ISSN 2723-6471

IoT based Intrusion Detection for Edge Devices using Augmented System

R. Nagarajan[1,*], M. Batumalay[2], Zhengrui Xu[3]

[1]*PG and Research Department of Computer Science, Sri Ramakrishna College of Arts and Science, Coimbatore, Tamil Nadu 641006*

[2]*Faculty of Data Science and Information Technology, INTI International University, 71800 Nilai, Negeri Sembilan, Malaysia*

[3]*Faculty of Liberal Arts, Shinawatra University (SIU) Pathum Thani 12160 Thailand*

(Received: June 19, 2024; Revised: July 22, 2024; Accepted: August 26, 2024; Available online: September 17, 2024)

Abstract

The Edge Computing (EC) paradigm is gaining popularity among users due to its inherent characteristics and expeditious delivery approach. Users may get information from the network's edge thanks to this feature of network architecture. The security of this edge network design, however, is a major issue. Through the Internet and in a shared setting, users can access all EC services. Intrusion detection is a method of network security that searches for threats. It is ineffective to monitor real-time network data, and current detection techniques are unable to identify known dangers. To address this problem, a technique known as augmentation oversampling is proposed, which incorporates the minority classes in the dataset. Our Sort-Augment-Combine (SAC) approach divides the dataset into subsets of the class labels, from which synthetic data is generated for each group. The developed synthetic data was then used to oversample the minority classes. After the oversampling process was complete, the distinct classes were combined to provide improved training data for model fitting. When compared to the original dataset, the models trained using the enhanced datasets perform better in terms of accuracy, recall (sensitivity), and true positives (specificity). SAC fared best in a UNSW-NB15 dataset when compared to the Synthetic Minority Oversampling Technique (SMOTE) and Generative Adversarial Network-Data Augmentation (GAN-DA). Additionally, SAC points to improvements in general sensitivity, specificity, and accuracy. SMOTE, datasets with ROSE enhancements, and Random Over-Sampling Examples for process innovation.

Keywords: Edge Computing, Sort-Augment-Combine, Intrusion Detection, Generative Adversarial Network- Data Augmentation, Synthetic Minority Oversampling Technique, Random Over-Sampling Examples, Process Innovation

1. Introduction

The Internet of Things is the network where real-world objects or things will link with the current Internet infrastructure to connect to computers, stuff, household appliances, technology, automobiles, and other stuff that are all considered to be things. The Internet of Things (IoT) is a global system of networked computers, mechanical and digital gadgets, household items, animals, and people with unique identities (UIDs) and the capacity to communicate data across a network without the need for direct human or computer interaction [1]. The idea of the Internet of Things has evolved as a result of the convergence of several technologies, including real-time analytics, system reading, widespread sensors, and embedded structures [2]. Traditional fields, such as embedded systems, wireless sensor networks, control systems, automation, and others, are what enable the Internet of Things [3].

As a consequence of IoT, attack probability and attack surface area will simultaneously increase [4]. Since most IoT applications require security, IoT intrusion detection systems must be created to protect communications made possible by such technologies [5]. Recently, improvements in IoT IDS (Intrusion Detection System) have been due to developments in Artificial Intelligence (AI) [6], particularly deep learning and machine learning techniques. Additionally, a number of machine learning (ML) [7] methods exist, such as the use of support vector machine (SVM) models in [8], decision tree algorithms (DT) in [9], k-nearest neighbor (kNN) [10], k-means [11], and many more artificial intelligence techniques. Numerous deep neural network technologies exist [12]. Fog, clouds, and other IoT-based technologies have been used as a platform for IDS solutions. The deep recurrent neural network (RNN) model

***Corresponding author: R Nagarajan (rnagarajan.snr@gmail.com)**
DOI: https://doi.org/10.47738/jads.v5i3.358

is particularly noteworthy [13], A lot of research has been done on using different types of neural networks, such as the convolutional neural network (CNN) model [14], the restricted Boltzmann machines (RBMs) model [15], and the multi-layered perceptron neural network [16].

IDS focuses primarily on edge devices, utilized by numerous smart applications at once by various users on the same edge device. Due to the possibility of providing misleading output data with wrong input data and producing false results, if an edge device is compromised, this worsens its security vulnerabilities. The edge device's data may also be exploited for unethical actions. Firewalls safeguard such networks. However, it is unable to block the malicious inside packets produced by insider attacks. Such firewall-based solutions don't work because of the edge nodes' diversity and complexity. Additionally, it is expensive and difficult to manage a large number of firewalls on most edge computing devices in order to implement security solutions.

To overcome this problem, the paper proposed a new algorithm for edge computing, the Augmented Intrusion Detection System (AIDS), which can detect unknown assaults in real time with a low False Alarm Rate (FAR). The recommended detection framework has many detection modules and classifiers. In order to enhance the model's feature representation capacity to analyze network traffic efficiently, an intrusion detection data enhancement dataset is needed to produce varied samples from several perturbations. To evaluate, the proposed approach with edge device computing, datasets such asUNSW-NB15 dataset are used. When compared with other existing methods, the results demonstrate and improve accuracy, reducing the FAR.

2. Literature Review

Almogren [17] suggested a novel method based on an enhanced intrusion detection strategy called the Deep Belief Network (DBN).In order to compare them with current detection methods, several detection models were developed utilising various DBN architectures. The recommended strategy surpasses state-of-the-art procedures by a large margin, according to the test findings.

Singh et al [18] proposed a Mobile Edge Computing (MEC)-based Hybrid Intrusion Detection Framework (EHIDF) that used ML to detect intrusive traffic. The proposed structure has three intrusion detection modules with three classifiers. The hybrid detection module (HDM) uses the Meta-Adaboost M1 algorithm, whereas the signature detection module (SDM) uses the C4.5 classifiers. The suggested solutions were created to address the current detection issues by quickly and accurately identifying previously unidentified assaults. The experimental findings might enhance the suggested techniques' accuracy and FAR. Additionally, the proposed framework's security strength is examined using a game-theoretical approach.

