observed that SD-regularization is not best for feature selection because some of the essential features are lost with any dimensionality reduction algorithm. 2) Computational time to select

						the best shrinking factor for SD-regularization increases.
[58] 2021	CNN	CIC-IDS2017 and CSE-CIC-IDS2018	Acc, Pre, Re, F1-score, ROC	Botnet, DDos,DoS, Infiltration, Bruteforce, and SQL-Injection	It has three convolution layers with one fully connected layer and a flatten layer	1)The models did not perform better with an attack with few records like Infiltration attacks. 2) The data set contains an imbalance that was not addressed
[69] 2022	AE	KDD-99	Acc, Pre, Re, F1-score, FAR	DoS, R2L, U2R, Probe	It contains three non-symmetric deep auto-encoder (SNDAE) with three hidden layers of 12,24,24 neurons. It uses sigmoid as an activation function,	1. The model does not perform better with U2R and R2L.
[60] 2018	NADE+RF	KDD-99 and NSL-KDD	Acc, Pre, Re, F1-score	DoS, R2L, U2R, Probe	The model uses 2 AE, which is termed stacked AE. They contain 3 hidden layers of 14,28,28 neurons, each with sigmoid activation. Further, for classification, they used RF classifier.	1. The model did not perform better with fewer samples. 2. The performance of U2R and R2L attacks was very less.
[65] 2019	LSTM	CIDDS-001	Acc, Pre, Re, FPR	Binary	The model contains six hidden layers with a learning rate of 0.01 for Rmsprop optimizer	1. FPR of the proposed models was high, leading to inappropriate attack detection.
[25] 2021	LSTM	NSL-KDD, CIDDS-001, and CICIDS2017	Acc, Pre, Re, F1-sore, ROC	DoS, R2L, U2R, Probe, Bruteforce, portscan, Ping scan, Infiltration, web attacks, and patator	It contains two layers. In layer one, LSTM detects network data as an attack or normal. Further in, layer two is used to detect the type of attack by the Improved One-vs-One technique	The model performance was good for CIDDS-001 when compared with NSL-KDD, and CICIDS2017
[70] 2022	Contractive Auto Encoder	NSL-KDD and BoT-IoT	AC, Pre, Re, F1-sore	DoS, R2L, U2R, Probe and IG-OS-Fingerprint	1.The optimal features are retrieved by a generalized Mean Grey Wolf Algorithm. 2.For classification, Contractive Auto Encoder is used. In order to reduce the impact of input changes, a CAE uses ElasticNet regularization, a combination of L1 and L2 norm regularization.	Binary classification accuracy is lower than that of five-class categorizations.

4.2.5 Hybrid DL techniques in IDS

AE+LSTM

Zhang et al.[71] presented a hybrid DL technique to decrease high dimensional space to low dimensional space and increase the detection rate. To reduce the training error and dimensionality reduction, they used three-layer AE. The reduced data is fed into LSTM, which contains an input layer, four hidden layers, and an output layer with sigmoid as an activation function. The loss is calculated by binary cross entropy. Further, for optimization, they used the Adam optimizer. It is used to extract the efficient features and classify the samples as Dos, Normal, Probe, R2L, and U2R with an accuracy of 97.6%, 96.8%, 95.3%, 94.8%, and 94.7%, respectively.

Sparse Autoencoder +DNN

Narayana Rao et al. [72] proposed a hybrid DL model which contains a sparse autoencoder with L1 regularization to obtain latent features of the data by imposing sparsity of weights. Further extracted features

are trained by using DNN. Further, to enhance the performance of the suggested model, they identified optimal parameters for the DNN classifier with an input layer, an output layer with sigmoid activation, and a hidden layer with ReLU activation function. By varying the learning rate, they fixed the learning rate as 0.01, and loss is calculated by sparse categorical cross-entropy. Finally, the suggested model was examined on datasets like KDDCup99, UNSW-NB15, and NSL-KDD for classifying binary and multi-class attacks.

DNN+LSTM

Raneem et al. [73] suggested a DL model using the Single-hidden Layer Feed-forward Neural Network (SLFFN) technique in stage one, and DNN is used to predict the sort of intrusive activity in the second stage. To address the data imbalance, they used SMOTE, an oversampling technique. Further, the balanced data is fed into SLFFN to identify the pattern as an attack or nonattack. Further, in stage two, DNN with sequential and LSTM techniques are used to determine the attack types of the IoTID20 dataset. There is no extensive examination of various kinds of assaults in the IoTID20 dataset. They have not provided a comparative analysis of the existing works.

LSTM+CNN

Sun et al. [74] suggested a hybrid DL-based IDS to accurately retrieve temporal and spatial features from network traffic to identify attacks. To detect attacks effectively, they build a classifier by combining CNN and LSTM. CNN takes the input and transforms it into a high-dimensional vector, and it is passed to the LSTM section. LSTM extracts temporal features, and they are fed into a fully connected layer for detecting attack types. The proposed model was tested on CIC-IDS 2017. The experimental results show that CNN-LSTM performs well when compared with single DL models like CNN and LSTM. However, from the experiments, it is observed that it does not perform better with Heartbleed and SSH-Patator attacks because of class imbalance. The performance of minority samples can be improved by balancing the classes.

Lo et al.[75] suggested a hybrid IDS named HyDL-IDS to detect attacks in-vehicle network traffic. To retrieve spatial and temporal features, they used CNN and LSTM. The optimal features are passed to a fully connected layer to detect network attacks. It was tested on a car-hacking dataset, and experimental results show the hybrid model outperformed with an accuracy of 100%. A Summary of various hybrid DL models in IDS is shown in Table 7.

Table 8, shows the comparison of the various ML and DL models for IDS in terms of benefits, and drawbacks

Table 7. Summary of Applications of various hybrid DL models in IDS.

Paper/ year	Method	Dataset	Performance metrics	Attack types	Model details	Limitations
[71] 2020	AE+LSTM	KDD-99	Acc	DoS, R2L, U2R, Probe	1. An AE with three-layer structure is used to extract the features. 2. Further, the optimal features are trained with LSTM for multi-class classification	1. Encoder and decoder part of the Auto Encoder takes more time to reduce the dimensionality of the dataset.
[74] 2020	CNN+ LSTM	CICIDS2017	Acc, FPR, TPR, F1-Score	DoS, Infiltration, PortScan, Heartbleed, and patator	1.To extract spatial and temporal features, initially, CNN is used, which has three convolution layers,5 pooling layers ,5 fully connected layers 2.Further, 2 LSTM channels are used to extract more relevant features. The first LSTM uses linear activation whereas the second LSTM uses RELU activation.	1. Class balancing was not addressed. 2.The model did not perform well with Heartbleed, and patator assaults
[72] 2021	Sparse AE+DNN	KDDCup99, NSL-KDD, and	Acc, DR, F1-Score, FPR	DoS, R2L, U2R, Probe, Generic,	1. Sparse AE is used to reduce the feature dimensions. It was built with one hidden layer with 22 neurons, Relu activation at hidden and an output layer	1.The detection rate of U2R attacks was not reasonable.

