Advances in Intrusion Detection Systems: Integrating Machine Learning, Deep Learning, IoT, and Federated Learning

Ziadul Amin Chowdhury
School of Science, Engineering and Technology, East Delta University, Bangladesh

M.M. Rahman
School of Science, Engineering and Technology, East Delta University, Bangladesh

Tanvir Azhar
School of Science, Engineering and Technology, East Delta University, Bangladesh

ABSTRACT

The integration of Machine Learning (ML) and Deep Learning (DL) techniques has ushered in a new era of Intrusion Detection Systems (IDS). These advanced approaches significantly enhance detection accuracy, enabling the identification of novel cyber threats and processing massive datasets to ensure robust and reliable network security. The synergy between IoT devices and Federated Learning empowers IDSs to handle distributed data sources and secure edge environments effectively. By leveraging diverse datasets, including network traffic, system logs, and user behavior, IDSs can construct comprehensive threat models and improve their overall effectiveness. This paper investigates cutting-edge methodologies and models based on ML, DL, IoT, and Federated Learning. The challenges associated with deploying DL and ML in IDS have been discussed, and potential avenues for future research have been proposed. This survey aims to guide researchers in adopting contemporary network security and intrusion detection techniques.

General Terms

Intrusion Detection Techniques – IDS; IoT – Internet of Things, XAI – Explainable Artificial Intelligence; Federated Learning – FL.

Keywords

IDS, Deep Learning, Machine Learning, Federated Learning, Cyber security, IoT.

1. INTRODUCTION

The rapid digitization of the world has led to a surge in cyber threats, making cybersecurity a critical concern. As the internet continues to expand, so does the attack surface, exposing individuals and organizations to many risks, including phishing, malware, ransomware, and hacking attempts. To mitigate these risks, robust cybersecurity measures are essential. Regular security assessments and penetration testing can help identify vulnerabilities before malicious actors can exploit them. Additionally, keeping software up-to-date, using strong passwords, and implementing multi-factor authentication can significantly strengthen security defenses [1]. To strengthen security, people and businesses should prioritize software upgrades, strong passwords, and understanding of the current security risks. Furthermore, putting firewalls, multi-factor authentication, and frequent data backups into place helps boosting defenses against cyberattacks [2].

Intrusion Detection Systems (IDS) play a crucial role in cybersecurity by monitoring networks and systems for unauthorized or malicious activity. By analyzing network traffic and system logs, IDS is capable of identifying potential hazards, including malware, DoS attacks, and unprotected access. This proactive approach enables timely alerts and rapid incident response, improve overall cybersecurity defenses [3].

James Anderson further introduced the Intrusion Detection System [4]. Dorothy Denning developed the theoretical foundation for IDS [5]. The first feasible IDS was rule-based; after that, it developed to a rule-based & anomaly-based mixture, also known as a hybrid [6, 7]. Change in detection systems over the last few decades was steady. A significant change through the use of ML However, this change began with the introduction of neural networks in IDS [8]. After that, several other ML Models were used to improve this field. ML techniques include DT algorithms, KNN, SVM models, K-Means, AI techniques, etc. [9, 10]. More recently, there is a propensity towards using of deep neural network technologies, such as the RBM, RNN, MPNN, and others. IDS is being advanced with the aid of these DL models by deploying them in fog, cloud, and IoT-based systems [11-14].

In modeling an IDS, the traditional classifier was used for feature selection. Sometimes it became complex to handle data. To get rid of it, different optimizing algorithms were used, like genetic algorithm (GA), meta-heuristic (MH) optimizing algorithm, etc. These optimizing algorithms enrich the capability of making predictions through feature selections [15, 16]. These optimization algorithms are used in data mining techniques in general.

However, developing and testing any IDS model in a network environment may be difficult and risky. So, all these testing was in lab environment. Though the accuracy & efficiency was showing quite good, still it may suffer in real world. Because data might be too close optimize. Federated learning is a valuable machine learning approach where the data is dispersed among numerous devices or organizations. It is the exercise of training a machine learning model to prevent network intrusion using the collaborative power of multiple devices or entities. It is the opposite of centralized on-premise fashion as it moves the model training to the devices themselves [17].

It can easily adapt to the processing of vast and distributed datasets; besides, it benefits from parallel computing from several devices, boosting both reliability and low response time. Federated Learning helps to keep updating the knowledge of the IDS making it able to respond to emerging threats immediately and adapt to a particular environment, thus enhancing the functionality, productivity, and overall strength of IDS when it comes to addressing the current and future threats in the realm of cybersecurity [18].

In computer network security, IDS is both a defensive and offensive role player against different malicious attacks & cyber threats. Technologies related to IDS are upgrading besides developments and a variety of threats & attacks. Recent technology advances have accelerated the use of FL, ML, DL, and XAI to improve IDS. It helps to keep changing the approaches and win the challenges & blockade of network security.

