Intrusion Detection in IoT Under Various Machine Learning Models

Arfa Farooqi[1], M. Junaid Arshad[2]

[1,2]Department of Computer Science, University of Engineering and Technology (UET), Lahore-Pakistan

[1]arfa.farooqi1122@gmail.com

***Abstract*—Internet of Things refers to the network of physical objects embedded with sensors, software, and other technologies that enable them to connect and exchange data with other devices and systems over the internet. These objects can range from simple household items like refrigerators and thermostats to complex industrial machinery. IoT allows for greater automation, control, and data analysis in various domains, including home automation, healthcare, transportation, agriculture, and manufacturing, bringing in addition to many benefits, challenges related to security issues. Intrusion Detection Systems (IDS) have been an important tool for the protection of networks and information systems. Many machine learning models have been used to enhance its performance and accuracy. In this paper, we present a survey of IDS research efforts under machine learning models for IoT. Our objective is to identify issues in previous models and review leading trends. We classified the IDSs proposed in the literature according to the following attributes: machine learning models, datasets and accuracy.**

***Keywords*—Internet of Things, Intrusion Detection System, Machine Learning, Datasets, Transformers and Accuracy**

I. INTRODUCTION

OUR lives are impacted by the Internet of Things in many different ways, including our homes, vehicles, trains, streets, travel, businesses, and agriculture [2]. But as these gadgets become more linked, questions have been raised about the reliability and security of IoT communications. Ensuring se- cure and dependable device interactions become increasingly important as IoT systems get larger and more sophisticated.

Any illegal or hostile activity that aims to jeopardize the network's availability, confidentiality, or integrity is referred to as an intrusion [10] in the context of computer networks. Network intrusions can take many different forms, and de- pending on their traits and goals, they are usually divided into several groups. Typical forms of network intrusions include the following:

Denial-of-Service (DoS) Attack: A denial-of-service (DoS) assault occurs when an attacker overwhelms a network or certain network resources, such as servers or routers, with an excessive amount of traffic. The targeted resources are overloaded by this large amount of traffic and are unable to reply to valid user requests. A denial-of-service (DoS) [2] attack aims to interfere with the regular operation of a network or service, resulting in downtime and depriving authorized users of access to resources. DoS assaults are more difficult to counteract since they might originate from a single source or from numerous sources (DDoS).

Distributed Denial-of-Service (DDoS) Attack: DDoS assaults are identical to DoS attacks, but they involve the coordination of several infected devices, often known as bots or zombies, to conduct a coordinated attack against a target [11]. By dispersing traffic from several sources, DDoS assaults increase their effect and make it more challenging for network administrators to monitor or remove suspicious content.

Malware: Any program intended to interfere with, harm, or get unauthorized access to computer systems or networks is referred to as malware, short for malicious software. Trojan horses [12], worms, viruses, ransomware, and spyware are examples of common malware. Email attachments, malicious websites, and infected portable media are just a few of the ways that malware may infect networked devices. Malware may carry out a broad range of malicious actions after it is placed on a device, such as stealing confidential data, interfering with network functions, or giving attackers access without authorization.

Intrusion by Unauthorized Access: Unauthorized access happens when a hacker enters a network or system without authorization by taking advantage of flaws in pass- words, incorrect security configurations, or vulnerabilities [10]. To obtain access to network resources, attackers might use a number of strategies, including brute-force attacks, password guessing, and taking advantage of known flaws. Once within the network, attackers can try to steal confidential data, advance their privileges, or move widely.

Port Scanning and Probing: Port scanning and probing [10] involve scanning network ports and probing for vulnerabilities or weaknesses in networked devices or services. Attackers find open ports, services, and possible points of access into a network by using port scanning tools. Cyber criminals can get data about the target network and pinpoint possible targets for exploitation by using port

scanning and probing as the initial stage of their reconnaissance process.

Data Breaches: When private or sensitive data is accessed, taken, or made public by uninvited people, it is called a data breach. A variety of attack methods, such as SQL injection, phishing, social engineering, or taking advantage of unpatched vulnerabilities, can lead to data breaches. Organizations may face serious consequences from data breaches, such as monetary losses, harm to their reputation, regulatory fines, and legal obligations. [10] All things considered, network breaches present serious risks to both persons and companies, emphasizing the significance of establishing strong security measures in place and routinely checking network traffic for unusual activities. A thorough network security plan must include firewalls, access restrictions, encryption, intrusion detection and prevention systems (IDS/IPS), and security awareness training. An intrusion detection system (IDS) [7], [12] is a security tool that monitors system activity and network traffic to look for indications of malicious activity, unauthorized access, or policy breaches. An intrusion detection system's main goal is to identify possible security issues and notify administrators or security staff so they may take the necessary action.

