AN ENSEMBLE INTRUSION DETECTION SYSTEM USING GRADIENT BOOSTING ALGORITHM FOR EDGE-BASED COMPUTING

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

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

ABSTRACT

Intrusion detection systems (IDS) are essential for maintaining the security of edge computing environments, where traditional centralized approaches often fail to meet the demands for real-time processing and low latency. This paper proposes an Ensemble Intrusion Detection System (EIDS) using the Gradient Boosting Algorithm to enhance detection accuracy and performance in edge-based computing environments. The system's effectiveness was evaluated against AdaBoost, another ensemble learning algorithm, using key performance metrics such as accuracy, recall, precision, F1-score, and time efficiency. The results indicate that Gradient Boosting achieved superior performance, with an accuracy of 97.68%, recall of 97.68%, precision of 97.69%, and an F1-score of 97.68%, while training in just 4.5 seconds. In contrast, AdaBoost achieved an accuracy of 95.85%, with a significantly longer training time of 58.0 seconds. Gradient Boosting's faster training time and higher accuracy make it a more suitable choice for real-time intrusion detection in edge computing environments. These findings highlight the potential of ensemble learning techniques, particularly Gradient Boosting, in enhancing the security and efficiency of edge-based systems.

Keyword: - *Gradient Boosting; AdaBoost; IDS and Ensemble*

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 [1]. 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 [2]. 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 [3]. To address these challenges, Intrusion Detection Systems (IDS) are increasingly being deployed at the edge, offering a first line of defence 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 centres. This technology can enhance the efficiency and capabilities of mobile networks [3].

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 [4]. 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. [4]

As information technology takes over the globe, security has become an inextricable problem [5]. 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 [6]. 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 [7]. 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

A. **Intrusion Detection Systems (IDS)**

Intrusion Detection Systems (IDS)[8], 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 [9]

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) [10], random forest (RF) [11], and support vector machine (SVM) [12]. With the development of deep learning, convolutional neural network (CNN) [13] recurrent neural network (RNN) [14], and long short-term memory (LSTM) [15] 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 [16].

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.

Traditional Intrusion Detection Systems (IDS), which rely on centralized data processing, face several challenges in this distributed architecture. Centralized systems result in high latency and bandwidth consumption as data must be

sent from edge devices to a central server for analysis. Moreover, edge devices typically have limited computational resources, making it difficult to deploy conventional IDS that require significant processing power and storage. This severely impacts the effectiveness of traditional IDS in edge environments, where resources are constrained, and security breaches can escalate rapidly.

In addition to resource limitations, traditional IDS are also inadequate for providing real-time detection and response to security threats in edge computing. The delay caused by centralizing detection tasks often hinders the ability to respond quickly to attacks, leaving edge systems exposed. As threats become more sophisticated and frequent, it becomes critical to develop IDS solutions that are lightweight, decentralized, and capable of operating within the resource constraints of edge devices. These systems must also enable real-time threat detection and response to ensure the security of edge environments without compromising performance or efficiency.

The primary objective of this research is to develop and Implement gradient boosting algorithms to enhance detection accuracy. Evaluating the system's performance in terms of detection rate, false positive rate, and computational efficiency. It focuses on three different machine learning algorithms for intrusion detection including Decision tree, Naive bayes and Gradient Boosting. Decision tree is used in designing SIDS while Naïve bayes is used in designing AIDS and Gradient boosting is used to integrate the two methods

B. Signature detection module (SDM)

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.

C. 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 [17] 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.

D. AdaBoost algorithm:

Reference [18] 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.

E. Gradient boosting

The 'gradient' in gradient boosting refers to the optimization process used to iteratively improve 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.

F. 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 [19] 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 [20].

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 [1]. 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 [2]. 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 [3]. To address these challenges, Intrusion Detection Systems (IDS) are increasingly being deployed at the edge, offering a first line of defence 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 centres. This technology can enhance the efficiency and capabilities of mobile networks [3].

