STATISTICS, OPTIMIZATION AND INFORMATION COMPUTING
Stat., Optim. Inf. Comput., Vol. **15**, February 2026, pp 1416–1432.
Published online in International Academic Press (www.IAPress.org)

Comprehensive Study: Machine Learning and Deep Learning Approaches in Intrusion Detection Systems

Azizi Najoua [1,*], Jamali Abdellah [1], Naja Najib [2]

[1] *Computer, Networks, Mobility and Modeling Laboratory-Faculty of Science and Technology, Hassan 1st University, Settat, Morocco; National School of Applied Sciences of Berrechid (ENSA), Settat, Morocco*

[2] *Department of Mathematics, Computer Science and Networking Institut National des Postes et Télécommunications (INPT), Rabat, Morocco*

Abstract This paper presents a synthesis of approaches from various studies aimed at enhancing attack classification using machine learning (ML) and deep learning (DL) models. The works studied cover diverse aspects of cybersecurity, with a particular focus on intrusion detection systems (IDS) and Internet of Things (IoT) security. The paper provides an overview of the datasets used to train ML and DL models, the metrics used to evaluate the performance of these techniques, outlines the process for implementing them, and discusses perspectives and future research directions.

Keywords IoT, Cyber-attacks, ML, DL, Fault tolerance, IDS, Cybersecurity, Intrusion Detection, Prevention Mechanisms, Advanced Algorithms, Performance Metrics, Hybrid Models, Vulnerability Detection

DOI: 10.19139/soic-2310-5070-2782

1. Introduction

Nowadays, the use of connected devices has grown exponentially, generating massive volumes of data exchanged between IoT devices or between these devices and processing servers.
As data exchange continues to grow at scale, cyberspace is facing an increasing number of threats, particularly with the rise of more sophisticated attacks. In response to this rapid evolution, it has become imperative to invest in improving intelligent solutions for intrusion detection, prevention, and attack prediction. In this context, Machine Learning (ML) and Deep Learning (DL) offer particularly promising solutions.
This paper presents an in-depth exploration of ML and DL applications aimed at strengthening cybersecurity, with a particular focus on the Internet of Things (IoT) ecosystem and network traffic monitoring and analysis. The studies examined aim to improve intrusion detection and prevention mechanisms using various datasets, advanced algorithms, and robust performance metrics. Researchers are increasingly combining traditional machine learning (ML) techniques with innovative deep learning (DL) models to better address the evolving nature of cyber threats. This integration highlights the need for continuous adaptation and improvement in intrusion detection strategies. Existing approaches range from the development of effective intrusion detection systems (IDS) within IoT ecosystems and network traffic analysis, to the design of hybrid ML–DL frameworks and advanced vulnerability detection methodologies. The common thread of this research is its commitment to strengthening the resilience and effectiveness

*Correspondence to: Azizi Najoua (Email: n.azizi.doc@uhp.ac.ma). Computer, Networks, Mobility and Modeling Laboratory-Faculty of Science and Technology Hassan 1st University, Settat, Morocco

ISSN 2310-5070 (online) ISSN 2311-004X (print)
Copyright © 2026 International Academic Press

of security frameworks in the face of increasingly dynamic and sophisticated cyber challenges.
The structure of this paper is as follows: The second section presents the process of preparation and normalization of training and validation data for machine learning (ML) and deep learning (DL) models aimed to detecting attacks and intrusions. The first step in this process involves collecting training data. The second step involves the preparation and normalization of the data so that it can be used effectively by the models. The third phase involves selecting the most relevant and crucial features for predicting attacks, an essential step, particularly for ML algorithms. Finally, the last step is dedicated to training and evaluating the model.
The third section focuses on the metrics commonly used to assess the performance of ML and DL models. The fourth section provides detailed information on the data sources, the information presented, and the topologies used to collect training datasets for the models.
The fifth section presents a list of related works on attack classification and detection using machine learning and deep learning models. Finally, the last section discusses future research.

2. Implementation Process of ML and DL Based Intrusion Detection Systems

Intrusion detection poses a significant challenge due to the constantly evolving nature of cyber threats. Machine learning (ML) and deep learning (DL) techniques present a promising solution to tackle this issue. The following method outlines a standard approach, from data collection to the operational implementation of an intrusion detection system, to create robust and adaptive detection frameworks.
Data collection and preprocessing : The first step involves collecting relevant data from diverse sources, ensuring the inclusion of both normal and intrusion data to maintain a balanced dataset. Once data is collected, preprocessing is essential, which includes cleaning the data by removing missing or outlier values and correcting any inconsistencies. Data transformation follows, normalizing numerical data and encoding categorical variables as needed.
After data collection and preprocessing, one of the critical challenges that arises concerns the distribution of classes within the datasets.
Unbalanced datasets can severely affect machine learning performance, as algorithms tend to favor the majority class and produce biased evaluations. Balancing techniques offer potential solutions, but each comes with trade-offs. Undersampling reduces the size of majority classes, which may cause information loss, while oversampling methods such as SMOTE can increase the risk of overfitting by replicating minority patterns. Recent studies confirm that these strategies have a strong impact on performance metrics. For instance, applying SMOTE to the CSE-CIC-IDS2018 dataset increased Recall for minority attacks like FTP and SSH brute force by nearly 15%, although it slightly reduced Precision due to synthetic data variability.[1]. These results highlight the need to adapt preprocessing strategies depending on whether false positives or false negatives are more critical in the IDS context [2] [3]

Feature selection and Detection Latency: Feature selection is a crucial step to identify the most relevant features for intrusion detection. Several techniques can be used for this purpose, including Information Gain (IG), Gain Ratio (GR), ReliefF, Symmetric Uncertainty, Chi-square, and ANOVA (F-test). The comparative analysis of these methods is summarized in table 1.

After feature selection, the model training and evaluation phase begins, where the dataset is divided into training, validation, and test sets. Appropriate ML or DL algorithmsare chosen and trained. Model performance is evaluated using metrics like accuracy, recall, and F1-score, with an emphasis on improving hyperparameters.
Finally, continuous validation and refinement are essential to sustaining the effectiveness of the system. This involves monitoring performance in real-world conditions, collecting feedback for model refinement, and adapting the system to address emerging threats.

