WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Investigating the Role of Intrusion Detection Systems in Edge Computing: A Comprehensive Review

[1]Saleh Fahad Siyi, [2]Abdusalam Ya'u Gital, [3]Fatima Umar Zambuk, [4]Sani Umar, [5]Odulu Lydia, [6]M A. Lawal & [7]I. Z. Yakubu*

[1,2,3,]Department of Computer Science, Abubakar Tafawa Balewa University Bauchi
[4]Department of computer science federal university of agriculture mubi, Adamawa state
[5,6]Ground Receiving Station, National Center For Remote Jos, Plateau State
[7]Department of Computing Technology,
SRM Institute of Science and Technology, Chennai, India

A B S T R A C T :

Edge computing (EC) has emerged as a transformative paradigm, bringing computational power closer to data sources to enhance efficiency and reduce latency. However, this decentralized architecture introduces new security challenges, necessitating robust Intrusion Detection Systems (IDS). This review explores the evolution of IDS in edge environments, analyzing traditional and contemporary detection techniques, including machine learning-based approaches such as Gradient Boosting, Decision Trees, and Long Short-Term Memory (LSTM) networks. We further assess the limitations of existing IDS frameworks and propose an ensemble-based detection system for improved accuracy and reduced computational overhead. Finally, we discuss emerging threats, research gaps, and future directions in securing edge-based infrastructures.

Keywords: Machine Learning, Gradient Boosting, Anomaly Detection, Signature-Based IDS, Ensemble Learning, Cybersecurity, Federated Learning, Blockchain-Based Security.

1. INTRODUCTION

The rapid proliferation of Edge Computing (EC), which brings computation and data storage closer to the location where it is needed, has significantly enhanced the performance and efficiency of various applications. However, this paradigm shift also introduces new security challenges, particularly in terms of intrusion detection. Traditional centralized intrusion detection systems (IDS) are not well-suited for edge environments due to latency issues, bandwidth constraints, and the distributed nature of edge devices (Abbas et al., 2017).

Edge Computing (EC) is a technology that brings computing resources and services closer to the edge of the network, typically at or near base stations or access points in a mobile network (Khan et al., 2019). Edge-based computing is transforming the landscape of data processing, bringing computational power closer to where data is generated. While this model enhances efficiency and reduces latency, it also introduces new vulnerabilities, as the devices at the edge are often less secure and more exposed to cyber-attacks. Traditional centralized security measures may be inadequate due to limited bandwidth, computational resources, and real-time requirements in these distributed environments (Khan et al., 2019). To address these challenges, Intrusion Detection Systems (IDS) are increasingly being deployed at the edge, offering a first line of defense against malicious activity. This allows for faster data processing, reduced latency, and improved user experiences for applications that require real-time processing, such as augmented reality, virtual reality, gaming, and Internet of Things (IoT) devices. Edge Computing (EC) aims to alleviate network congestion by processing data locally, rather than sending it all the way to centralized data centers. This technology can enhance the efficiency and capabilities of mobile networks (singh et al., 2021).

In today's interconnected world, securing networks and devices at the edge of computing systems has become increasingly vital. With the rise of the Internet of Things (IoT) and edge-based computing, the attack surface for malicious actors has expanded, necessitating advanced methods of intrusion detection (Abdulsahib & Khalaf, 2018). An Ensemble Intrusion Detection System (IDS) leveraging the Gradient Boosting Algorithm offers a promising solution to enhance security in edge-based environments. This approach combines the strengths of multiple machine learning models to detect and mitigate various forms of cyber threats more accurately and efficiently. By utilizing Gradient Boosting, a powerful machine learning technique known for its predictive accuracy and ability to handle complex data, this IDS is capable of identifying both known and novel threats in real-time, ensuring robust protection for edge-based systems where traditional security mechanisms may fall short. (Abdulsahib & Khalaf, 2018)

As information technology takes over the globe, security has become an inextricable problem (Abdulsahib & Khalaf, 2021). More knowledge is distributed to all parts of the globe from everywhere across the internet due to the remarkably rapid development of different Internet technology types (Alkhafaji et al., 2021). Any device sent across the World wide web could contain sensitive information, and in those instances, the sender and receiver must recognize information security issues before enjoying the ease and efficiency of the system (Al-Khanak et al., 2021). Network security measures such as Firewalls, IDS, Access Control, and Encryption are critical in protecting the confidentiality, integrity, and availability of data and resources in a

computer network. These measures help safeguard networks from various threats, including unauthorized access, data breaches, malware, and other cyber-attacks. IDS is the most essential security measure widely used in today's network

Intrusion Detection Systems (IDS) (AymanDawood et al., 2019), also identified as Intrusion Detection and Prevention Systems, are network appliances that record malicious behavior, record information about it, take action to stop it, and then report it. Intrusion detection systems will notify you if your network is being hacked, drop packets, and reconfigure the link to prohibit the client's IP from being blacklisted (Carlos et al., 2021).

Research of intrusion detection is evolving rapidly with the development of machine learning. Traditional machine learning techniques have been widely used in intrusion detection, such as decision tree (DT) (Safavian and Landgrebe, 1991), random forest (RF) (Zhang et al., 2008), and support vector machine (SVM) (Hsu et al., 2003). With the development of deep learning, convolutional neural network (CNN) (Vinayakumar et al., 2017), recurrent neural network (RNN) (Yin et al., 2017), and long short-term memory (LSTM) (Roy et al., 2017) are becoming popular in intrusion detection. These techniques are based on different principles, and how to effectively exploit their advantages to address intrusion detection tasks in particular domains remains an open research question. Existing IDSs can be divided into two categories based on the detection method: signature-based detection and anomaly-based detection (Ghorbani et al., 2009)

Signature-based detection is typically best used for identifying known threats. It operates by using a pre-programmed list of known threats and their indicators of compromise. An indicator of compromise might be a specific behavior that generally precedes a malicious network attack, file hashes, malicious domains, known byte sequences, or even the content of email subject headings. As a signature-based IDS monitors the packets traversing the network, it compares these packets to the database of known indicators of compromise or attack signatures to flag any suspicious behavior.