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 [4]. 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. [4]

As information technology takes over the globe, security has become an inextricable problem [5]. 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 [6]. 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 [7]. 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

G. Intrusion Detection Systems (IDS)

Intrusion Detection Systems (IDS)[8], 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 [9]

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) [10], random forest (RF) [11], and support vector machine (SVM) [12]. With the development of deep learning, convolutional neural network (CNN) [13] recurrent neural network (RNN) [14], and long short-term memory (LSTM) [15] 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 [16].

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.

Traditional Intrusion Detection Systems (IDS), which rely on centralized data processing, face several challenges in this distributed architecture. Centralized systems result in high latency and bandwidth consumption as data must be sent from edge devices to a central server for analysis. Moreover, edge devices typically have limited computational resources, making it difficult to deploy conventional IDS that require significant processing power and storage. This severely impacts the effectiveness of traditional IDS in edge environments, where resources are constrained, and security breaches can escalate rapidly.

In addition to resource limitations, traditional IDS are also inadequate for providing real-time detection and response to security threats in edge computing. The delay caused by centralizing detection tasks often hinders the ability to respond quickly to attacks, leaving edge systems exposed. As threats become more sophisticated and frequent, it becomes critical to develop IDS solutions that are lightweight, decentralized, and capable of operating within the resource constraints of edge devices. These systems must also enable real-time threat detection and response to ensure the security of edge environments without compromising performance or efficiency.

The primary objective of this research is to develop and Implement gradient boosting algorithms to enhance detection accuracy. Evaluating the system's performance in terms of detection rate, false positive rate, and computational efficiency. It focuses on three different machine learning algorithms for intrusion detection including Decision tree, Naive bayes and Gradient Boosting. Decision tree is used in designing SIDS while Naïve bayes is used in designing AIDS and Gradient boosting is used to integrate the two methods

H. Signature detection module (SDM)

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.

I. 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 [17] 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.

J. AdaBoost algorithm:

Reference [18] 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.

K. Gradient boosting

The 'gradient' in gradient boosting refers to the optimization process used to iteratively improve 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.

L. 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 [19] 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 [20].

2. RELATED WORKS

A literature review of new discoveries about IDS solutions in EC networks has been conducted [21]. To safeguard the edge network from insider attack, a firewall architecture has been created. [22]. 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 [14]. 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 [23]. A Distributed Intrusion Detection Systems (DIDS) has been proposed in [24]. 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 [25]. 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 [26], 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 [27].

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 [28] 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 [29], 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 [30]. 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 [31] for the IoT applications.

3. ANALYSIS OF RESULT

Gradient Boosting outperformed AdaBoost in terms of all the key metrics: accuracy, recall, precision, and F1-score. It achieved an accuracy of 97.68%, while AdaBoost had a lower accuracy of 95.85%. The recall, precision, and F1-score of Gradient Boosting were all consistently higher than those of AdaBoost, indicating that Gradient Boosting not only correctly identified more instances but also had fewer false positives and false negatives. Training Time: Gradient Boosting demonstrated significantly faster training time (4.5 seconds) compared to AdaBoost, which took 58.0 seconds to train. Prediction Time: Although both models exhibited negligible prediction times, Gradient Boosting’s overall time was still far more efficient due to its faster training phase. Total Time: Gradient Boosting had a total runtime of 4.6 seconds, making it 12.6 times faster than AdaBoost, which took 58.0 seconds in total