		UNSW-NB15		Backdoor, Fuzzers, worms, shellcode, exploits, reconnaissance	3. The extracted features are fed into DNN for classification, which has one hidden layer with Relu activation, optimizer as Adam further to calculate loss sparse categorical loss function is used.	
[75] 2022	CNN+ LSTM	car-hacking	Acc, Pre, Re, F1-score	DoS, Fuzzy, Spoofing	CNN+LSTM is used to extract temporal and spatial features. The model contains two 1D-convolution layers, Max pool layers 2, dropout layers 3, LSTM layer 1, flatten layer 1, dense layer 1.	1.The model cannot detect zero-day attacks

Table 8. Comparison of the various ML and DL models with advantages and disadvantages.

Model	Learning method	Advantages	Limitations
SVM	Supervised	1. Overfitting is less likely because models are more generalized. 2. It can work with a non-linear transformation	1. Requires more training and testing time 2.Kernal selection is difficult 3. Requires more memory
DT	Supervised	1. These are easy to understand 2. It works well with continuous and discrete data	1. More prone to overfitting. 2. Training time is more. 3. Small variations in data may produce different decision trees
KNN	Supervised	1. Retraining is not required. Can add additional data for predictions. 2.It is simple and easy to understand	1. It is computationally expensive 2. Sensitive to missing values and outliers 3. Finding the optimal K value is difficult May overfit
RF	Supervised	1. Works well with more data 2. It maintains accuracy even when a significant portion of the data is missing and has a good technique for forecasting missing data.	1. When the number of trees increases, it requires more training 2. May prone to overfit
LGBM	Supervised	1. It requires less memory 2. Works better with larger datasets 3. It is a histogram-based model which fastens the training process	1. Prone to overfit 2. Does not perform well with smaller datasets
MLP	Supervised	1. Works well with larger data. 2. Predictions are faster once training is completed.	1. Requires more training time 2. Can overfit 3. Difficult to select the number of neurons and layers
CNN	Supervised	1. Extract optimal features automatically.	1. Requires large data to train. 2. It is difficult to implement 3. Sometime, it will overfit
AE	Unsupervised	1. It can be used for feature extraction. 2. It can learn non-linear data	1. It may remove important information. 2. If the parameters are less than the data, there are chances of overfitting
LSTM	unsupervised	1. Avoid the vanishing gradient problem 2. It can give high accuracy in predictions	1. Requires more training time 2. Easy to overfit 3. Requires more memory to train
GAN	unsupervised	1. Used to generate synthetic data 2. It learns internal representations of the data	1. Requires more time to train

			2. Learning to create discrete data, like text, is challenging.

5. EXPLAINABLE ARTIFICIAL INTELLIGENCE IN INTRUSION DETECTION SYSTEMS

There are specific issues in intrusion detection, particularly with the transparency of the systems. Cybersecurity specialists now typically base their conclusions on IDS suggestions. Therefore, model predictions should be clear and understandable. XAI is a specialized field of study to explain the logic behind predictions generated by ML and DL models. XAI is initiated in the year 1970. XAI is gaining more popularity among the research community and application users. Despite its inability to reveal the reasons behind critical decisions, the renaissance of XAI studies is related to the combination of AI with ML across businesses and its impact on the crucial decision-making process [76]. The XAI is categorized into model dependency and scope of Explainability, as depicted in Fig. 14.

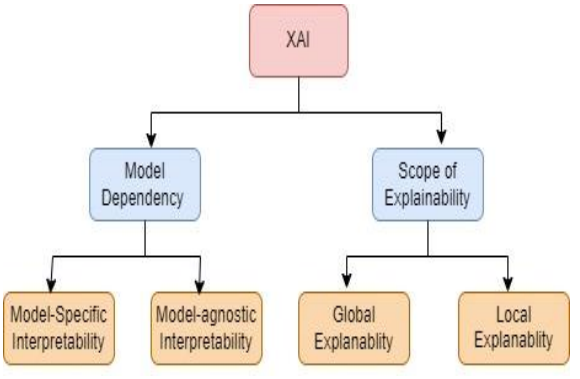

Fig. 14. Approaches of Explainable Artificial Intelligence.

5.1. Model-specific interpretability:

These techniques address a single type of model or a collection of models. They are based on the features and purposes of the particular model, such as tree interpreters. These approaches limit our possibilities for more accurate and representative models since we can only use models that provide a specific interpretation. The limitation of the model is that we can only use models that provide the specific type of interpretation we need, possibly at the expense of utilizing a more predictive and representative model. Therefore model-agnostic interpretability approaches have recently attracted a lot of attention.

5.2. Model-agnostic interpretability

Regardless of complexity, model-agnostic methods and techniques can be used with any ML model. These impartial methods frequently work by analyzing feature input and output combinations. In addition, it may interpret local or global interpretations of the model.

5.3. Global Explainability

Understanding the reasoning behind all potential outcomes is more straightforward when a model is globally explicable. In addition, these models provide insight into the model's overall decision-making process, allowing for a better understanding of the attributions for various input data.

To create a global interpretation tree for various ML models based on their local explanations, Yang et al.[77] suggested model interpretation through recursive partitioning. Their experiments showed that their approach could determine whether a specific ML model is acting rationally or is overfitting to an irrational pattern.

A method based on activation maximization was suggested by Nguyen et al. [78] to synthesize the preferred inputs in neural networks using a learned prior in the form of a deep generator network. Even though literature employs a wide range of approaches to facilitate global interpretability, it may be challenging to establish global model interpretability, especially for models with more parameters.

5.4. Local Explainability

Local interpretation techniques explain how the features contribute to predicting the individual instance. For example, some of the models, like Local interpretable model-agnostic explanations(LIME) [79], Leave One Covariate Out (LOCO) [80], etc.

Barli et al.[81] utilized a local surrogate model LIME to estimate the black-box model predictions. It concentrates on creating local surrogate techniques that can be applied to define specific forecasts.

Lundberg et al. [82] suggested Shapely Additive Explanations (SHAP) as a game-theoretic optimal solution based on Shapley values for model explainability. The relevance of each feature in each forecast is calculated using SHAP. The authors have shown that this model is equivalent to many local interpretable models, including LIME [79] and Layer Wise Relevance Propagation [83].

5.5. Evaluation Methods for Explainability Techniques in IDS

Although there is a significant expansion of XAI in IDS, the literature shows few papers in this field over the past decade.

Alenezi e al.[84] used XGBoost, RF, and Sequential Keras classifiers on two data sets such as Malicious URLs, and Android Malware which contains 5 classes. Further, they utilized XAI techniques such as Kernel, tree, and Deep SHAP to explain the contribution of each feature to build the model.

Mahbooba et al.[85] to improve trust management, they tackled the XAI concept by investigating the DT model in the context of IDS. Specifically, the DT algorithm has investigated its choices using feature engineering and a rule-based model that professionals can understand. The limitation of the work is that they did not handle missing and categorical values; therefore, the model may overfit due to noise, and information gain in DT is biased in favor of features with more levels of data. Further, they did not pay attention to adversarial attacks or IDS development through XAI tools' explanations.

Liu et al.[86] suggested a framework called FAIXID, which integrates XAI and data cleaning methods to aid experts in monitoring the assaults. The framework has 5 modules. Initially, to improve the quality of the data per module, explainability is utilized. Further, the model explainability was handled by an XAI technique such as Boolean Rule Column Generation. In post-module explainability, they used Contrastive Explanations to give a sample-based explanation. Nex,t in attribute module it looks at the features from many perspectives and chooses the interpretive features to give analysts varied levels of explainability that are appropriate for their needs. Finally, the evaluation module assesses the justifications and collects analyst responses.