II. TYPES OF IDS

There are two main types of Intrusion Detection Systems:

Network-based IDS (NIDS): Network Intrusion Detection Systems (NIDS) [5] continuously observe network traffic, examining packets as they move through network interfaces to identify any suspicious behaviors or recognizable attack patterns. NIDS sensors are strategically positioned throughout the network, often at key locations like the network perimeter, behind firewalls, or on critical segments, to provide comprehensive coverage. NIDS are capable of identifying various types of network-based attacks, such as port scans, denial-of- service (DoS) attacks, malware infiltration, and unauthorized access endeavors. Network-based Intrusion Detection Systems (NIDS) [9] can be categorized based on their deployment architecture and monitoring approach. Here are the main types:

Traditional NIDS: Traditional NIDS are deployed at specific points within the network infrastructure, such as at network gateways or behind firewalls. They monitor all network traffic passing through their designated monitoring points. Traditional NIDS typically use signature-based detection [4] methods to identify known patterns of malicious activity within network packets.

Inline NIDS: Inline NIDS are positioned directly in the network traffic flow, allowing them to inspect and potentially block or modify network packets in real-time. Inline NIDS actively participate in the network traffic flow and can take immediate action to block or mitigate identified threats. These systems often include Intrusion Prevention System (IPS) functionality, combining intrusion detection and prevention capabilities in a single device.

Passive NIDS: Passive NIDS operate in a non- intrusive manner, monitoring network traffic passively without actively participating in the traffic flow. They analyze copies of network packets or traffic feeds obtained from network taps, port mirroring, or other passive monitoring techniques. Passive NIDS are less likely to impact network performance but may have limited visibility into encrypted or encrypted traffic.

Distributed NIDS: Distributed NIDS consist of multiple sensors or probes distributed throughout the network infrastructure. These sensors work together to monitor network traffic across different network segments or geographical locations. Distributed NIDS provide comprehensive coverage of the network and can scale to accommodate large and complex network environments.

Cloud-based NIDS: Cloud-based NIDS [15] are deployed in cloud environments to monitor network traffic and detect threats within cloud-based infrastructure and services. These systems are designed to provide security monitoring and threat detection capabilities for cloud-based applications, platforms, and virtualized environments. Cloud-based NIDS can integrate with cloud-native security services and platforms to provide centralized visibility and control over cloud-based networks.

Each type of NIDS has its own advantages and limitations, and the choice of deployment architecture depends on factors such as network architecture, security requirements, performance considerations, and compliance requirements. Organizations often deploy a combination of NIDS types to provide comprehensive network security coverage and address specific security needs.

The types of Network-based Intrusion Detection Systems (NIDS) described earlier focus on deployment architecture and monitoring approach. In contrast, packet-based and flow-based network IDS differ primarily in the level of granularity at which they analyze network traffic. Here's how they differ: *i) Packet-based NIDS:* Packet-based NIDS inspect in- dividual network packets in real-time as they traverse the network. These systems analyze the contents of each packet, including header information and payload data, to detect signs of malicious activity. Packet-based NIDS are well-suited for detecting specific network attacks that can be identified based on characteristics within individual packets, such as signature-based attacks or anomalies in packet headers. *ii) Flow-based NIDS:* Flow-based NIDS analyze aggregated network traffic flows instead of individual packets. Network flows represent sequences of related packets between specific source and destination endpoints, typically defined by common attributes such as IP addresses, port numbers, and transport protocols. Flow-based NIDS focus on identifying patterns and anomalies in network traffic flows, such as sudden increases in traffic volume,

unusual communication patterns, or deviations from normal traffic behavior. Flow-based NIDS are often used for detecting distributed denial-of-service (DDoS) attacks, identifying network scanning activities, and monitoring overall network performance and usage. In summary, while packet- based NIDS inspect each network packet individually, flow- based NIDS analyze aggregated network flows to identify patterns and anomalies in network traffic. Both approaches have their advantages and limitations, and organizations may choose to deploy one or both types of NIDS depending on their specific security requirements and monitoring objectives. Additionally, packet-based and flow-based NIDS can be further classified based on their deployment architecture, such as traditional, inline, passive, distributed, or cloud-based, as described earlier.