Table 1. Comparative analysis of feature selection methods in IDS

Method	Description	Strengths	Limitations
Information Gain (IG)	Measures entropy reduction with respect to target variable.	Simple, effective, widely used.	Biased toward attributes with many values.
Gain Ratio (GR)	Normalized IG to reduce bias.	Corrects IG bias, good for IoT/SDN.	Less frequently applied; favors attributes with few values.
ReliefF	Evaluates feature quality by distinguishing similar instances of different classes.	Captures complex dependencies; robust.	Higher computational cost.
Symmetric Uncertainty	Normalized mutual information (balanced measure).	Symmetric, normalized measure of association.	Rarely used explicitly in IDS experiments.
Chi-square	Tests independence between feature and class label.	Effective for categorical features; efficient.	Requires discretization of continuous variables.
ANOVA (F-test)	Compares group means for continuous variables.	Useful for continuous attributes.	Assumptions of normality and equal variances.

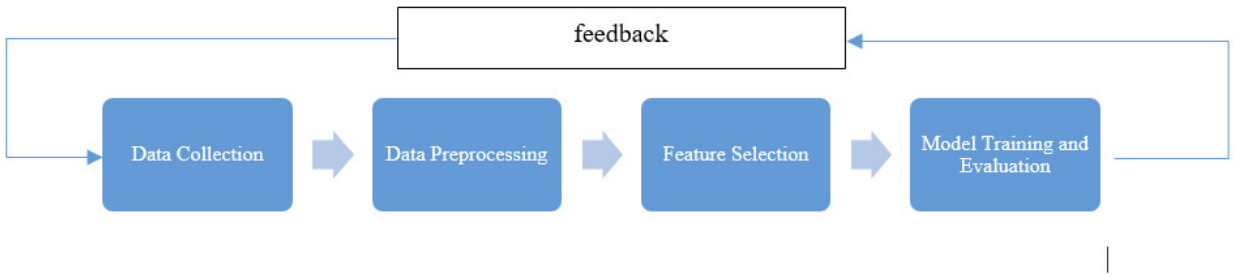

Figure 1. Workflow for Implementing Machine Learning and Deep Learning Models

Figure 1 provides a structured framework for the development and deployment of ML and DL-based intrusion detection systems, emphasizing data collection,data preprocessing, relevant feature selection, and ongoing system refinement. It is important to highlight that each stage must be adapted to IDS-specific requirements, particularly with regard to real-time preprocessing, balanced feature selection, and continuous retraining.

Figure 1 illustrates the ML/DL workflow stages. It is important to highlight that each stage must be adapted to IDS-specific requirements, particularly with regard to real-time preprocessing, balanced feature selection, and continuous retraining.

2.1. IoT-Specific Adaptations for Intrusion Detection System

: The deployment of Intrusion Detection Systems (IDS) in Internet of Things (IoT) environments introduces unique challenges that differ from traditional IT networks. IoT devices are resource-constrained, rely on various communication protocols, and are exposed to both cyber and physical-layer

threats. To ensure that Machine Learning (ML) and Deep Learning (DL) approaches are effective in IoT ecosystems, several specific adaptations have been identified in the literature.

1. Resource limitations and lightweight models : IoT devices generally have limited processing power, memory, and energy. As a result, complex deep learning architectures such as CNNs or large ensembles, although accurate, are often impractical for edge deployment. Several studies emphasize the need for lightweight solutions. For example, study [4] notes that CNNs and RNNs are powerful but resource intensive, and therefore recommends hybrid approaches where DL is applied for feature extraction while lightweight ML handles classification.
The study [5] stresses the importance of resource-efficient IDS for IoT devices and suggests compression and optimization techniques.

2. Distributed and Federated Learning : Centralized training is often impractical in IoT due to bandwidth and privacy concerns. Federated Learning (FL) has been proposed as an alternative, where models are trained locally and only updates are aggregated. While not yet widely applied in IDS, its potential is noted in study [3], which emphasizes distributed resilience in IDS for IoT frameworks.

3. Protocol Diversity and Heterogeneous Traffic Most datasets such as CICIDS2017, CSE-CIC-IDS2018 do not represent IoT protocols such as Zigbee, Z-Wave, MQTT, or CoAP. The lack of protocol diversity is acknowledged in [2], which calls for dataset enrichment to cover IoT traffic. Similarly, [6] highlights the difficulty of building datasets that reflect IoT heterogeneity.

4. Physical and Device-Level Threats IoT devices are exposed to hardware-layer attacks such as side-channel leakage, fault injection, and backdoors. [5] provides a taxonomy of ML / DL techniques to detect vulnerabilities and discusses hardware-assisted mechanisms such as TPMs, PUFs, and ARM TrustZone as complementary layers.

5. Evaluation Criteria for IoT IDS Beyond accuracy, IDS for IoT must be assessed on false positive rate (FPR), inference latency, memory footprint, and energy consumption. As pointed out in [7], accuracy alone can be misleading, especially for imbalanced datasets common in IoT scenarios.

In summary, intrusion detection for IoT requires shifting from accuracy-centric designs toward lightweight, protocol-aware, resource-efficient, and hardware-conscious solutions. Integrating compression techniques, federated learning, and protocol-specific feature engineering provides a foundation for practical IDS in IoT ecosystems.

3. Performance Metrics for Evaluating ML and DL Techniques

In machine learning and data science, evaluating the performance of classification models is vital to ensure their effectiveness and reliability. Classification tasks focus on predicting the category of a given data point, and performance metrics offer a quantitative evaluation of how closely these predictions match the actual class labels. Selecting and understanding performance metrics is essential to compare different models, fine-tuning their parameters, and ultimately choosing the most suitable model for a specific application. This section delves into various performance metrics used in classification.
Accuracy measures the proportion of correct predictions (true positives and true negatives) relative to the total number of predictions.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

Where:

- **TP (True Positive)**: number of instances correctly classified as positive.
- **TN (True Negative)**: number of correctly classified negative instances.
- **FP (False Positive)**: number of instances incorrectly classified as positive.
- **FN (False Negative)**: number of instances incorrectly classified as negative.

Precision is defined as the ratio of correct positive results to the total number of positive results predicted by the model. The mathematical function of precision is defined by the function below.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{2}$$

Recall refers to the ratio of valid positive outcomes to the total number of relevant samples. The mathematical function of recall is defined by the function below.

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{3}$$

Where:

- **TP (True Positive)**: number of instances correctly classified as positive.
- **FN (False Negative)**: number of instances incorrectly classified as negative.

The F1 score is a performance metric used in classification tasks that combines precision and recall into a single value. It is especially useful when there is an imbalance between classes or when both false positives and false negatives carry significant consequences. The mathematical function of the F1 score is defined by the function below.

$$\text{F1-score} = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \tag{4}$$

Confusion matrix presents values for true positives, false positives, true negatives, and false negatives. It allows us to understand, on one hand, the different errors made by a prediction algorithm, but more importantly, to find the several types of errors committed. By analyzing them, it is possible to verify the results that show how these errors occurred.