Meanwhile, Anomaly-based intrusion detection systems can alert you to suspicious behavior that is unknown. Instead of searching for known threats, an anomaly-based detection system utilizes machine learning to train the detection system to recognize a normalized baseline. The baseline represents how the system normally behaves, and then all network activity is compared to that baseline. Rather than searching for known indicators of compromise, anomaly-based IDS simply identifies any out-of-the-ordinary behavior to trigger alerts.

With an anomaly-based IDS, anything that does not align with the existing normalized baseline such as a user trying to log in outside of standard business hours, new devices being added to a network without authorization, or a flood of new IP addresses trying to establish a connection with a network will raise a potential flag for concern. The disadvantage here is that many non-malicious behaviors will get flagged simply for being atypical. The increased likelihood for false positives with anomaly-based intrusion detection can require additional time and resources to investigate all the alerts to potential threats.

At the same time, this potential disadvantage is also what makes anomaly-based intrusion detection able to detect zero-day exploits signature-based detection cannot. Signature-based detection is limited to a list of known, existing threats. On the other hand, it also has a high processing speed and greater accuracy for known attacks. These two detection methods have advantages and disadvantages that generally complement each other well, and are often used best in tandem. Edge computing extends cloud capabilities to localized processing units, reducing latency and network congestion. However, this distributed nature increases the attack surface, making traditional security solutions inadequate. IDSs have become crucial in detecting and mitigating cyber threats in EC. This review presents an in-depth analysis of IDS mechanisms, their applications in edge environments, and future security trends. The remainder of this paper is organized as follows: Section 2 provides literature review and related work on edge computing and IDS; Section 3 concludes with key findings and recommendations.

2. LITRETURE REVIW

2.1Edge-based Computing

Edge-based computing, often referred to as edge computing, involves processing data closer to the source of generation rather than relying solely on centralized data centers. This approach enhances efficiency, reduces latency, and improves the performance of applications, especially in scenarios like IoT (Internet of Things), real-time analytics, and autonomous systems. Edge Computing (EC) is a paradigm that brings computing resources and services closer to the network edge, enabling low-latency and high-bandwidth applications. In this literature review, we explore the potential threats and their preventions in Edge Computing and highlight key research works in this field.

Edge Computing often encounters two main types of attacks. One is, after authentication, the edge device is frequently exploited. Since the device has particular privileges for some inside network activity, any insider can use it to engage in undesirable behavior. This is regarded as insider attacks or unauthorized attacks in which the intruder tries to infiltrate the edge layer or cloud layer. Two, Unauthenticated edge devices occasionally attempt to attack the device layer or cloud layer using cutting-edge tactics (Zhang and Liu, 2020). This is known as unauthenticated attack or outside attack. Due to the multitenant architecture used in mobile edge computing environments, where resources are shared across numerous apps, insider attacks are exceedingly difficult to detect. Consequently, hostile edge device attacks are the main issue in mobile edge computing setting. The firewall serves as the initial layer of defense for such networks. However, a conventional firewall solution can only stop internet traffic coming from the outside. However, it is unable to filter the malicious inside packets produced by insider attack. (Mahesh, 2019). Such firewall-based solutions don't work because of the edge nodes' complexity and diversity. Additionally, it is expensive and difficult to operate a sizable number of firewalls on most EC devices to apply security solutions. Edge Computing holds great potential for a wide range of applications, including augmented reality, IoT services, video streaming, mobile gaming, and intelligent transportation.

2.2 Edge Computing Applications

2.2.1 Augmented Reality (AR) and Virtual Reality (VR):

Edge Computing has been identified as a promising solution for delivering immersive AR and VR experiences. By offloading computationally intensive tasks to edge servers, EC reduces latency and improves the quality of AR/VR content rendering. Researchers have proposed frameworks and architectures for EC-enabled AR/VR systems (Chen et al., 2019; Hu et al., 2020). Augmented Reality (AR) is a technology that integrates computer-generated content into the real-world environment, enhancing the user's perception and interaction with the surroundings. Over the years, AR has gained significant attention across various domains, including gaming, education, healthcare, and industrial applications. This literature review provides an overview of the key advancements, challenges, and potential applications of augmented reality.

2.2.2 Internet of Things (IoT) Services:

The combination of Edge Computing and IoT enables real-time data processing and analytics at the network edge. Edge servers provide low-latency and localized services for IoT devices, enabling efficient data processing, event detection, and response. Research has focused on MEC-based IoT platforms and applications, including smart cities, industrial automation, and healthcare (Bonomi et al., 2014; Ning et al., 2018).

2.2.3 Video Streaming and Content Delivery:

Edge Computing can significantly improve the performance of video streaming services by caching and delivering content from edge servers. By reducing the distance between the user and the content source, EC reduces latency and improves video quality. Research efforts have explored EC-based video delivery architectures, adaptive streaming algorithms, and content caching strategies (Mao et al., 2017; Wang et al., 2019).

2.2.4 Mobile Gaming:

EC offers opportunities for enhancing mobile gaming experiences by providing low-latency communication and computation capabilities at the network edge. Edge servers can offload game processing tasks, reduce network congestion, and enable real-time multiplayer gaming. Studies have proposed EC architectures and algorithms for game offloading and resource management (Hu and Mao, 2020).