Table 9. Various XAI-based applications in IDS.

Author/Year	Model		XAI model	Dataset	Contribution	Limitations
	ML	DL				
[83] / 2018	-	DNN	Layer wise relevance propagation	NSL-KDD	1.Experiments were done only on Dos attacks. 2. They do not provide a fully functional implementation with full-textual explanations.	1.Experiments were done only on Dos attacks. 2. They do not provide a fully functional implementation with full-textual explanations.
[87] / 2020	one-vs-all and multi class classifier	-	SHAP	NSL-KDD	1. A unique dataset is treated with domain knowledge. 2.SHAP can be explored on more attacks. 3. SHAP does not work in real-time environments.	1. A unique dataset is treated with domain knowledge. 2.SHAP can be explored on more attacks. 3. SHAP does not work in real-time environments.
[88] / 2021	-	CNN, AE	LIME	NA	1. focuses primarily on time series data with a single variable.	1. focuses primarily on time series data with a single variable.
.[81] / 2021	-	Varatio nal Autoenc oder (VAE)	LIME	CSECICIDS 2018 and CICIDS2017	1. Loss-Based Detection on a VAE is not efficient for the mitigation of anomalies. 2. Considers only DoS and DDoS attacks	1. Loss-Based Detection on a VAE is not efficient for the mitigation of anomalies. 2. Considers only DoS and DDoS attacks
[89] / 2021	-	DNN	LIME, Contrastive Explanations Method, SHAP, Boolean Decision Rules and ProtoDash.	NSL-KDD	1.A unique dataset is treated with domain knowledge. 2. They did not use any XAI framework's explanation to validate the accuracy of the predicted results.	1.A unique dataset is treated with domain knowledge. 2. They did not use any XAI framework's explanation to validate the accuracy of the predicted results.
[90] / 2021	Gradient-boosting, logistic	-	SHAP	RegSOC–KES2021	1. Consider only one attack for the explanation.	1. Consider only one attack for the explanation.

	regression					
[91] / 2021	Random Forest	-	SHAP	CSE-CIC IDS 2018	1. Only tested on a single dataset	1. Only tested on a single dataset
[92] / 2022	Random forest, Decision tree	-	SHAP	NF-BoT-IoT-v2, and NF-ToN-IoT-v2	1. They did not handle an imbalance in the dataset by which the model may bias towards the majority of samples.	1. They did not handle an imbalance in the dataset by which the model may bias towards the majority of samples.
[93] / 2022	Stacked Random Forest	-	SHAP	CIRA-CIC-DoHBrw-2020	1. Only tested on DoH attacks from browser data. 2. Can be tested on other DoH attacks.	1. Only tested on DoH attacks from browser data. 2. Can be tested on other DoH attacks.
[94] / 2022	KNORA-E and KNORA-U	-	SHAP, LIME	CICDDoS-2019	2. Tested only DDoS attacks 3. Experiments were done on a single dataset.	2. Tested only DDoS attacks 3. Experiments were done on a single dataset.

To provide interpretability of ML models, Hariharan et al.[95] suggested an XAI for IDS by comparing the explanations based on the local and global scope on LGBM, Random Forest, and eXtreme Gradient Boosting. They utilized Permutation Importance and SHAP explanation algorithms for the global explanation. They utilized evaluation metrics such as precision, accuracy, and recall for global comparisons. Furthermore, to provide consistency and stability of the model, they employed local explanations such as Contextual Importance and Utility algorithms (CIUA), and Local Interpretable Model-Agnostic Explanation algorithms (LIME) on the NSL-KDD dataset. The limitation of the work is that they focused on a single attack explanation. Table 9 represents relevant XAI-based IDS, including their strengths and weakness.

6. EXISTING SURVEYS WITH ML OR DL TECHNIQUES

Buczak et al. [96] described a thorough and focused literature review of ML and Data Mining techniques for intrusion detection in cyberspace. A short note of the methods like Artificial Neural Networks, Fuzzy Association Rules, Bayesian networks, Association Rules, Clustering, Ensemble Learning, Inductive Learning, Decision tree, Hidden Markov Models, Naive Bayes, Sequential Pattern Mining, and SVM are discussed along with dataset comparisons and performance metrics.

Liao et al. [97] examined various network intrusion methodologies on a cloud platform. They studied and summarized several pattern-based IDS and a Rule-based approach. Moreover, it was stated that security, communication, and administration difficulties are all raised by wireless IDSs. In addition, most wireless IDSs must be evaluated under various mobility and topology situations to ensure protection capacity.

Amna Riaz et al. [98] discussed existing concepts and solutions for intrusion detection in the cloud environment. The limitations and the unique capabilities of the cloud-based IDS with general IDS are mapped and analyzed. The issues related to the lack of datasets for evaluating the performance of intrusion detection in the cloud environment were mentioned.

Ankit et al.[99] discussed the features and limitations of datasets such as CIC-IDS-2017 and CIC-IDS 2018. They outlined the desirable elements, such as Network Configuration, Labeled Dataset, protocols, etc., for creating an optimal dataset. Ansam et al. [100] and studied several machine-learning techniques for detecting intrusions, especially zero-day attacks. Furthermore, the most popular public datasets were used for intrusion detection. Further the benefits and limitations have been discussed. This survey identifies that a new dataset is required for intrusion detection systems and the existing learning techniques are trained by using old datasets like KDDCUP99. In general, KDDCUP99 does not include newer malware attacks.

Leevy et al. [101] provided a detailed survey report on intrusion detection techniques based on CSE-CIC-IDS 2018 dataset. Ilhan et al. [102]provided a comparative study on various ML methods on intrusion detection by using various datasets like UNSW-NB15, ISCX-2012, NSL-KDD, CIC IDS-2018, and CIDDS-001.

Yirui Wu et al. [103] provided a survey report on deep learning methods used for network intrusion detection. In this, Autoencoder, RNN, and Boltzmann Machines were studied. This paper also discusses the difficulties associated while comparing different

datasets like NSL- KDD, KDDCup 99, and metrics for evaluation.

Zeeshan et al. [1] described a thorough and focused literature review based on the ML and DL methods. It informs new researchers about current trends, domain expertise, and field advancement in network security. This paper shows that 60% of the existing algorithms were tested using old datasets. Those datasets failed to address the new modern attacks and proved that existing methodologies limit performance in the real-time environment. The list of surveys carried out on ML and DL approaches in Intrusion Detection is shown in Table 10.

Table 10. List of Surveys compared with our survey.

Paper	Year	Models	Data set	Class balancing	XAI
[96]	2016	√	x	√	x
[97]	2013	x	x	x	x
[98]	2017	x	x	x	x
[42]	2020	√	x	√	x
[101]	2020	√	x	x	√
[103]	2020	x	√	√	x
[1]	2021	√	√	√	x
Proposed review	2022	√	√	√	√

7. DISCUSSION AND RESEARCH CHALLENGES

These days, IDS are an essential part of daily life. However, developing an IDS that recognizes and reacts to various threats and attacks is challenging. As a result, researchers have conducted many studies in the field of IDS for various applications. Some researchers contend that DL, via a neural network, will provide IDS additional flexibility, enabling it to detect and categorize hazardous attacks more successfully.