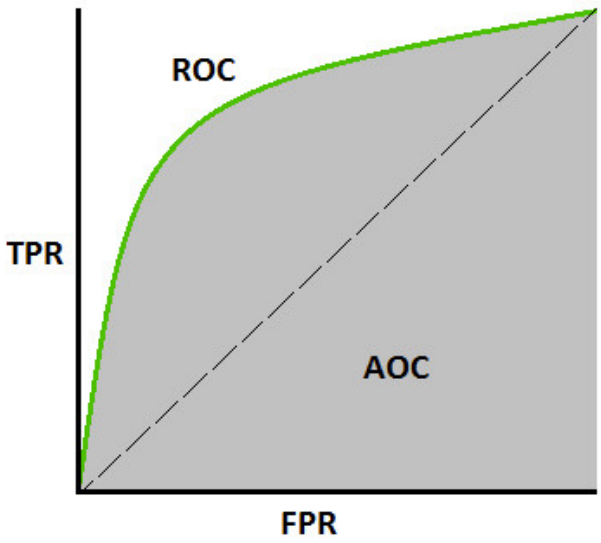

Figure 2. AUC-ROC Curve for Model Performance Evaluation

ROC curve (Receiver Operating Characteristic curve) and AUC curve (Area Under the Curve): The ROC curve shows the true positive rate versus the false positive rate. The AUC measures the model's ability to distinguish between classes.

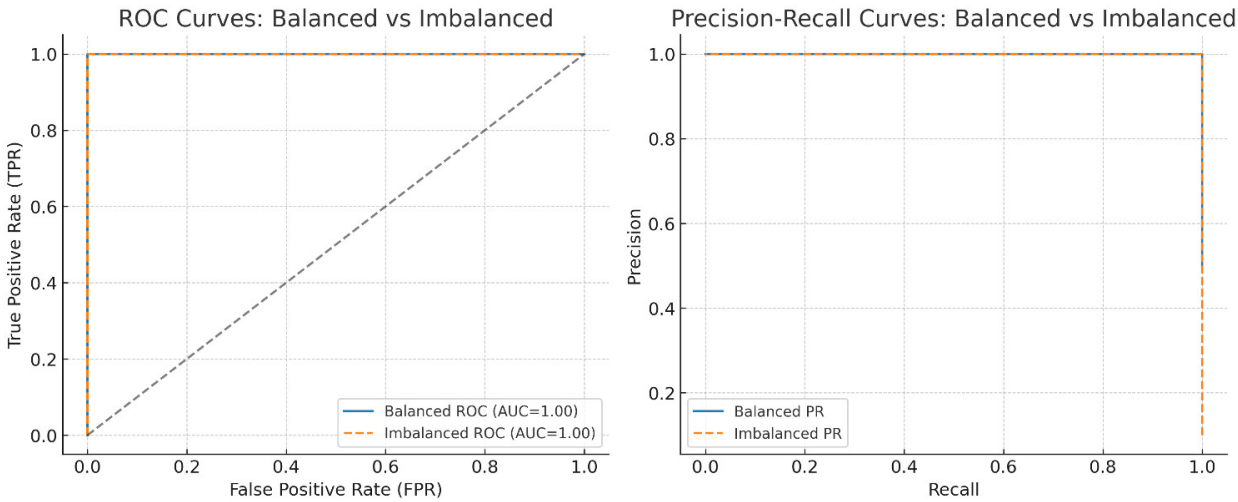

Figure 3. Comparison of ROC and Precision–Recall Curves under Balanced and Imbalanced Class Distributions

The ROC curve is plotted with TPR (True Positive Rate) against the FPR (False Positive Rate) where TPR is on the y-axis and FPR is on the x-axis.

Accuracy is widely used to evaluate models, but it can be misleading in imbalanced datasets where predictions are dominated by the majority class. For example, study [1] shows that in intrusion detection, a model with 99% accuracy may still fail to detect rare and critical attacks such as U2R or Web-based intrusions. In such cases, Recall is more informative because it reflects the ability to detect minority classes. Precision, on the other hand, is essential when minimizing false alarms is a priority, as in the real-time monitoring of critical infrastructures [3]. The F1 score provides a compromise between precision and recall, but its usefulness decreases when the class distributions are heavily skewed, as noted in analyses of the CSE-CIC-IDS2018 dataset [1]. To capture trade-offs more effectively, researchers often rely on the ROC and AUC curves, which illustrate the balance between the True Positive Rate (TPR) and the False Positive Rate (FPR), Figure 3. However, even a high AUC does not always imply practical utility, since small increases in FPR can overwhelm security analysts with excessive false alerts [2] [6].

4. Datasets used to train ML and DL models

This section explores the popular datasets used to train and test Intrusion Detection Systems (IDS), security algorithms, and other cybersecurity and anomaly detection applications.

The CICIDS2017 dataset was created by the Canadian Institute for Cybersecurity (CIC) and network traffic captures with simulated attacks and normal traffic. This dataset covers several types of attacks, such as Denial of Service (DoS), Identity Theft (Impersonation), SQL Injection attacks, and brute-force attacks. This dataset is commonly used to evaluate IDS performance for cybersecurity research.

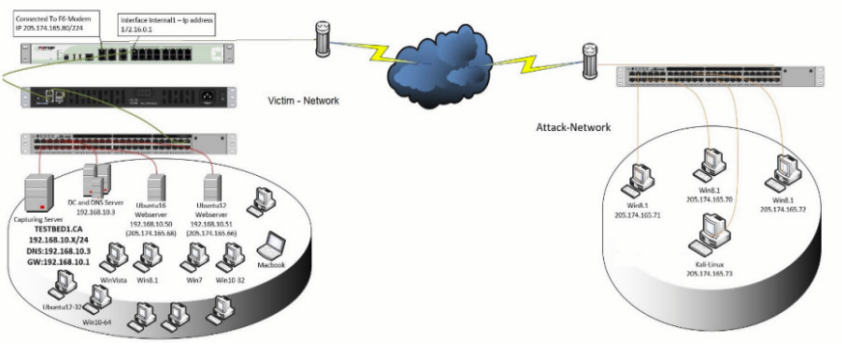

Figure 4. Architecture of the CICIDS2017 Dataset Generation Framework

Figure 4. illustrates the topology used to simulate attack tools and benign activities in the CICIDS2017 dataset. Although diverse, its laboratory generation limits its representativeness for noisy large-scale IoT networks.

The CTU-13 is a dataset compiled by the research group at the Czech Technical University (CTU). It has network traces generated by both real and simulated attack scenarios, with a focus on malicious activities associated with botnets. CTU-13 is widely used for research on intrusion and anomaly detection as it covers several types of botnet behaviors.

