International Journal of Advances in Scientific Research and Engineering (ijasre)

E-ISSN : 2454-8006

DOI: 10.31695/IJASRE.2023.9.12.8

Volume 9, Issue 12
December - 2023

An Improved Intrusion Detection in Wireless Sensor Networks Using Hybrid Multiclass Over-Sampling and Deep Neural Networks

Omeiza Aliyu. O[1], Bisallah Hashim. I[2], Okike Bnjamin[3] and Sanusi Muhammad[4]

Research Scholar[1] and Lecturer [2-4]

Department of Computer Science

University of Abuja, Abuja, Nigeria

ABSTRACT

With the emergence of new attacks, there is a continual need for innovative approaches that can closely monitor and swiftly adapt to evolving threats. IDSs can be broadly categorized into misuse detection and anomaly detection, each utilizing machine-learning methods. Machine learning algorithms, particularly those relying on datasets like DARPA and KDD Cup 1999, have gained popularity. However, challenges include dataset limitations, overfitting, and the requirement for substantial computational power. This study focuses on the specific problem of class imbalance in Wireless Sensor Networks (WSN) datasets for intrusion detection. Existing techniques, such as over sampling and under sampling, have limitations, and the imbalance poses challenges for accurate intrusion detection model development. The research aims to develop an enhanced intrusion detection model addressing multiclass imbalance through an optimized KNN-SMOTE oversampling technique integrated with a Deep Learning model. The significance lies in its potential to greatly enhance the accuracy of intrusion detection systems, contributing to the security of computer networks. The study's scope involves static dataset analysis and multiclass classification of intrusion attacks. The research questions revolve around the development of a multiclass oversampling technique, a hybrid model, and the performance evaluation of the developed system in comparison to existing IDS. The research proposes a novel OK-SMOTE-DL model combining the potential of the KNN model, firefly algorithm, and the SMOTE model fused with Deep Learning techniques to help create a better model for handling multiclass intrusion datasets that can similarly detect minority classes effectively in wireless sensor networks.

Key Words: Deep Learning model, Intrusion detection model development, Machine-learning methods, Wireless Sensor Networks.

1.0 INTRODUCTION

1.1 Background of the Study

IDSs provide critical tools for defenders to recognize and react to offensive incidents. The continuous struggle between attackers and defenders can be likened to an "arms race" where both sides constantly enhance their techniques to surpass each other. As fresh attacks emerge, scholars and professionals in the industry are motivated to examine new approaches that can closely monitor this competition and adapt swiftly to the advancements in the field [1]. The primary objective of an IDS is to detect any harmful behavior by analyzing network traffic or resource usage and raising an alarm if necessary. Based on the method of intrusion detection utilized, IDSs can be categorized into two main groups. The first group matches observed activities with a database of intrusion techniques, while the second group tracks regular behavior and notifies about any unusual incidents [2].

IDS solutions are crucial security elements that may successfully counter various security assaults when used in conjunction with firewalls. IDS strategies can be broadly divided into two groups: misuse detection and anomaly detection, both of which can be carried out using various machine-learning methods. Attack and malicious activity signatures are a major component of signature-based systems used for misuse detection, which supports multi-class

classification. Nevertheless, these systems are unable to recognize novel attacks or their modifications for which there are no signatures. On the other hand, IDS techniques based on anomaly detection can only allow binary classifications and can identify new threats by leveraging user profiles of expected behavior. Yet, profiles for users must be updated appropriately in dynamic companies where positions constantly change. Moreover, false positives for the IDS may be experienced with anomaly detection methods [3].

An established security technology called a Network Intrusion Detection System (NIDS) analyses network traffic to find security problems at the network level. A Host-based Intrusion Detection System (HIDS), on the other hand, is another common security solution that detects security threats directly on computer hosts by inspecting system logs, processes, files, or network interfaces. An IDS can become a Protocol-based Intrusion Detection System by focusing on particular protocols, such as the Hypertext Transfer Protocol (HTTP) of a web server (PIDS). A Structured Query Language (SQL) protocol for a database would be an example of an application protocol-based intrusion detection system and IDSs may also focus on monitoring such protocols (APIDS). IDSs can vary in several ways, much as the range of security event sources, such as networks and diverse host types [4].

Machine learning algorithms have become widely popular for developing effective Intrusion Detection Systems (IDSs). However, it is crucial to accurately learn and predict data before applying these methods. Many studies have used datasets such as DARPA and KDD Cup 1999, which account for a significant portion of IDS research. Some datasets, however, have drawbacks, including old attack versions that might not accurately reflect actual network attacks, a lack of data, and an excessive amount of redundant records that induce a bias when training Network Intrusion Detection Systems (NIDSs). The NSL-KDD dataset has been proposed to get around these restrictions. Recent studies have used ML classifiers to assess IDSs' efficacy in protecting web applications in an IoT setting, using web attacks as a metric [5].

Over time, Intrusion Detection Systems (IDSs) that rely on machine learning have been developed, utilizing state-of-the-art advances from the machine learning research community as well as the extensive datasets gathered in cybersecurity research. However, a significant challenge with these methods is that they are often trained on a single large dataset, making them prone to over-fitting when exposed to novel attack types. Moreover, these techniques typically necessitate vast amounts of data, which may not always be readily accessible and may take considerable time to collect. Additionally, several commonly employed machine learning algorithms in IDSs do not support online learning, implying that they need to be retrained from the beginning when new data is introduced, which necessitates substantial computational power [6].

1.2. Statement of the Problem

Wireless Sensor Networks (WSN) datasets for intrusion detection often exhibit class imbalance, where the number of data samples for each class is unequal, particularly in real-world WSN datasets with multiple classes. This imbalance complicates intrusion detection model development, leading to increased complexity and higher error rates, especially for the minority class. Various techniques, such as oversampling, undersampling, or a combination of both, have been proposed to address class imbalance, but they still have limitations. Few algorithms consider multiclass imbalance, and even when they do, accuracy in detecting the minority class remains a challenge [7-11].

To tackle this issue, several intrusion detection algorithms have been implemented, but the imbalanced nature of attack types can still introduce bias during training. Methods that successfully increase the minority class may introduce noise and duplication of irrelevant data, leading to overfitting. Common oversampling techniques, like ASN-SMOTE and ADASYN, fail to address performance drops in the majority class due to the addition of synthetic samples from the minority class. Additionally, the ASN-SMOTE technique is sensitive to the K value, impacting the performance of generated synthetic samples [12].