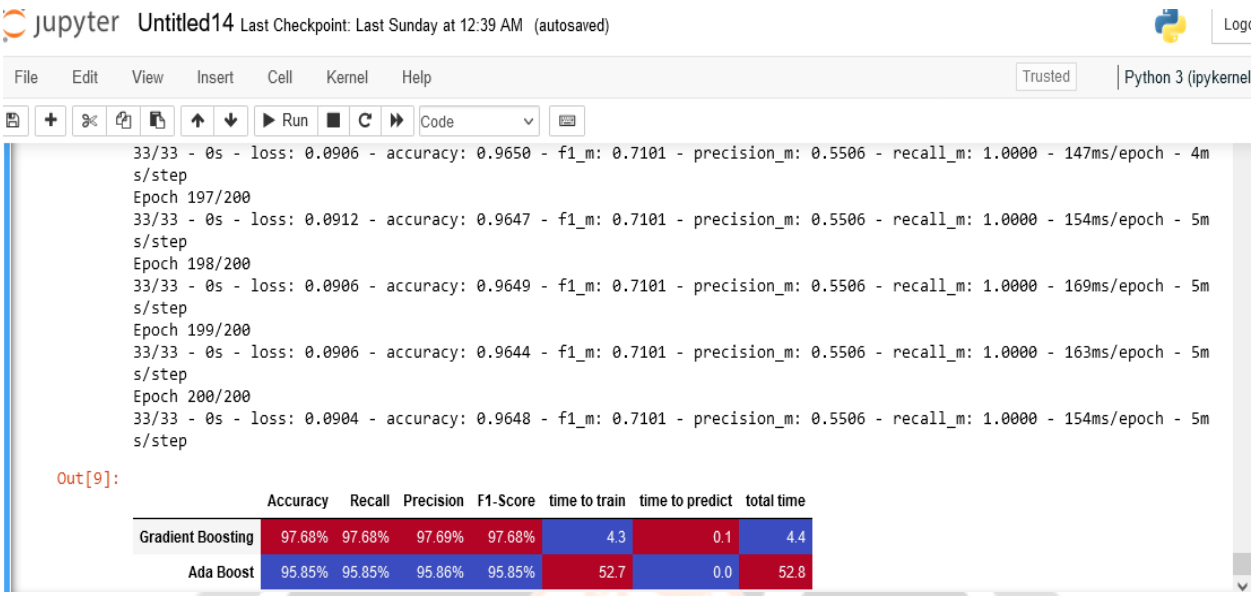

Fig. 1. Implementation window

G. Implementation setup

The proposed EIDS is implemented on Intel(R) Core(TM) i7-3632QM CPU @ 2.20GHz 2.20 GHz and NvidiaGeforce GT 650M-2GB GPU processor with 4.00 GB RAM. In the data preprocessing phase, the original UNSW-NB15 dataset underwent size reduction by eliminating redundant data. The k-means clustering algorithm was employed, with the value of k systematically varied to explore different cluster configurations.

H. Feature selection and ranking

The proposed EIDS that includes SDM, ADM, and EDM. It uses UNSW-NB 15 dataset for the experiment and testing purposes. The SDM uses a C4.5 classifier, ADM uses Naive-based classifier, and HDM uses the Gradient Boosting Algorithm for the classification and attack detection purpose. The UNSW-NB15 dataset consists of total 47 features. The high information gain value of a feature contains high significance compared to other features. The Information Gain (IG (S, F)) value of a feature (F) in a given dataset (size=S) is calculated using Eq. 1.

$$I_G(S,F) = E_N(S) - (E_N)_F(S) \qquad (1)$$

Where Entropy (EN (S)) is the defined degree of non-homogeneity in the given dataset (size=S). It is computed using Eq. 2, where pk is the proportion of instances for class k. (EN) F (S) is the extra needed information for classification when feature F is selected. It can be defined by using Eq. 2, where V(F) is the all-distinct values of feature F, and Sv is the number of tuples for which feature F has value v.

$$E_N(S) = \sum_k - p_k \log 2\, p_k \qquad (2)$$

$$(E_N)_F(S) = \sum_{v=V(F)} \frac{|S_v|}{|S|} * E_N(S_v) \qquad (3)$$

The information gain value of all 47 features of the UNSWNB15 dataset is presented in Table 6. Table 6 illustrated that IG (S, F) value of 25 features is 0. Thus, these features are not important because they do not provide any information or contribution to the IDS. The remaining 22 features IG (S, F) value are also tabulated. It is also observed

that a smaller number of features shows significantly less accuracy compared to a greater number of features. The features are removed whose IG (S, F) value is near to 0 and consider the features whose IG (S, F) value are significantly high. Based on this phenomenon, the bold black (blue shaded) feature (15 features) is considered for evaluation.