2.2.5 Vehicular Networks and Intelligent Transportation:

EC has implications for intelligent transportation systems by enabling real-time data processing and decision-making at the network edge. Edge servers can support applications such as traffic management, collision detection, and autonomous vehicle control. Research has explored EC-based vehicular networks, edge analytics, and cooperative sensing (Zhang et al., 2019; Zhang and Liu, 2020).

2.2.6 Edge-based IDS security architecture

The majority of today's enterprise security systems are cloud-based, and service providers are responsible of meeting all security standards. However, current security solutions are not scalable due to the ad hoc environment and low-latency requirements of applications like EC and IoT. For the EC environment, the edge computing idea offers a fresh approach to designing and deploying innovative security solutions. Any edge-based security solution's primary goal is to fulfill all security criteria at the network's edge. Instead of using the cloud, this paradigm provides security solutions at the edge computing layer. Additionally, by allowing processing and storage capabilities at the edge network, this can offload the processing task from the end devices. Thus, the edge computing paradigm minimizes network latency and congestion in addition to enhancing system security. User-centric, device-centric, and end-to-end security are the three basic classifications of the edge-based security architecture (Sha, 2020). When there is uncertainty and a high demand for a quick response, an edge-based IDS can function. The end-user layer, mobile edge networking layer, and data storage layer are the three core layers of the edge-based mobile computing architecture. Resources, data, and services with security features make up the data storage layer. The smart end-user is initially connected to the system through edge devices at the end-user layer. The mobile edge networking layer then receives all of the raw traffic from edge applications. The majority of the crucial security elements are implemented in this layer. In this work, the Edge-based Hybrid IDF is deployed in this layer. This layer's duties include lowering network latency, meeting several real-time requirements, offloading difficult computational operations, processing data rapidly, offering security measures, keeping track of all traffic data, and many other things. The data storage layer saves the data in the cloud after processing it.

2.5 IDS overview and limitations

There are generally four different IDS types that are used to secure the EC environment. These are Host-Based (HIDS) (Besharatiet al.,2019), Network-Based (NIDS), Hypervisor-Based, and Distributed IDS (DIDS). Host-based IDS are in responsible of keeping tabs and examining the data gathered from a particular host machine. Network-based IDS are employed to identify network intruders by comparing in real-time the behavior of network traffic with that which has previously been observed. DIDS is made up of several HIDS/NIDS for monitoring network traffic across a large network. It enables a user to observe and analyze communications between virtual machines (VM). Hypervisor-based IDS are primarily employed in cloud computing for the purpose of detecting intrusions in a virtual environment. These IDS use different types of intrusion detection techniques based on: Signature (Bakshiet al.,2010), Anomaly (Aljawarnehet al., 2018), Artificial Neural Network (ANN) (Akashdeepet al.,2017), Fuzzy Logic (Dovomet al., 2019), Association Rule (Aljawarnehet al., 2018), Support Vector Machine (SVM) (Gauthamaet al.,2017), Signature based IDS primarily uses pattern matching to find intrusions by comparing captured patterns to databases of previously created patterns.

Table 1: The intrusion detection modules

Modules	Functions	Limitations
Signature based IDS	uses pattern matching to find intrusions by comparing captured patterns to databases of previously created patterns	Unable to detect unknown attacks. high FAR

Anomaly based IDS	Detects unidentified attacks	Poor accuracy
ANN-based IDS	Categorizes unstructured network packets effectively using a number of hidden layers.	Takes more time Requires more sample training
SVM-based IDS	Handles a lot of preprocessed data with great accuracy	Can only detect intrusions for a specific sample of data.
Fuzzy Logic	Yields positive results for several ambiguous issues	Less detection rate
Association Rule-based IDS	Yields positive results for several ambiguous issues	Less detection rate

As a result, they are unable to detect unknown attacks, which results in a high FAR for such attacks. Anomaly has poor accuracy in detecting unidentified attacks. As a result of this flaw, ANN-based IDS was identified in the literature. ANN-based IDS (Akashdeepet al., 2017) effectively categorize unstructured network packets using a number of hidden layers. But, It takes more time and sample training. Fuzzy logic-based IDS and Association Rule-based IDS it yields positive results for several ambiguous issues. But the detection rate is less. SVM-based IDS can correctly categorize intrusions for a specific sample of data. A lot of preprocessed data can be handled by this kind of IDS with great accuracy. In GA-based IDS, complexity is increased with computational cost, whereas Hybrid IDS is an efficient approach to accurately classifying rules. The mobile IDS consist of handheld wireless devices (Smartphone's or mobile devices) with intrusion detection capabilities (Abdenacer et al., 2021). It has a self-configurable network where the system deploys very quickly and autonomously without the assistance of any third party. Several design and implementation issues emerged as the IDS technology was migrated from a standalone computer to a mobile device. Battery-powered, limited energy supply, constrained resources, limited computing power, poor memory, and low sensing range are some of the difficult problems with such a mobile-based IDS. The mobile IDS system is more vulnerable because of its ad hoc nature and independent functioning nodes, which may not be sufficient to detect intrusions. In some cases, some mobile applications run on a particular platform and permit third-party applications. Thus, Smartphone's cannot detect new intrusive traffic and unique vulnerabilities. Due to resource limitations, the nature of mobile IDS may have a high response time for suspicious activities and limited traffic pattern visualization. While attempting to identify intrusive packets using a mobile device, some limitations of mobile IDS should be kept in mind. These include the physical security of mobile nodes, the lack of a central security management point, single points of failure, an undefined network boundary, and low cryptographic support.