The comparative analysis of various ML and DL models was provided by Zhang et al.[104]. They stressed that ensemble learning has a good effect on intrusion detection research. Further, DL models require more training time than traditional ML models because DL models are deep in the structure. The performance of DL models depends on the design, hyperparameters, and the number of iterations. Further, Ring et al . [105] thoroughly analyzed network-based intrusion detection systems and emphasized the importance of labeling data while evaluating and training the intrusion detection systems.

To summarize, several researchers used various ML and DL models to detect intrusion in the network effectively. Furthermore, to enhance the performance, some researchers utilized hyperparameter tuning as in [20],[94],[19]. To reduce computational time, some scholars utilized feature selection and feature extraction methods as in [106], [107], [108], [109], [110], [20],[18]. Some researchers looked into combining algorithms to get improved accuracy or a reduced false alarm rate to enhance model implementation, as in [111],[35],[72], and [112]. Finally, over the past few years, a small amount of work has been done by researchers to explain their black box models with XAI techniques like layer-wise relevance propagation [83], SHAP [91], LIME [94], etc.

The subsection provides various challenges in IDS.

7.1. Some open issues and Research Challenges

The majority of the literature was focused on offline analysis. However, there is no advantage of conducting offline research unless we do not test our models in a real-time environment. Therefore, IDS models must be verified using real-world circumstances.

A high-quality IDS dataset is essential for testing and validating IDS Models. However, as mentioned in the previous section, most public datasets have missing values, incomplete network features, raw pcap files, and incomplete CSV files, which may reduce the model's performance. This can be addressed by preprocessing the data by removing duplicate and noisy data.

Literature uses conventional methods like SMOTE and variations of GAN to generate synthetic samples. Future research can enhance classification performance for minority classes with novel class balancing methods.

The IDS datasets contain redundant features, which reduces the performance and increases the training and testing time. Therefore, it is essential to investigate novel techniques to reduce the dimensionality of the dataset. Further, novel nature-inspired algorithms can be explored more in the future.

In order to identify new attacks, detection methods need to be retrained using new training data with minimum training time.

The training time of ML/DL methods can be reduced by selecting appropriate hyperparameters using optimization methods without compromising performance.

Enormous amounts of unlabeled data can be obtained from a network with little effort. Hence novel semi-supervised machine learning algorithms can be proposed to detect intrusions using unlabeled data.

Deep learning models are extremely complex and require a large amount of training time. To address this issue, high-performance computing environment is recommended. However, these environments are highly expensive. As a result, there is a trade-off between performance and cost. Cloud-based GPU platforms and model training services should be investigated.

Another difficulty with ML and DL methods is model overfitting. Overfitting occurs when algorithms are heavily influenced by training data characteristics and attempt to learn patterns that are noisy, non-generalized, and limited to the training data set.

Only a few IDS methods can detect both signature and anomaly-based attacks. In the future, investigations should be done to develop efficient hybrid models to handle known and unknown attacks with low FAR and less computational time.

Blockchain technology has been used by several distributed IDS systems to increase the security of data exchanged during the intrusion detection process. However, the investigated deep learning-based IDS techniques have ignored this problem. As a result, further studies can concentrate on integrating blockchain or other security technologies [113].

Techniques such as online learning and incremental learning can be used to detect intrusions in streaming data. In addition, transfer learning, semi-supervised learning, and reinforcement learning (RL) techniques should be investigated further when creating an IDS to achieve critical goals such as quick training, real-time, and unified models for anomaly detection.

The literature shows that the researchers use black box models like RF, SVM KNN, AE, CNN, etc. Most works exhibit an excellent detection rate and low FPR. However, there are still significant issues in intrusion detection, particularly with the systems' transparency. Nevertheless, the model's predictions should be understandable since cybersecurity specialists now base their decisions on the recommendations of an IDS. Further, there is an opportunity for academics to develop novel XAI in IDS to interpret their model with better explanations.

8. CONCLUSION

The usage of cyberspace increases daily, leading to new and complex attacks. It becomes challenging to detect them with traditional techniques. However, there is now ongoing research in creating novel applications of learning models, such as creating new datasets or combining algorithms. Hence, in this paper, we have reviewed various recent IDS which are based on the ML, DL, and XAI methods. It provides updated relevant information to new researchers and records upcoming signs of progress in the field of Intrusion Detection. Firstly, this study highlights the concept of IDS, the classification of DL and ML algorithms, how DL and ML algorithms are used to design the IDS framework, and the usage of XAI in IDS. It was evident that datasets significantly impact this field because some consider them outdated or to contain redundant information. Choosing an appropriate dataset is a challenging task. Hence, the research provides the most popular datasets used to identify intrusions. Several challenges have been identified in the present review and can be addressed in future work.

9. ACKNOWLEDGMENT

The authors state that they are aware of no personal or professional conflicts that might have appeared to have impacted the findings provided in this study. There are no funding organizations. As part of my doctorate committee, I am working on this project. Observance of ethical guidelines

REFERENCES

[1] Z. Ahmad, A. Shahid Khan, C. Wai Shiang, J. Abdullah, and F. Ahmad, **"Network intrusion detection system: A systematic study of machine learning and deep learning approaches,"** *Trans. Emerg. Telecommun. Technol.*, vol. 32, no. 1, pp. 1–29, 2021, doi: 10.1002/ett.4150.

[2] "• Number of ransomware attacks per year 2022 | Statista." https://www.statista.com/statistics/494947/ransomware-attacks-per-year-worldwide/ (accessed Aug. 09, 2022).

[3] P. A. Cuadra, **"In memoriam / In Memoriam,"** *Songs Cifar Sweet Sea*, no. November, pp. 76–79, 2019, doi: 10.7312/cuad92890-037.

[4] M. V. Brahmam, K. R. Sravan, and M. S. Bhavani, **"Pearson Correlation based Outlier detection in spatial-temporal data of IoT Networks,"** pp. 1–10.

[5] M. Mohammadi *et al.*, **"A comprehensive survey and taxonomy of the SVM-based intrusion detection systems,"** *J. Netw. Comput. Appl.*, vol. 178, no. December 2020, p. 102983, 2021, doi: 10.1016/j.jnca.2021.102983.

[6] R. Kulhare and D. D. Singh, **"Survey paper on intrusion detection techniques,"** *Int. J. Comput. Technol.*, vol. 6, no. 2, pp. 329–335, 2013, doi: 10.24297/ijct.v6i2.3498.

[7] S. Hajj, R. El Sibai, J. Bou Abdo, J. Demerjian, A. Makhoul, and C. Guyeux, **"Anomaly-based intrusion detection systems: The requirements, methods, measurements, and datasets,"** *Trans. Emerg. Telecommun. Technol.*, vol. 32, no. 4, pp. 1–36, 2021, doi: 10.1002/ett.4240.

[8] N. Moustafa and J. Slay, **"UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set),"** *2015 Mil. Commun. Inf. Syst. Conf. MilCIS 2015 - Proc.*, no. November, 2015, doi: 10.1109/MilCIS.2015.7348942.

[9] A. Sonule, M. Kalla, A. Jain, and D. S. Chouhan, **"Unsw-Nb15 Dataset and Machine Learning Based Intrusion Detection Systems,"** *Int. J. Eng. Adv. Technol.*, vol. 9, no. 3, pp. 2638–2648, 2020, doi: 10.35940/ijeat.c5809.029320.