Therefore, this research aims to develop an enhanced intrusion detection model capable of handling multiclass imbalance and improving accuracy in a wireless sensor network. This will be achieved through the creation of an optimized KNN-SMOTE oversampling technique integrated with a Deep Learning model.

DOI: 10.31695/IJASRE.2023.9.12.8

1.3. AIM AND OBJECTIVES OF STUDY

This research work aims to drive and propose Novel algorithms and techniques for intrusion detection in wireless sensor networks using hybrid multiclass over-sampling and deep neural networks.

The specific objectives are:

i. To design and develop an optimized K Nearest Neighbour and Synthetic Minority Oversampling Technique (OK-SMOTE).

ii. To develop a hybrid of OK-SMOTE and an optimized Deep Learning Model for enhanced Intrusion Detection (ID).

iii. To apply the developed hybrid OK-SMOTE-DL model on the multiclass intrusion dataset for multiclass intrusion detection.

iv. To validate the performance of the developed hybrid model by comparing it with state-of-the-art IDS.

1.4. Significance of Study

The development of an advanced multiclass intrusion detection system, utilizing a novel hybrid KNN-SMOTE multiclass oversampling technique and deep neural network model, holds significant implications for researchers, society, and government. This approach has the potential to greatly enhance the accuracy and effectiveness of intrusion detection systems, crucial for ensuring the security of computer networks. By combining the strengths of KNN-SMOTE and a deep neural network, the proposed method adeptly handles imbalanced and multiclass datasets, leading to more precise and reliable detection of network intrusions.

This research is particularly vital for society given the persistent threat of cyber-attacks. Strengthening intrusion detection systems through the proposed approach contributes to safeguarding sensitive data, preventing financial losses, and protecting critical infrastructure from evolving cyber threats.

The implications extend to government and policymakers. With the increasing frequency and sophistication of cyber-attacks, governments globally are focusing on cybersecurity policies. The proposed approach serves as a valuable tool for governments, aiding in improving detection and response capabilities to protect national security interests in the face of growing cyber threats.

1.5. Scope of Study

This research work is constrained to detecting intrusion using a static dataset collected online. Additionally, we will be concentrating on the multi-class classification of intrusion attacks.

This research is expected to address the following questions;

i. Can a multiclass oversampling technique be developed and what strategy can be used?

ii. Can a hybrid multiclass oversampling technique and Deep Learning model be developed?

iii. How will the developed hybrid model perform when applied to multiclass intrusion detection?

iv. Can the developed hybrid system perform better or similar when compared with existing IDS?

2.0 LITERATURE REVIEW

2.1. Intrusion Detection in WSN with DNNs

Wireless Sensor Networks (WSNs) are cost-effective devices used for data sensing and transmission in various applications. Despite their utility, WSNs are susceptible to security threats, including intrusion attacks, where unauthorized users compromise network integrity or steal sensitive data. Intrusion detection, vital for network security, can be achieved using Deep Neural Networks (DNNs). DNNs, adept at pattern recognition, are well-suited for WSNs due to their ability to handle large datasets, robustness to noise, and adaptability to changing environments [13].

DOI: 10.31695/IJASRE.2023.9.12.8

Intrusion detection with DNNs involves training a model on normal and abnormal network traffic data. The model learns patterns associated with normal traffic and identifies deviations as potential intrusions. Common approaches include using Recurrent Neural Networks (RNNs) to model sequences of data, learning patterns indicating intrusions, and employing Convolutional Neural Networks (CNNs) to analyze sensor data for intrusion patterns in WSNs [3].

2.2 Related Studies

Several research works have been presented on intrusion detection. This review focused on intrusion detection systems showing the methods adopted, data used, and results obtained. [14] proposed unsupervised deep learning and Generative Adversarial Networks (GAN) to address data imbalance in network intrusion detection. The method used involves re-sampling training data using the GAN after studying the rare class and then using a Random Forest (RF) model to classify the re-sampled data. The CICIDS 2017 dataset was used in their study, and they found that the GAN model was effective in detecting imbalanced data in network intrusion. In addition, they compared the performance of GAN-RF with SMOTE-RF and found that the former outperformed the latter. The authors also plan to improve their intrusion detection system by including an autoencoder model that compresses data characteristics to a low level before resampling with GAN.

First quote the authors name like, Gonzalez-Cuautle, D et al., introduced a technique named SMOTE

Dhanabal, L et al., introduced a technique named SMOTE to overcome the problem of imbalanced datasets in two collections of data, CIDDS-001 and ISCX-Bot-2014, which involve genuine data related to botnets and intrusion detection systems. The technique includes various stages such as obtaining and labeling data, oversampling minority-class samples by creating synthetic examples, selecting important features using Principal Component Analysis (PCA), employing supervised machine-learning classification algorithms to train a fully balanced set, improving the classification accuracy by grid search, and evaluating the classification models' performance by measures such as Accuracy, Recall, and F1-Score [15].

Tavallaee, M et al., introduced a new method for identifying attacks in a dataset that is not balanced. The technique is a combination of an LSTM autoencoder and an OC-SVM. An OC-SVM is a type of support vector that is often used to identify anomalies in an unsupervised manner. It is trained solely on data that is considered "normal" to learn the boundaries of those points [16].

Mulyanto, M et al., put forward a method for intrusion detection using a 1DCNN on the UNSW NB15 dataset. Their approach involved using 32 convolution filters, each with a kernel size of 5 and a 'sigmoid' activation function. The generated 32 filters enable the model to learn multiple features in the first convolution layer [17].

Mijalkovic, J et al., utilized the UGR'16 dataset for data wrangling purposes to effectively train their neural network model and conduct experiments on various input sizes such as 10000, 50000, and 1 million. Additionally, a generative adversarial network (GAN) model was used to balance the dataset by producing attack label samples such as blacklist, anomaly spam, and SSH scan. The classifier was trained using 60% of the dataset as the training set and 40% as the test set. The model was trained using Sklearn, the neural network library, and the Broyden-Fletcher-Goldfarb-Shanno optimization solver to minimize error [18].

Fu, Y et al., proposed a novel approach to network intrusion detection using a genetic algorithm (GA) and fuzzy C-means clustering (FCM) to enhance feature subset selection. The algorithm combines a bagging (BG) classifier with a convolutional neural network (CNN) model to extract deep features. The GA is employed with 5-fold cross-validation (CV) to determine the optimal CNN model structure, and the extracted features are evaluated using a BG classifier with a 5-fold CV [19].