Host-based IDS (HIDS): Host Intrusion Detection Systems (HIDS) [10] observe and analyze the actions taking place on individual host systems, such as servers, workstations, or network devices, to identify any suspicious activities or signs of compromise. HIDS agents are installed directly onto host systems, allowing them to monitor a range of system-level activities including system logs, file integrity, system calls, and more, in order to detect any potential intrusions. HIDS are capable of recognizing various types of attacks that target individual host systems, such as unauthorized access attempts, efforts to escalate privileges, alterations to files, and the installation of malicious software. Here are some common types of HIDS:

i. ***System Log Monitoring:*** HIDS can monitor system logs generated by operating systems, applications, and services running on the host. This type of HIDS analyzes log entries for signs of unauthorized access, system misconfigurations, software vulnerabilities, or other suspicious activities.

ii. ***File Integrity Monitoring (FIM):*** FIM HIDS monitor changes to critical system files, directories, and configurations. These HIDS compare the current state of files and configurations against a known baseline to detect unauthorized modifications, tampering, or malware infections.

iii. ***Registry Monitoring:*** HIDS can monitor changes to system registries on Windows-based hosts. Registry monitoring HIDS analyze registry entries for modifications or additions that may indicate unauthorized changes or the presence of malware.

iv. ***Kernel Module Monitoring:*** HIDS can monitor the loading and unloading of kernel modules (drivers) on the host operating system. Kernel module monitoring HIDS detect attempts to load unauthorized or malicious kernel modules, which can provide attackers with elevated privileges or back- door access to the system.

v. ***Application Behavior Monitoring:*** HIDS can monitor the behavior of applications running on the host, including process creation, network connections, file accesses, and system calls. Application behavior monitoring HIDS detect anomalous or suspicious behavior that may indicate the presence of malware, exploits, or unauthorized activities.

vi. ***Anomaly Detection:*** Some HIDS employ anomaly detection techniques to establish a baseline of normal behavior for the host system [4]. Anomaly detection HIDS monitor deviations from this baseline to detect unusual or suspicious activities that may indicate a security incident.

vii. ***Endpoint Detection and Response (EDR):*** EDR solutions combine HIDS capabilities with additional features for incident response, investigation, and remediation. EDR HIDS provide real-time visibility into endpoint activities, automated response actions, and advanced threat hunting capabilities. These are some of the common types of HIDS, each providing specific capabilities for monitoring and detecting security threats on individual host systems. Organizations often deploy a combination of HIDS types to provide comprehensive coverage and address various security requirements and use cases.

Detection Techniques: Intrusion Detection Systems can operate using different detection techniques, including:

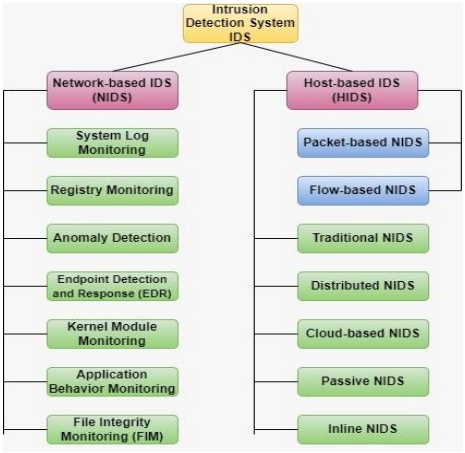

Fig. 1. IDS and their types

Signature-based detection: This method involves comparing network traffic or system actions with a repository of recognized attack patterns or signatures [9]. Upon discovering a match, an alert is triggered.

Anomaly-based detection: Anomaly detection involves establishing a standard pattern of normal behavior for the network or system and then scrutinizing deviations from this norm [9]. These deviations could indicate potential security threats or unusual activities.

Heuristic-based detection: Heuristic detection depends on predefined algorithms or rules to identify suspicious behaviors or actions that might signal an intrusion. While more adaptable than signature-based detection, heuristic-based detection may yield a higher rate of false positives. Once an IDS detects something fishy going on, it sends out alerts or messages to

let administrators or security folks know. These alerts usually give details about what's happening, which part of the system or network is affected, and suggestions on what to do next.

In the world of cybersecurity, IDS play a crucial role that spot trouble before it gets out of hand. They're crucial for organizations to catch and deal with security issues quickly, making sure attacks don't cause too much damage and keeping the network safe. Often, they team up with Intrusion Prevention Systems (IPS), which can automatically respond to detected threats by blocking or lessening the impact of malicious activities. So, in simpler terms, when an IDS spots trouble, it tells the right people so they can take action. This helps keep networks safe from cyber threats and ensures everything runs smoothly.

III. COMPONENTS AND OPERATIONS OF IDS

Components of an IDS: Sensors or Agents [10]: These are the components responsible for collecting data and monitoring network traffic or system activities. In Network-based IDS (NIDS) [9], sensors are strategically placed at key points in the network to monitor traffic. In Host-based IDS (HIDS), agents are installed directly on individual host systems to monitor their activities.

Detection Engine: The detection engine is the core component of the IDS responsible for analyzing the collected data and detecting suspicious or malicious activities. Depending on the detection technique employed (signature-based, anomaly-based, heuristic-based), the detection engine compares observed patterns against known signatures, baseline behaviors, or predefined rules to identify potential intrusions.