Eskandari et al [19] presented the intelligent intrusion detection system (IDS) known as Passban, which is capable of protecting the Internet of Things devices that are directly linked to it. The uniqueness of the suggested solution is in its ability to be installed immediately on relatively affordable IoT gateways, following which it would fully deploy edge computing to detect cyberthreats as close to data sources as possible. It illustrates that passban can identify several malicious traffic types, such as Port Scanning, HTTP, Secure Shell (SSH), Brute Force, and Synchronize (SYN) flood assaults, with acceptable accuracy and minimal false positive rates.

Celesti [20] supported the creation of a cost-effective Non-Intrusive Appliance Load Monitoring (NIALM) infrastructure by using IoT characteristics and AI technologies. NIALM, IoT-based Cognitive Smart Metres (CSMs) and an Edge-based Accumulator that collects CSM data and extracts features to train an on-board ML model minimise costs and latency. The results demonstrate strategy and ML model effectiveness.

Sandhu et al [21] The proposed architecture employs three technologies to identify malicious edge devices in a fog computing environment: IDS, Virtual Honeypot Device (VHD), and the Markov model. Using a two-stage Markov model, we classify edge devices into four categories. The VHD's log repository will store logs from recognized malicious devices. A simulated environment is used to evaluate the suggested model, which shows how well the system performs. In addition to lowering the percentage of erroneous IDS alarms, the suggested model effectively detected the malicious device.

ISSN 2723-6471

Sudqi Khater et al. [22] presented a lightweight IDS Multi-Layer Perceptron (MLP) model based on vector space representation. The Australian Defence Force Academy Linux Dataset (ADFA-LD) and Windows Dataset (ADFA-WD) are used to test the intrusion detection system. These datasets include novel system calls, exploits, and assaults on various programs. Simulation results demonstrate one hidden layer and a few nodes, attaining accuracy, recall, and F1-measure. Raspberry Pi is evaluating performance.

Cao et al. [23] propose a framework for edge computing where physical devices distributed on a bus would serve as mobile edge nodes. Real-time transit data streams are analysed using descriptive analytics to find significant trends. Descriptive analytics at a mobile node and transit information are provided by the recommended platform management, which is evaluated for advantages and disadvantages using an application experiment.

Mudgerikar et al. [24] presented an IDS for IoT devices that is divided at the device-edge level. IDS analyses IoT devices based on their "behavior" by leveraging system-level data such as running process settings and their system calls in an autonomous, effective, and scalable way, and then it looks for aberrant behavior suggestive of intrusions. Effective attack detection is made possible with the help of an innovative device-edge split architecture and the modular design of IDS.8, powerful file-less attack methods that have recently been seen against IoT devices are included in the E-Spion collection of 3,973 typical IoT malware samples. The evaluation's findings enhance computational and detection effectiveness.

Aljumah [25] developed a model for the Internet of Things Intrusion Detection System (IoT-IDS) that combines CNN and general convolution and is called the Temporal Convolution Neural Network (TCNN). Synthetic minority oversampling with nominal continuity is used to collect TCNN to accommodate imbalanced datasets. Attribute transformation and reduction are two efficient feature engineering strategies that are combined. On the Bot-IoT data repository, the given model is compared to the Logistic Regression (LR) and Random Forest (RF) machine learning algorithms, as well as the CNN deep learning and Long Short-Term Memory (LSTM) algorithms. According to the findings of the experiments, the performance of TCNN strikes an excellent compromise between its efficiency and its ability to recognise many classes of traffic.

Cao et al. [26] The Bayesian topic model Latent Dirichlet Allocation (LDA) for mobile edge computing was recently introduced as a novel approach. This algorithm is utilised in network IDS. Multiple features are extracted from the packet headers using the tcpdump technique. Using the features, tcpdump transferred the packets into papers. To understand the behavior patterns of typical traffic, a trained topic model is only used in the absence of attacks. The level to which the test traffic matches the regular traffic is determined by comparing it to the acquired behavioural patterns. A minimal threshold probability is set during the training phase. The intrusion is referred to as traffic when a host's test traffic has a probability that is lower than the host's threshold during the test phase. DARPA 1999 dataset is used for intrusion detection system is validated. According to the experimental results, Mobile Edge Computing (MEC) security may be preserved using the suggested strategy

3. Proposed Methodology

The key part of our technique to address the problem of class imbalance in datasets is data augmentation through deliberate oversampling of the minority class. Security measures must be performed to preserve data integrity and confidentiality while utilising these technologies to communicate with or store data near the user. For instance, these hacks by spammers or other attackers may add unneeded data and strain the system. These are listed in the following order: Data security problems at the edge layer and with edge computing 2. SAC augmentation, also known as sort-adjust-combine.

3.1. Advances In Edge Layer Computing And Data Security

Supporting standard interfaces, this framework lets services share contextual data. Topology Master (TM), a task designer, and a repository for Docker images are all parts of service management. While Task Designer provides a web interface for tracking IoT services, the Docker repository oversees managing all Docker images.Units. Service orchestration is the responsibility of TM, and edge node service requests and topologies. Worker nodes or edge nodes in close proximity to IoT devices perform tasks assigned by TM for data processing. TM and staff communicate via

 ISSN 2723-6471

the RabbitMQ protocol. IoT brokers, federated brokers, and IoT discovery are all part of context management. These components manage contextual data, such as worker availability, task and produced data stream management, topology, and task management, in addition to facilitating data flow across tasks [27]. Figure 1 displays the data flow and procedure of the framework.