Jung, I. et al., proposed BMCD, a novel technique that addresses the class imbalance in large-scale multiclass datasets. The approach employs a modified version of the Synthetic Minority Over-Sampling Technique (SMOTE) designed specifically for multiclass datasets to enhance detection rates of minority classes while maintaining efficiency. The study utilized the CICIDS2017 dataset, which includes benign data and various attack types, such as DoS, DDoS, brute force SSH, brute force FTP, heartbleed, infiltration, and botnet, making it one of the most current datasets available. CICFlowMeter was used to analyze network traffic results [20].

DOI: 10.31695/IJASRE.2023.9.12.8

introduced a technique that utilizes machine learning methods and information from user and kernel spaces to identify security breaches in smart devices. Their approach involves tracing device behavior using a low-overhead LTTng tracer to collect system information and then processing the data into numeric arrays, which can be used to train machine learning algorithms. The technique employs an analysis system consisting of sensors, actuators, and an analysis engine to detect unusual behavior and issue alerts in the event of an intrusion. The trace data is transmitted in optimized binary format (CTF) through the network to facilitate rapid processing by the machine learning algorithms.

[22] proposed a machine learning-based lightweight intrusion detection model for IoT networks that have limited resources. Their approach includes several steps, such as data sampling, segmentation, feature ranking, dimensionality reduction, adding non-linearity, ensemble learning, and data processing. The model they developed demonstrated better performance with higher detection rates and lower false positive and negative rates compared to conventional IDS approaches. The proposed model was tested on two publicly available datasets, namely CICIDS2017 and NSL-KDD, and was trained using the B-Stacking algorithm, taking only 0.28 seconds for training and 0.02 seconds for prediction.

[23] presented an intrusion detection scheme that merges feature dimensionality reduction with an enhanced Long Short-Term Memory (LSTM) model. To reduce the model's complexity, a Stacked Autoencoder (SAE) is employed to compress high-dimensional feature inputs into low-dimensional feature outputs. The approach utilizes a Bi-directional LSTM model that includes an Attention Mechanism to identify bidirectional structural features that LSTM alone is unable to detect.

The suggested approach by Andresini, G et al., includes the following steps: The training set (T) is first transformed into the training set (T2D) by converting each sample from a non-image form to a grayscale image form. Second, it processes (T2D, Y) and creates fresh images of fabricated assaults to balance the training set using the ACGAN architecture. Third, it uses the enhanced training set (T2D, Y) to train a 2D CNN architecture to identify attacks from regular network traffic. The study's two datasets, CICIDS2017 and AAGM17, both feature an uneven amount of malicious traffic, with 80% of regular flows and 20% of attacks. Categorical features in the KDDCUP99 and UNSW-NB15 datasets need to be transformed using one-hot encoding 24].

Qaddoura, R et al., described a method that uses a single hidden layer feed-forward neural network and consists of three steps: clustering with reduction, oversampling, and classification (SLFN). The approach is shown as Algorithm 1, which requires inputs like the dataset, the number of clusters, the oversampling ratio, and the reduction percentage. On line 1, the data are first divided into training and test sets, after which the three stages of reduction with clustering, oversampling, and classification are carried out on lines 2 to 4, 5, and 6, respectively. The testing set on Line 7 is then predicted using the generated classification model. To assess the effectiveness of the suggested strategy, the accuracy, precision, recall, and G-mean are computed on Line 8 [25].

Zhou, F., et l., developed a model, named NIDD (Network Intelligent Data Detection), to identify network intrusions using three main elements: Deep Convolution Generation Adversarial Network (DCGAN), Light Gradient Boosting Machine (LightGBM), and Shapley Additive exPlanations (SHAP). The NIDD model can create new attack samples by studying the current attack sample data, which can assist in increasing the number of rare attack samples. The LightGBM algorithm serves as the base classifier to train the dataset and build the intrusion detection model [26].

Khanam, S et al., presented a new method called CFLVAE to address the problem of data imbalance in intrusion detection. This method is based on a deep generative model that creates new samples for the minority attack classes using the CFL objective function. The generated data is high-quality, varied, and authentic which leads to a well-balanced intrusion dataset and enhances the accuracy of intrusion detection using learning-based classifiers. The proposed approach includes training a distinct architecture DNN classifier using the balanced intrusion dataset to improve detection performance [27].

2.3. Research Gap

From all the reviews conducted, the following issues were observed;

i. Multiclass Intrusion Detection is still an issue that needs to be addressed due to the difficulty in detecting minority instances.

DOI: 10.31695/IJASRE.2023.9.12.8

ii. From the literature reviewed, multiclass data imbalance is still an open problem that requires attention. Most of the data balancing techniques are for binary class imbalance problems.

iii. The application of machine learning models in detecting intrusions is an ongoing research that still requires improvements.

3.0 RESEARCH METHODOLOGY

3.1. Research Design

A summary of the proposed methodology for achieving each objective is shown in Figure 3.1. For objective one, the optimized K-Nearest neighbor (KNN) Synthetic Minority Oversampling Technique (SMOTE) named (OK-SMOTE) minority over-sampling technique will be designed, the mathematical models will be formulated and the model developed using a MATLAB programming environment. For objective two, the Deep Learning (DL) model will be designed and hybridized with the developed OK-SMOTE. For objective three, the WSN NSL-KDD dataset will be collected, applying OK-SMOTE-DL. Finally, the performance of the developed system will be evaluated using accuracy, precision, recall, FPR, TPR, and F1-Score performance metrics. The validation of the performance will be conducted in this section.