2.8 Signature detection module (SDM)

A signature detection module is a component or algorithm used in various domains to identify and detect specific patterns or signatures of interest. It is commonly used in areas such as network security, intrusion detection, malware analysis, fraud detection, and more. In network security and intrusion detection systems, a signature detection module analyzes network traffic or system logs to identify known patterns of malicious activity or attacks. These patterns are typically represented as signatures, which are specific sequences of bytes, strings, or behavior associated with known threats or vulnerabilities. The signature detection module compares the observed data against a database of pre-defined signatures and raises an alert if a match is found.
Similarly, in malware analysis and antivirus systems, a signature detection module scans files or memory for specific patterns or sequences that are characteristic of known malware. These signatures can include file hashes, byte sequences, or behavioral indicators associated with malicious software. When a file or system is scanned, the signature detection module compares the observed data against a database of known malware signatures to identify any matches. The specific implementation and techniques used in a signature detection module can vary depending on the application and domain. Some techniques employed include pattern matching algorithms (e.g., string matching, regular expressions), machine learning approaches (e.g., classification algorithms, anomaly detection), and heuristics.

2.8.1 C4.5 Algorithm

Ross Quinlan Published a paperTitled "Induction of decision trees"Machine Learning, 1986introduces the C4.5 algorithm, which is a widely-used decision tree learning algorithm. Decision trees are a popular machine learning technique for classification tasks, and the C4.5 algorithm builds upon its predecessor, the ID3 algorithm, to address some of its limitations. The main focus of the paper is to present the C4.5 algorithm and its unique features. Quinlan discusses the decision tree learning process, including attribute selection, tree construction, and tree pruning. The C4.5 algorithm uses an information-theoretic approach, specifically the concept of information gain, to decide the most informative attributes for splitting the data at each step. It can handle both discrete and continuous attribute values and handles missing data as well. Quinlan's paper provides a detailed description of the C4.5 algorithm's key steps, along with explanations of the information gain calculation, attribute selection heuristics, and pruning techniques. The author also discusses experimental results comparing C4.5 with other decision tree algorithms and demonstrates its effectiveness in handling real-world datasets. The significance of the C4.5 algorithm lies in its ability to handle both categorical and continuous attributes, handle missing data, and its flexibility in tree construction and pruning. The algorithm has been widely adopted in various fields and has influenced subsequent research in decision tree learning.Top of Form

C4.5 is a successor of ID3 used an extension to information gain known as gain ratio to overcomes the bias of Information gain and applies a kind of normalization to information gain using a split (Ashaet al 2010). The C4.5 algorithm is an improvement of the ID3 algorithm, developed by Quinlan Ross in 1993. It is based on Hunt's algorithm and like the ID3, it is serially implemented. Pruning takes place in C4.5 by replacing the internal node with a leaf node, thereby reducing the error rate (Podgorelecet al, 2015). Unlike the ID3, the C4.5 accepts both continuous and categorical attributes in building the decision tree. It has an enhanced method of tree pruning that reduces misclassification errors, due to noise or too-much detail in the training data set. Like the ID3 the data is sorted at every node of the tree, in order to determine the best splitting attribute. It uses the gain ratio impurity method to evaluate the splitting attribute (mousafaet al 2017).

In order to avoid the multi-value bias problem, the C4.5 method used gain ratio instead of the information gain equation, which is a good solution to the multi-value bias problem, but more logarithmic expressions are brought into the computation process by gain ratio, which will influence the running time, in the developing the model with has very significant impact on the evaluation performance and good quality to predict the feature data. The model is a fraction between information gain and its splitting information.

Decision Tree is a non-parametric supervised machine learning model that is used for both classification and regression applications (Quinlan, 1986). It splits the data after a particular attribute repeatedly. The deduced decision rules from the data attributes are learned by the decision tree. It forecasts the value of the target variable based on those rules. This model's decision-making process can be represented as a tree, which makes it simpler for the user to understand. The result of DT can be seen using a variety of ML tools. (Safavian and Landgrebe 1991). The core ideas of DT are two units: leaves and decision nodes. Data is split depending on a specific parameter in the decision node, and the result or decisions are acquired in the leaves unit. Entropy (Shannon, 1948), which gauges the impurity of the split, was used as the splitting criterion in this experimental investigation. The following equation can be used to determine the entropy equation for each internal decision node in the decision tree:

2.9 Anomaly detection module (ADM)

Anomaly detection is a technique used to identify patterns or instances that deviate significantly from the norm or expected behaviour within a dataset. An anomaly detection module is a component or algorithm designed to detect and flag such anomalies. It is employed in various domains, including cyber security, fraud detection, system monitoring, and quality control, among others. According to Akoglu and Koutra, (2015) the goal of an anomaly detection module is to distinguish normal or expected patterns from abnormal or anomalous ones. This is typically achieved by learning patterns from a training dataset and then applying them to new, unseen data to identify deviations. Anomaly detection can be performed using different techniques, including statistical methods, machine learning algorithms, or a combination of both.

2.9.1 Common approaches used in anomaly detection modules include:

1. Statistical Methods: Statistical methods involve calculating statistical metrics such as mean, standard deviation, or percentile values from the training data. Data points that fall outside certain statistical thresholds are flagged as anomalies.
2. Unsupervised Machine Learning: Unsupervised machine learning techniques aim to discover patterns or clusters within the data without relying on pre-labeled training examples. Anomalies are identified as data points that do not conform to the learned patterns or clusters.
3. Supervised Machine Learning: In some cases, anomaly detection can be formulated as a supervised learning problem if labeled examples of anomalies are available. Supervised machine learning algorithms are trained on both normal and anomalous instances to learn the characteristics of anomalies and distinguish them from normal patterns.
4. Time Series Analysis: Anomaly detection in time series data involves analyzing the temporal patterns and identifying data points that deviate significantly from expected trends or seasonal patterns.