[10] D. Protić, **"Review of KDD Cup '99, NSL-KDD and Kyoto 2006+ datasets,"** *Vojnoteh. Glas.*, vol. 66, no. 3, pp. 580–596, 2018, doi: 10.5937/vojtehg66-16670.

[11] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, **"A detailed analysis of the KDD CUP 99 data set,"** *IEEE Symp. Comput. Intell. Secur. Def. Appl. CISDA 2009*, no. Cisda, pp. 1–6, 2009, doi: 10.1109/CISDA.2009.5356528.

[12] T. Murovič and A. Trost, **"Genetically optimized massively parallel binary neural networks for intrusion detection systems,"** *Comput. Commun.*, vol. 179, no. July, pp. 1–10, 2021, doi: 10.1016/j.comcom.2021.07.015.

[13] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, **"2015 IEEE Symposium on Computational Intelligence for Security and Defense Applications, CISDA 2015 - Proceedings,"** *2015 IEEE Symp. Comput. Intell. Secur. Def. Appl. CISDA 2015 - Proc.*, no. Cisda, pp. 1–6, 2015.

[14] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, **"Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization,"** no. Cic, pp. 108–116, 2018, doi: 10.5220/0006639801080116.

[15] I. Sharafaldin, A. H. Lashkari, S. Hakak, and A. A. Ghorbani, **"Developing realistic distributed denial of service (DDoS) attack dataset and taxonomy,"** *Proc. - Int. Carnahan Conf. Secur. Technol.*, vol. 2019-Octob, no. Cic, 2019, doi: 10.1109/CCST.2019.8888419.

[16] M. Montazerishatoori, L. Davidson, G. Kaur, and A. Habibi Lashkari, **"Detection of DoH Tunnels using Time-series Classification of Encrypted Traffic,"** *Proc. - IEEE 18th Int. Conf. Dependable, Auton. Secur. Comput. IEEE 18th Int. Conf. Pervasive Intell. Comput. IEEE 6th Int. Conf. Cloud Big Data Comput. IEEE 5th Cybe*, pp. 63–70, 2020, doi: 10.1109/DASC-PICom-CBDCom-CyberSciTech49142.2020.00026.

[17] F. Last, G. Douzas, and F. Bacao, **"Oversampling for Imbalanced Learning Based on K-Means and SMOTE,"** pp. 1–19, 2017, doi: 10.1016/j.ins.2018.06.056.

[18] H. S. Raj Kumar Batchu, "**A Hybrid Detection System for DDoS Attacks Based on Deep Sparse Autoencoder and Light Gradient Boost Machine,"** *J. Inf. Knowl. Manag.*, p. 2250071, 2022, doi: RAJ KUMAR BATCHU, Now https://doi.org/10.1142/S021964922250071X.

[19] R. K. Batchu and H. Seetha, **"On Improving the Performance of DDoS attack detection system,"** *Microprocess. Microsyst.*, vol. 93, no. December 2021, p. 104571, 2022, doi: 10.1016/j.micpro.2022.104571.

[20] R. K. Batchu and H. Seetha, **"A generalized machine learning model for DDoS attacks detection using hybrid feature selection and hyperparameter tuning,"** *Comput. Networks*, vol. 200, no. June, p. 108498, 2021, doi: 10.1016/j.comnet.2021.108498.

[21] H. Han, W. Y. Wang, and B. H. Mao, **"Gavel,"** *Lect. Notes Comput. Sci.*, vol. 3644, no. PART I, pp. 878–887, 2005, doi: 10.1007/11538059_91.

[22] J. H. Lee and K. H. Park, **"GAN-based imbalanced data intrusion detection system,"** *Pers. Ubiquitous Comput.*, vol. 25, no. 1, pp. 121–128, 2021, doi: 10.1007/s00779-019-01332-y.

[23] R. Panigrahi and S. Borah, **"Dual-stage intrusion detection for class imbalance scenarios,"** *Comput. Fraud Secur.*, vol. 2019, no. 12, pp. 12–19, 2019, doi: 10.1016/S1361-3723(19)30128-9.

[24] P. Bedi, N. Gupta, and V. Jindal, **"Siam-IDS: Handling class imbalance problem in Intrusion Detection Systems using Siamese Neural Network,"** *Procedia Comput. Sci.*, vol. 171, no. 2019, pp. 780–789, 2020, doi: 10.1016/j.procs.2020.04.085.

[25] N. Gupta, V. Jindal, and P. Bedi, **"LIO-IDS: Handling class imbalance using LSTM and improved one-vs-one technique in intrusion detection system,"** *Comput. Networks*, vol. 192, no. December 2020, 2021, doi: 10.1016/j.comnet.2021.108076.

[26] H. Zhang, L. Huang, C. Q. Wu, and Z. Li, **"An effective convolutional neural network based on SMOTE and Gaussian mixture model for intrusion detection in imbalanced dataset,"** *Comput. Networks*, vol. 177, no. May, 2020, doi: 10.1016/j.comnet.2020.107315.

[27] D. Li, D. Kotani, and Y. Okabe, **"Improving Attack Detection Performance in NIDS Using GAN,"** *Proc. - 2020 IEEE 44th Annu. Comput. Software, Appl. Conf. COMPSAC 2020*, pp. 817–825, 2020, doi: 10.1109/COMPSAC48688.2020.0-162.

[28] **"Understanding AUC - ROC Curve | by Sarang Narkhede | Towards Data Science."** https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5 (accessed Nov. 29, 2021).

[29] A. Q. Review, "MIND," pp. 433–460, 1950.

[30] C. Security, C. Technology, D. I. Edeh, and A. Hakkala, **"Network Intrusion Detection System using Deep Learning Technique,"** no. June, 2021.

[31] Z. Ullah, F. Al-Turjman, L. Mostarda, and R. Gagliardi, **"Applications of Artificial Intelligence and Machine learning in smart cities,"** *Comput. Commun.*, vol. 154, no. December 2019, pp. 313–323, 2020, doi: 10.1016/j.comcom.2020.02.069.

[32] G. M. Borkar, L. H. Patil, D. Dalgade, and A. Hutke, **"A novel clustering approach and adaptive SVM classifier for intrusion detection in WSN: A data mining concept,"** *Sustain. Comput. Informatics Syst.*, vol. 23, pp. 120–135, 2019, doi: 10.1016/j.suscom.2019.06.002.

[33] R. Wazirali, **"An Improved Intrusion Detection System Based on KNN Hyperparameter Tuning and Cross-Validation,"** *Arab. J. Sci. Eng.*, vol. 45, no. 12, pp. 10859–10873, 2020, doi: 10.1007/s13369-020-04907-7.

[34] S. Amaran and R. Madhan Mohan, **"Intrusion Detection System using Optimal Support Vector Machine for Wireless Sensor Networks,"** *Proc. - Int. Conf. Artif. Intell. Smart Syst. ICAIS 2021*, pp. 1100–1104, 2021, doi: 10.1109/ICAIS50930.2021.9395919.

[35] A. Khraisat, I. Gondal, P. Vamplew, J. Kamruzzaman, and A. Alazab, **"Hybrid intrusion detection system based on the stacking ensemble of C5 decision tree classifier and one class support vector machine,"** *Electron.*, vol. 9, no. 1, 2020, doi: 10.3390/electronics9010173.