The KDD CUP 99 is one of the oldest and most widely used datasets for intrusion detection research created from the DARPA Intrusion Detection Evaluation Program in 1998. The dataset has network connections labeled as normal or malicious, covering a wide range of attacks. Although KDD CUP 99 is still used, it has been criticized for its inherent biases and the presence of redundant data, leading to the development of newer datasets like NSL-KDD.

The NSL-KDD is an enhanced version of KDD CUP 99, designed to address some of the weaknesses of the original dataset. NSL-KDD is more representative of real-world scenarios and is widely used to evaluate the performance of intrusion detection systems.

The CSE-CIC-IDS2018 dataset is a collaboration between the Canadian Institute for Cybersecurity (CIC) and the Communications Security Establishment (CSE) of Canada. It has labeled network traffic captures, covering several types of attacks such as denial-of-service attacks, SQL injection attacks, and brute-force attacks. This dataset is used to train and test modern IDS and security systems.

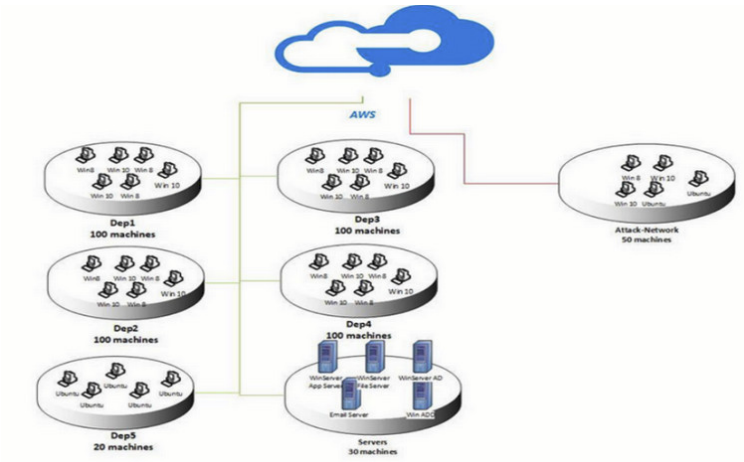

Figure 5. Architecture of the CSE-CIC-IDS2018 Intrusion Detection Framework

Figure 5. illustrates the topology used to simulate attack tools and benign activities in the CSE-CIC-IDS2018 dataset This dataset includes a wider diversity of attacks (DDoS, botnets, web attacks). However, it still lacks native IoT traffic patterns such as Zigbee or MQTT, limiting its direct applicability to IoT deployments.

The UNSW-NB15 dataset consists of raw network packets that were generated by a tool called IXIA Perfect Storm in the Cyber Range Laboratory of the Australian Center for Cyber Security (ACCS). It has a mix of modern normal activities and synthetic contemporary attack behaviors. The dataset includes nine types of attacks, including Fuzzers, Analysis, Backdoors, Denial of Service (DoS), Exploits, Generic, Reconnaissance, Shellcodes, and Worms. The tools Argus and Bro-IDS were used, and 12 algorithms were developed to generate 49 features along with the class label. The dataset has a total of 2,540,044 records stored in four CSV files, with the training set and the test set having 175,341 and 82,332 records, respectively. The dataset has been used in different studies for intrusion detection, cyber forensics, privacy preservation, and threat intelligence approaches in different systems such as network systems, the Internet of Things, SCADA, Industrial IoT, and Industry 4.0.

As shown in Table 2, modern datasets such as CSE-CIC-IDS2018 and UNSW-NB15 offer broader attack diversity and greater realism than older datasets such as KDD Cup 99 and NSL-KDD.

All datasets summarized in this paper are publicly available and were originally introduced in the cited studies. While this review does not present new experiments, it offers a consolidated overview of the characteristics of the dataset as reported in the literature. Although these datasets play a crucial role in ensuring reproducibility and enabling comparative analysis, they are not fully representative of real-world intrusion detection scenarios. Table 2 presents the analysis and highlights four key limitations that should guide the selection of datasets for IDS research. To assist researchers in choosing appropriate datasets for training ML/DL models, we outline below four major challenges that limit the applicability of existing datasets to real-world IDS scenarios.

- **Severe Class Imbalance:** Many widely used datasets suffer from skewed class distributions. For example, NSL-KDD and CICIDS2017 are dominated by DoS traffic, while minority classes such as U2R or web-based attacks are underrepresented, leading to biased classifiers [8]. Studies show that oversampling and SMOTE techniques improve recall on minority classes but often at the expense of precision [8], highlighting the challenge of designing balanced IDS models.
- **Lack of Realistic Network Environments:** Several datasets, including CICIDS2017 and CSE-CIC-IDS2018, were generated in controlled laboratory settings [1]. While they provide diverse attack types (e.g., brute force, DDoS, infiltration), they fail to capture the heterogeneity and unpredictability of real IoT and Industry 4.0 environments. IoT-focused datasets such as Bot-IoT [2] or those generated in testbeds [6, 4] attempt to address this gap but often remain small in scale or limited in attack diversity.
- **Aging and Obsolescence:** Legacy datasets such as KDD Cup 99 and NSL-KDD are still frequently used despite their outdated traffic patterns and unrealistic artifacts [3]. Their continued use risks producing IDS models that perform well in benchmarks but fail against modern attack vectors, particularly in IoT ecosystems [5].
- **Absence of Standardized Evaluation Protocols:** Different studies use the same datasets with inconsistent preprocessing, feature selection, and train–test splits, making results difficult to compare [8]. For example, some works use random splits while others rely on temporal splits, producing divergent outcomes. This lack of uniformity undermines the reliability of reported accuracy metrics across IDS literature.