4. CONCLUSION

This research focused on evaluating the performance of an Ensemble Intrusion Detection System (EIDS) in edge-based computing environments using two popular ensemble learning algorithms: Gradient Boosting and AdaBoost. The results clearly demonstrate that Gradient Boosting is the superior choice for an Intrusion Detection System in edge-based environments. It provides better accuracy and precision in detecting intrusions while being significantly more time-efficient, especially in terms of training. This makes it an ideal candidate for edge computing applications where both performance and speed are crucial for real-time threat detection. Based on the research findings, Adopt Gradient Boosting for Edge-based IDS, given its superior accuracy and time efficiency, its ability to provide fast, accurate threat detection makes it highly suitable for real-time applications. It is also more efficient in terms of both accuracy and time, it is important to ensure that edge devices have sufficient computational resources to handle the algorithm's demands without performance degradation. Implementing distributed processing across edge nodes could help manage computational load. Consider AdaBoost for Lightweight Applications, the latter might still be useful for lightweight applications where training time is less of a concern, or where edge devices have limited computational power and storage.

5. 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]. Khan WZ, Ahmed E, Hakak S, Yaqoob I, Ahmed A (2019) Edge computing: a survey. Future GenerComputSyst 97:219–235

[3] Singh BN, Khari M (2021) A survey on hybrid intrusion detection techniques. Research in intelligent and computing in engineering. Springer, Berlin, pp 815–825

[4] Ghaida Muttashar Abdulsahib, Osamah Ibrahim Khalaf, An improved Algorithm to Fire Detection in Forest by Using Wireless Sensor Networks, International Journal of Civil Engineering and Technology, 9(10), 2018, pp. 369–377

[5] Khalaf OI, Abdulsahib GM. Optimized dynamic storage of data (ODSD) in IoT based on blockchain for wireless sensor networks. *Peer-to-Peer Netw Appl*. (2021) 14:2858–73. doi: 10.1007/s12083-021-01115-4

[6] Alkhafaji, A. A., Sjarif, N. N. A., Shahidan, M. A., Azmi, N. F. M., Sarkan, H. M., Chuprat, S., & Al-Khanak, E. N. (2021). Payload Capacity Scheme for Quran Text Watermarking Based on Vowels with Kashida. CMC Computer, Materials and Continua, 67(3).

[7] Al-Khanak, E. N., Lee, S. P., Khan, S. U. R., Behboodian, N., Khala, O. I., Verbraeck, A., & van Lint, J. W. C. (2021). A Heuristics-Based Cost Model for Scientific Workflow Scheduling in Cloud. CMC Computer. Materials and Continua, 67(3), 3265–3282. doi:10.32604/cmc.2021.015409

[8] Dawood, A., Salman, O. I. K., & Muttashar, G. (2019). An adaptive intelligent alarm system for wireless sensor network. Indonesian Journal of Electrical Engineering and Computer Science, 15(1), 142–147. doi:10.11591/ijeecs.v15.i1.pp142-147

[9] Tavera, C. A., Ortiz, J. H., Khalaf, O. I., Saavedra, D. F., & Aldhyani, T. H. H. (2021). Wearable Wireless Body Area Networks for Medical Applications. Computational and Mathematical Methods in Medicine, 2021, 1–9. doi:10.1155/2021/5574376 PMID:33986824

[10] Safavian, S. R., & Landgrebe, D. A. (1991). A survey of decision tree classifier methodology. IEEE Transactions on Systems, Man, and Cybernetics, 21(3), 660-674. https://doi.org/10.1109/21.97458