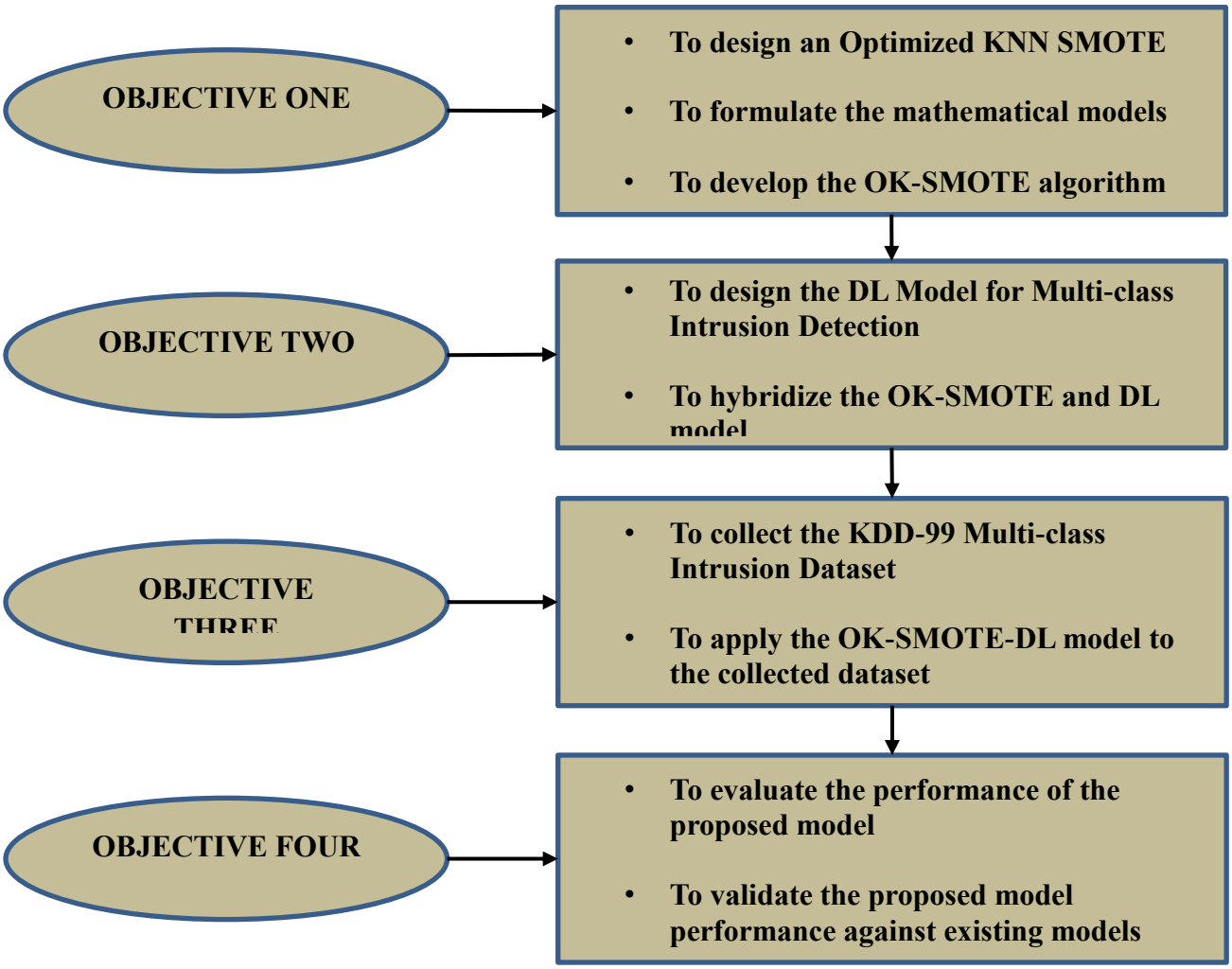

Figure 3.1: Proposed Research Methodology

3.2. Design of Optimized Multiclass KNN SMOTE Algorithm

In the first phase, the proposed optimized KNN SMOTE (OK-SMOTE) algorithm is a combination of SMOTE minority oversampling, KNN, and firefly algorithm fused to produce a better-balanced dataset for multiclass imbalance data. The new minority oversampling technique proposed in this research is inspired by the ASN-SMOTE

DOI: 10.31695/IJASRE.2023.9.12.8

minority oversampling with adaptive qualified synthesizer selection proposed by [28]. The samples produced by this algorithm are affected by the value of k. An appropriate tuning of k is capable of producing better synthetic samples and a balanced dataset that will eventually lead to improved detection and classification of minority attacks in WSN. Figure 3.2 shows the architecture of the proposed OK-SMOTE is proposed.

The algorithm is divided into three steps; optimized multiclass noise filtering, Optimized Neighbor Instance Selection, and generation of synthetic instances.

i. Multiclass Noise Filtering

Noise filtering from data is a vital step in data mining and machine learning training. The success of ML algorithms is largely affected by the amount of noise in the data. In the imbalance data concept, noise are outliers that are around the decision boundary. The algorithm is to detect noise and minority instances and filters accordingly using the KNN algorithm.

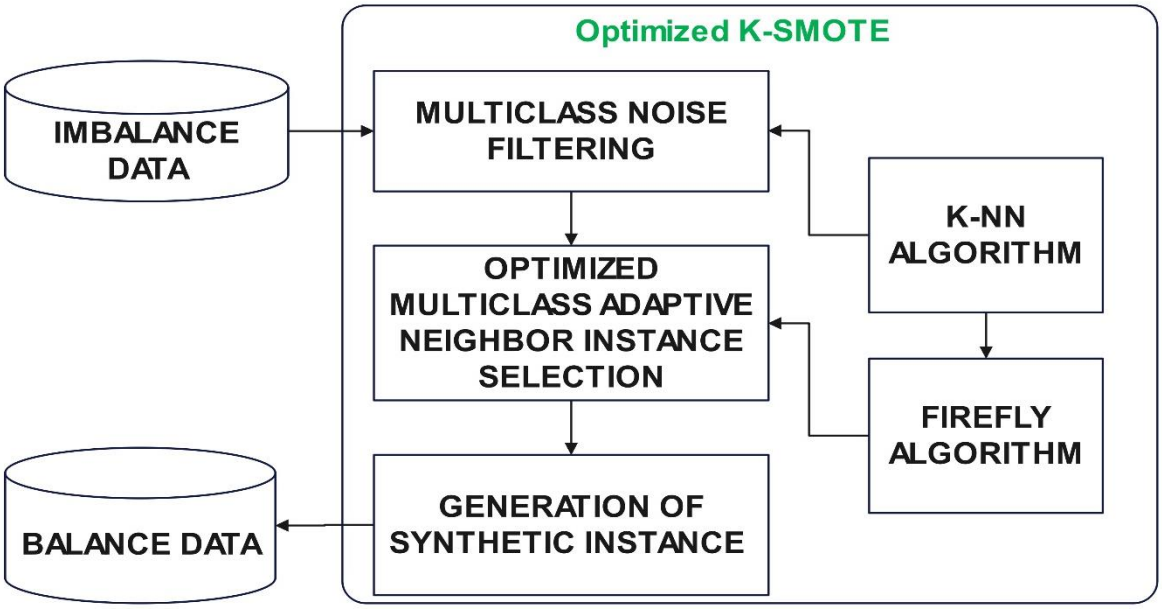

Figure 3.2: OK-SMOTE Model Architecture

Mathematically, the filtering technique is represented as follows:

Let $\mathbf{V}$ be the complete data, $\mathbf{N_c}$ be the number of classes, $\mathbf{M}$ be the majority class, **and** $\mathbf{R}$ be the minority class. $\mathbf{N_m}$ and $\mathbf{N_r}$ are the numbers of majority and minority class instances. For each minority class $\mathbf{R_i} \in \mathbf{R}, (\mathbf{i = 1, 2}, \ldots, \mathbf{N_c - 1})$, and for each minority class instance $\mathbf{R_{ij}} \in \mathbf{R_i}, (\mathbf{j = 1, 2}, \ldots, \mathbf{N_r})$, find the nearest instance $\boldsymbol{\delta}(\mathbf{R_{ij}})$ by calculate the Euclidean distance to each instance in $\mathbf{Q}$ as follows:

$$\delta(R_{ij}) = \arg_{v \epsilon V, v \neq R_{ij}} \min \left\| R_{ij} - v \right\|_2 \tag{3.1}$$

Where $\left\| R_{ij} - v \right\|_2$ returns the Euclidean distance between two points R_{ij} and v.

Given the qualified and unqualified set of minority instances to be $\mathbf{S_q}, \mathbf{S_u}$, the unqualified minority instances and noise are obtained as follows:

$$R_{ij} \in S_{iq}, \text{if } \delta(R_{ij}) \notin M \tag{3.2}$$

$$R_{ij} \in S_{iu}, \text{if } \delta(R_{ij}) \in M \tag{3.3}$$

ii. Optimized Multiclass Adaptive Neighbor Instance Selection

Selecting appropriate neighboring instances is vital for the effective generation of synthetic samples. The existing SMOTE algorithm selects neighboring instances randomly, which affects its performance. While algorithms like ASN-SMOTE propose an adaptive neighbor instance selection strategy, its performance was also affected by the **k**

DOI: 10.31695/IJASRE.2023.9.12.8

value [28]. To address this problem, an optimized adaptive neighbor instance selection is proposed. The implementation is highlighted as follows:

a. The firefly algorithm will be used to first generate the optimal **k** value to be used to obtain the k-nearest neighbor of the minority class.

b. For each instance in S_{iq} given as $q_{it}, t = 1, 2, \dots, N_q$, in minority class i, where N_q is the number of qualified instances. The Euclidean distance between q_{it} and V is calculated to determine the **k** nearest neighbor instances given as $q_{it}{}^{1}, q_{it}{}^{2}, \dots, q_{it}{}^{k}$.

c. If all neighboring instances are minority instances, they are added to a set Q_{it}. Given as:

$$q_{it} \in Q_{it} \quad , \text{if } q_{it} \in R_{ij} \tag{3.4}$$

d. If the k nearest neighbor instance contains at least one majority instance, find the nearest instance ($q_{it}{}^{near}$).

e. Lastly, calculate the Euclidean distance of each neighbor in the minority class using:

$$\delta' = \|q_{it} - q_{it}{}^{near}\|_2 \tag{3.5}$$

$$q_{it} \in Q_{it} \quad , \text{if } \delta(R_{ij}) < \delta' \tag{3.6}$$

iii. Optimized Synthetic Instances Generation

The final step of the proposed optimized multiclass oversampling technique is to generate synthetic instances. This will be achieved using the Linear Interpolation method due to its simplicity and accuracy in generating actual minority class instances. Linear interpolation is a method used in minority oversampling techniques to generate synthetic data points for the minority class. Linear interpolation addresses this imbalance by creating new synthetic data points that lie between existing minority class instances. The process of linear interpolation involves selecting two neighboring minority class instances and creating a new data point along the straight line connecting them. The position of the new data point is determined by a weighting factor, typically ranging from 0 to 1. A weighting factor of 0 corresponds to replicating the first instance, while a factor of 1 corresponds to replicating the second instance. Intermediate values produce synthetic data points lying at different positions along the line.

The reason for using linear interpolation in minority oversampling is to introduce new data points that reflect the underlying distribution of the minority class. By generating synthetic instances within the range of existing minority class examples, the algorithm attempts to capture the patterns and characteristics of the minority class more accurately. This process helps to alleviate the class imbalance and improve the performance of machine learning models, particularly those sensitive to imbalanced data. Some advantages of using linear interpolation in minority oversampling include:

i. **Preserving Data Distribution**: Linear interpolation ensures that the synthetic data points are created within the range of existing minority class instances, maintaining the overall distribution and characteristics of the minority class.

ii. **Smooth Interpolation**: The linear nature of interpolation results in a smooth transition between neighboring minority class instances. This can be beneficial when the underlying data exhibits a gradual change or progression.

iii. **Controlled Generation**: The weighting factor used in linear interpolation allows for control over the position of the synthetic data points. This flexibility enables the algorithm to create diverse examples at different locations along the line, increasing the variability of the minority class.

However, some limitations may include Potential overfitting which is likely to occur when generated data points closely resemble existing instances. The synthetic data may not capture the full diversity of the minority class and may introduce redundancy. Also, interpolated data points only rely on the information present in the two neighboring instances. If there are sparse or inadequate examples near a certain region, the interpolation may not adequately capture the true characteristics of that area.

DOI: 10.31695/IJASRE.2023.9.12.8

It is worthy of note that the proposed technique in this research has reduced the potential of the limitations by removing noisy instances before interpolation making it very suitable for this research work. To achieve this, the number of synthetic instances required to be generated will be calculated using:

$$N_S = \text{round}\left[\frac{N_m - N_r}{N_q}\right] \quad (3.7)$$

Where N_m and N_r are the numbers of majority and minority class instances and N_q is the number of qualified instances. Equation 3.7 is necessary to calculate the number of instances required to be generated for each minority sample class. For each minority class, a qualified neighbor (q_{it}') will be randomly selected from the set of all qualified instances (Q_{it}), and the linear interpolation is performed to generate the instances using the equation:

$$S_i = q_{it} + w \times (q_{it}' - q_{it}) \quad (3.8)$$

Where the weight is a random number between 0 and 1 ($w \in (0,1)$). This is repeated for N_S times and N_C classes. Figure 3.3 shows the OK-Smote algorithm.

3.3. Hybridization of KNN-SMOTE and Deep Learning Model

In the second phase, the proposed OK-SMOTE will be hybridized with deep learning models to classify the required datasets. Figure 3.4 shows the proposed optimized hybrid OK-SMOTE and Machine Learning (ML) models for enhanced intrusion detection in WSN. The imbalance data will first be balanced using the proposed multiclass OK-SMOTE minority oversampling technique. The balanced data will then be pre-processed to put in a format suitable for the ML model. The data will then be portioned in the ratio 70:30 for training and testing respectively.