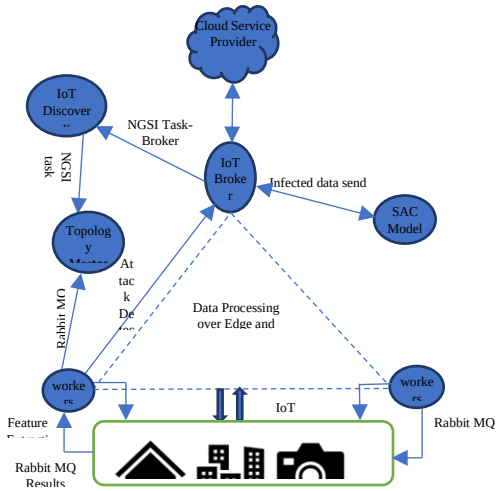

Figure 1. Framework for edge computing and data flow

3.2. Data Security Issues

Edge computing handles data from several sensors, devices, servers, and data centres. Edge nodes are in charge of gathering data from numerous ubiquitous devices, sending it to servers or other edge nodes, and analysing it.The task deployment is carried out by a mobile agent that is dynamically deployed on the edge nodes during the data analysis. These operations can occasionally be completed in parallel, asynchronously, and without the support of the other nodes [28]. Data collection, transmission, and job assignment are what make edge data analytics, also known as real-time interaction and application monitoring, possible. Decentralised edge data analytics are vulnerable to security risks due to edge nodes' heterogeneity, mobility, and geographic spread. The vulnerabilities may result in attacks that affect data integrity, confidentiality, authentication, and access control. Figure 2 depicts the many dangers that result in these problems with data security.

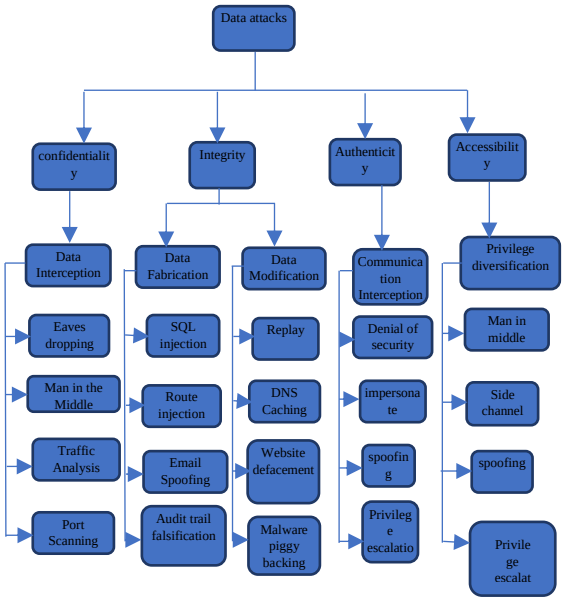

Figure 2. Data Attack Types in Edge Computing

Data Confidentiality: Users and data owners need to have safe access to confidential information at the edge layer. Before transferring data to other edge nodes or cloud data centres, edge nodes in the core network architecture receive

and process data [29]. Through the intercepting of data packets during a chat, attackers can gain access to sensitive information such as credit card numbers, usernames, and passwords. An attacker may wiretap the system and decode data packets if one or more edge nodes are compromised [30]. Therefore, it is necessary to handle and store data inside security frameworks without jeopardising edge layer confidentiality. Typical approaches to guaranteeing data secrecy include cryptography techniques, secure data capture, and secure Internet of Things devices or edge nodes. Using dynamic keys, data input in cryptographic techniques is encrypted, authenticated, and divided. In a secure data collection technique, retransmission of communications options enhances the data. IoT edge nodes also use blockchain technology and AI to continuously validate the data.

Data Integrity: A process of verifying that the data is genuine, complete, and accurate at every stage of the data analysis process, beginning with the data source. Outsourcing to edge nodes gives data owners less control and exposes the data to security threats. Attacks with the purpose of removing, modifying, or updating data can result in integrity issues [31]. Inadequate data integrity and accuracy solutions could impair edge computing's efficacy and performance. There are two possible ways to achieve data integrity: batch auditing and dynamic auditing, which uses homomorphic tags on the externalised data. There aren't many privacy-preserving processes because the main purpose of system tracking is to boost security.

Data Authentication: Enabling users to access cloud servers or edge nodes requires confirming identity numbers. Checks must be made on edge data centres and nodes. In edge computing, authentication can greatly enhance data quality [32]. Typically, passwords, smart cards, and biometric-based authentication are employed in cloud and edge computing applications.

Accessibility: Using this technique, the rights and privileges that users and edge nodes possess within the system are determined. Sensitive data access is limited to specific network users. Specification of local rules is necessary to define access criteria for users and resources within the infrastructure. Decisions about accessibility in edge computing may be influenced by three factors: edge nodes for specialised resource access, virtual machines, and computation and storage services [33]. Access control techniques used in edge computing include necessary, optional, role-based, and attribute-based approaches.

3.3. Sac Data Augmentation of Intrusions In Edge Devices

Based on the SAC [34] data augmentation methodology, the strategy can be used to both binary and multiclass datasets. The explanation of the three SAC steps is provided below:

3.3.1. Sort

Pre-processing creates an instant class subset from the original dataset. Attack and benign subsets of a binary class dataset will be created. A data frame, S, with classes is given: $A, B, C, \ldots$using a power set, we can represent it, $P(S) = \{A, B, C, \ldots\}$, where $A \subseteq S, B \subseteq S, and\ C \subseteq S$.The formula suggests that S's instant classes, A, B, and C, are A, B, and C respectively. S may also be expressed as a set as shown in “equations (1) - (3)”.

$$A = \{a_1, a_2, a_{3,\ldots\ldots}\} \quad (1)$$

$$B = \{b_1, b_2, b_3, \ldots\ldots\} \quad (2)$$

$$c = \{c_1, c_2, c_3, \ldots\ldots\} \quad (3)$$

Where $a1, \ldots\ldots, b1, \ldots\ldots, c1 \ldots\ldots$ are elements of the subsets A, B, and C.