[36] G. Pu, L. Wang, J. Shen, and F. Dong, "**A hybrid**

unsupervised clustering-based anomaly detection method," *Tsinghua Sci. Technol.*, vol. 26, no. 2, pp. 146–153, 2021, doi: 10.26599/TST.2019.9010051.

[37] A. Thakkar and R. Lohiya, **"A survey on intrusion detection system: feature selection, model, performance measures, application perspective, challenges, and future research directions",** vol. 55, no. 1. Springer Netherlands, 2022.

[38] M. Mohammadi, A. Al-Fuqaha, S. Sorour, and M. Guizani, **"Deep learning for IoT big data and streaming analytics: A survey,"** *IEEE Commun. Surv. Tutorials*, vol. 20, no. 4, pp. 2923–2960, 2018, doi: 10.1109/COMST.2018.2844341.

[39] S. Hosseini and M. Azizi, **"The hybrid technique for DDoS detection with supervised learning algorithms,"** *Comput. Networks*, vol. 158, pp. 35–45, 2019, doi: 10.1016/j.comnet.2019.04.027.

[40] M. i, B. Shirazi, and I. Mahdavi, **"Anomaly network-based intrusion detection system using a reliable hybrid artificial bee colony and AdaBoost algorithms,"** *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 31, no. 4, pp. 541–553, 2019, doi: 10.1016/j.jksuci.2018.03.011.

[41] T. S. Yange, O. Onyekware, and Y. M. Abdulmuminu, "**A Data Analytics System for Network Intrusion Detection Using Decision Tree,"** vol. 8, no. 1, pp. 21–29, 2020, doi: 10.12691/jcsa-8-1-4.

[42] M. Wang, Y. Lu, and J. Qin, **"A dynamic MLP-based DDoS attack detection method using feature selection and feedback,"** *Comput. Secur.*, vol. 88, 2020, doi: 10.1016/j.cose.2019.101645.

[43] J. Gu and S. Lu, **"An effective intrusion detection approach using SVM with na ¨ ıve Bayes feature embedding,"** *Comput. Secur.*, p. 102158, 2020, doi: 10.1016/j.cose.2020.102158.

[44] N. V. Sharma and N. S. Yadav, **"An optimal intrusion detection system using recursive feature elimination and ensemble of classifiers,"** *Microprocess. Microsyst.*, vol. 85, no. July 2020, p. 104293, 2021, doi: 10.1016/j.micpro.2021.104293.

[45] J. Liu, Y. Gao, and F. Hu, **"A fast network intrusion detection system LightGBM,"** *Comput. Secur.*, vol. 106, p. 102289, 2021, doi: 10.1016/j.cose.2021.102289.

[46] S. Gavel, A. S. Raghuvanshi, and S. Tiwari, **"Maximum correlation based mutual information scheme for intrusion detection in the data networks,"** *Expert Syst. Appl.*, vol. 189, no. January 2020, p. 116089, 2022, doi: 10.1016/j.eswa.2021.116089.

[47] C. A. de Souza, C. B. Westphall, and R. B. Machado, **"Two-step ensemble approach for intrusion detection and identification in IoT and fog computing environments,"** *Comput. Electr. Eng.*, vol. 98, no. December 2021, p. 107694, 2022, doi: 10.1016/j.compeleceng.2022.107694.

[48] D. Kwon, H. Kim, J. Kim, S. C. Suh, I. Kim, and K. J. Kim, **"A survey of deep learning-based network anomaly detection,"** *Cluster Comput.*, vol. 22, pp. 949–961, 2019, doi: 10.1007/s10586-017-1117-8.

[49] M. Maithem and G. A. Al-Sultany, **"Network intrusion detection system using deep neural networks,"** *J. Phys. Conf. Ser.*, vol. 1804, no. 1, 2021, doi: 10.1088/1742-6596/1804/1/012138.

[50] A. E. Cil, K. Yildiz, and A. Buldu, **"Detection of DDoS attacks with feed forward based deep neural network model,"** *Expert Syst. Appl.*, vol. 169, no. November 2020, p. 114520, 2021, doi: 10.1016/j.eswa.2020.114520.

[51] F. Folino, G. Folino, M. Guarascio, F. S. Pisani, and L. Pontieri, **"dee,"** *Inf. Fusion*, vol. 72, no. December 2020, pp. 48–69, 2021, doi: 10.1016/j.inffus.2021.02.007.

[52] C. Szegedy *et al.*, **"Going deeper with convolutions,"** *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 07-12-June, pp. 1–9, 2015, doi: 10.1109/CVPR.2015.7298594.

[53] K. Simonyan and A. Zisserman, **"Very deep convolutional networks for large-scale image recognition,"** in *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, 2015, pp. 1–14.

[54] K. He and J. Sun, **"Deep Residual Learning for Image Recognition,"** pp. 1–9.

[55] M. V. O. de Assis, L. F. Carvalho, J. J. P. C. Rodrigues, J. Lloret, and M. L. Proença, **"Near real-time security system applied to SDN environments in IoT networks using convolutional neural network,**" *Comput. Electr. Eng.*, vol. 86, p. 106738, 2020, doi: 10.1016/j.compeleceng.2020.106738.

[56] J. Chen, Y. tao Yang, K. ke Hu, H. bin Zheng, and Z. Wang, **"DAD-MCNN: DDoS attack detection via multi-channel CNN,"** *ACM Int. Conf. Proceeding Ser.*, vol. Part F1481, no. February 2019, pp. 484–488, 2019, doi: 10.1145/3318299.3318329.

[57] M. S. ElSayed, N. A. Le-Khac, M. A. Albahar, and A. Jurcut, **"A novel hybrid model for intrusion detection systems in SDNs based on CNN and a new regularization technique,"** *J. Netw. Comput. Appl.*, vol. 191, no. March, p. 103160, 2021, doi: 10.1016/j.jnca.2021.103160.

[58] L. Yu *et al.*, **"PBCNN: Packet Bytes-based Convolutional Neural Network for Network Intrusion Detection,"** *Comput. Networks*, vol. 194, no. March, p. 108117, 2021, doi: 10.1016/j.comnet.2021.108117.

[59] M. A. Agalit, A. Sadiqui, Y. Khamlichi, and E. M. Chakir, **"Hybrid Intrusion Detection System for Wireless Networks,"** *Lect. Notes Electr. Eng.*, vol. 745, no. June, pp. 507–513, 2022, doi: 10.1007/978-981-33-6893-4_47.

[60] N. Shone, T. N. Ngoc, V. D. Phai, and Q. Shi, **"A Deep Learning Approach to Network Intrusion Detection,"** vol. 2, no. 1, pp. 41–50, 2018.

[61] C. Kim and J. S. Park, **"Designing online network intrusion detection using deep auto-encoder Q-learning,"** *Comput. Electr. Eng.*, vol. 79, 2019, doi: 10.1016/j.compeleceng.2019.106460.

[62] X. K. Li, W. Chen, Q. Zhang, and L. Wu, **"Building Auto-Encoder Intrusion Detection System based on random forest feature selection,"** *Comput. Secur.*, vol. 95, p. 101851, 2020, doi:

10.1016/j.cose.2020.101851.

[63] R. Yao, N. Wang, Z. Liu, P. Chen, D. Ma, and X. Sheng, **"Intrusion detection system in the Smart Distribution Network: A feature engineering based AE-LightGBM approach,"** *Energy Reports*, vol. 7, pp. 353–361, 2021, doi: 10.1016/j.egyr.2021.10.024.