Mechanism: When the IDS detects suspicious activity, it generates alerts or notifications to notify administrators or security personnel. Alerts typically contain information about the detected activity, including the type of intrusion, the affected system or network, and any relevant contextual information.

Logging and Reporting: IDS systems often maintain logs of detected events and activities for further analysis, forensic investigation, or compliance purposes. Detailed reporting capabilities allow administrators to review and analyze security events over time, identify trends, and assess the effectiveness of security measures.

Response Mechanism (optional): Some IDS systems include the capability to respond to detected threats automatically. For example, an Intrusion Prevention System (IPS) can block or mitigate malicious activities in real- time by taking predefined actions, such as blocking network traffic, isolating compromised systems, or triggering security alerts.

Operation of an IDS:

Data Collection: The IDS collects data from various sources [13], such as network traffic (NIDS) or system logs (HIDS), using sensors or agents deployed throughout the network or on individual host systems. ***Analysis and Detection:*** The collected data is analyzed by the detection engine using predefined detection techniques (signature-based, anomaly-based, heuristic-based) to identify potential intrusions or security incidents. ***Alert Generation:*** When suspicious activity is detected, the IDS generates alerts or notifications to notify administrators or security personnel. Alerts contain detailed information about the detected activity, including the type of intrusion, severity level, affected system or net- work, and any recommended actions. ***Alert Handling and Response:*** Upon receiving alerts, administrators or security personnel review and prioritize them based on their severity and potential impact. Depending on the organization's security policies and procedures, they may take appropriate actions to investigate, contain, and mitigate the detected threats. ***Logging and Reporting:*** The IDS logs detected events and activities for further analysis, forensic investigation, or compliance purposes. Detailed reporting capabilities provide administrators with insights into security events, trends, and the overall security posture of the network. In summary, an Intrusion Detection System (IDS) is a critical security technology that helps organizations detect and respond to potential security threats, such as unauthorized access, malicious activities, or policy violations. By continuously monitoring network traffic or system activities and analyzing them for signs of intrusion, IDS systems play a crucial role in maintaining the security and integrity of computer networks.

Working of ML Models:

Intrusion Detection using Machine Learning (ML) involves leveraging ML algorithms and techniques to detect and classify anomalous or malicious activities within a computer network or system. ML-based intrusion detection systems (IDS) can analyze large volumes of data to identify patterns and deviations indicative of unauthorized access or malicious behavior. Their work is also shown in Fig. 2. Here's how it typically works:

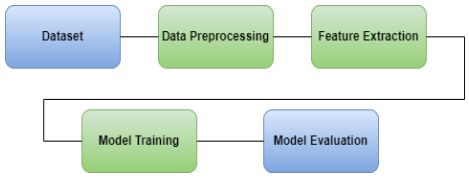

Fig. 2. ML Models Working

1. ***Data Collection:*** ML-based IDS systems collect data from various sources within the network, such as network traffic logs, system logs, packet captures, and other security-related data sources [13]. The data collected may include information about network connections, system activities, user behavior, and other relevant metadata.
2. ***Data Preprocessing:*** Before feeding the data into ML algorithms, preprocessing steps are performed to clean, normalize, and transform the data into a suitable format for analysis. Preprocessing may involve tasks such as handling missing values [5], encoding categorical variables, scaling numerical features, and extracting relevant features from raw data.

3. ***Feature Extraction:*** Feature extraction involves selecting or deriving relevant features from the raw data that capture meaningful information about network behavior or system activities [14]. Features may include network protocol types, source and destination IP addresses, port numbers, packet sizes, timestamps, user login/logout events, file access patterns, etc.
4. **Model Training:** ML algorithms are trained on labeled datasets containing examples of normal and malicious activities. [14] Supervised learning algorithms, such as Support Vector Machines (SVM), Random Forests, Gradient Boosting Machines (GBM), or Deep Learning models (e.g., Convolutional Neural Networks, Recurrent Neural Networks), can be trained using labeled data to classify network traffic or system events as normal or anomalous. Unsupervised learning algorithms, such as K-means clustering, DBSCAN, Isolation Forest, or Autoencoders, can detect anomalies in the data without requiring labeled examples of malicious activity.
5. **Model Evaluation:** The trained ML models are evaluated using separate test datasets to assess their performance in detecting intrusions. [8] Evaluation metrics such as accuracy, precision, recall, F1-score, and area under the ROC curve (AUC-ROC) are used to quantify the effectiveness of the models in distinguishing between normal and malicious activities.
6. **Deployment and Monitoring:** Once trained and evaluated, the ML-based IDS models can be deployed in production environments to continuously monitor network traffic or sys- tem activities in real-time. The IDS system generates alerts or notifications when it detects suspicious behavior, enabling security analysts to investigate and respond to potential security incidents promptly.