3.3.2. Augment

After dividing and sorting the data frame into subsets of the proximate classes, R's synthon package's Sync function generator was used to synthesize the data value from the original dataset's latent space to boost the minority classes. Each variable in a dataset is synthesized sequentially by the generator using regression modelling. In order to estimate its parameters, it uses conditional distributions to fit data to the anticipated distribution and generate synthetic values.

Consider a dataset of variables as an example$(Z_1, Z_2, \ldots, Z_n)$. $Z_1$is the initial variable that has to be synthesized in this situation; however, since there aren't any predictors for it earlier, its synthetic values are instead produced via

 ISSN 2723-6471

replacement from its original values. Based on preceding variables' conditional distributions, the distributions of the following variables are subsequently calculated and synthesized [20]. With the help of preset parameters and the function generator, we were able to produce high-quality synthesized data in our work.

Each class subset, for instance, was provided to the function with m = 1 and *k* taking various values based on the amount of the incoming synthetic data. The variables the subset inherits from the universal set are practically preserved during formation because other subsets share them (data co-location).Consequently, this supports preserving the distribution underlying the initial data variables. For instance, new values for the synthetic data are created to produce "equations (4-6)" based on this synthesis.

$$A = \{\overline{a_1}, \overline{a_2}, \overline{a_3}, \ldots..\} \quad (4)$$

$$B = \{\overline{b_1}, \overline{b_2}, \overline{b_3}, \ldots\ldots\} \quad (5)$$

$$C = \{\overline{c_1}, \overline{c_2}, \overline{c_3}, \ldots..\} \quad (6)$$

Minority courses renew themselves. The enhanced subsets of A, B, and C are formed by combining "equations (1), (2), and (3)". The new enhanced subsets are:

$$\bar{A} = \{a_1, a_2, a_3, \overline{a_1}, \overline{a_2}, \overline{a_3}, \ldots..\} \quad (7)$$

$$\bar{B} = \{b_1, b_2, b_3, \overline{b_1}, \overline{b_2}, \overline{b_3}, \ldots..\} \quad (8)$$

$$\bar{C} = \{c_1, c_{2,} c_3, \overline{c_1}, \overline{c_2}, \overline{c_3}, \ldots..\} \quad (9)$$

3.3.3. Combine

A new training dataset is generated by combining the newly enhanced subsets. This dataset is formed through the merging of the results obtained from applying equations (7), (8), and (9). The combination of these equations allows for the creation of a more comprehensive training dataset, which integrates the enhanced subsets, leading to improved model training and performance.

$$P(S) = \{\{\} + \{A\} + \{B\} + \{C\}\}. \quad (10)$$

Note: Row-binding is used to combine the enhanced subgroups since the subsets share variables.

The Sort-Augment-Combine (SAC) algorithm follows a structured process for handling datasets. It begins by loading the dataset and splitting it into subsets based on class labels. The algorithm then enters a loop where, for each column of the dataset, it applies a synthetic function generator to produce new data. This process is repeated for each subset of class labels. Once the synthetic data is generated for all columns, the algorithm combines the results. Steps 3 through 8 are repeated for each class label until new class labels are formed. Finally, the data is grouped, and the augmented dataset is returned, completing the process.The work flow process of SAC is provided below. Initially the datasets are loaded. The dataset is split into subsets of class Labels, next dividing and sorting the data frame into subsets of the proximate classes. Then the Sync function generator was used to synthesize the data value from the original dataset's latent space to boost the minority classes. The subsets are combined together and produced new enhanced subsets. The new enhanced subsets are merged and subgroups are formed using row-binding process.

4. Results and Discussion

An early study proposed worker edge node intrusion detection with data collection. To achieve high-degree spatial separation, worker nodes analyse high-dimensional input. This dimensional reduction provides updates to the IoT broker. Updating updates is mostly intended to identify potential assaults and enhance dubious devices and data. Data is forwarded to the upgraded model or edge nodes for processing by the IoT broker based on signs of an impending attack. This device attack notification and data augmentation method doesn't load edge nodes or alert other resources.

4.1. Datasets

The intrusion detection investigations for this article make use of the UNSW-NB15 dataset [35]. First, we use the dataset samples to train an enhanced model that generates synthetic samples. After that, the generated samples, and

 ISSN 2723-6471

samples from the UNSW-NB15 dataset were used to train the network traffic classifier. They evaluate the performance of the proposed model on UNSW-NB15 testing dataset. These experiments are conducted on a PC with an Inter(R) Core (TM) i7-6700 CPU operating at 3.40GHz and 16 GB of RAM using Python and the Tensor Flow Framework version 1.14.0.

4.2. Evaluation Metrics

This article evaluates our proposed model using the following metrics, including the F1 score in the "equations (11–16)" and detection accuracy, precision, recall, and F1. Accuracy is the percentage of correct classifications. Accuracy shows the relationship between right and wrong classifications and the number of accurate classifications. The recall is the ratio that represents the amount of time it took for the number of right categories to catch up to the number of incorrect classifications. And last, the F-score calculates an average that considers both accuracy and recall. Compare it with some methods such as, Generative Adversarial Network- data Augmentation **(**GAN-DA) [36], Synthetic Minority Oversampling Technique (SMOTE) [37], Random Over-Sampling Examples (ROSE). Compare the sensitivity and specificity for intrusion detection using the original dataset and the enhanced data from table 1.
Accuracy: The proportion of samples with accurate classification to all samples. When the dataset is balanced, accuracy is a suitable statistic to use. Accuracy may not be the best statistic since in actual network systems, compared to aberrant samples, normal samples are far more prevalent.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{FN} + \text{TN}} \tag{11}$$

Precision (P): The ratio of true positive samples to predicted positive samples and represents attack detection confidence.

$$\text{P} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{12}$$

Recall (R)detection rate. It is the ratio of positive samples to all positive samples. The detection rate is a key IDS metric.

$$\text{R} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{13}$$

F-measure (F) Harmonic average of accuracy and recall

$$\text{F} = \frac{2 * \text{P} * \text{R}}{\text{P} + \text{R}} \tag{14}$$

The false negative rate (FNR) is the sample ratio of false negatives to positives. FNR is attack detection's missed alarm rate.

$$\text{FNR} = \frac{\text{FN}}{\text{TP} + \text{FN}} \tag{15}$$

The false positive rate (FPR) false positive to expected positive ratio. The false alarm rate (FPR) in assault detection is determined as follows:

$$\text{FDR} = \frac{\text{FP}}{\text{TP} + \text{FP}} \tag{16}$$

Note: true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN).