However, this study is conducted to decide appropriate approaches & methodologies of IDS in different circumstances. In short, it is a try to find the answers of some specified situations as following:

- Strengths & weaknesses of different approaches like ML, DL, AI, FL & XAI, etc. in different situations.
- Differences between input data & results according to that.
- Comparative discussion between earlier studies according to their results & approaches.

Comparative discussion between earlier studies according to their results & approaches.

2. BACKGROUND STUDIES

IDS is a security utility that monitors a computer network or systems for malicious activities or policy violations. IDS can be classified into five types.

Network Intrusion Detection System (NIDS): These systems are deployed at strategic points within a network to monitor all network traffic for signs of malicious activity. NIDS analyze network packets to identify patterns which may identify an attack, such as unauthorized access attempts or data breaches.

Host Intrusion Detection System (HIDS): HIDS are software applications that run on individual hosts or devices, such as servers and workstations. They monitor system-level activities, including file system changes, system calls, and network connections, to detect any suspicious behavior.

Protocol-Based Intrusion Detection System (PIDS): PIDS focus on analyzing specific protocols, such as TCP/IP, to identify protocol-specific attacks. They examine the syntax and semantics of protocol messages to detect anomalies or deviations from expected behavior.

Application Protocol-Based Intrusion Detection System (APIDS): APIDS are designed to monitor application-level traffic, such as HTTP, FTP, and SMTP. They analyze the content of application-specific messages to identify potential attacks, such as SQL injection or cross-site scripting.

Hybrid Intrusion Detection System (HIDS): Hybrid IDS combines the capabilities of multiple IDS types to provide a more comprehensive and effective security solution. They may incorporate NIDS, HIDS, PIDS, and APIDS elements to detect a broader range of threats.

Intrusion detection can be based on two methods: Signature detection & Anomaly detection. The mixture of these two types of detection methods, there is another method can be developed, known as hybrid detection method [20]. To select the proper model for IDS, some of the popular approaches to ML and DL were studied. Below are a few of those algorithms.

2.1. Machine Learning for IDS:

Machine learning, a subset of artificial intelligence, involves training algorithms to learn patterns from data. Supervised learning utilizes labeled datasets to train models to map inputs to correct outputs. It encompasses regression, predicting numerical values, and classification, categorizing data into predefined classes [21]. On the other hand, unsupervised learning works with unlabeled data, discovering patterns and insights without explicit guidance. It is employed for tasks such as clustering, grouping similar data points, association rule mining, identifying relationships between variables, and dimensionality reduction, reducing the number of features while preserving data integrity [22]. Semi-supervised learning combines the best of both worlds, utilizing a small amount of labeled data and a large amount of unlabeled data to train models. It is suitable for both regression and classification problems [23]. Inspired by behavioral psychology, reinforcement learning trains algorithms to make decisions in an environment by learning from rewards and penalties. It can be categorized into model-based and model-free approaches and is commonly used to develop automated systems [24].

Some popular ML and DL approaches were studied to select a proper model. A few of those algorithms have been discussed below.

2.1.1. Logistic Regression:

Logistic Regression is a widely used supervised learning algorithm that estimates the probability of a binary outcome. It assumes a linear relationship between the dependent and independent variables, which may limit its performance in complex scenarios [25]. Logistic regression is used in Intrusion Detection Systems (IDS) to classify network traffic as normal or malevolent. It models the probability of an event (e.g., an intrusion) using a logistic function, making it effective for binary classification tasks. Enhancements like Principal Component Analysis (PCA) can enhance its performance by reducing data dimensionality [26].

2.1.2. Support Vector Machines:

SVMs are powerful supervised learning algorithms for classification and regression. SVMs aim to find the optimal hyperplane that separates data points into different classes. Kernel tricks allow SVMs to handle nonlinearly separable data. While SVMs are effective in high-dimensional spaces, they can be computationally expensive and sensitive to hyperparameter tuning. Enhancements like feature scaling and kernel tricks can further enhance their performance in IDS [27].

2.1.3. Decision Trees:

DT are non-parametric supervised learning techniques that generate a tree-like model of decisions and their possible repercussions. They are easy to understand and can handle both numerical and categorical data. Moreover, they can be prone to overfitting, particularly when the tree is too deep. In IDS, DT categorize network traffic as legitimate or malicious by recursively separating data depending on feature values. Their simplicity, scalability, and simple decision-making processes make them a popular choice [28].

2.1.4. Random Forest:

RF is an ensemble learning method that combines different decision trees to improve prediction accuracy. By averaging the predictions of various trees, Random Forests can reduce overfitting and improve generalization performance. They are resistant to noise and outliers and can handle high-dimensional data. This makes them more successful in IDS, where they can handle large datasets and provide robust predictions [29].

2.2. Deep Learning for IDS:

Deep learning (DL) is a class of neural network algorithms implemented with multiple hidden layers. In this article, some of the widely used DL algorithms will be discussed.

2.2.1. Convolutional Neural Network (CNN):

Convolutional Neural Networks (CNNs) are a class of artificial neural networks most frequently used for image analysis. They can handle high-dimensional data and are noise-resistant, which makes them appropriate for identifying intricate patterns in network traffic. Convolution, pooling, fully linked layers, and activation functions are the four fundamental processes that make up CNN [30].