ML-based intrusion detection offers several advantages, including the ability to adapt to evolving threats, handle large volumes of data, and detect previously unseen or zero-day attacks. However, it also poses challenges such as the need for labeled training data, the risk of false positives/negatives, and the interpretability of complex ML models. Continuous monitoring, model updating, and collaboration between ML experts and cybersecurity professionals are essential for building effective ML-based IDS solutions.

IV. RELATED WORK

This section provides a brief overview of ML based intrusion detection methods from the literature. We focus on the latest published papers for comparison in this study because they are state of the art making them currently latest to review. While there is other ML intrusion detection-based algorithms available, we couldn't explore them due to time constraints.

The dataset utilized [1] in this research was obtained from IEEE Data port which included over 800 samples of normal and malicious traffic in binary visualization format for the purpose of training models. This study examined seventeen models using various feature extractors and classification methods. Recall, accuracy, precision, and F1-score were used to evaluate the models, with accuracy and precision having the most significance. Three models—KNN, SMO, and random forest—were built using individual learning methods, and a stacked model that combined KNN and SMO was also developed. An additional five models were developed using auto-correlogram and FcTH filters. Three models were based on individual learning algorithms, like the first four models mentioned above, in addition to two stacked models. Four additional models were developed using the DenseNet transfer model. Three of these models were based on individual learning algorithms, like the first four models mentioned above, in addition to a stacked model, KNN and SMO, and KNN as the meta classifier. Then four additional models were developed using VGG-16 as the transfer model. These four models are like the DenseNet models regarding training algorithms. When VGG-16 combines the stacked model, KNN and SMO, and KNN as the meta classifier with k = 3, for 90 percent to 10 percent data split, the highest precision and accuracy was obtained. This model was chosen as the best model as a result. Using characteristics from the UNSW-NB15 dataset, Ah- mad et al. [5] suggest feature clusters regarding its flow, Mes- sage Queuing Telemetry Transport (MQTT), and Transmission Control Protocol (TCP). An imbalanced dataset, dimensionality, and overfitting are no longer problems. To overcome missing values of features, imputation was done. The proposed method used supervised machine learning (ML) methods such as random forest (RF), support vector machine, and artificial neural networks on the clusters. The model reaches 98.67% and 97.37% accuracy using RF in binary and multiclass classification. Utilizing RF on flow and MQTT features, TCP features, and top features from both clusters, classification accuracies of 96.96%, 91.4%, and 97.54% were obtained using cluster-based approaches.

In [6] IDS-FMLT model was used to predict malicious and regular traffic in the networks. It was tested on many heterogeneous datasets, which are CUP-99, KDD, and NetML- 2020. The experimental results of this model obtained an accuracy of 96.73% for training and 95.18% for validation with 4.82% miss rate in intrusion detection.

This paper [8] presented SecurityBERT for cyber threat detection in IoT networks, a new structure that utilized the Bidirectional Encoder Representations from Transformers (BERT) model. It is a 15-layer BERT-based architecture having only 11 million parameters for multi-category classification. A novel privacy-preserving encoding technique called Privacy- Preserving Fixed-Length Encoding (PPFLE) was integrated along with-it during training. It utilizes the Byte-level BytePair Encoder (BBPE) Tokenizer. In cyber threat detection, it performed better than usual ML and Deep Learning (DL) methods, such as Convolutional Neural Networks (CNNs) or Re- current Neural Networks (RNNs). EdgeIIoTset cybersecurity dataset was used and it identified 14 different attack types with 98.2% overall accuracy outperforming previous records set

by hybrid solutions such as GAN-Transformer-based architecture and CNN-LSTM models. It needs less computational power compared to other ML models with an inference time of less than 0.15 seconds on an average CPU and a compact model size of just 16.7MB. It can be implemented for resource-constrained IoT devices for real life traffic analysis.

Z. Wang et al. [3] introduced BERT-of-Theseus, Vision Transformer, and PoolFormer (BT-TPF), an IoT intrusion detection model with a knowledge distillation technique designed for IoT environments with limited computing resources. The model uses a Vision Transformer to train a small Poolformer model and a Siamese network to reduce features, obtaining a notable parameter reduction while maintaining high accuracy.

[15] They presented a transformer based NIDS model by using attention mechanism especially for cloud environments and used CIC-IDS 2018 dataset for model training and testing. It achieved accuracy above 93 percent and was compared against CNN-LSTM model. They trained the model in different scenarios (the number of encoder layers were 3, 4, or 5) and evaluated its results.