Attack samples are often seen as positives and normal samples are typically considered as negatives since an IDS's goal is to identify assaults. Accuracy, recall, FNR, and FPR are regularly used measures in attack detection. The results of the enhanced intrusion were compared with the accuracy of results from other established techniques, as shown in figure 3 and the results of the augmented incursion were compared to the recall outcomes of other established techniques, as shown in figure 4.

ISSN 2723-6471

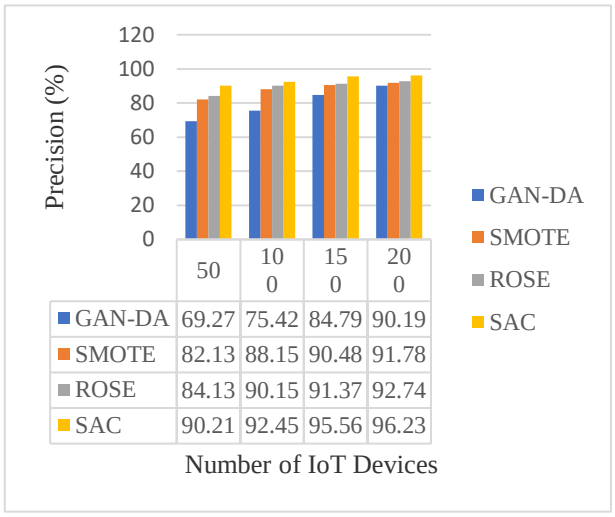

Figure 3. Augmented value in Precision

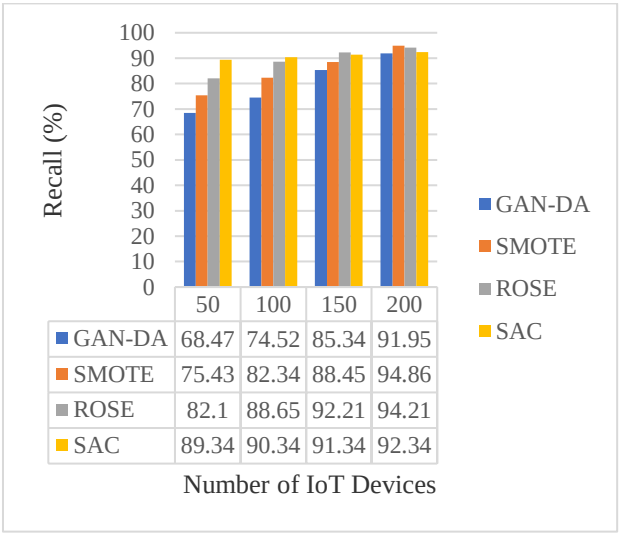

Figure 4. Augmented value in Recall

While other approaches like GAN-DA, SMOTE, and ROSE also yield higher precision results for proposed SAC based diverse methods like 79%, 88%, 89.59%, and so on for UNSW-NB15 datasets, the proposed method yields higher precision results of 93.435%. The suggested algorithm outperforms the current approaches in terms of results. While other approaches like GAN-DA, SMOTE, and ROSE also yield higher accuracy for the proposed SAC-based datasets in different ways, like 80.07%, 85.29%, and 89.29%, and accordingly for UNSW-NB15 datasets, the proposed method yields higher recall results of 90.84%. The suggested algorithm outperforms the current approaches in terms of result.

Figure 5 shows the comparison between the results of the augmented incursion and the F-measure score of other established approaches. While other approaches like GAN-DA, SMOTE, and ROSE also provide higher precision for proposed SAC based diverse ways like 75.39%, 79.89%, 84.45%, and so on for UNSW-NB15 datasets, the proposed method yields higher F-Measure score values of 89.79%. As seen in Figure 6, the suggested algorithm produces results that are higher than those of the current approaches.

The results of the augmented incursion are shown in figure 6. The accuracy results of other established approaches are shown in figure 5. While other approaches like GAN-DA, SMOTE, and ROSE also yield higher precision for proposed SAC based diverse ways like 75.48%, 80.06%, 83.08%, and so on for UNSW-NB15 datasets, the proposed method yields superior accuracy results of 87.41%. The suggested algorithm outperforms the current approaches in terms of results.