Table 2. Enhanced comparative analysis of public intrusion detection datasets

Dataset	**Realism / Recency**	**Attack Diversity**	**Scalability**	**IoT Focus**	**Imbalance Severity**	**Limitations**	**Recommended Use Case / Attack types included**
KDD Cup 99	Outdated (1999)	DoS, U2R, R2L, Probe	High	No	High	Redundant records, unrealistic traffic	Baseline benchmarking only; avoid for modern IDS. Attacks: DoS, privilege escalation (remote-to-local and user-to-root), probing.
NSL-KDD	Moderate (2009)	Similar to KDD Cup	Medium	No	High	Still outdated, lacks IoT attacks	Academic teaching and algorithm prototyping. Attacks: DoS, privilege escalation (remote-to-local and user-to-root), probing.
CICIDS2017	Recent (2017)	Diverse (DoS, Heartbleed, SQLi, Brute force, Botnet, etc.)	Medium	Partial	High (dominated by DoS)	Generated in lab, not fully real-world	General IDS benchmarking; limited IoT relevance. Attacks: DDoS (LOIC), Heartbleed, SSH brute force, SQL injection, botnet (Ares), XSS, infiltration. *Reference:* https://www.unb.ca/cic/datasets/ids-2017.html
CSE-CIC-IDS2018	Very recent (2018)	Broad (DoS, brute-force, web, infiltration, botnet)	High	Limited	Moderate	Lacks emerging IoT-specific traffic	Evaluating ML/DL models on multi-class attacks. Attacks: DDoS, Brute force, DoS, Web attack, Infiltration, Botnet, PortScan. *Reference:* https://www.unb.ca/cic/datasets/ids-2018.html
UNSW-NB15	Contemporary (2015)	9 attack types (DoS, Exploits, Reconnaissance, Worms, etc.)	High	Limited	Moderate	Synthetic, not IoT-native	Suitable for modern IDS and cyber-forensics. Attacks: Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode, Worms.
CTU-13	Contemporary (2011)	Botnet-focused	Medium	No	Moderate	Limited to botnet scenarios	Botnet detection studies. Attacks: Virut, Menti, Rbot, Murlo, Sogou, NSIS, Neris.
IoT-specific datasets (e.g., Bot-IoT)	Recent (2020+)	IoT-relevant (botnets, scanning, brute force)	Medium	Strong	Moderate	Often small-scale, limited device diversity	IDS in IoT/Industry 4.0; anomaly detection in constrained devices. Attacks: IoT botnet traffic (Mirai), scanning, brute-force, DoS.

5. Related works

This section reviews a selection of research papers focused on enhancing cybersecurity in IoT environments and network traffic analysis. The works discussed contribute to the improvement of digital corpus analysis, attack classification, and detection using Machine Learning (ML) and Deep Learning (DL) models.
The selected papers highlights various approaches to strengthening security measures in IoT ecosystems and network traffic monitoring. They highlight how ML and DL techniques are being applied to address the unique challenges posed by IoT devices and the complex nature of modern network traffic. These studies collectively highlight ongoing efforts to develop more robust and efficient cybersecurity solutions in the rapidly evolving landscape of IoT and network security.
[2] The study demonstrates that voting techniques (hard and soft voting) enhance the performance of algorithms such as Random Forest (100 estimators), Decision Tree, KNN (K=5), SVM, Logistic Regression, and XGBoost (500 trees, learning rate 0.1), achieving 100% accuracy with the XGBoost model. The models were trained to detect Bot-IoT and Ton-IoT attacks. Bot oversampling and undersampling techniques are used to balance classes with a similar proportion of labels. In this study the GNN algorithm is recommended to be used for the detection and classification of such attacks.

[6] The study focuses on capturing the network traffic of IoT devices and analyzing their behavior in various states and during targeted attacks. The data collected are utilized for several applications including the identification of IoT devices where the data set trains machine learning algorithms to recognize different types of IoT devices and their behaviors. Behavioral analysis is conducted to understand how IoT devices operate in different scenarios, enabling the detection of unusual or suspicious behavior. Intrusion detection systems are developed using the dataset to identify malicious activities and unauthorized devices within IoT networks. Furthermore, researchers evaluated the performance of the Random Forest classifier in identifying devices and their types while also assessing the transferability of trained models across different laboratories. The dataset also facilitates studies on the transferability of trained models across diverse datasets which helps evaluate the generalization of security measures in various IoT configurations. To further their work, the researchers plan to test the case study on their own datasets and devices that were not included in this study. They aim to expand their research to include IoT devices using Zigbee and Z-Wave protocols for profiling and intrusion detection. They also intend to broaden the categories used in this experiment for a more granular analysis and to create a comprehensive dataset on IoT attacks to facilitate experimentation with anomaly detection in both benign and malicious traffic.
[1] The study improves IDS by finding crucial features to distinguish malicious and benign network traffic, particularly effective for FTP, SSH, WEB, XSS, and SQL attack traffic using the CSE-CIC-IDS2018 dataset.
The proposed methodology consists of several phases. Data preprocessing is first performed to remove invalid values, transform categorical features into numerical ones, and reduce data volume in order to optimize storage and processing time. Next, feature selection is applied to enhance classification and prediction performance through a workflow that incorporates six methods: information gain, gain ratio, Relief, symmetric uncertainty, chi-squared, and ANOVA. For each data subset, normalized scores are computed and compared against a defined threshold to retain the most relevant features. Finally, five classification algorithms are evaluated using the Orange tool: Logistic Regression (LR) as a robust baseline, Naive Bayes (NB) with categorical features, Support Vector Machine (SVM) with an RBF kernel, Decision Tree (DT) with restricted depth to prevent overfitting (achieving 0.99 accuracy), and Random Forest (RF) with 100 trees and a maximum depth of 20. These models are applied to classify benign and malicious traffic across the selected subsets. The researchers see that the accuracy of the algorithms slightly improves with the increase in the number of features. However, ehighly satisfactory results were obtained in most cases with a small feature number. Future research will focus on different aggregation techniques that could replace the mean score in calculating the most crucial features. [3] In the context of Industry 4.0, this study proposes a new intrusion detection system that uses machine

learning to find routing attacks against the RPL protocol. Network traffic is generated by the Cooja-Contiki simulator for various topologies, then transformed into CSV files. The researchers developed a process to create relevant and optimized datasets, using Random Forest (RF) methods and Pearson's correlation to select the most relevant features, eliminate redundancies and perform specific labeling for each type of attack. The results show that for binary classification, the Decision Tree (DT), Random Forest (RF), and K-Nearest Neighbors (KNN) algorithms achieved the best performance with accuracy exceeding 99%. For multiclass classification (7 classes), KNN obtained 99% accuracy with a 98% detection rate, closely followed by RF and DT. RF proved to be the most balanced choice in terms of performance and execution time. Based on these results, the researchers proposed RF-IDSR, an RF-based IDS for Industry 4.0 networks that use RPL. This system aims to provide fault tolerance and intrusion while detecting attacks. In addition, they introduced slight improvements to the RPL protocol to prevent specific attacks and network failures.
For future work, the researchers plan to implement RF-IDSR and evaluate its performance in both simulation and experimental environments. They also intend to generate new datasets that incorporate the proposed improvements and extend their study to other types of attacks against RPL.
[4] The paper provides an overview of several types of intrusion detection systems (IDS) and discusses the advantages of using machine learning (ML) and deep learning (DL) approaches for IDS.Performance metrics used to evaluate IDS are discussed, including accuracy, precision, recall, and F1 score.
Suggestions for future work include using real-time training on live network data, developing hybrid ML-DL models, testing against zero-day attacks, and exploring distributed processing frameworks like Spark.