V. DISCUSSIONS: FEATURES AND LIMITATIONS

Synthetic minority [1] oversampling technique (SMOTE) approach was used to handle the imbalanced dataset which is widely used to oversample the minority class. It generates more synthetic examples of the minority class throughout the length of the line segments connecting some/all the minority class nearest neighbors to do oversampling. Two image filters were used: auto-color correlogram filter and fuzzy color and texture histogram (FcTH) Filter. Unlike color histograms, which only represent an image's color distribution, the auto- color correlogram filter shows how the spatial correlation between colors changes with distance. Lack of spatial knowledge could result in inaccurate predictions. A histogram finds it difficult to identify differences between the two images due to their similar color context. However, because a correlogram uses spatial information, it can easily identify the difference. The goal of the fuzzy color and texture histogram (FcTH) filter is to ensure that the features are sufficiently descriptive of the class while mapping an image's visual attributes to feature space. The FcTH filter uses and combines color and texture information from images, just like the auto-color correlogram filter. A fuzzy linkage histogram, formed by a fuzzy system, contains several pins, each of which represents a distinct color in the image. There are three fuzzy units in FcTH. The first fuzzy unit produces a hue saturation value (HSV) color space in 10 bins. The 10 bins are increased to 24 bins in the second fuzzy unit, then to 192 bins in the third unit. Next, the Gustafson–Kessel fuzzy classifier is used to map the 192-bin histogram into eight regions in the interval 0–7. Many filters were used to test a number of models. The auto-correlogram filter was used to create the first four models.

ref	Year	Accuracy	Precision	F1 Score	Recall	Transformer based	ML based	DL based
[1]	2023	✓	✓	✓	✓	✗	✓	✗
[3]	2024	✓	✓	✓	✓	✓	✗	✗
[5]	2021	✓	✗	✗	✗	✗	✓	✗
[6]	2023	✓	✗	✗	✗	✗	✗	✓
[8]	2024	✓	✓	✓	✓	✓	✗	✗
[15]	2024	✓	✓	✓	✓	✓	✗	✗

Fig. 3. Comparison analysis of models

But whether the results apply to other datasets or real-world situations is not made clear. Evaluating the model's generalizability outside of the experimental context is questionable.

BERT-of-Theseus [3] presents a Siamese network-based dimensionality reduction technique is presented that uses deep metric learning's benefits to encode input features. It increases the feature similarity between samples from various categories and decreases it across samples in the same category. It uses ViT-based 9-layer network intrusion detection model. This model can handle the challenges like limited storage capacity, poor communication environment, fewer computing resources, limited power of nodes and limited model generalization ability in the IoT network. CIC-IDS2017 and TON-IoT datasets were used for its training.

While the effectiveness of this model is shown on two particular datasets (TON_IoT and CIC-IDS2017), there can be concerns about how well it generalizes to other datasets or situations from the real world that have distinct features. Experiments on particular datasets are mentioned in the text, but it doesn't go into depth on how diverse the experiments were or how well the model held up in various situations especially where there are mobility scenarios.

In fused machine learning approach [6] during the validation stage, the suggested IDS-FMLT model is assessed using the KDD, CUP-99, and NetML-2020 datasets. To predict network traffic, the fused model is loaded from the cloud. Normal and malicious attacks are the two categories of network traffic that the suggested IDS-FMLT model predicts. Access is allowed to traffic if the suggested IDS-FMLT model predicts regular traffic. If malicious traffic is predicted by the model, then traffic is blocked and recorded as a noted attack in the cloud database.

On the other hand it causes increased computational complexity of system. Other methods such as federated learning, long short-term memory (LSTM), and hybrid computational intelligence can be used, for less cost and improving the system's accuracy. KDD, CUP-99, and NetML-2020 datasets are used but to make this system more reliable in real time application, other latest published intrusion detection data sets like CIC IoT Dataset 2022 and other bench marks dataset as well like NSL-KDD or UGR16 or UNSW-NB15[13] or CICDS-17, 18, and 19 during training phase can be used. Also is focused on limited

evaluation metrics such as F1 score and recall.

[5] Data preprocessing was done on the dataset. Its main contributions are imputation of missing values by three distinct methods: multiple imputations, linear regression, and mean. It performed binary and multi-class classification of malicious and regular packets by utilizing full features (37), TCP features (18), Flow and MQTT features (13) and top contributing features selected from TCP and flow and MQTT features set (11) and used three distinct supervised learning classifiers RF, SVM, and ANN for it.

More collection of appropriate features related to other IoT protocols could be used to increase the detection accuracy of known and unknown attacks by using suggested methodology. In Privacy-preserving BERT-based Lightweight Model [8] they introduced privacy into the training data through crypto- graphic hash functions and named this technique as Privacy-Preserving FixedLength Encoding (PPFLE). It has two main objectives. One is to maintain privacy by making sure that only encoded data is observed, which results in hiding sensitive information in the network data while preserving key classification features. Other is to convert unstructured network data into a structured format making it like natural English language, so that the BERT model can be implemented effectively. This dataset includes fifteen (15) types of attacks related to Internet of Things (IoT) and Industrial IoT (IIoT) connectivity protocols. These attacks are grouped into five main categories: DoS/DDoS attacks, Information Gathering, Man-in-the-middle (MITM) attacks, Injection attacks, and Malware attacks. In the DoS/DDoS attack category, examples like TCP SYN Flood, UDP flood, HTTP flood, and ICMP flood attacks are included. The Information Gathering category covers activities such as port scanning, operating system fingerprinting, and vulnerability scanning. MITM attacks involve attacks like DNS Spoofing and ARP Spoofing. Injection attacks include incidents such as Cross-Site Scripting (XSS), SQL injection, and file-uploading attacks.