[64] L. Zhang, H. Yan, and Q. Zhu, **"An Improved LSTM Network Intrusion Detection Method,"** *2020 IEEE 6th Int. Conf. Comput. Commun. ICCC 2020*, pp. 1765–1769, 2020, doi: 10.1109/ICCC51575.2020.9344911.

[65] S. A. Althubiti, E. M. Jones, and K. Roy, **"LSTM for Anomaly-Based Network Intrusion Detection,"** *2018 28th Int. Telecommun. Networks Appl. Conf. ITNAC 2018*, pp. 1–3, 2019, doi: 10.1109/ATNAC.2018.8615300.

[66] T. Pooja and S. Purohit, **"Evaluating Neural Networks using Bi-Directional LSTM for Network IDS (Intrusion Detection Systems) in Cyber Security,"** *Glob. Transitions Proc.*, pp. 0–13, 2021, doi: 10.1016/j.gltp.2021.08.017.

[67] Y. Imrana, Y. Xiang, L. Ali, and Z. Abdul-Rauf, **"A bidirectional LSTM deep learning approach for intrusion detection,"** *Expert Syst. Appl.*, vol. 185, no. June, p. 115524, 2021, doi: 10.1016/j.eswa.2021.115524.

[68] L. Zhiqiang, G. Mohiuddin, Z. Jiangbin, M. Asim, and W. Sifei, **"Intrusion detection in wireless sensor network using enhanced empirical based component analysis,"** *Futur. Gener. Comput. Syst.*, vol. 135, pp. 181–193, 2022, doi: 10.1016/j.future.2022.04.024.

[69] E. ul H. Qazi, M. Imran, N. Haider, M. Shoaib, and I. Razzak, **"An intelligent and efficient network intrusion detection system using deep learning,"** *Comput. Electr. Eng.*, vol. 99, no. February 2021, p. 107764, 2022, doi: 10.1016/j.compeleceng.2022.107764.

[70] M. D. Moizuddin and M. V. Jose, **"A bio-inspired hybrid deep learning model for network intrusion detection,"** *Knowledge-Based Syst.*, vol. 238, p. 107894, 2022, doi: 10.1016/j.knosys.2021.10789**4.**

[71] **Y. Zhang, Y. Zhang, N. Zhang, and M. Xiao, "A network intrusion detection method based on deep learning with higher accuracy,"** *Procedia Comput. Sci.*, vol. 174, no. 2019, pp. 50–54, 2020, doi: 10.1016/j.procs.2020.06.055.

[72] K. Narayana Rao, K. Venkata Rao, and P. R. P.V.G.D., **"A hybrid Intrusion Detection System based on Sparse autoencoder and Deep Neural Network,"** *Comput. Commun.*, 2021, doi: 10.1016/j.comcom.2021.08.026.

[73] R. Qaddoura, A. M. Al-Zoubi, H. Faris, and I. Almomani, **"A multi-layer classification approach for intrusion detection in iot networks based on deep learning,"** *Sensors*, vol. 21, no. 9, pp. 1–21, 2021, doi: 10.3390/s21092987.

[74] P. Sun *et al.*, **"DL-IDS: Extracting features using CNN-LSTM hybrid network for intrusion detection system,"** *Secur. Commun. Networks*, vol. 2020, 2020, doi: 10.1155/2020/8890306.

[75] W. Lo, H. Alqahtani, K. Thakur, A. Almadhor, S. Chander, and G. Kumar, **"A hybrid deep learning based intrusion detection system using spatial-temporal representation of in-vehicle network traffic,"** *Veh. Commun.*, vol. 35, p. 100471, 2022, doi: 10.1016/j.vehcom.2022.100471.

[76] S. Neupane *et al.*, "**Explainable Intrusion Detection Systems (X-IDS): A Survey of Current Methods, Challenges, and Opportunities,"** pp. 1–25, 2022, [Online]. Available: http://arxiv.org/abs/2207.06236.

[77] C. Yang, A. Rangarajan, and S. Ranka, **"Global Model Interpretation Via Recursive Partitioning,"** *Proc. - 20th Int. Conf. High Perform. Comput. Commun. 16th Int. Conf. Smart City 4th Int. Conf. Data Sci. Syst. HPCC/SmartCity/DSS 2018*, pp. 1563–1570, 2019, doi: 10.1109/HPCC/SmartCity/DSS.2018.00256.

[78] A. Nguyen, A. Dosovitskiy, J. Yosinski, T. Brox, and J. Clune, **"Synthesizing the preferred inputs for neurons in neural networks via deep generator networks,"** *Adv. Neural Inf. Process. Syst.*, no. Nips, pp. 3395–3403, 2016.

[79] M. T. Ribeiro, S. Singh, and C. Guestrin, **"'Why Should I Trust You?' Explaining the Predictions of Any Classifier,"** *NAACL-HLT 2016 - 2016 Conf. North Am. Chapter Assoc. Comput. Linguist. Hum. Lang. Technol. Proc. Demonstr. Sess.*, pp. 97–101, 2016, doi: 10.18653/v1/n16-3020.

[80] J. Lei, M. G'Sell, A. Rinaldo, R. J. Tibshirani, and L. Wasserman, **"Distribution-Free Predictive Inference for Regression,"** *J. Am. Stat. Assoc.*, vol. 113, no. 523, pp. 1094–1111, 2018, doi: 10.1080/01621459.2017.1307116.

[81] E. M. Bårli, A. Yazidi, E. H. Viedma, and H. Haugerud, **"DoS and DDoS mitigation using Variational Autoencoders,"** *Comput. Networks*, vol. 199, no. February, p. 108399, 2021, doi: 10.1016/j.comnet.2021.108399.

[82] S. M. Lundberg, P. G. Allen, and S.-I. Lee, **"A Unified Approach to Interpreting Model Predictions."** [Online]. Available: https://github.com/slundberg/shap.

[83] K. Amarasinghe and M. Manic, **"Improving user trust on deep neural networks based intrusion detection systems,"** *Proc. IECON 2018 - 44th Annu. Conf. IEEE Ind. Electron. Soc.*, no. Ml, pp. 3262–3268, 2018, doi: 10.1109/IECON.2018.8591322.

[84] R. Alenezi and S. A. Ludwig, **"Explainability of Cybersecurity Threats Data Using SHAP."**

[85] B. Mahbooba, M. Timilsina, R. Sahal, and M. Serrano, **"Explainable Artificial Intelligence (XAI) to Enhance Trust Management in Intrusion Detection Systems Using Decision Tree Model,"** *Complexity*, vol. 2021, 2021, doi: 10.1155/2021/6634811.

[86] H. Liu, C. Zhong, A. Alnusair, and S. R. Islam, **"FAIXID: A Framework for Enhancing AI Explainability of Intrusion Detection Results Using Data Cleaning Techniques,"** *J. Netw. Syst.*

Manag., vol. 29, no. 4, pp. 1–30, 2021, doi: 10.1007/s10922-021-09606-8.

[87] **M. Wang, K. Zheng, Y. Yang, and X. Wang, "An Explainable Machine Learning Framework for Intrusion Detection Systems,"** *IEEE Access*, vol. 8, pp. 73127–73141, 2020, doi: 10.1109/ACCESS.2020.2988359.