The model hyper-parameters will be optimized using the FA algorithm. Figure 3.5 shows the hybrid OK-SMOTE and DL algorithms for enhanced intrusion detection. After training, the 30% of data not used for training will be used for testing.

Hyperparameter optimization is a crucial step in building an accurate and efficient deep neural network model. The choice of hyperparameters such as learning rate, batch size, and the number of hidden layers can significantly impact the performance of the model. The Firefly algorithm is a metaheuristic optimization algorithm inspired by the flashing behavior of fireflies. It is widely used for optimization problems in various domains, including machine learning. The Firefly algorithm can effectively optimize the hyperparameters of a deep neural network model.

The algorithm starts by initializing a swarm of fireflies, each representing a candidate solution. The fireflies are attracted to other brighter fireflies, and their movement is governed by a set of rules based on their brightness and distance from other fireflies. The algorithm iteratively updates the position of each firefly until a satisfactory solution is found.

In this research, the firefly algorithm will be used to search for optimal values of hyperparameters. The algorithm starts by generating an initial set of hyperparameters, which are used to train the model. The performance of the model is then evaluated using a validation set, and the brightness of each firefly is calculated based on its accuracy. The algorithm then updates the position of each firefly to search for better hyperparameters, and the process repeats until an optimal set of hyperparameters is found.

DOI: 10.31695/IJASRE.2023.9.12.8

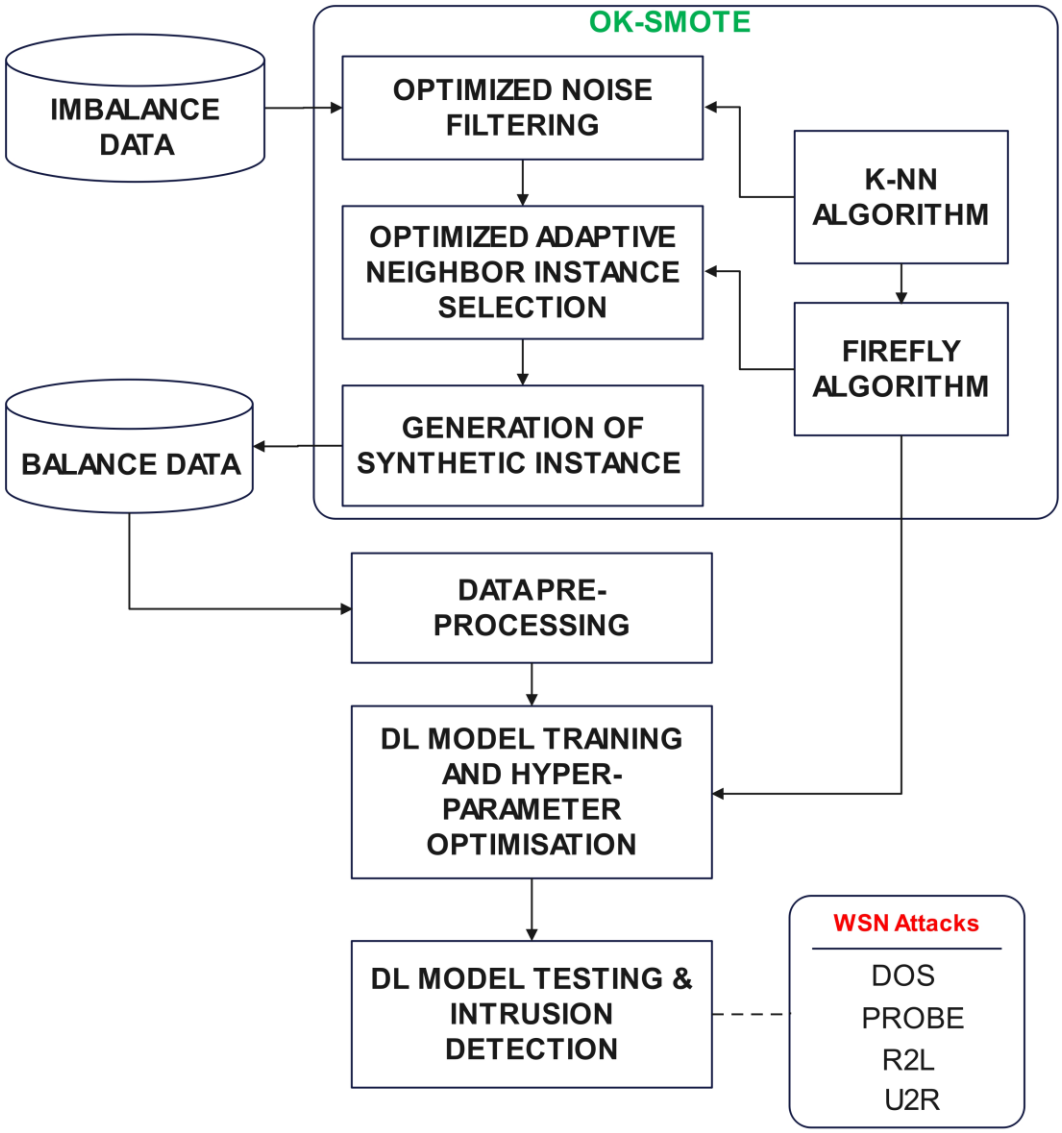

Figure 3.4: Hybrid KNN-SMOTE and DL Model

The objective function (**F**) to be used by FA to evaluate the fitness of each firefly is the root mean squared error mathematically represented as:

$$\mathbf{F} = \sqrt[2]{\frac{1}{T}\sum_{i=1}^{T}(Y - X)^2} \qquad (\mathbf{3.10})$$

Where T is the total test sample, Y is the actual category and X is the predicted category.

RMSE is a widely used objective function that offers several advantages, including sensitivity to errors, interpretability, continuity, and differentiability. RMSE penalizes larger errors more than smaller ones, making it suitable for scenarios where minimizing significant errors is crucial. Its expression in the same units as the target variable allows easy interpretation and evaluation of the model's performance by stakeholders. Additionally, the

DOI: 10.31695/IJASRE.2023.9.12.8

continuity and differentiability of RMSE enable efficient optimization using gradient-based algorithms. It considers all data points, providing a comprehensive evaluation of the model's performance across the entire dataset and encouraging improvements in prediction accuracy.