[8] This study presents a comprehensive survey-based classification of Intrusion Detection Systems (IDS), focusing on the taxonomy of machine learning-based IDS (ML). It includes a comparative analysis of various ML algorithms employed in IDS implementations, highlighting their strengths and weaknesses. In addition, it addresses key research challenges in the field, providing insight into the complexities and limitations faced by current ML techniques to improve IDS effectiveness and IDS reliability.
[7] The study focuses on intrusion detection systems to classify network traffic as normal or malicious using ML techniques trained on the KDD-CUP-99 data set. The study analyzes the following ML algorithms: LR, Decision Tree, K-Nearest Neighbor, Naïve Bayes, Bernoulli Naïve Bayes, Multinomial Naïve Bayes, XG-Boost Classifier, AdaBoost, Random Forest, SVM, Rocchio Classifier, Ridge, Passive-Aggressive Classifier, ANN, and Perceptron. The results show that SVM achieves the highest accuracy at 98.08%. Future work will focus on improving the adaptability of these classifiers to large-scale datasets. MFFNN, CNN, and RNN, as well as ensemble learning models and extreme learning machines, have become unavoidable directions for future research.

[5] Bin Hulayyil et al. (2023) present a comprehensive analysis of potential vulnerabilities in IoT architectures across the hardware, network, and application layers. The authors propose a taxonomy of machine learning (ML) and deep learning (DL) techniques that have been employed to detect vulnerabilities, threats, and attacks in the IoT ecosystem, while also reviewing the most recent detection frameworks. Their study concludes that ML and DL approaches are essential for strengthening IoT security by ensuring integrity, availability, authentication, and authorization. In addition to software- and network-level threats, the authors emphasize hardware-layer vulnerabilities such as side-channel attacks (power analysis, electromagnetic leakage, timing attacks), fault injection (voltage glitches, laser fault injection), and hardware backdoors, which are particularly critical in resource-constrained IoT devices. To mitigate these risks, hardware-assisted mechanisms such as Trusted Platform Modules (TPMs), Physical Unclonable Functions (PUFs), and trusted execution environments (e.g., ARM TrustZone) are discussed as promising solutions. For future directions, the authors highlight two key research challenges: (i) enhancing IoT system intelligence by adopting advanced ML/DL techniques for proactive vulnerability detection, and (ii) addressing resource constraints in IoT devices by optimizing computations, employing data-sharing mechanisms, and applying compression techniques to reduce the footprint of ML/DL models.

[9] The paper presents a framework for implementing Machine Learning (ML) and Deep Learning (DL) techniques aimed at improving Intrusion Detection Systems (IDS) in the context of Network Traffic Monitoring and Analysis (NTMA). The study uses the CSE-CIC-IDS2018 database as the training dataset for the CNN model. The CNN model achieves an accuracy of 92% after 30 iterations, highlighting its potential for intrusion detection.
For future work, the authors propose testing the studied model in a real NTMA environment to confirm its performance under concrete operational conditions and evaluate its practical applicability.
[10] the study proposes a robust network intrusion detection system using ML and DL models.
The Decision Tree classifier achieves a remarkable accuracy of 99.05% and is particularly adept at finding various attack categories. Ensemble models also perform well, with Random Forest achieving 98.96% accuracy, Adaboost at 97.87%, and XGBoost at 98.08%. The K-Nearest-Neighbor (KNN) classifier achieves its best performance with K=7, reaching an accuracy of 95.58%. The DL model, including two dense layers with ReLU activation and a third layer with Sigmoid activation, reaches an accuracy of 98.44% using the ADAM optimizer, with an 80:20 Train-Test Split Ratio. Notably, XGBoost demonstrates 95% accuracy in detecting network attack exploits, while Random Forest excels in finding Fuzzers (90%), Generic attacks (99%), and Reconnaissance attacks (79%).

As shown in Table 3, most studies report high accuracy (more than 95%), but very few provide crucial operational metrics such as False Positive Rates (FPR), training time, or inference latency. This omission limits the practical relevance of the reported results, since FPR is critical in real deployments where excessive alerts overwhelm security analysts. Furthermore, computational cost is rarely discussed, although it is a determining factor for deploying IDS models in IoT and edge environments. Lightweight ML models (RF, DT, KNN) generally offer fast inference with limited resource consumption, while ensemble and deep learning models yield higher accuracy but are unsuitable for constrained devices due to training and inference overheads.
Another critical gap is generalizability: most studies train and evaluate on a single dataset, without testing robustness across unseen attack variants or different environments. This raises concerns about dataset bias and overfitting. Only a few works discuss cross-dataset validation or domain adaptation.

Algorithmic Strengths and Weaknesses in IDS Contexts

Although numerous related works provide accuracy values for different machine learning (ML) and deep learning (DL) approaches, a deeper algorithmic analysis is essential to understand why certain methods consistently perform better in intrusion detection systems (IDS), especially within IoT environments.
Tree-based models such as Decision Trees (DT) and Random Forests (RF) are frequently reported as achieving strong results across diverse datasets, often exceeding 99% accuracy [1]. Their robustness to noise, ability to handle both categorical and continuous features, and interpretability make them well suited for network intrusion tasks. In addition, RF's ensemble nature reduces overfitting and provides stable performance across attack types such as FTP, SSH, and SQL injection. However, these models may become computationally expensive as the dataset grows and are less adapted to continuous traffic streams in real-time IoT monitoring.
Ensemble methods such as XGBoost and voting classifiers further enhance detection performance by aggregating multiple learners. Studies on IoT-focused datasets confirm that boosting and ensemble strategies achieve superior precision and recall for complex attacks, including botnets and denial-of-service [2]. Nevertheless, the increased computational cost makes these models more suitable for IoT gateways or cloud deployment than for highly resource-constrained devices.