2.2.2. Long Short-Term Memory (LSTM):

A normal LSTM unit has three gates: an input gate, an output gate, and a forget gate. These gates control how information gets into and out of the memory cell [31]. Because LSTM includes feedback connections, it can handle complete data sequences rather than simply individual data points, in contrast to standard neural networks. Because of this, it is very good at finding trends in sequential data like time series, text, and speech.

It is perfect for sequence prediction problems because of its exceptional ability to capture long-term dependencies. The LSTM network design is made up of three parts, and each one does a different job [32]. These three parts of an LSTM unit are typically called gates. They govern the data that passes through and out of the memory cell, referred to as an LSTM cell. These three gates are the input, output, and forget gates. The primary purpose of this section is to establish whether the information from the previous timestamp should be kept or discarded as irrelevant. In Section 2, the cell attempts to acquire new knowledge by analyzing the input. Finally, the cell copies the altered data from the current timestamp to the succeeding timestamp in the third part. This LSTM cycle is viewed as a single-time step.

Each LSTM cell in a standard feedforward neural network may be thought of as a layer of neurons, each with its own current state and hidden layer.

2.2.3. Restricted Boltzmann Machine (RBM):

RBMs are neural networks that are used to learn probability distributions from input data. They consist of visible and hidden layers, taught using contrastive divergence. RBMs excel in dimensionality reduction, feature extraction, and anomaly detection, making them useful in machine learning and cybersecurity. In Intrusion Detection Systems (IDS), RBMs spot anomalies and detect cyber-attacks by analyzing network traffic patterns. Their ability to process large datasets and learn from unlabeled data improves the efficiency and accuracy of IDS, adding to more effective real-time threat detection and response [33].

2.2.4. Deep Belief Network (DBN):

DBN is a probabilistic generative model that is composed of multiple layers of stochastic, latent variables [34]. It is an array of multiple RBM structures. DBNs operate in two main phases: pre-training and fine-tuning phases. The network finds representations of the input data channel by channel during the pre-training stage. The layers are separately trained in an RBM fashion so the RBMs can learn the representations of the data rapidly. In this phase, the network begins to understand the probability distribution of the inputs and hence the structure of data that is available to it.

In the fine-tuning phase the DBN modifies the parameters already learned by the network in order to specialize in a certain job say classification or regression. This is often done with a process called back propagation whereby the performance of the network on a given task is assessed and the resultant errors are used to change some of the characteristics of the network. In this phase, the neural network is trained using labelled data and it commonly involves supervised learning.

2.2.5. Deep Belief Networks:

Deep Belief Networks (DBNs) enhance Intrusion Detection Systems (IDS) by learning hierarchical data representations and leveraging unsupervised pre-training for improved accuracy. Their scalability and robustness make them effective for real-time, large-scale intrusion detection.

2.2.6. Auto-Encoder (AE):

Autoencoders are self-supervised machine learning models that replicate input data in order to minimize its size. These models are trained as supervised machine learning models, and during inference, they function as unsupervised models. That’s why they are called self-supervised models.

Autoencoders are successful in IDS by learning normal network behavior to identify abnormalities through high reconstruction mistakes. They help identify known and unexpected assaults by extracting important characteristics. Their unsupervised learning reduces manual categorization, and they adapt to evolving traffic patterns. Efficient in real-time computation, they suit high-speed networks well [35].

2.2.7. Generative Adversarial Networks (GAN):

GAN are an approach to generative modeling using DL methods, such as CNN [36]. GANs are an inventive means of training a model that regenerates itself by defining a problem as a supervised learning problem involving two sub-models: the generator model that to be trained to generate new examples, and the discriminator model that tries to classify examples as either real (from the field of study) or fake (produced).

Using a fixed-length arbitrary vector as input, a generator model provides a sample inside the domain. The vector is selected from freely from a Gaussian distribution, and the vector is utilized to seed the generating process. After learning, points in this multivariate vector space will line up with points in the issue domain, providing a shortened version of the data distribution. The discriminator model takes an example from the domain as input (genuine or produced) and offers a binary class label of real or fake (produced). The training dataset offers the real example. The generator model outputs the developed examples. A standard categorization model performs as the discriminator.

2.2.8. Transformers:

Transformers are a modern state-of-the-art NLP model and are considered the evolution of the encoder-decoder architecture [37]. While this architecture relies mainly on RNNs to retrieve ordered data, Transformers completely lack this recurrency. The encoder and decoder are both made up of a stack of N = 6 identical layers. Each of the layers is made of two & three sublayers respectively. Unlike classic DL models like RNNs, transformers employ a technique called self-focus to process input.

2.3. Ensemble Learning:

Ensemble learning is a machine learning technique that aggregates two or more learners in order to produce better predictions. Bias-variance tradeoff is a well-known problem in machine learning. To reduce these errors, ensemble technologies are introduced.