Lastly, the Malware category covers threats like backdoors, password crackers, and ransomware attacks.

Apart from its advantages, it requires a very large dataset for training. As it includes 15 layers of encoders and it extracts 11 million features, small datasets can not be used for its training purposes. If too many features are extracted without enough information, the model may not perform well on test data and may not generalize well to new samples. A smaller dataset increases the likelihood of overfitting, a phenomenon in which the model becomes more adept in memorization of the training set than at generalizing to new information. Overfitting can occur when there is a high probability of capturing noise in the data due to feature extraction. Developing models on a small dataset with lots of features can be time-consuming and computationally costly. As the number of features grows, so does the complexity of the model, resulting in longer training times. Local databases cannot be used for dataset as they may lack packet network data. Edge-IIoTset dataset is used for training purpose which includes 15 attacks but this model was able to detect 14 attacks on same dataset.

[15] This model used 3,4 and 5 encoders. When the number of encoder layers was 3, the values of four evaluation indexes accuracy, precision, recall and F1-score were 93.38%, 91.7%, 93.38%, and 92.39% respectively. When the number of encoder layers were 4, they were 93.36%, 92.16%, 93.36%, and 92.10%. And when the number of encoder layers were 5, they were 93.46%, 92.19%, 93.4%, and 92.16%. It was designed especially for cloud-based environments.

Although it was designed for cloud environments but in diverse cloud environments such as edge cloud systems, it will face challenges due to the distributed nature of environment.

VI. EVALUATION METRICS

Evaluation metrics which are chosen for the comparison of these papers are mostly accuracy, precision, recall and F1 score.

Accuracy: The ratio of accurately predicted occurrences to all instances in the dataset is known as accuracy. It evaluates how accurate the model's predictions are overall as shown in Eq. (1).

$$Accuracy = (TP + TN)/(TP + TN + FP + FN) \quad (1)$$

Precision: Precision evaluates how well the model predicts positive predictions. It is the proportion of true positive predictions to all of the model's positive predictions. The precision of a model indicates its ability to avoid false positives as shown in Eq. (2).

$$Precision = TP/(TP + FP) \quad (2)$$

Recall (Sensitivity): Recall, sometimes referred to as sensitivity or true positive rate, assesses how well the model can distinguish positive examples from all of the dataset's actual positives. It is the proportion of true positive predictions to the total number of actual positive instances as shown in Eq. (3).

$$Recall = TP/(FN + TP) \quad (3)$$

F1 Score (F-measure): The F1 score is the harmonic mean of precision and recall as shown in Eq. (4).

$$F1 = 2TP/(2TP + FP + FN) \quad (4)$$

where True Positive (TP) represents the number of positive samples correctly predicted as positive; True Negative (TN) denotes the number of negative samples correctly predicted as negative; False Positive (FP) signifies the number of negative samples incorrectly predicted as positive; and False Negative (FN) signifies the number of positive samples incorrectly predicted as negative. These values can be obtained from the confusion matrix.

Table I: Summary of reference papers

Ref.	Dataset	Year	Model	Accuracy	Training Sample	Testing Sample	Advantages	Limitations	Performance Metrics
[1]	IEEE Dataport	2023	ML based	98.30%	90%	10%	Takes advantage of VGG-16 combined with stacked model	Limited generalizability. Results applicable on other datasets or real-world situations are not made clear.	F1 score, Recall, accuracy and precision
[3]	CIC-IDS2017 and TON-IoT	2024	BERT based	99.6% and 99.4% respectively	75%	25%	Siamese network-based dimensionality reduction technique and ViT	not discussed properly in mobility scenarios	F1 score, Recall, accuracy and precision
[5]	UNSW-NB15	2021	ML based	98.67% and 97.37% of accuracy in binary and multi-class classification	60% of original data was used from dataset	_	Data preprocessing, 3 types of imputations (mean, multiple and regression) on dataset was done	More collection of appropriate features related to other IoT protocols could be used	Accuracy
[6]	CUP-99, KDD, and NetML-2020	2023	DL based	95.18%	70%	30%	IDS-FMLT model was used to predict malicious and regular traffic in the networks.	more computational complexity, not trained on latest datasets	Accuracy
[8]	Edge-IIoTset	2024	LLM based	98%	80%	20%	Privacy-Preserving FixedLength Encoding (PPFLE) and 15 layers of encoders	Large dataset is required for testing. Was trained to detect 15 types of attacks but was able to detect 14.	F1 score, Recall, accuracy and precision
[15]	CIC-IDS 2018	2024	Transformer based	93%	70%	30%	Designed for cloud environment.	In diverse/distributed cloud environments such as edge cloud systems it will face challenges due to the distributed nature of environment.	F1 score, Recall, accuracy and precision