2.9.2 Adaptive boosting (AdaBoost)

AdaBoost, short for Adaptive Boosting, is a machine learning algorithm used for classification tasks. It is an ensemble learning method that combines multiple weak classifiers to create a strong classifier. AdaBoost was introduced by Yoav Freund and Robert Schapire in 1996. The main idea behind AdaBoost is to iteratively train a series of weak classifiers, where each weak classifier focuses on the misclassified instances from previous iterations. The final strong classifier is a weighted combination of these weak classifiers.

2.9.3 High-level overview of the AdaBoost algorithm:

Schapire (2003) first introduced the AdaBoost algorithm. It is an ensemble learning methodology that uses an iterative process to fix the errors made by weak learners. In order to improve the performance of the model, it continuously invokes a basic learning algorithm or a weak learner. Reassigning weights to each instance and giving incorrectly identified instances higher weights is the core idea behind AdaBoost. Briefly stated, when training the Adaboost model, the basic classifier (such as DT) is first trained, and it then makes use of that classifier to make predictions using the training data. The second classifier is then trained by increasing the weight of improperly categorized training instances, and using the newly updated weights, it once more makes a prediction on the training set. The weights of the instances are then updated once more, and so on. Up until the very last basic learner, this process will be carried out.

Initialization: Assign equal weights to all training examples.

Iterative Training:

a. Train a weak classifier on the training data using the current weights.

b. Evaluate the performance of the weak classifier and calculate the weighted error rate, which represents the misclassification rate weighted by example weights.

c. Update the weights of the training examples. Increase the weights of misclassified examples and decrease the weights of correctly classified examples.

d. Repeat steps a-c for a predefined number of iterations or until a certain condition is met.

Final Classifier:

Combine the weighted predictions of all weak classifiers into a final strong classifier. The weights of the weak classifiers are determined based on their performance during training.

During the prediction phase, the final strong classifier classifies new instances based on the weighted votes of the weak classifiers.

AdaBoost has been widely used in various applications, including face detection, object recognition, text categorization, and bioinformatics, among others. It is known for its ability to handle complex classification tasks by combining multiple weak classifiers and achieving high accuracy.

2.9.4 Overview of Gradient Boosting

Gradient boosting is a popular machine learning technique used for both regression and classification tasks. It is an ensemble method that combines multiple weak prediction models, typically decision tress, to create a stronger predictive model. The 'gradient' in gradient boosting refers to the optimization process used to iteratively improvr the model's performance. It works by fitting the weak models to the errors or residuals of the previous models in the ensemble, with each subsequent model trying to minimize the remaining errors. Boosting on the other hand, refers to the process of subsequently adding weak models to the ensemble, with each model learning from the mistakes of its predecessors. This iterative process continues until a predefined stopping criterion is met, such as reaching a certain number of models or when the models performance plateaus. Gradient boosting has proven to be highly effective and is widely used in various domains, including machine learning competitions, finance, and healthcare. Some popular implementations of gradient boosting include XGBoost, LightBoost, LightGMB and CatBoost, each with its own unique features and optimization. By combining multiple weak models, gradient boosting can provide rebust predictions, handle complex data patterns, and reduce bias and variance. However, it is important to carefully tune hyper parameters and avoid over fitting, as gradient boosting can be prone to capturing noise if not properly regularized

2.10 Ensemble Detection Module (EDM)

The two traditional IDSs stated above cannot adequately safeguard our information systems against the constantly changing types of threats. There is a need for new methods of combining different intrusion detection systems to improve their effectiveness. Hence, the proposed Ensemble intrusion system as several researches have shown that combined algorithms perform better than single algorithms (Musa et al.,2021) The goal of Ensemble intrusion detection systems is to combine several detection models to achieve better results. A hybrid intrusion detection system consists of two components. The first component processes the unclassified data. The second component takes the processed data and scans it to flag out intrusion activities (Khari and Karar,2013).

2.11 Related works

A literature review of new discoveries about IDS solutions in EC networks has been conducted (Chen and Ran, 2019). To safeguard the edge network from insider attack, a firewall architecture has been created. (Markham and Payne 2001). This architecture supports accurate, insurmountable, and tamper-resistant features to be present in any security system. A deep learning approach for intrusion detection in online communities has been explored in (Yin et al., 2017). In order to evaluate the effectiveness of the system, Recurrent Neural Networks (RNN) is in the lead for binary and multi-class classification. High computational processing was observed in this system, which will lower its efficiency. (Muna and Moustafa, 2018). A Distributed Intrusion Detection Systems (DIDS) has been proposed in (Menget al., 2018). This research intends to reduce the false alarm rate in DIDSs-based edge computing systems. Additionally, they reduced the response time and energy consumption. A deep belief network for the Edge-of-Things (EoT) has been proposed in (Almogren, 2020). The proposed system is capable of identifying intrusive behavior in the EoT network. Data collection, feature extraction, and classification modules make up the proposed framework. But this model has a high cost and computational requirement. A key concern is the network security of the Internet of Things (IoT). To view this security issue, in [Liang et al., 2020), proposed a robust IDS. A multi-agent system, blockchain, and deep learning algorithms make up this approach. Although the system is highly efficient, combining three separate approaches makes the system more complex and increases response time. Device-edge-based IDS for the IoT infrastructure has been proposed in (Mudgerikar et al., 2020). Behavioral profiles and system-level data are used to create the IDS. Effective detection is supported by the special split architecture, which has very little delay. But the system architecture's complexity caused a computational overload.