2.3.1. Simple Ensemble Techniques:

Simple ensemble techniques combine predictions from multiple models to produce a final prediction.

Max Voting: The max voting method is usually used for classification problems. This method makes predictions for every data point using a variety of models. The predictions by each model are treated as a 'vote'. The final guess is determined by utilizing the predictions that is obtained from most models.

Averaging: Averaging involves choosing the average of predictions from multiple models. This can be particularly beneficial for regression problems where the final forecast is the mean of predictions from all models. For classification, averaging can be applied to the predicted probabilities for a more confident prediction.

Weighted Average: Weighted averaging is similar, but each model's prediction is given a different weight. The weights can be assigned based on each model's performance on a validation set or tuned using grid or randomized search techniques. This allows models with higher performance to have a greater influence on the final prediction.

2.3.2. Advanced Ensemble Techniques:

Advanced ensemble techniques go beyond basic methods like bagging and boosting to enhance model performance further.

Stacking: Stacking is an ensemble method that creates a new model by adding predictions from several ML models. Predictions on the test set are made using this model.

Blending: Although it only employs a validation set from the train set to generate predictions, blending uses the same methodology as stacking. Stated differently, the predictions are made on the holdout set exclusively, in contrast to stacking. A model is constructed using the holdout set and the predictions, and it is then applied to the test set.

Bagging (Bootstrap Aggregating): This technique makes use of these subsets (bags) in order to obtain a reasonable representation of the distribution (full set). The original set may be larger than the subsets generated for bagging.

Boosting: Boosting is a continuous process where each new model tries to fix the mistakes of the previous model. Models that succeed the preceding model are contingent upon it.

3. CASE STUDIES AND COMPARATIVE ANALYSIS

Here, some specific articles have been studied based on their variety of models, techniques & depth of the analysis. Then all of the studies are summarized in 3 different tables for the ease of comparative analysis.

In table 1, all reviewed studies based on ML have been summarized. In these articles, several ML algorithms have been proposed and tested in this area of IDS, like K Nearest Neighbor (KNNs) [38] and Linear Regressions (LR) [39].

Some research claim that optimized ANNs are able to identify patterns in input data and use those patterns to inform predictions [40]. Data may be efficiently divided into many classes using SVM. Due to their ability to handle both continuous and labeled data, DTs and RFs are widely used machine learning methods for IDS. Additionally, compared to the previous research, their accuracy is slightly higher. However, a variety of variables, the particular issue being addressed, and the training and implementation resources available, impact which ML is optimum for IDS [41].

Table 2 provides a thorough summary of the several DL-based IDSs used in cybersecurity. Among the methods are DNN, FDDNN, RNN, and others. In certain studies, multiple algorithms were used. In that instance, the model in this table that has the highest accuracy has been used as a reference. Table 3 illustrates how FL has become a viable strategy for IDS. enabling collaboration between several stakeholders in the training of a global model without compromising privacy. By lowering the chance of data breaches and safeguarding sensitive data, FL provides benefits over conventional centralized machine learning techniques. FedSVM, FedELM, FedAE, and other FL techniques have been proposed for IDS.

From these studies, few earlier studies will be discussed in a bit detail. Such as, N. Bindra and M. Sood is one of those studies which used [42] unprocessed data for intrusion detection. The primary focus of this study is to identify the most effective algorithm for intrusion detection. The other focal point of conducting this study is to identify DDoS attack. They have chosen CIC IDS 2017 as their dataset.

According to the authors, they built DDoS Detector, five ML based models for network classifications. Dataset was included Web-based, brute force, DDoS, Infiltration, Heart-bleed, Bot, Scan, etc. In this study, primarily the experiment was run without data pre-processing. It is found that the time taken by most of the classifiers was astonishingly high. Then the problem areas were narrowed down. After processing & fine tuning the data, the program was executed nicely. There may have some issue with the model due to data overfitting or underfitting. In this study, authors used k fold cross validation, where k=10. In between LR, KNN, RF, Gaussian NB, Linear SVM & some other ML model, RF (Random Forest) had the highest accuracy of classification, and it is about 0.961. Standard deviation is 0.1.

Table 1: Different ML studies and their comparisons

Authors	ML Approach	Data Set	Accuracy	Attack Type	Year	Ref
Ü Çavuşoğlu	Hybrid layered IDS	NSL-KDD	99.5%	Dos, U2R, R2L, Probe	2019	[43]
J. Ren et al.	DO_IDS	UNSW-NB15	93%	Fuzzers, Backdoors, DoS, Shellcode, Worms, etc.	2019	[44]
N. Bindra et al.	Random Forest	CICIDS2017	96%	DDoS	2019	[42]
Alqahtani et al.	Random Forest (RF),	KDD'99	94%	DoS, U2R, R2L, PROBE	2020	[45]
T. Saranya et al.	LDA, RF and CART	KDD'99	98%	DoS, U2R, R2L, PROBE	2020	[46]
K. Sai Kiran	SVM, Adaboost	Sensor480	98%	Generic	2020	[47]
M. Asif et al.	MR-IMID	NSL-KDD	95.7%	DoS, U2R, R2L, PROBE	2021	[48]
M. Sarhan et al.	Decision Tree	NF-ToN-IoT	99.6%	DoS, Fuzzers, Generic, Infiltration, Worms, etc.	2021	[49]
A. Raghuvanshi et al.	SVM	NSL-KDD	98%	DoS, U2R, R2L, PROBE	2022	[50]
M.B. Pranto et al.	RF	NSL-KDD	99.5%	DoS, U2R, R2L, PROBE	2022	[41]