Table 3. Comparative synthesis of ML/DL approaches for IDS in IoT

Ref.	**Dataset**	**Algorithm / Model**	**Performance** (Acc./FPR)	**Complexity & Resource Usage**	**Generalizability & Limitations**	**Explainability (XAI)**
[1] Göcs & Johanyák (2023)	CSE-CIC-IDS2018	Feature selection + RF, DT, SVM, LR	99% of all metrics for the models (RF, DT, KNN)	Low (traditional ML)	Limited to dataset; no cross-validation; no scalability details	Not addressed
[2] Jarjis et al. (2022)	Bot-IoT, Ton-IoT	Voting ensemble (RF, DT, KNN, SVM, XGBoost)	Acc. 100% (XGBoost)	High (ensemble costly at edge)	Risk of overfitting; not tested on unseen IoT traffic	Not addressed
[3] Medjek et al. (2021)	Cooja-Contiki (simulation)	RF, DT, KNN	Acc. 99%	simulator cooja-contiki 3.0 6LoWPAN-IoT VM WITH 48 GB RAM, 8 VCPUs	Narrow scope; simulation only	Not addressed
[4] Sharafali et al. (2022)	CICIDS2017, CTU-13, etc.	Hybrid ML-DL (CNN, RNN, LSTM, RF, SVM)	DL ¿ ML; hybrid best	High (DL resource intensive)	Not evaluated across datasets; unsuitable for IoT edge	Not addressed
[6] Dadkhah et al. (2022)	IoT dataset (lab, 60 devices)	RF, DT, XGBoost, AdaBoost, KNN	Acc. 98% (AdaBoost best)	Medium (tree ensembles)	Poor transferability; laboratory-only	Not addressed
[7] Tripathy & Behera (2023)	KDD CUP-99	LR, DT, NB, KNN, SVM, etc.	SVM highest with Acc. of 98.08%)	Low (traditional ML)	Outdated dataset; no IoT focus; dataset bias	Not addressed
[9] Azeroual et al. (2022)	CSE-CIC-IDS2018	CNN	Acc. 92% (30 iterations)	High (DL training intensive)	Not tested in real networks; Limited explainability	Not addressed
[10] Kumar et al. (2022)	UNSW-NB15	DT, RF, AdaBoost, XGBoost, Dense NN	Acc. 95–99% (DT best), RF FPR 10%	Medium to High (DL needs GPU)	No cross-dataset validation; reproducibility details missing	Not addressed

In contrast, Naïve Bayes remains attractive for IoT edge devices due to its simplicity and low computational cost. However, its assumption of feature independence is rarely satisfied in network traffic, which can lead to suboptimal accuracy, typically below ensemble or tree-based models [7]. Similarly, Support Vector Machines (SVM) have shown high accuracy (up to 98%) in classic datasets [7], particularly when using nonlinear kernels. Yet, their poor scalability with large-scale data and the complexity of kernel tuning limit their applicability to modern high-volume IoT traffic.

Deep learning models have also been widely explored. Convolutional Neural Networks (CNNs), for example, achieved promising results on CSE-CIC-IDS2018, reaching around 92% accuracy after 30 iterations [8]. However, this plateau suggests limitations: CNNs excel at extracting spatial patterns but lack sequential memory, making them less effective for detecting long-term traffic dependencies. Moreover, their training complexity and reliance on GPUs restrict their feasibility in IoT edge deployments. Recurrent Neural Networks (RNNs) and Long Short-Term Memory networks (LSTMs) better capture temporal correlations in traffic sequences [4], which is crucial for identifying slow or stealthy attacks, but they suffer from high training cost and convergence challenges.

Overall, the choice of algorithm reflects a trade-off between accuracy, interpretability, and resource efficiency. Lightweight ML models such as RF, DT, or Naïve Bayes remain attractive for IoT edge nodes, where interpretability and speed are critical. In contrast, DL models such as CNNs and LSTMs offer higher capacity for complex and zero-day attacks but require deployment on cloud servers or more powerful gateways. Hybrid ML-DL frameworks, combining deep feature extraction with efficient ML classifiers, have emerged as a promising direction to balance these trade-offs [4, 10].

Hybrid ML–DL models will be implemented in two stages: deep learning architectures such as CNNs or autoencoders are first employed for feature extraction and dimensionality reduction. The resulting representations are then fed into machine learning classifiers, including Random Forest, SVM, or XGBoost, to perform the final decision making. This approach leverages the capacity of DL to capture complex high-level features while benefiting from the interpretability and efficiency of ML classifiers.

Figure 6 shows this complete pipeline, where preprocessing, DL feature extraction, ML classification, and evaluation form a unified IDS framework.

Table 4. Comparative strengths and weaknesses of algorithms in IDS/IoT contexts

Algorithm	**Strengths**	**Weaknesses**	**Suitable IDS/IoT Use Cases**
RF / DT	Interpretable, robust, suitable for IoT edge	Sensitive to class imbalance	Multi-class (FTP, SSH, SQL) [1]
CNN	Captures complex patterns	Performance plateau, computationally expensive	Brute-force attack detection on large datasets [8]
RNN / LSTM	Sequential traffic analysis	Heavy training cost	Detection of persistent traffic patterns [4]
Naïve Bayes	Ultra-fast, lightweight	Unrealistic independence assumption	Low-power IoT edge devices [7]
XGBoost / Ensembles Learning	High accuracy, resilience	Very computationally expensive	Cloud IoT, gateways [2]

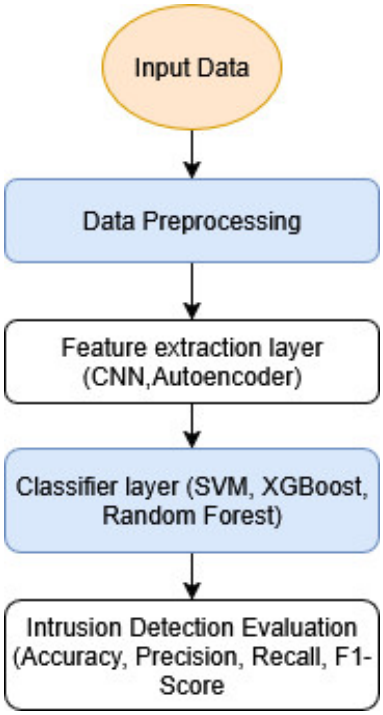

Figure 6. DL feature extraction followed by ML classification

6. Perspectives and future work

Future cybersecurity research is expected to advance in several key directions. A major avenue concerns the application of Graph Neural Networks (GNNs) for cyberattack detection and classification [2], supported by case studies on diverse datasets, including underexplored sources, and extended to IoT devices using Zigbee and Z-Wave protocols [6]. Parallel efforts will focus on constructing comprehensive IoT attack datasets to strengthen anomaly detection [6], as well as on exploring advanced aggregation techniques for identifying salient features [1]. Research will also prioritize real-time training with live network data, hybrid ML–DL models, robustness against zero-day attacks, and distributed processing frameworks such as Spark [4]. In the hybrid paradigm, deep models such as CNNs or autoencoders serve for feature extraction and dimensionality reduction, while machine learning classifiers such as Random Forest, SVM, or XGBoost perform the final decision-making. This design leverages the representation power of DL and the interpretability and efficiency of ML, resulting in a unified IDS pipeline (Figure 6) [3, 8, 4]. Finally, enhancing classifier adaptability to large-scale datasets and assessing models such as MFFNN, CNN, RNN, ensemble learning, and extreme learning machines remains a crucial challenge [7].