An IDS has been developed in (Vimalet al., 2020) for the internet industry. Additionally, they developed the Cloudlet concept, which is used to deploy Edge-based IoT devices in cities. A database module, a mobile application module, and a microcontroller module make up the proposed model. But The model's security effectiveness and performance are poor. In (Cao et al.,2020), a network IDS for mobile edge computing was proposed. This method collects all of the tcp dump packets, analyzes and extracts the features, and then forwards the packet into the network if it is determined to be an authentic packet. To learn the behavioral pattern of a typical packet, a topic model is trained. However, the accuracy of detection is degraded as new packet types enter networks. Data-driven mimicry and game theory-based IDS have been proposed in (Li et al., 2020). In the edge computer networks, the new attacks are examined based on participant game income and player game balance points. They also try to reduce the cost of the IDS. Traffic inspection and classification-based distributed attack model has been proposed in (Koziket al. 2018) for the IoT applications.

Table: 2 Comparative analyses of IDS solutions in different areas

Paper	Title	Year	Working environment	Aim of the work	Proposed approach	Limitation
Azeroual and Nikiforova	Apache Spark and MLlib-Based Intrusion Detection System or How the Big Data Technologies Can Secure the Data	2022	Apache Spark and MLlib	utilizing machine learning to find data anomalies	k-means algorithm for clustering analysis implemented in Sparks MLlib	Anomaly detection is not actively and thoroughly checked.
Yusuf Musa	Artificial Neural	2022	MATLAB	use neural network	Multi-Layer Perceptron	Can not scale

Malgwi et al	Network Model for Intrusion Detection System		Neural Network Toolbox version 2017b	technique to create an Intrusion Detection System (IDS).	was used to build the Artificial Intelligence	data
DheerajBasavaraj and ShahabTayeb	Towards a lightweight intrusion detection framework for in-vehicle networks	2022	Google collab	To develop efficient IDS that detects anomalies in the vehicular system.	Artificial neural network model	
Hind Bangui et al.,	**A hybrid machine learning model for intrusion detection in VANET**	**2022**	MATLAB R2019a	a hybrid machine learning model for intrusion detection to address the real-time attack detection in VANET	Random Forest and a posterior detection based on coresets to improve the detection accuracy and increase detection efficiency	a large number of trees can make the algorithm too slow and ineffective for real-time predictions. In general, these algorithms are fast to train, but quite slow to create predictions once they are trained.
Adeel Abbas	A new ensemble-based intrusion detection system for internet of things	2021		An ensemble-based intrusion detection model and Cross-comparison of several feature selection methods	three supervised classification algorithms; decision tree, naive Bayes and logistic regression. Stacking classifier is used for ensemble learning with hard voting	REQUIRES TOO MANY ALGORITHMS
SumeghTharewal et al.,	Intrusion detection system for industrial Internet of Things based on deep reinforcement learning	2022	Stable (2.10.0)	a near-end strategy optimization method for the Industrial Internet of Things intrusion detection system	deep reinforcement learning algorithm	It requires very large amount of data in order to perform better than other techniques
SitiMaesaroh	Wireless Network Security Design And Analysis Using Wireless Intrusion Detection System	2022	Linux operating system	Wireless Intrusion Detection System	Implementing the Linux operating system with Snort as a sensor engine and Iptables as an attack handler	WORKS ONLY ON LINUX OS
YakubuImrana	χ2-BidLSTM: A Feature Driven Intrusion Detection System Based on χ2 Statistical Model and Bidirectional LSTM	2022	Python's TensorFlow and Keras libraries ON	novel feature-driven intrusion detection system, χ2-BidLSTM	χ2 statistical model and bidirectional long short-term memory (BidLSTM)	higher complexity and needs more training time
Wei Wang Lo et al.,	Representation learning-based network intrusion detection system by capturing explicit and implicit feature interactions	2022	Linux os	a novel Network Intrusion Detection System(, RL-NIDS)	, unsupervised Feature Value Representation Learning module (FVRL) and supervised Neural Network for object Representation Learning (NNRL)	Hard ware comlexity
Doaa N. Mhawi et al.,	Advanced Feature-Selection-Based Hybrid Ensemble Learning Algorithms for Network Intrusion Detection Systems	2022	Colabplatform	novel Ensemble Learning (EL) algorithm-based network IDS model to enhance the detection capabilities of IDS	CFS–FPA–ensemble method	Inability to tackle infrequent traffic problems

				to protect service providers against attacks		
Emad-ul-HaqQazi et al.,	An intelligent and efficient network intrusion detection system using deep learning	2022	TensorFlow library and GPU framework	new approach for network intrusion detection based on the DL approach	non-symmetric deep auto-encoder for network intrusion detection problems and presents its detailed functionality and performance	Hardware intensive
WaiWeng Lo	E-GraphSAGE: A Graph Neural Network based Intrusion Detection System for IoT	2022	GNNs	E-GraphSAGE	Graph Neural Networks	TIME COMLEXITY
Vinayakumar Ravi	Recurrent deep learning-based feature fusion ensemble meta-classifier approach for intelligent network intrusion detection system	2022	Kaggle GPU environment with K80 GPU	end-to-end model for network attack detection and network attack classification	deep learning-based recurrent models. Scikit-learn5 and Keras6 with TensorFlow7	Not tested in adversarial environment
Qusay M Alzubi et al.,	Intrusion Detection System Based On Hybridizing A Modified Binary Grey Wolf Optimization And Particle Swarm Optimization	2022	MATLAB	a hybridization of modified binary Grey Wolf Optimization and Particle Swarm Optimization	SVM AND DECISION TREE CLASSIFIER	DATASET NOT CURRENT
Ruizhe Zhao et al.,	A hybrid intrusion detection system based on feature selection and weighted stacking classifier	2022	Python and Scikit-learn on Windows10 system	A Hybrid Intrusion Detection System	Feature Selection and Weighted Stacking Classifier	

2.12 Research Gap

In today's MEC environment, several technical challenges arise with applications requiring real-time detection and decision-making, such as autonomous vehicles, industrial monitoring, healthcare diagnostics, and IoT systems, these environments demand swift and accurate responses, often in scenarios where the consequences of delays or errors can be critical. However, achieving this level of performance is hampered by three interconnected issues as identified from the literature;

High Latency and Bandwidth Constraints: Many systems rely on transmitting large volumes of data to centralized servers (e.g., cloud computing) for processing. This introduces delays due to the time required for data transfer, especially in networks with limited bandwidth or high latency. Furthermore, relying on constant communication to the cloud can be expensive and impractical in remote or bandwidth-limited environments.