G. Logeswari et al.	HFS-LGBM	NSL-KDD	98.7%	DoS, U2R, R2L, PROBE	2023	[51]
Avtar Singh et al.	SVM-RF	CICDDoS2019	99.1%	DoS, U2R, R2L, PROBE	2024	[52]

Table 2: Different DL studies and their comparisons

Authors	DL Algorithm	Dataset	Accuracy	Attack Type	Year	Ref
F A Khan et al.	TSDL	KDD99	99.99%	DoS, U2R, R2L, PROBE	2019	[53]
Ge et al.	FFNN	BoT-IoT	99.4%	DoS, DDoS, Info Teft, etc.	2019	[54]
Su et al.	BAT-MC	NSL-KDD	84.25%	DoS, R2L, U2R, Probe	2020	[55]
Boukhalfa et al.	LSTM	NSL-KDD	99.98%	DoS, R2L, U2R, Probe	2020	[56]
Mighan et al.	SAE–SVM	UNB ISCX 2012	95.98%	Fuzzers, Backdoors, DoS, Shellcode, Worms, etc.	2021	[57]
Ashiku et al.	FNN-CNN	UNSW-NB15	95.4%	Fuzzers, Backdoors, DoS, Shellcode, Worms, etc.	2021	[58]
A Kumaar et al.	ImmuneNet	CIC-Bell-DNS 2021	99.19%	Phishing, Malware, Spam	2022	[59]
Houda et al.	XAI	UNSW-NB15	99%	Fuzzers, Backdoors, DoS, Shellcode, Worms, etc.	2022	[60]
Figueiredo et al.	Stacked-LSTM	CICIDS2017	99%	DDoS	2023	[61]
EUH Qazi et al.	HDLNIDS	CICIDS2018	98.90%	Brute-Force, Infiltration, BotNet	2023	[62]
Devendiran et al.	Dugat-LSTM	NSL-KDD	99.65%	DoS, R2L, U2R, Probe	2024	[63]

Table 3: Different FL studies and their comparisons

Authors	DL Algorithm	Dataset	Accuracy	Attack Type	Year	Ref
Chen et al.	FedAGRU	CICIDS2017	98.82%	DoS Attack	2020	[64]
Mothukuri, V., et al.	FedAVG	CICFlowmeter	95.5%	Man in the Middle, DDoS	2021	[65]
Rey V. et al.	FedAVG	N-BaIoT	98.59%	DDoS, Worm, SQL Injection	2022	[66]
Sharan et al.	FedAVG	NF-BoT-IoT-v2	93.08%	DoS, DDoS, Info Theft, etc.	2023	[67]
Bukhari et al.	Fed-SCNN-Bi-LSTM	CICIDS2017	99.99%	DoS Attack	2024	[68]

4. EVALUATION OF ML MODELS:

During the evaluation of an ML model, it is necessary to analyze its prediction capacity, generalize potential, and overall quality. Evaluation metrics provide objective criteria to measure these aspects.

4.1.Confusion Metrics:

A confusion metrics is a summary of correct and incorrect predictions and helps visualize the outcomes. It categorizes predictions into four categories:

True Positive: Correctly estimated favorable circumstances.

True Negative: Correctly estimated negative instances.

False Positive (FP): Incorrectly projected positive instances.

False Negative (FN): Incorrect predictions of negative instances.

4.2.Accuracy:

Accuracy is a metric that measures how often a machine learning model correctly predicts the outcome. Calculating accuracy, the following equation can be used,

$$Accuracy = \frac{Number\ of\ Currect\ Predictions\ (TP + TN)}{Total\ Number\ of\ Predictions\ (TP + TN + FP + FN)}$$

4.3.Precision:

Precision is the ratio of correctly predicted favorable circumstances (TP) to a total number of classified positive samples.

$$Precision = \frac{TP}{TP + FP}$$

4.4.Recall:

The recall is calculated as the ratio between the numbers of Positive samples correctly identified as positive instances to the total number of affirmative samples. It helps us to measure how many positive samples were correctly classified by the ML model.

$$Recall = \frac{TP}{TP + FN}$$

4.5.F_β Score:

The F_β score is a variant of the F_1 score. F_1 score is the harmonic mean of precision and recall. When β = 1, F_β = F_1. Mathematically, Fβ score is given by:

$$F\beta\ Score = \frac{(1 + \beta^2) \cdot P \cdot R}{\beta^2 \cdot P + R}$$

Here, P = Precision, R = Recall. If β is 1, then, F_β = F1 &

$$F1\ Score = \frac{2PR}{P + R}$$

5. CHALLENGES OF EXISTING STUDIES:

There are a lot of studies conducted on network IDS based on machine learning, deep learning, and federated learning. In this article, goal is set to analyze almost all of the areas to find the loopholes & try to find some solutions of it.