3.4. Data Collection and Pre-processing

For this experiment, the NSL-KDD network-based Intrusion dataset will be utilized. The NSL-KDD dataset is a benchmark dataset widely used for evaluating intrusion detection systems [29]. It is a modified version of the original KDD Cup 1999 dataset, which is preprocessed to remove redundant and irrelevant features [30]. The dataset contains both normal and attack traffic and is used to train and test machine-learning models for intrusion detection. The dataset contains KDD Train+, and KDD Test+ consisting of 40 features and one categorical class. The dataset contains five classes of intrusion attacks - Normal, Dos, Probe, R2L, and U2R, with a varying number of samples in each class as shown in Figure 3.1. A sample of 10 records with 5 features from the NSL-KDD dataset is shown in Table 3.2, including 3 different class attributes.

Table 3.1: NSL-KDD dataset sample numbers

Data Set Type	Total Samples	Normal	DOS	PROBE	U2R	R2L
KDD Train+	125973	67343	45927	11656	52	995
KDD Test+	22544	9711	7458	2421	200	2754

Table 3.2: NSL-KDD dataset sample

Sample No	Duration	Src_Byte	Dst_Byte	Land	Wrg_Frag	Attack Type
1	0	300	13788	0	0	'Dos'
2	0	233	616	0	0	'Dos'
3	0	343	1178	0	0	'Dos'
4	0	253	11905	0	0	'Dos'
5	507	437	14421	0	0	'Dos'
6	0	227	6588	0	0	'Dos'
7	0	215	10499	0	0	'normal'
8	0	241	1400	0	0	'R2L'
9	0	303	555	0	0	'Dos'
10	0	45	45	0	0	'Dos'

As part of the pre-processing step in this research, all textual features were converted to numeric and then subjected to denoising to remove features with a certain percentage of zeros. The number of non-zero elements for each feature was calculated, and any feature with less than 30% non-zero elements was filtered out. This threshold is experimental and can be adjusted during the experiment. Finally, the features were normalized, particularly those with a large difference among their members.

The developed hybrid OK-SMOTE and Deep Learning model will be applied to the collected NSL-KDD dataset to classify the data into different categories. Using MATLAB2022b programming language on a Windows 10, 64-bit

DOI: 10.31695/IJASRE.2023.9.12.8

Operating System, 2.6Ghz coreTMi7 processor. This software and systems specification will be optimal for carrying out the experiments and producing the desired results.

The use of MATLAB software will be crucial in the development of a hybrid data oversampling and deep neural network for enhanced intrusion detection. The oversampling technique (such as SMOTE) and optimized KNN SMOTE (OK-SMOTE) will be implemented in MATLAB to generate synthetic samples in the regions where the minority class is underrepresented. The deep neural network (such as LSTM or 1DCNN) will also be developed in MATLAB to classify the network traffic as normal or abnormal and the various classes. The oversampled data will be used to train the deep neural network, which will then be tested on a dataset (such as NSL-KDD) to evaluate its performance. The output will be a model that can accurately detect and classify network attacks, which can help improve the security of telecommunication infrastructure.

3.5. Performance Evaluation and Validation

To evaluate the performance of the proposed model, the following metrics will be deployed:

i. Accuracy (Acc): Acc measures the number of attacks and samples correctly classified divided by the total number of samples in the test dataset. It is mathematically given as:

$$\text{Acc} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}} \tag{3.10}$$

ii. Precision: This is the proportion of attacks that are classified as true attacks. It is mathematically represented as:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{3.11}$$

iii. Recall: This is the proportion of actual attacks that are correctly classified. It is mathematically represented as:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{3.12}$$

iv. F1-Score: is the harmonic mean of precision and recall represented mathematically as:

$$\text{F1} - \text{score} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3.13}$$

Where; TP is True Positive, TN is True Negative, FP is False Positive and FN is False Negative.

3.5.1. Performance Validation

To validate the performance of the proposed hybrid intrusion detection model, experiments will be performed using the NSL-KDD datasets with DNN only as the ID model. The performance obtained will be compared with the proposed technique. Furthermore, our proposed model will be validated by comparing it with the existing IDS proposed by [31; 7; 32; 33; 34; 26; & 27]. The validation will be based on the evaluated metrics such as accuracy, precision, recall, and F1-score.

4.0. CONCLUSION

The utilization of the SMOTE model as well as deep learning models in intrusion detection has provided promising results. By developing an improved hybrid multiclass data oversampling technique called OK-SMOTE which will be hybridized with a deep learning model for enhanced intrusion detection in wireless sensor networks, we can improve the accuracy in detecting minority classes while also retaining the ability to detect intrusions on a multiclass scale.

REFERENCES

1. Papamartzivanos, D., & Kambourakis, G. (2019). Introducing Deep Learning Self-Adaptive Misuse Network Intrusion Detection Systems. 7. https://doi.org/10.1109/ACCESS.2019.2893871