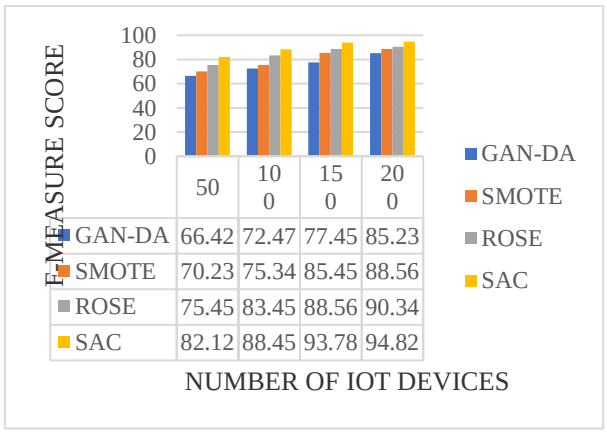

Figure 5. Augmented value in F-Measure Score

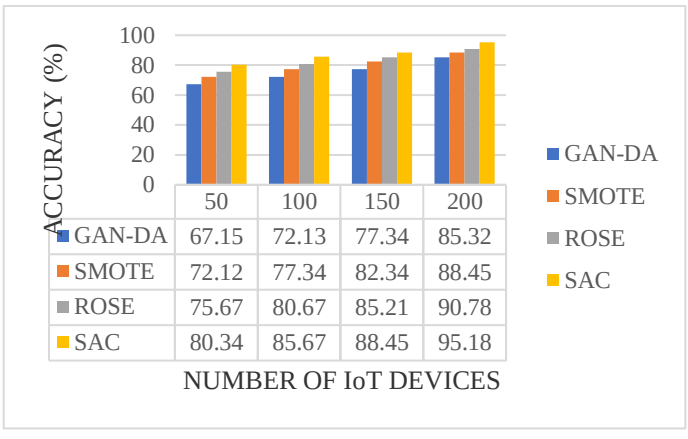

Figure 6. Augmented value in Accuracy

Table 1 presents a comparison between the sensitivity and specificity of the Original Data (OD) and Augmented Data (AD) in an intrusion detection system. Sensitivity measures the system's ability to correctly identify attacks, natural events, and no events, while specificity assesses how well the system recognizes the absence of these events. For the Original Data, the sensitivity scores are 97 for attacks, 72 for natural events, and 85 for no events. This indicates that

ISSN 2723-6471

while the system is highly effective in detecting attacks (97), it struggles more with identifying natural events (72). The Augmented Data, on the other hand, shows improved sensitivity, with scores of 90 for attacks, 96 for natural events, and 99 for no events, indicating better overall detection accuracy, particularly for natural events and no events. In terms of specificity, the Original Data achieves scores of 76 for attacks, 85 for natural events, and 78 for no events. The Augmented Data again outperforms the Original Data, with specificity scores of 85 for attacks, 92 for natural events, and 97 for no events. These results suggest that augmenting the data enhances both the sensitivity and specificity of the system, particularly in recognizing natural events and no events.

Table 1. The Comparison Between the Sensitivity and Specificity of Original Data (OD) and Augmented Data (AD) Intrusion System Datasets

Data	Sensitivity			Specificity		
	Attack	Natural	No Events	Attack	Natural	No Events
Original Data	97	72	85	76	85	78
Augmented Data	90	96	99	85	92	97

Figure 7, which implements the sensitivity, shows that the 99% specificity of the supplemented dataset is slightly higher than the 85% specificity of the original dataset. This is especially important because newer assaults employ evasive techniques to evade intrusion detection. Figure 8 shows that the specificity of the supplemented dataset is 97%, which is marginally higher than the original dataset's 78%. This is especially important because evasive tactics are being used in recent attacks to avoid intrusion detection.

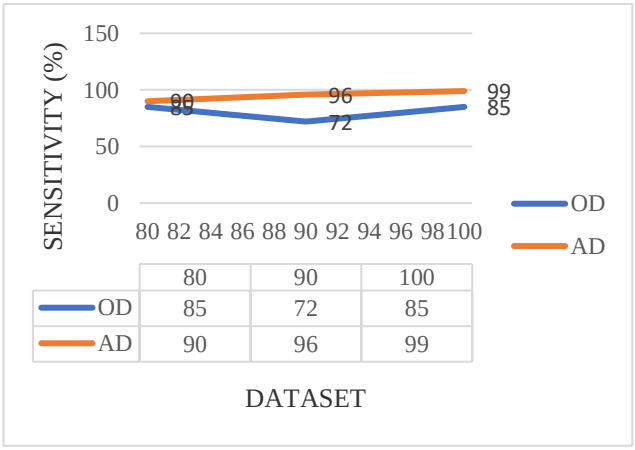

Figure 7. Sensitivity Intrusion Dataset

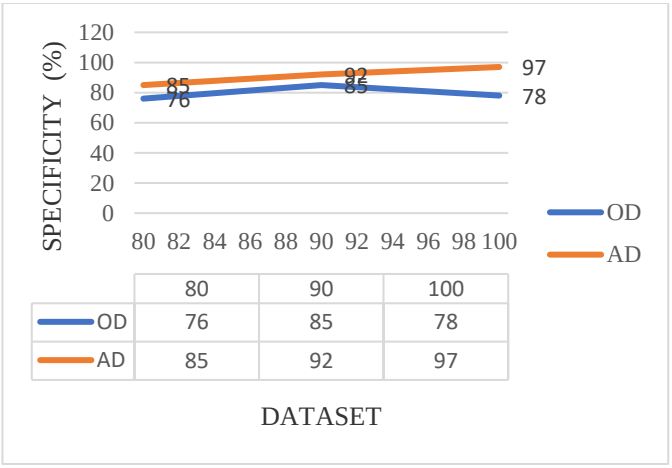

Figure 8. Specificity Intrusion Dataset

5. Conclusion

This work has led to the development of a unique approach for data augmentation in intrusion detection and an edge device-based intrusion detection model. a technique for augmenting data that involves oversampling minority classes in binary and multiclass datasets to enhance generalisation and classification. We refer to our method as Sorting, Augmenting, and Combining. We separated the dataset into discrete groups of edge devices and instant classes before creating synthetic data. Use a compare function to compare the two datasets' structures to ensure that the distribution of the synthetic data values is the same. By employing generated data, they enhanced minority class identification. In large server infrastructures, the suggested technique enhances intrusion detection attacks by 94%. Several metrics, including detection accuracy, precision, recall, accuracy, F-Measure Score, and false alarm rate, have been assessed and contrasted between the suggested system and the current approach. This is vital and significant for intrusion detection, and this technique may be repeated later on for additional validation with benchmark datasets from various sectors.