Most of the studies used the dataset NSL-KDD, KDD'99, UNSW-NB15, etc. This is due to the lake of dataset. Almost all of the studies conducted were based on the dataset that is used several times & most of the studies followed a similar algorithm. A few studies tried to go through the preprocessed dataset. But later, they had to modify & filter it due to processing time.

Most of the studies have shown that proposed IDS according to the study is capable of detecting threats in the network. But almost all these studies conducted in a controlled environment & with the labeled dataset. Nevertheless, excellent performance in actual contexts is not assured even if the models attain high precision on test sets [69].

Most research prioritize the detection outcomes; consequently, they frequently apply intricate models and expensive data preparation approaches, resulting to poor efficiency. IDS should detect any attack real-time. Currently different approaches of federated learning are using to reduce processing time.

6. CONCLUSION:

In a nutshell, incorporating Machine Learning (ML) and Deep Learning (DL) in Intrusion Detection System (IDS) has led to a significant improvement in detecting anomalies with a lower number of false positives. IoT devices and FL make this platform even better than before. With these advancements, IDS have become a smart and highly flexible component of today cybersecurity. Focusing on the earlier discussion, it can be concluded that, it is must to work on processing & execution time of models. In future, the feasibility of collecting and forwarding packet data directly to a machine learning model should be checked. This feasibility checking might introduce some new platform. Besides this, it is time to invent or discover some model that will process data real-time. Processing time of data should be the main focus area to improve. Some of the state-of-the-art technologies like Transformers, GAN has already been involved to some extent to bring some improvements in this area. Most of the current studies gone through a repetitive use of similar dataset. To avoid the loop of using similar dataset, recently developed datasets like OD-IDS2022, which contains almost 4M records of 28 types of attacks must be needed to be studied, analyzed, tested & implemented. It'll open a new path of IDS. Almost all of the earlier studies were developed using a single machine. It is time to use this technology in an uncontrolled but optimized environment or controlled real network where gaining accuracy with low detection rate should not be prioritized.

7. REFERENCES

[1] Wagh, S.K., V.K. Pachghare, and S.R. Kolhe, *Survey on intrusion detection system using machine learning techniques.* International Journal of Computer Applications, 2013. **78**(16): p. 30-37.

[2] Wei, J., et al., *An intrusion detection algorithm based on bag representation with ensemble support vector machine in cloud computing.* Concurrency and Computation: Practice and Experience, 2020. **32**(24): p. e5922.

[3] Othman, S.M., et al., *Survey on intrusion detection system types.* International Journal of Cyber-Security and Digital Forensics, 2018. **7**(4): p. 444-463.

[4] Anderson, J.P., *Computer security threat monitoring and surveillance.* Technical Report, James P. Anderson Company, 1980.

[5] Denning, D.E., *An intrusion-detection model.* IEEE Transactions on software engineering, 1987(2): p. 222-232.

[6] Smaha, S.E. *Haystack: An intrusion detection system*. in *Fourth Aerospace Computer Security Applications Conference*. 1988. Orlando, FL, USA.

[7] Lunt, T.F. *IDES: An intelligent system for detecting intruders*. in *Proceedings of the symposium: computer security, threat and countermeasures*. 1990. Rome, Italy.

[8] Ryan, J., M.-J. Lin, and R. Miikkulainen, *Intrusion detection with neural networks.* Advances in neural information processing systems, 1997. **10**.

[9] Peng, K., et al., *Intrusion detection system based on decision tree over big data in fog environment.* Wireless Communications and Mobile Computing, 2018. **2018**.

[10] Schueller, Q., et al. *A hierarchical intrusion detection system using support vector machine for SDN network in cloud data center*. in *2018 28th International Telecommunication Networks and Applications Conference (ITNAC)*. 2018. IEEE.

[11] Lo, C.-C., C.-C. Huang, and J. Ku. *A cooperative intrusion detection system framework for cloud computing networks*. in *2010 39th International Conference on Parallel Processing Workshops*. 2010. IEEE.

[12] Deshpande, P., et al., *HIDS: A host based intrusion detection system for cloud computing environment.* International Journal of System Assurance Engineering and Management, 2018. **9**: p. 567-576.

[13] Modi, C., et al., *A survey of intrusion detection techniques in cloud.* Journal of network and computer applications, 2013. **36**(1): p. 42-57.

[14] Mohamed, T., et al. *Intelligent Hand Gesture Recognition System Empowered With CNN*. in *2022 International Conference on Cyber Resilience (ICCR)*. 2022. IEEE.