2. Vaigandla, K., Azmi, N., & Karne, R. (2022). Investigation on Intrusion Detection Systems (IDSs) in IoT. 10(3).

DOI: 10.31695/IJASRE.2023.9.12.8

3. Lansky, J. A. N., Ali, S., Mohammadi, M., Majeed, M. K., & Rahmani, A. M. (2021). Deep Learning-Based Intrusion Detection Systems : A Systematic Review. IEEE Access, 9, 101574–101599. https://doi.org/10.1109/ACCESS.2021.3097247
4. Zuech, R., Khoshgoftaar, T. M., & Wald, R. (2015). Intrusion detection and Big Heterogeneous Data : a Survey. https://doi.org/10.1186/s40537-015-0013-4
5. Maseer, Z. K., Yusof, R., Bahaman, N., Mostafa, S. A., Feresa, C. I. K., & Foozy, M. (2021). Benchmarking of Machine Learning for Anomaly Based Intrusion Detection Systems in the CICIDS2017 Dataset. 9, 22351–22370. https://doi.org/10.1109/ACCESS.2021.3056614
6. Dang, Q. (2019). Studying machine learning techniques for intrusion detection systems To cite this version : HAL Id : hal-02306521.
7. Muhuri, P. S., Chatterjee, P., Yuan, X., Roy, K., & Esterline, A. (2020). Using a Long Short-Term Memory Recurrent Neural Network (LSTM-RNN) to Classify Network Attacks. 1, 1–21.
8. Nagaraja, A., Boregowda, U. M. A., & Khatatneh, K. (2020). Similarity-Based Feature Transformation for Network Anomaly Detection. IEEE Access, 8, 39184–39196. https://doi.org/10.1109/ACCESS.2020.2975716
9. Feng, S., Zhao, C., & Fu, P. (2020). A cluster-based hybrid sampling approach for imbalanced data classification. Review of Scientific Instruments, 91(5). https://doi.org/10.1063/5.0008935.
10. Xu, Z., Shen, D., Nie, T., & Kou, Y. (2020). A hybrid sampling algorithm combining M-SMOTE and ENN based on Random forest for medical imbalanced data. Journal of Biomedical Informatics, 107(June), 103465. https://doi.org/10.1016/j.jbi.2020.103465
11. Yi, X., Xu, Y., Hu, Q., Krishnamoorthy, S., Li, W., & Tang, Z. (2022). ASN-SMOTE: a synthetic minority oversampling method with adaptive qualified synthesizer selection. Complex and Intelligent Systems, 8(3), 2247–2272. https://doi.org/10.1007/s40747-021-00638-w
12. Susan, S., & Kumar, A. (2021). The balancing trick: Optimized sampling of imbalanced datasets—A brief survey of the recent State of the Art. Engineering Reports, 3(4). https://doi.org/10.1002/eng2.12298
13. Jia, H., Liu, J., Zhang, M., He, X., & Sun, W. (2021). Network intrusion detection based on IE-DBN model ☆ . Computer Communications, 178(July), 131–140. https://doi.org/10.1016/j.comcom.2021.07.016
14. Lee, J., & Park, K. (2019). GAN-based imbalanced data intrusion detection system.
15. Gonzalez-Cuautle, D., Hernandez-Suarez, A., Sánchez-Pérez, G., Toscano-Medina, L.K., Portillo-Portillo, J., Olivares-Mercado, J., Perez-Meana, H., & Sandoval-Orozco, A.L. (2020). Synthetic Minority Oversampling Technique for Optimizing Classification Tasks in Botnet and Intrusion-Detection-System Datasets. Applied Sciences.
16. Elsayed, M. S., Dev, S., & Jurcut, A. D. (2020). Network Anomaly Detection Using LSTM-Based Autoencoder. 37–45. https://doi.org/10.1145/3416013.3426457.
17. Azizjon, M., & Kim, W. (2020). 1D CNN-based network intrusion detection with normalization on imbalanced data. 218–224.
18. Yilmaz, I., Masum, R., & Siraj, A. (2020). Addressing Imbalanced Data Problem with Generative Adversarial Network For Intrusion Detection.
19. Nguyen, B. H., Xue, B., & Zhang, M. (2020). A survey on swarm intelligence approaches to feature selection in data mining. Swarm and Evolutionary Computation, 54(April 2019), 100663. https://doi.org/10.1016/j.swevo.2020.100663
20. Abdulrahman, A. A., & Ibrahem, M. K. (2020). Toward Constructing a Balanced Intrusion Detection Dataset Based on CICIDS2017. 2(3), 132–142.

DOI: 10.31695/IJASRE.2023.9.12.8

21. Gassais, R., Ezzati-jivan, N., Fernandez, J. M., Aloise, D., & Dagenais, M. R. (2020). Multi-level host-based intrusion detection system for Internet of things.

22. Roy, S., Li, J., Choi, B., & Bai, Y. (2021). A Lightweight Supervised Intrusion Detection Mechanism for IoT Networks. 1–13.

23. Lu, G., & Tian, X. (2021). An Efficient Communication Intrusion Detection Scheme in AMI Combining Feature Dimensionality Reduction.

24. Andresini, G., Appice, A., Rose, L. De, & Malerba, D. (2021). GAN augmentation to deal with the imbalance in imaging-based intrusion detection. Future Generation Computer Systems, 123, 108–127. https://doi.org/10.1016/j.future.2021.04.017.

25. Qaddoura, R., Al-zoubi, A. M., & Almomani, I. (2021). applied sciences A Multi-Stage Classification Approach for IoT Intrusion Detection Based on Clustering with Oversampling.

26. Zhou, F., Du, X., Li, W., Lu, Z., & Wu, J. (2022). NIDD : an intelligent network intrusion detection model for nursing homes. Journal of Cloud Computing. https://doi.org/10.1186/s13677-022-00361-y

27. Khanam, S., Ahmedy, I., Idris, M. Y. I., & Jaward, M. H. (2022). Towards an Effective Intrusion Detection Model Using Focal Loss Variational Autoencoder for Internet of Things (IoT). Sensors, 22(15), 5822. MDPI AG. Retrieved from http://dx.doi.org/10.3390/s22155822

28. Yi, X., Xu, Y., Hu, Q., Krishnamoorthy, S., Li, W., & Tang, Z. (2022). ASN-SMOTE: a synthetic minority oversampling method with adaptive qualified synthesizer selection. Complex and Intelligent Systems, 8(3), 2247–2272. https://doi.org/10.1007/s40747-021-00638-w

29. Dhanabal, L., & Shantharajah, S. P. (2015). A study on NSL-KDD dataset for intrusion detection system based on classification algorithms. International Journal of Advanced Research in Computer and Communication Engineering, 4(6), pp. 446-452.

30. Tavallaee, M., Bagheri, E., Lu, W., & Ghorbani, A. (2009). A Detailed Analysis of the KDD CUP 99 Data Set. Second IEEE Symposium on Computational Intelligence for Security and Defense Applications (CISDA). https://ieeexplore.ieee.org/document/5356528

31. Mulyanto, M., Faisal, M., Prakosa, S.W., Leu, J.-S. (2021). Effectiveness of Focal Loss for Minority Classification in Network Intrusion Detection Systems. Symmetry, 13, 4. https://doi.org/10.3390/sym13010004

32. Mijalkovic, J., & Spognardi, A. (2022). Reducing the False Negative Rate in Deep Learning Based Network Intrusion Detection Systems. Algorithms, 15(8), 258. MDPI AG. Retrieved from http://dx.doi.org/10.3390/a15080258

33. Fu, Y., Du, Y., Cao, Z., Li, Q., & Xiang, W. (2022). A Deep Learning Model for Network Intrusion Detection with Imbalanced Data. Electronics, 11(6), 898. MDPI AG. Retrieved from http://dx.doi.org/10.3390/electronics11060898

34. Jung, I., Ji, J., & Cho, C. (2022). EmSM: Ensemble Mixed Sampling Method for Classifying Imbalanced Intrusion Detection Data. Electronics, 11(9), 1346. MDPI AG. Retrieved from http://dx.doi.org/10.3390/electronics11091346

DOI: 10.31695/IJASRE.2023.9.12.8