In the IoT context, research priorities include enhancing system intelligence for vulnerability detection through advanced ML/DL methods, while simultaneously improving resource efficiency by addressing time and memory constraints [5]. These challenges cannot be overcome without reconsidering the datasets and evaluation practices that underpin the IDS models. Dataset realism and performance metrics are therefore central to assessing the applicability of ML/DL approaches under realistic IoT conditions. Accordingly, future work must extend beyond algorithmic refinements to include dataset representativeness, evaluation standards, and deployment strategies.

Datasets and Realism : Although benchmark datasets such as CICIDS2017 and CSE-CIC-IDS2018 offer a diversity of attack types, they remain limited by class imbalance, synthetic traffic, and insufficient

IoT-specific protocols. Standardized datasets capturing heterogeneous protocols (e.g., Zigbee, BLE, MQTT), constrained device behaviors, and real-world noise are essential for meaningful IDS evaluation.

Metrics Beyond Accuracy : Accuracy alone is insufficient to capture IDS utility. Metrics such as false positive rate (FPR), detection latency, and resource usage (CPU, memory, energy) are critical in IoT deployments. Developing resource-aware evaluation criteria tailored to IoT IDS should therefore be a priority.

Continuous Learning and Concept Drift : IDS must adapt to evolving threats. Promising directions include semi-supervised and online learning techniques capable of addressing concept drift without relying solely on extensive labeled datasets.

Model Compression and Resource-Aware Deployment: Strict memory and computational limits on the IoT edge motivate the research into model compression. Techniques such as pruning and quantization applied to Random Forests, Decision Trees, or lightweight neural networks should aim to preserve detection rates above 95%, thus enabling deployment on microcontrollers.

IoT-Specific Adaptations: Future IDS designs must account for protocol heterogeneity, bandwidth constraints, and limited energy availability. Approaches such as federated learning and hybrid ML–DL—where feature extraction is performed in the cloud and lightweight classification at the edge—offer promising avenues.

Reproducibility and Open Science: A major limitation of existing work lies in its lack of reproducibility, with many studies omitting preprocessing details, hyperparameters, or dataset splits. Advancing IDS research requires explicit publication of scripts, models, and configurations in open repositories to foster transparency and replicability.

In summary, advancing IDS for IoT requires shifting from generic algorithmic propositions to concrete and actionable solutions: realistic and standardized datasets, evaluation metrics tailored to IoT constraints, efficient continuous learning strategies, and resource-aware model optimization. These directions specifically address the weaknesses identified in this study and outline a structured roadmap for impactful future research.

7. Conclusion

In conclusion, the reviewed body of work makes significant contributions to cybersecurity, particularly in intrusion detection and prevention for IoT environments and in network traffic analysis. Using both traditional ML and modern DL approaches, these studies address the evolving nature of cyber threats while underscoring the importance of realistic datasets, advanced algorithms, and meaningful performance metrics. Hybrid models that combine the representational power of DL with the interpretability of ML emerge as a promising avenue, particularly for feature extraction and classification tasks. Likewise, the introduction of novel datasets such as CICIoT2023 and UNSW-NB15, together with the exploration of diverse attack scenarios, enhances the robustness and adaptability of IDS. Common research trends include optimizing ML models, investigating alternative aggregation strategies, and applying computational intelligence methods. Equally emphasized is the need for real-world testing and validation to ensure practical deployment. Some works focus on specific IoT attack categories, while others broaden their scope to encompass general challenges in ML-based IDS, including real-time training, model retraining against zero-day threats, and distributed frameworks such as Spark. Collectively, these contributions provide valuable insight and methodologies that advance IDS development and lay the foundation for future research in the continuously evolving landscape of cybersecurity.

Declarations

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Ethical Approval

This article does not contain any studies with human participants or animals performed by any of the authors.

Data Availability

All datasets referenced and analyzed in this survey are publicly available through their official repositories. No new datasets were generated in this study.

Code Availability

As this is a survey paper, no original implementation code was produced. All experimental results mentioned are taken from publicly available studies cited in the paper.

REFERENCES

1. L. Göcs and Z. C. Johanyák, "Identifying relevant features of cse-cic-ids2018 dataset for the development of an intrusion detection system." https://arxiv.org/abs/2307.11544, 2023.
2. A. H. Jarjis, N. Y. S. Alzubaidi, and M. K. Pehlivanoğlu, "Cyber attacks classification on enriching iot datasets," *EAI Endorsed Transactions on Internet of Things*, vol. 9, no. 3, 2022.
3. F. Medjek, D. Tandjaoui, N. Djedjig, and I. Romdhani, "Fault-tolerant ai-driven intrusion detection system for the internet of things," *International Journal of Critical Infrastructure Protection*, vol. 34, p. 100436, 2021.
4. S. A. Sharafali, N. H. Fallooh, and M. H. Ali, "Intrusion detection system based on machine learning and deep learning techniques: A review," in *3rd International Conference of Engineering Sciences (ICES' 2022)*, 2022.
5. S. B. Hulayyil, S. Li, and L. Xu, "Machine-learning-based vulnerability detection and classification in internet of things device security," *Electronics*, vol. 12, no. 18, p. 3927, 2023.
6. S. Dadkhah, H. Mahdikhani, P. Danso, and A. Zohourian, "Towards the development of a realistic multidimensional iot profiling dataset," in *International Conference on Privacy, Security and Trust (PST)*, 2022.
7. S. S. Tripathy and B. Behera, "Performance evaluation of machine learning algorithms for intrusion detection system," *Journal of Biomechanical Science and Engineering*, 2023.
8. A. Ali, S. Naeem, S. Anam, and M. M. Ahmed, "Machine learning for intrusion detection in cyber security: Applications, challenges, and recommendations," *Innovative Computing Review (ICR)*, vol. 2, no. 2, 2022. ISSN(E): 2791-0032.
9. H. Azeroual, I. D. Belghiti, and N. Berbiche, "A framework for implementing an ml or dl model to improve intrusion detection systems (ids) in the ntma context, with an example on the dataset (cse-cic-ids2018)," in *ITM Web of Conferences*, vol. 46, 2022.
10. D. K. A., S. Vajipayajula, K. Srinivasan, A. Tibrewal, T. S. Kumar, and T. G. Kumar, "Detection of network attacks using machine learning and deep learning models," in *International Conference on Machine Learning and Data Engineering*, 2022.