6. Declarations

6.1. Author Contributions

Conceptualization: R.N., M.B., and Z.X.; Methodology: Z.X.; Software: R.N.; Validation: R.N. and Z.X.; Formal Analysis: R.N. and Z.X.; Investigation: R.N.; Resources: Z.X.; Data Curation: Z.X.; Writing Original Draft Preparation: R.N. and Z.X.; Writing Review and Editing: Z.X. and R.N.; Visualization: R.N.; All authors have read and agreed to the published version of the manuscript.

6.2. Data Availability Statement

The data presented in this study are available on request from the corresponding author.

6.3. Funding

The authors received no financial support for the research, authorship, and/or publication of this article.

6.4. Institutional Review Board Statement

Not applicable.

6.5. Informed Consent Statement

Not applicable.

6.6. Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] A Ahmad Bilal Zia, "A Research Paper on Internet of Things based upon Smart Homes with Security Risk Assessment using OCTAVE Allegro," *International Journal of Engineering Research & Technology (IJERT)*, vol. V9, no. 06, pp. 940-948, Jun. 2020, doi: 10.17577/ijertv9is060692.

[2] M. M. Rashid, S. U. Khan, F. Eusufzai, Md. A. Redwan, S. R. Sabuj, and M. Elsharief, "A Federated Learning-Based Approach for Improving Intrusion Detection in Industrial Internet of Things Networks," *Network*, vol. 3, no. 1, pp. 158–179, Jan. 2023, doi: 10.3390/network3010008.

[3] A. Singh, K. Chatterjee, and S. C. Satapathy, "An edge based hybrid intrusion detection framework for mobile edge computing," *Complex and Intelligent Systems*, vol. 8, no. 1, pp. 3719–3746, Aug. 2021, doi: 10.1007/s40747-021-00498-4.

[4] P. Spadaccino and F. Cuomo, "Intrusion detection systems for IoT: Opportunities and challenges offered by edge computing," *ITU Journal on Future and Evolving Technologies*, vol. 3, no. 2, pp. 408–420, Sep. 2022, doi: 10.52953/wnvi5792.

[5] A. Dahou, M. A. Elaziz, S. A. Chelloug, M. A. Awadallah, M. A. Al-Betar, M. A. A. Al-qaness, A. Forestiero "Intrusion Detection System for IoT Based on Deep Learning and Modified Reptile Search Algorithm," Computational Intelligence and Neuroscience, vol. 2022, no. Jun., pp. 1–15, Jun. 2022, doi: 10.1155/2022/6473507.

[6] H. Alkahtani and T. H. H. Aldhyani, "Intrusion Detection System to Advance Internet of Things Infrastructure-Based Deep Learning Algorithms," Complexity, vol. 2021, no. 1, pp. 1–18, Jul. 2021, doi: 10.1155/2021/5579851.

[7] S. Bagui, X. Wang, and S. Bagui, "Machine Learning Based Intrusion Detection for IoT Botnet," International Journal of Machine Learning and Computing, vol. 11, no. 6, pp. 399–406, Nov. 2021, doi: 10.18178/ijmlc.2021.11.6.1068.

[8] R. Aditya, H. H. Nuha and S. Prabowo, "Intrusion Detection using Support Vector Machine on Internet of Things Dataset," *2022 IEEE International Conference on Communication, Networks and Satellite (COMNETSAT)*, vol. 1, no. 1, pp. 62-66, Nov. 2022, doi: 10.1109/comnetsat56033.2022.9994392.

[9] N. Boonsatit, S. Rajendran, C. P. Lim, A. Jirawattanapanit, and P. Mohandas, "New Adaptive Finite-Time Cluster Synchronization of Neutral-Type Complex-Valued Coupled Neural Networks with Mixed Time Delays," *Fractal and Fractional*, vol. 6, no. 9, pp. 515–515, Sep. 2022, doi: 10.3390/fractalfract6090515.

[10] S. S. Sivatha Sindhu, S. Geetha, and A. Kannan, "Decision tree based light weight intrusion detection using a wrapper

approach," *Expert Systems with Applications*, vol. 39, no. 1, pp. 129–141, Jan. 2012, doi: 10.1016/j.eswa.2011.06.013.

[11] Z. H. Abdaljabar, O. N. Ucan and K. M. Ali Alheeti, "An Intrusion Detection System for IoT Using KNN and Decision-Tree Based Classification," *2021 International Conference of Modern Trends in Information and Communication Technology Industry (MTICTI)*, vol. 1, no. 1, pp. 1-5, 2022.

[12] C. Liu, Z. Gu, and J. Wang, "A Hybrid Intrusion Detection System Based on Scalable K-Means+ Random Forest and Deep Learning," *IEEE Access*, vol. 9, no. 1, pp. 75729–75740, 2021, doi: 10.1109/access.2021.3082147.

[13] B. Susilo and R. F. Sari, "Intrusion Detection in IoT Networks Using Deep Learning Algorithm," *Information*, vol. 11, no. 5, p. 279, May 2020, doi: 10.3390/info11050279.

[14] M. Almiani, A. AbuGhazleh, A. Al-Rahayfeh, S. Atiewi, and A. Razaque, "Deep recurrent neural network for IoT intrusion detection system," *Simulation Modelling Practice and Theory*, vol. 101, no. 1, pp. 1-12, Nov. 2019, doi: 10.1016/j.simpat.2019.102031.

[15] A. Aljumah, "IoT-based intrusion detection system using convolution neural networks," *PeerJ Computer Science*, vol. 7, no. Sep., pp. 1-19, Sep. 2021, doi: 10.7717/peerj-cs.721.

[16] A. Elsaeidy, K. S. Munasinghe, D. Sharma, and A. Jamalipour, "Intrusion detection in smart cities using Restricted Boltzmann Machines," *Journal of Network and Computer Applications*, vol. 135, pp. 76–83, Jun. 2019, doi: 10.1016/j.jnca.2019.02.026.

[17] R. Qaddoura, A. M. Al-Zoubi, H. Faris, and I. Almomani, "A Multi-Layer Classification Approach for Intrusion Detection in IoT Networks Based on Deep Learning," *Sensors*, vol. 21, no. 9, p. 2987, Apr. 2021, doi: 10.3390/s21092987.