[11] Zheng, X., Ping, F., Pu, Y., Montenegro-Marin, C. E., & Khalaf, O. I. (2021). Recognize and regulate the importance of work-place emotions based on organizational adaptive emotion control. Aggression and Violent Behavior, 101557. doi:10.1016/j.avb.2021.101557

[12] Chen W-H, Hsu S-H, Shen H-P (2005) Application of SVM and ANN for intrusion detection. Comput Oper Res 32(10):2617– 2634

[13] R. Vinayakumar, K. Soman, P. Poornachandran, 2017. Applying convolutional neural network for network intrusion detection International Conference on Advances in Computing, Communications and Informatics (ICACCI) (2017), pp. 1222-1228

[14] Yin C, Zhu Y, Fei J, He X (2017) A deep learning approach for intrusion detection using recurrent neural networks. IEEE Access 5:21954–21961
[15] S.S. Roy, A. Mallik, R. Gulati, M.S. Obaidat, P.V. Krishna, 2017 A deep learning based artificial neural network approach for intrusion detection International Conference on Mathematics and Computing (2017), pp. 44-53
[16] JH.H. Jazi, H. Gonzalez, N. Stakhanova, A.A. Ghorbani, 2017 Detecting http-based application layer dos attacks on web servers in the presence of sampling
[17] Akoglu, Leman, Hanghang Tong, and Danai Koutra, 2015. "Graph based anomaly detection and description: a survey." *Data mining and knowledge discovery* 29 (2015): 626-688.
[18] Schapire RE (2003) The boosting approach to machine learning: an overview Nonlinear estimation and classification. Springer, Berlin, pp 149–171
[19] Salihu Sabiu Musa , Sania Qureshi , Shi Zhao , Abdullahi Yusuf , Umar Tasiu Mustapha , Daihai He, (2021). Mathematical modeling of COVID-19 epidemic with effect of awareness programs. Revised 26 January 2021, Accepted 29 January 2021, Available online 18 February 2021, Version of Record 24 February 2021.
[20] Manju Khari, and A. Karar, 2013.Analysis on Intrusion Detection by Machine Learning Techniques: A Review, Published 2013.Computer Science
[21] Chen J, Ran X (2019) Deep learning with edge computing: a review. Proc IEEE 107(8):1655–1674
[22] Markham T, Payne C (2001) Security at the network edge: a distributed firewall architecture. In: Proceedings DARPA Information Survivability Conference and Exposition II. DISCEX'01, volume 1, IEEE, pp 279–286
[23] Muna AL-H, Moustafa N (2018) Identification of malicious activities in industrial Internet of Things based on deep learning models. J Inform Secur Appl 41:1–11
[24] Meng W, Wang Y, Li W, Liu Z, Li J, Probst CW (2018) Enhancing intelligent alarm reduction for distributed intrusion detection systems via edge computing. Australasian conference on information security and privacy. Springer, Berlin, pp 759–767
[25] Almogren AS (2020) Intrusion detection in Edge-of-Things computing. J Parallel Distrib Comput 137:259–265
[26] Liang C, Shanmugam B, Azam S, Karim A, Islam A, Zamani M, Kavianpour S, Idris NB (2020) Intrusion detection system for the Internet of Things based on blockchain and multi-agent systems. Electronics 9(7):1120
[27] Mudgerikar A, Sharma P, Bertino E (2020) Edge-based intrusion detection for IoT devices. ACM Trans Manag Inform Syst (TMIS) 11(4):1–21
[28] 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
[29] Cao X, Fu Y, Chen B (2020) Packet-based intrusion detection using Bayesian topic models in mobile edge computing. Secur Commun Netw. https://doi.org/10.1155/2020/8860418
[30] Li Z, Chen J, Zhang J, Cheng X, Chen B (2020) Detecting advanced persistent threat in edge computing via federated learning. International conference on security and privacy in digital economy. Springer, Berlin, pp 518–532
[31] Kozik R, Chora´s M, Ficco M, Palmieri F (2018) A scalable distributed machine learning approach for attack detection in edge computing environments. J Parallel Distrib Comput 119:18–26.