Limited Computational Resources: The devices deployed in these environments often have constrained hardware capabilities, including low-power processors, limited memory, and restricted storage. These limitations prevent the deployment of computationally intensive algorithms, such as those used in advanced machine learning or image processing, directly on the devices.

Need for Real-Time Detection: Real-time systems demand immediate analysis and response to data inputs. This is particularly important in safety-critical applications, such as preventing industrial equipment failures or detecting anomalies in medical systems. However, meeting these requirements is challenging when devices cannot perform complex processing locally or depend on slow and unreliable network connections for remote computation. The combination of these challenges results in delays, reduced accuracy, and inefficiencies in critical systems, potentially leading to missed opportunities for timely interventions or even catastrophic failures. Addressing this problem requires innovative solutions that balance the need for computational efficiency, minimal latency, and effective bandwidth usage, enabling robust real-time performance in constrained environments.

CONCLUSION AND RECOMMENDATIONS FOR FURTHER STUDIES

Intrusion Detection Systems (IDS) are essential for securing Edge Computing (EC) environments, which are increasingly vulnerable to sophisticated cyber threats. This review has highlighted the evolution of IDS mechanisms, including traditional and machine learning-based approaches, such as Decision Trees, Gradient Boosting, and Long Short-Term Memory (LSTM) networks. While these methods enhance detection accuracy and efficiency, challenges such as high false alarm rates, computational overhead, and adaptability to emerging threats persist.

The integration of ensemble-based IDS frameworks has shown promise in mitigating some of these limitations by combining multiple detection techniques. Additionally, emerging technologies such as federated learning and blockchain-based security solutions offer new pathways for enhancing EC security. However, these methods require further optimization to ensure real-time intrusion detection with minimal resource consumption. Overall, while significant advancements have been made, the evolving nature of cyber threats necessitates continuous innovation in IDS frameworks to improve accuracy, reduce computational burdens, and ensure the robustness of security solutions in EC environments.

3.2 Open Issues for Further Studies

Despite progress in IDS for EC, several open issues remain, requiring further investigation:

1. Explainable AI for IDS – The complexity of deep learning-based IDS models makes them difficult to interpret. Future research should focus on developing explainable AI techniques to enhance transparency and trustworthiness in decision-making.
2. Lightweight IDS for Resource-Constrained Edge Devices – Many IDS models require significant computational power, making them unsuitable for edge devices with limited resources. Optimizing IDS for energy efficiency and low computational overhead is a crucial research direction.
3. Adaptive and Self-Learning IDS – Existing IDS solutions often struggle to adapt to new attack patterns. Implementing adaptive learning mechanisms that can continuously update their models without extensive retraining remains a challenge.
4. Federated Learning for Privacy-Preserving IDS – Traditional IDS solutions often require centralized data processing, which raises privacy concerns. Federated learning can enable collaborative intrusion detection while preserving data privacy, but its effectiveness and scalability need further exploration.
5. Blockchain-Based Security for Edge Computing – Blockchain technology offers potential solutions for ensuring data integrity and secure authentication in IDS. However, challenges such as scalability and latency must be addressed before widespread adoption.
6. Integration of IDS with 5G and IoT Networks – The growing deployment of 5G and IoT devices requires IDS models that can handle high-speed, low-latency environments. Future research should explore how IDS can be effectively integrated into next-generation networks.
7. Reducing False Positives and False Negatives – High false alarm rates remain a significant challenge in anomaly-based IDS. Developing more robust hybrid detection techniques that balance precision and recall is essential.
8. Cyber Threat Intelligence Sharing in IDS – Collaborative security frameworks that enable real-time sharing of threat intelligence between edge nodes can improve IDS effectiveness. Research on secure and efficient ways to facilitate such data sharing is needed.

Addressing these open issues will enhance the security of EC environments and contribute to the development of more resilient IDS frameworks for next-generation computing infrastructures

REFERENCES

1. Abbas N, Zhang Y, Taherkordi A, Skeie T (2017) Mobile edge computing: a survey. IEEE Internet Things J 5(1):450–465
2. AbidSalih A, Mohsin AA (2021) Evaluation of classification algorithms for intrusion detection system: a review. J Soft Comput Data Min 2(1):31–40
3. Ahmad, R. W., Gani, A., Hamid, S. H. A., Shiraz, M., Yousafzai, A., & Xia, F. (2015). A survey on virtual machine migration and server consolidation frameworks for cloud data centers. Journal of Network and Computer Applications, 52, 11–25. doi:10.1016/j.jnca.2015.02.002
4. Aljawarneh S, Aldwairi M, BaniYassein M (2018) Anomalybased intrusion detection system through feature selection analysis and building hybrid efficient model. J ComputSci 25:152–160
5. Ashish Singh, KakaliChatterjee,· Suresh Chandra Satapathy (2021):An edge based hybrid intrusion detection framework for mobile edge computing. Complex & Intelligent Systems https://doi.org/10.1007/s40747-021-00498-4
6. Bondada, M. B., &Bhanu, S. M. S. (2015).Analyzing user behavior using keystroke dynamics to protect cloud from malicious insiders.In 2014 IEEE International Conference on Cloud Computing in Emerging Markets, CCEM 2014 (pp. 1–8). Bangalore, India: IEEE. doi:10.1109/ CCEM.2014.7015481
7. Asif-Ur-Rahman M, Afsana F, Mahmud M, Kaiser MS, Ahmed MR, Kaiwartya O, James-Taylor A (2018) Toward a heterogeneous mist, fog, and cloud-based framework for the internet of healthcare things. IEEE Internet Things J 6(3):4049–4062
8. Azeroual, O.; Nikiforova (2022), A. Apache Spark and MLlib-Based Intrusion Detection System or How the Big Data Technologies Can Secure the Data. Information 2022, 13, 58. https://doi.org/10.3390/ info13020058
9. Cao X, Fu Y, Chen B (2020) Packet-based intrusion detection using Bayesian topic models in mobile edge computing. SecurCommunNetw. https://doi.org/10.1155/2020/8860418
10. Cao H, Wachowicz M, Cha S (2017) Developing an edge computing platform for real-time descriptive analytics. In: 2017 IEEE International Conference on Big Data (Big Data), IEEE, pp 4546– 4554
11. Eskandari M, HaiderJanjua Z, Vecchio M, Antonelli F (2020) Passban IDS: an intelligent anomaly-based intrusion detection system for IoT edge devices. IEEE Internet Things J 7(8):6882– 6897