[88] I. A. Khan, N. Moustafa, D. Pi, K. M. Sallam, A. Y. Zomaya, and B. Li, **"A New Explainable Deep Learning Framework for Cyber Threat Discovery in Industrial IoT Networks,"** *IEEE Internet Things J.*, vol. 9, no. 13, pp. 11604–11613, 2021, doi: 10.1109/JIOT.2021.3130156.

[89] S. Mane and D. Rao, **"Explaining Network Intrusion Detection System Using Explainable AI Framework,"** no. Ml, pp. 1–10, 2021, [Online]. Available: http://arxiv.org/abs/2103.07110.

[90] L. Wawrowski *et al.*, **"Detecting anomalies and attacks in network traffic monitoring with classification methods and XAI-based explainability,"** *Procedia Comput. Sci.*, vol. 192, no. 2019, pp. 2259–2268, 2021, doi: 10.1016/j.procs.2021.08.239.

[91] S. Wali, I. A. Khan, and S. Member, **"Explainable AI and Random Forest Based Reliable Intrusion Detection system Explainable AI and Random Forest Based Reliable Intrusion Detection system,"** 2021, doi: 10.36227/techrxiv.17169080.v1.

[92] T. T. H. Le, H. Kim, H. Kang, and H. Kim, **"Classification and Explanation for Intrusion Detection System Based on Ensemble Trees and SHAP Method,"** *Sensors*, vol. 22, no. 3, pp. 1–28, 2022, doi: 10.3390/s22031154.

[93] T. Zebin, S. Rezvy, and Y. Luo, "**An Explainable AI-Based Intrusion Detection System for DNS Over HTTPS (DoH) Attacks,"** *IEEE Trans. Inf. Forensics Secur.*, vol. 17, pp. 2339–2349, 2022, doi: 10.1109/tifs.2022.3183390.

[94] R. K. Batchu and H. Seetha, **"An Integrated Approach Explaining the Detection of Distributed Denial of Service Attacks,"** *Comput. Networks*, p. 109269, 2022, doi: 10.1016/j.comnet.2022.109269.

[95] S. Hariharan, R. R. R. Robinson, R. R. Prasad, and C. Thomas, **"XAI for intrusion detection system : comparing explanations based on global and local scope,"** *J. Comput. Virol. Hacking Tech.*, 2022, doi: 10.1007/s11416-022-00441-2.

[96] A. L. Buczak and E. Guven, **"A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection,"** *IEEE Commun. Surv. Tutorials*, vol. 18, no. 2, pp. 1153–1176, 2016, doi: 10.1109/COMST.2015.2494502.

[97] H. J. Liao, C. H. Richard Lin, Y. C. Lin, and K. Y. Tung, **"Intrusion detection system: A comprehensive review,"** *J. Netw. Comput. Appl.*, vol. 36, no. 1, pp. 16–24, 2013, doi: 10.1016/j.jnca.2012.09.004.

[98] A. Riaz *et al.*, **"Intrusion Detection Systems in Cloud Computing: A Contemporary Review of Techniques and Solutions *,"** *J. Inf. Sci. Eng.*, vol. 33, no. 160088, pp. 611–634, 2017, doi: 10.6688/JISE.2017.33.3.2.

[99] A. Thakkar and R. Lohiya, **"ScienceDirect A Review Review of of the the Advancement Advancement in in Intrusion Intrusion Detection Detection Datasets Datasets,"** vol. 00, no. 2019, 2020.

[100] A. Khraisat, I. Gondal, P. Vamplew, and J. Kamruzzaman, **"Survey of intrusion detection systems: techniques, datasets and challenges,"** *Cybersecurity*, vol. 2, no. 1, 2019, doi: 10.1186/s42400-019-0038-7.

[101] J. L. Leevy and T. M. Khoshgoftaar, **"A survey and analysis of intrusion detection models based on CSE-CIC-IDS2018 Big Data,"** *J. Big Data*, vol. 7, no. 1, 2020, doi: 10.1186/s40537-020-00382-x.

[102] I. F. Kilincer, F. Ertam, and A. Sengur, **"Machine learning methods for cyber security intrusion detection: Datasets and comparative study,"** *Comput. Networks*, vol. 188, no. December 2020, p. 107840, 2021, doi: 10.1016/j.comnet.2021.107840.

[103] Y. Wu, D. Wei, and J. Feng, **"Network attacks detection methods based on deep learning techniques: A survey,"** *Secur. Commun. Networks*, vol. 2020, 2020, doi: 10.1155/2020/8872923.

[104] C. Zhang, D. Jia, L. Wang, W. Wang, F. Liu, and A. Yang, **"Comparative research on network intrusion detection methods based on machine learning,"** *Comput. Secur.*, vol. 121, p. 102861, 2022, doi: 10.1016/j.cose.2022.102861.

[105] M. Ring, S. Wunderlich, D. Scheuring, D. Landes, and A. Hotho, **"A survey of network-based intrusion detection data sets,"** *Comput. Secur.*, vol. 86, pp. 147–167, 2019, doi: 10.1016/j.cose.2019.06.005.

[106] H. Alazzam, A. Sharieh, and K. E. Sabri, **"A feature selection algorithm for intrusion detection system based on Pigeon Inspired Optimizer,"** *Expert Syst. Appl.*, vol. 148, 2020, doi: 10.1016/j.eswa.2020.113249.

[107] Z. Halim *et al.*, **"An Effective Genetic Algorithm-Based Feature Selection Method for Intrusion Detection Systems,"** *Comput. Secur.*, vol. 110, p. 102448, 2021, doi: 10.1016/j.cose.2021.102448.

[108] H. Jiang, Z. He, G. Ye, and H. Zhang, **"Network Intrusion Detection Based on PSO-Xgboost Model,"** *IEEE Access*, vol. 8, pp. 58392–58401, 2020, doi: 10.1109/ACCESS.2020.2982418.

[109] S. Mohammadi, H. Mirvaziri, M. Ghazizadeh-Ahsaee, and H. Karimipour, **"Cyber intrusion detection by combined feature selection algorithm,"** *J. Inf. Secur. Appl.*, vol. 44, pp. 80–88, 2019, doi: 10.1016/j.jisa.2018.11.007.

[110] Saroj Kr. Biswas, **"Intrusion Detection Using Machine Learning: A Comparison Study,"** *Int. J. Pure Appl. Math.*, vol. 118, no. 2018, pp. 101–114, 2018, [Online]. Available: http://www.ijpam.eu.

[111] R. M. Elbasiony, E. A. Sallam, T. E. Eltobely, and M. M. Fahmy, **"A hybrid network intrusion detection framework based on random forests and weighted k-means,"** *Ain Shams Eng. J.*, vol. 4, no. 4, pp. 753–762, 2013, doi:

10.1016/j.asej.2013.01.003.

[112] F. Kuang, W. Xu, and S. Zhang, **"A novel hybrid KPCA and SVM with GA model for intrusion detection,"** *Appl. Soft Comput. J.*, vol. 18, pp. 178–184, 2014, doi: 10.1016/j.asoc.2014.01.028.

[113] S. W. Lee *et al.*, **"Towards secure intrusion detection systems using deep learning techniques:** Comprehensive analysis and review," *J. Netw. Comput. Appl.*, vol. 187, no. December 2020, p. 103111, 2021, doi: 10.1016/j.jnca.2021.103111