[18] A. S. Almogren, "Intrusion detection in Edge-of-Things computing," *Journal of Parallel and Distributed Computing*, vol. 137, no. Mar., pp. 259–265, Mar. 2020, doi: 10.1016/j.jpdc.2019.12.008.

[19] A. Singh, K. Chatterjee, and S. C. Satapathy, "An edge based hybrid intrusion detection framework for mobile edge computing," *Complex and Intelligent Systems*, vol. 8, no. Aug., pp. 3719–3746, Aug. 2021, doi: 10.1007/s40747-021-00498-4.

[20] G. Kalnoor and S. Gowrishankar, "IoT-based smart environment using intelligent intrusion detection system," *Soft Computing*, vol. 25, no. 17, pp. 11573–11588, Jul. 2021, doi: 10.1007/s00500-021-06028-1.

[21] A. Buzachis, M. Fazio, A. Galletta, A. Celesti, and M. Villari, "Intelligent IoT for Non-Intrusive Appliance Load Monitoring Infrastructures in Smart Cities.," *IRIS Institutional Research Information System - AIR Institutional Research Archive*, vol. 1, no. 1, pp. 97–106, Jan. 2019.

[22] R. Sandhu, A. S. Sohal, and S. K. Sood, "Identification of malicious edge devices in fog computing environments," *Information Security Journal: A Global Perspective*, vol. 26, no. 5, pp. 213–228, Jul. 2017, doi: 10.1080/19393555.2017.1334843.

[23] B. Sudqi Khater, A. W. B. Abdul Wahab, M. Y. I. B. Idris, M. Abdulla Hussain, and A. Ahmed Ibrahim, "A Lightweight Perceptron-Based Intrusion Detection System for Fog Computing," *Applied Sciences*, vol. 9, no. 1, p. 178, Jan. 2019, doi: 10.3390/app9010178.

[24] H. Cao, M. Wachowicz, and S. Cha, "Developing an edge computing platform for real-time descriptive analytics," *in 2017 IEEE International Conference on Big Data (Big Data)*, vol. 2017, no. 1, pp. 4546-4554, 2017, doi: 10.1109/BigData.2017.8258497

[25] A. Mudgerikar, P. Sharma, and E. Bertino, "Edge-Based Intrusion Detection for IoT devices," *ACM Transactions on Management Information Systems*, vol. 11, no. 4, pp. 1–21, Dec. 2020, doi: 10.1145/3382159.

[26] D. Sugianto and A. R. Hananto, "Geospatial Analysis of Virtual Property Prices Distributions and Clustering,"*Int. J. Res. Metav.*, vol. 1, no. 2, pp. 127-141, 2024.

[27] X. Cao, Y. Fu, and B. Chen, "Packet-Based Intrusion Detection Using Bayesian Topic Models in Mobile Edge Computing," *Security and Communication Networks*, vol. 2020, no. Aug., pp. 1–12, Aug. 2020, doi: 10.1155/2020/8860418.

[28] B. Cheng, G. Solmaz, F. Cirillo, E. Kovacs, K. Terasawa, and A. Kitazawa, "FogFlow: Easy Programming of IoT Services Over Cloud and Edges for Smart Cities," *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 696–707, Apr. 2018, doi: 10.1109/jiot.2017.2747214.

[29] A. R. Yadulla, M. H. Maturi, G. S. Nadella, and S. Satish, "Volatility Comparison of Dogecoin and Solana Using Historical Price Data Analysis for Enhanced Investment Strategies", *J. Curr. Res. Blockchain*., vol. 1, no. 2, pp. 91–111, Sep. 2024.

[30] J. Zhang, B. Chen, Y. Zhao, X. Cheng, and F. Hu, "Data Security and Privacy-Preserving in Edge Computing Paradigm: Survey and Open Issues," *IEEE Access*, vol. 6, pp. 18209–18237, 2018, doi: 10.1109/access.2018.2820162.

[31] B. Varghese and R. Buyya, "Next generation cloud computing: New trends and research directions," *Future Generation Computer Systems*, vol. 79, no. 3, pp. 849–861, Feb. 2018, doi: 10.1016/j.future.2017.09.020.

[32] R. K. Sadavarte and G. D. Kurundkar, "Data security and integrity in cloud computing : Threats and Solutions," *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, vol. 2020, no. Dec., pp. 356–363, Dec. 2020, doi: 10.32628/cseit206667.

[33] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated Machine Learning," *ACM Transactions on Intelligent Systems and Technology*, vol. 10, no. 2, pp. 1–19, Feb. 2019, doi: 10.1145/3298981.

[34] T. Wahyuningsih and D. Sugianto, "Temporal Patterns in User Conversions: Investigating the Impact of Ad Scheduling in Digital Marketing," *J. Digit. Mark. Digit. Curr*., vol. 1, no. 2, pp. 165-182, 2024.

[35] S. A. Ghaffarand W. C. Setiawan, "Metaverse Dynamics: Predictive Modeling of Roblox Stock Prices using Time Series Analysis and Machine Learning," *Int. J. Res. Metav.*, vol. 1, no. 1, pp. 77-93, 2024.

[36] D. Yuan et al., "Intrusion Detection for Smart Home Security Based on Data Augmentation with Edge Computing," ICC 2020 - 2020 *IEEE International Conference on Communications (ICC)*, vol. 2020, no. 1, pp. 1-6.

[37] A. M. Salman, Haider Rasheed Abdulshaheed, Z. S. Jabbar, Ahmed Dheyaa Radhi, and Poh Soon JosephNg, "Enhancing quality of service in IoT through deep learning techniques," *Periodicals of Engineering and Natural Sciences (PEN)*, vol. 11, no. 3, pp. 68–68, May 2023, doi: 10.21533/pen.v11i3.3577.