12. Farhin F, Shamim KM, Mahmud M (2021) Secured smart healthcare system: blockchain and bayesian inference based approach. Proceedings of international conference on trends in computational and cognitive engineering. Springer, Berlin, pp 455–465
13. Farhin F, Shamim KM, Mahmud M (2020) Towards secured service provisioning for the Internet of Healthcare Things. In: 2020 IEEE 14th International Conference on Application of Information and Communication Technologies (AICT), IEEE, pp 1–6
14. Furnell S (2004) Enemies within: the problem of insider attacks. Comput Fraud Secur 2004(7):6–11
15. Howard, J. ,Gugger, S. (2020). *Deep Learning for Coders with fastai&PyTorch*.O'Reilly Media.
16. Khan WZ, Ahmed E, Hakak S, Yaqoob I, Ahmed A (2019) Edge computing: a survey. Future GenerComputSyst 97:219–235
17. Li Y, Xia J, Zhang S, Yan J, Ai X, Dai K (2012) An efficient intrusion detection system based on support vector machines and gradually feature removal method. Expert SystAppl 39(1):424– 430
18. Mahesh Yadav YR (2019) Effective analysis of malware detection in cloud computing. ComputSecur 83:14–21
19. Mundie, D. A., Perl, S. J., &Huth, C. L. (2014). Insider threat defined: Discovering the prototypical case. Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications(Jowua), 5(2), 7–23.
20. Mijwel, M. (2018).Artificial Neural Networks Advantages and Disadvantages [Blog post].Reviewed by Springer Nature. Retrieved from https://www.linkedin.com/pulse/artificial-neural-networks-advantages-disadvantages-maad-m-mijwel/
21. RajinderSandhu, Amandeep Singh Sohal&Sandeep K. Sood (2017): Identification of malicious edge devices in fog computing environments, Information Security Journal: A Global Perspective, DOI: 10.1080/19393555.2017.1334843
22. Roman R, Lopez J, Mambo M (2018) Mobile edge computing, fog et al.: a survey and analysis of security threats and challenges. Future GenerComputSyst 78:680–698
23. Sabella D, Vaillant A, Kuure P, Rauschenbach U, Giust F (2016) Mobile-edge computing architecture: the role of MEC in the Internet of Things. IEEE Consum Electron Mag 5(4):84–91
24. Sazzadul HM, Mukit M, Bikas M, Naser A (2012) An implementation of intrusion detection system using genetic algorithm. Int J NetwSecurAppl (IJNSA) 4(2):109–120
25. Shah, J. (2017, Nov. 16). Neural Networks for Beginners: Popular Types and Applications [Blog Post]. Retrieved from https://blog.statsbot.co/neural-networks-for-beginners-d99f2235efca
26. Sha K, Yang TA, Wei W, Davari S (2020) A survey of edge computing-based designs for IoT security. Digit CommunNetw 6(2):195–202
27. Shahraki A, Abbasi M, Haugen O (2020) Boosting algorithms for network intrusion detection: a comparative evaluation of Real AdaBoost, Gentle AdaBoost and Modest AdaBoost. EngAppliArtifIntell 94:103770
28. Shamim KM, Zenia N, Tabassum F, Mamun SA, Arifur RM, Shahidul IM, Mahmud M (2021) 6G Access network for intelligent internet of healthcare things: opportunity, challenges, and research directions. Proceedings of international conference on trends in computational and cognitive engineering. Springer, Berlin, pp 317–328
29. Sharma P, Sengupta J, Suri PK (2019) Survey of intrusion detection techniques and architectures in cloud computing. Int J High Perform ComputNetw 13(2):184–198
30. Singh BN, Khari M (2021) A survey on hybrid intrusion detection techniques. Research in intelligent and computing in engineering. Springer, Berlin, pp 815–825
31. Siriwardhana Y, Porambage P, Liyanage M, Ylianttila M (2021) A survey on mobile augmented reality with 5G mobile edge computing: architectures, applications, and technical aspects. IEEE CommunSurv Tutor 23(2):1160–1192
32. Vimal S, Suresh A, Subbulakshmi P, Pradeepa S, Kaliappan M (2020) Edge computing-based intrusion detection system for smart cities development using IoT in urban areas. Internet of things in smart technologies for sustainable urban development. Springer, Berlin, pp 219–237
33. Yueyue DD, Maharjan S, Qiao G, Zhang Y (2019) Artificial intelligence empowered edge computing and caching for internet of vehicles. IEEE WirelCommun 26(3):12–18