VII. CONCLUSION

In conclusion, the application of machine learning (ML) models in IoT intrusion detection presents a promising avenue for enhancing cybersecurity in interconnected systems. Through the utilization of various ML techniques, such as supervised learning algorithms and deep learning architecture, researchers have made significant strides in detecting and mitigating cyber threats in IoT networks. However, challenges remain, including the need for more diverse and comprehensive datasets, the development of lightweight models suitable for resource-constrained IoT devices, and the continual adaptation to evolving cyber threats. Future research efforts should focus on addressing these challenges to further advance the effectiveness and efficiency of ML-based intrusion detection systems in IoT environments.

REFERENCES

[1] Musleh, D., Alotaibi, M., Alhaidari, F., Rahman, A. and Mohammad, R.M., 2023. Intrusion Detection System Using Feature Extraction with Machine Learning Algorithms in IoT. Journal of Sensor and Actuator Networks,12(2), p.29.

[2] Altulaihan, E., Almaiah, M.A. and Aljughaiman, A., 2024. Anomaly Detection IDS for Detecting DoS Attacks in IoT Networks Based on Machine Learning Algorithms. Sensors, 24(2), p.713.

[3] Wang, Z., Li, J., Yang, S., Luo, X., Li, D. and Mahmoodi, S., 2024. A lightweight IoT intrusion detection model based on improved BERT-of- Theseus. Expert Systems with Applications, 238, p.122045.

[4] Thapa, N., Liu, Z., Kc, D.B., Gokaraju, B. and Roy, K., 2020. Comparison of machine learning and deep learning models for network intrusion detection systems. Future Internet, 12(10), p.167.

[5] Ahmad, M., Riaz, Q., Zeeshan, M., Tahir, H., Haider, S.A. and Khan, M.S., 2021. Intrusion detection in internet of things using supervised machine learning based on application and transport layer features using UNSW-NB15 dataset. EURASIP Journal on Wireless Communications and Networking, 2021(1), pp.1-23.

[6] Farooq, M.S., Abbas, S., Sultan, K., Atta-ur-Rahman, M.A., Khan, M.A. and Mosavi, A., 2023. A fused machine learning approach for intrusion detection system.

[7] Garcia-Teodoro, P., Diaz-Verdejo, J., Macia´-Ferna´ndez, G. and Va´zquez, E., 2009. Anomaly-based network intrusion detection: Techniques, systems and challenges. computers & security, 28(1-2), pp.18-28.

[8] Ferrag, M.A., Ndhlovu, M., Tihanyi, N., Cordeiro, L.C., Debbah, M., Lestable, T. and Thandi, N.S., 2024. Revolutionizing Cyber Threat Detection with Large Language Models: A privacy-preserving BERT- based Lightweight Model for IoT/IIoT Devices. IEEE Access.

[9] Kumar, S., 2007. Survey of current network intrusion detection techniques. Washington Univ. in St. Louis, pp.1-18.

[10] Srivastava, R. and Richhariya, V., 2013. Survey of current network intrusion detection techniques. Journal of Information Engineering and Applications, 3(6), pp.27-33.

[11] Saied, M., Guirguis, S. and Madbouly, M., 2024. Review of artificial intelligence for enhancing intrusion detection in the internet of things. Engineering Applications of Artificial Intelligence, 127, p.107231.

[12] Mukherjee, B., Heberlein, L.T. and Levitt, K.N., 1994. Network intrusion detection. IEEE network, 8(3), pp.26-41.

[13] Moustafa, N. and Slay, J., 2015, November. The significant features of the UNSW-NB15 and the KDD99 data sets for network intrusion detection systems. In 2015 4th international workshop on building analysis datasets and gathering experience returns for security (BADGERS) (pp. 25-31). IEEE.

[14] Nicho, M. and Girija, S., 2022, July. Evaluating Machine Learning Methods for Intrusion Detection in IoT. In Proceedings of the 12th Inter- national Conference on Information Communication and Management (pp. 7-12).

[15] Long, Z., Yan, H., Shen, G., Zhang, X., He, H. and Cheng, L., 2024. A Transformer-based network intrusion detection approach for cloud security. Journal of Cloud Computing, 13(1), p.5.