[15] Ghosh, P., et al. *CS-PSO based intrusion detection system in cloud environment*. in *Emerging Technologies in Data Mining and Information Security: Proceedings of IEMIS 2018, Volume 1*. 2019. Springer.

[16] Nguyen, M.T. and K. Kim, *Genetic convolutional neural network for intrusion detection systems.* Future Generation Computer Systems, 2020. **113**: p. 418-427.

[17] Muneer, S., et al., *A Critical Review of Artificial Intelligence Based Approaches in Intrusion Detection: A Comprehensive Analysis.* Journal of Engineering, 2024. **2024**(1): p. 3909173.

[18] Agrawal, S., et al., *Federated learning for intrusion detection system: Concepts, challenges and future directions.* Computer Communications, 2022. **195**: p. 346-361.

[19] Suramwar, M.V. and S. Bansode, *A Survey on different types of Intrusion Detection Systems.* International Journal of Computer Applications, 2015. **122**(16).

[20] Biermann, E., E. Cloete, and L.M. Venter, *A comparison of intrusion detection systems.* Computers & Security, 2001. **20**(8): p. 676-683.

[21] Nasteski, V., *An overview of the supervised machine learning methods.* Horizons. b, 2017. **4**(51-62): p. 56.

[22] Naeem, S., et al., *An unsupervised machine learning algorithms: Comprehensive review.* International Journal of Computing and Digital Systems, 2023.

[23] Van Engelen, J.E. and H.H. Hoos, *A survey on semi-supervised learning.* Machine learning, 2020. **109**(2): p. 373-440.

[24] Qiang, W. and Z. Zhongli. *Reinforcement learning model, algorithms and its application*. in *2011 International Conference on Mechatronic Science, Electric Engineering and Computer (MEC)*. 2011. IEEE.

[25] Peng, C.-Y.J., K.L. Lee, and G.M. Ingersoll, *An*

introduction to logistic regression analysis and reporting. The journal of educational research, 2002. **96**(1): p. 3-14.

[26] Sathya, S.S., R.G. Ramani, and K. Sivaselvi, *Discriminant analysis based feature selection in kdd intrusion dataset.* International Journal of computer applications, 2011. **31**(11): p. 1-7.

[27] Gupta, M. and S. Shrivastava, *Intrusion detection system based on SVM and bee colony.* International Journal of Computer Applications, 2015. **111**(10).

[28] Quinlan, J.R., *Induction of decision trees.* Machine learning, 1986. **1**: p. 81-106.

[29] Breiman, L., *Random forests.* Machine learning, 2001. **45**: p. 5-32.

[30] O'shea, K. and R. Nash, *An introduction to convolutional neural networks.* arXiv preprint arXiv:1511.08458, 2015.

[31] Staudemeyer, R.C. and E.R. Morris, *Understanding LSTM--a tutorial into long short-term memory recurrent neural networks.* arXiv preprint arXiv:1909.09586, 2019.

[32] Qaddour, J. and N. Rajabi, *LSTM Deep Recurrent Neural Network Model for Voltage Abnormality Detection at IoT Gateway.* Int. J. Comput. Appl., 2019. **177**(9): p. 9-13.

[33] Hinton, G., *A Practical Guide to Training Restricted Boltzmann Machines.* Neural Networks: Tricks of the Trade/Springer, 2012.

[34] Hinton, G.E., *Deep belief networks.* Scholarpedia, 2009. **4**(5): p. 5947.

[35] Meyer, D., *Introduction to autoencoders*. 2015.

[36] 36. Creswell, A., et al., *Generative adversarial networks: An overview.* IEEE signal processing magazine, 2018. **35**(1): p. 53-65.

[37] Liu, Y., et al., *Transformer in convolutional neural networks.* arXiv preprint arXiv:2106.03180, 2021. **3**.

[38] Liu, G., et al., *An enhanced intrusion detection model based on improved kNN in WSNs.* Sensors, 2022. **22**(4): p. 1407.

[39] Khammassi, C. and S. Krichen, *A NSGA2-LR wrapper approach for feature selection in network intrusion detection.* Computer Networks, 2020. **172**: p. 107183.

[40] Choraś, M. and M. Pawlicki, *Intrusion detection approach based on optimised artificial neural network.* Neurocomputing, 2021. **452**: p. 705-715.

[41] Pranto, M.B., et al., *Performance of machine learning techniques in anomaly detection with basic feature selection strategy-a network intrusion detection system.* J. Adv. Inf. Technol, 2022. **13**(1).

[42] Bindra, N. and M. Sood, *Detecting DDoS attacks using machine learning techniques and contemporary intrusion detection dataset.* Automatic Control and Computer Sciences, 2019. **53**(5): p. 419-428.

[43] Çavuşoğlu, Ü., *A new hybrid approach for intrusion detection using machine learning methods.* Applied Intelligence, 2019. **49**: p. 2735-2761.

[44] Ren, J., et al., *Building an effective intrusion detection system by using hybrid data optimization based on machine learning algorithms.* Security and communication networks, 2019. **2019**(1): p. 7130868.

[45] Alqahtani, H., et al. *Cyber intrusion detection using machine learning classification techniques*. in *Computing Science, Communication and Security: First International Conference, COMS2 2020, Gujarat, India, March 26–27, 2020, Revised Selected Papers 1*. 2020. Springer.

[46] Saranya, T., et al., *Performance analysis of machine learning algorithms in intrusion detection system: A review.* Procedia Computer Science, 2020. **171**: p. 1251-1260.

[47] Kiran, K.S., et al., *Building a intrusion detection system for IoT environment using machine learning techniques.* Procedia Computer Science, 2020. **171**: p. 2372-2379.

[48] Asif, M., et al., *MapReduce based intelligent model for intrusion detection using machine learning technique.* Journal of King Saud University-Computer and Information Sciences, 2022. **34**(10): p. 9723-9731.

[49] Sarhan, M., et al. *Netflow datasets for machine learning-based network intrusion detection systems*. in *Big Data Technologies and Applications: 10th EAI International Conference, BDTA 2020, and 13th EAI International Conference on Wireless Internet, WiCON 2020, Virtual Event, December 11, 2020, Proceedings 10*. 2021. Springer.

[50] Raghuvanshi, A., et al., *Intrusion detection using machine learning for risk mitigation in IoT-enabled smart irrigation in smart farming.* Journal of Food Quality, 2022. **2022**(1): p. 3955514.

[51] Logeswari, G., S. Bose, and T. Anitha, *An intrusion detection system for sdn using machine learning.* Intelligent Automation & Soft Computing, 2023. **35**(1): p. 867-880.

[52] Singh, A., H. Kaur, and N. Kaur, *A novel DDoS detection and mitigation technique using hybrid machine learning model and redirect illegitimate traffic in SDN network.* Cluster Computing, 2024. **27**(3): p. 3537-3557.

[53] Khan, F.A., et al., *A novel two-stage deep learning model for efficient network intrusion detection.* IEEE Access, 2019. **7**: p. 30373-30385.

[54] Ge, M., et al. *Deep learning-based intrusion detection for IoT networks*. in *2019 IEEE 24th pacific rim international symposium on dependable computing (PRDC)*. 2019. IEEE.

[55] Su, T., et al., *BAT: Deep learning methods on network intrusion detection using NSL-KDD dataset.* IEEE Access, 2020. **8**: p. 29575-29585.

[56] Boukhalfa, A., et al., *LSTM deep learning method for network intrusion detection system.* International Journal of Electrical and Computer Engineering, 2020. **10**(3): p. 3315.

[57] Mighan, S.N. and M. Kahani, *A novel scalable intrusion detection system based on deep learning.* International Journal of Information Security, 2021. **20**(3): p. 387-403.

[58] Ashiku, L. and C. Dagli, *Network intrusion detection system using deep learning.* Procedia Computer Science, 2021. **185**: p. 239-247.

[59] Akshay Kumaar, M., et al., *A hybrid framework for intrusion detection in healthcare systems using deep learning.* Frontiers in Public Health, 2022. **9**: p. 824898.

[60] Abou El Houda, Z., B. Brik, and L. Khoukhi, *"why should i trust your ids?": An explainable deep learning framework for intrusion detection systems in internet of things networks.* IEEE Open Journal of the Communications Society, 2022. **3**: p. 1164-1176.

[61] Figueiredo, J., C. Serrão, and A.M. de Almeida, *Deep learning model transposition for network intrusion detection systems.* Electronics, 2023. **12**(2): p. 293.

[62] Qazi, E.U.H., M.H. Faheem, and T. Zia, *HDLNIDS: hybrid deep-learning-based network intrusion detection system.* Applied Sciences, 2023. **13**(8): p. 4921.

[63] Devendiran, R. and A.V. Turukmane, *Dugat-LSTM: Deep learning based network intrusion detection system using chaotic optimization strategy.* Expert Systems with Applications, 2024. **245**: p. 123027.

[64] Chen, Z., et al., *Intrusion detection for wireless edge networks based on federated learning.* IEEE Access, 2020. **8**: p. 217463-217472.

[65] Mothukuri, V., et al., *Federated-learning-based anomaly detection for IoT security attacks.* IEEE Internet of Things Journal, 2021. **9**(4): p. 2545-2554.

[66] Rey, V., et al., *Federated learning for malware detection in IoT devices.* Computer Networks, 2022. **204**: p. 108693.

[67] Sarhan, M., et al., *Cyber threat intelligence sharing scheme based on federated learning for network intrusion detection.* Journal of Network and Systems Management, 2023. **31**(1): p. 3.

[68] Bukhari, S.M.S., et al., *Secure and privacy-preserving intrusion detection in wireless sensor networks: Federated learning with SCNN-Bi-LSTM for enhanced reliability.* Ad Hoc Networks, 2024. **155**: p. 103407.

[69] Liu, H. and B. Lang, *Machine learning and deep learning methods for intrusion detection systems: A survey.* applied sciences, 2019. **9**(20): p. 4396.