Journal of Neonatal Surgery
ISSN(Online): 2226-0439
Vol. 14, Issue 11s (2025)
https://www.jneonatalsurg.com

OPEN ACCESS

Quantile Recursive-Random Outlier Imputation With Wrapper Boruta For Instrusion Detection In WSNs

Ms. D. Priyadarshini[1], Dr. K. Sarojini[2]

[1]Assistant Professor, Department of BCA, Dr. N. G. P. Arts and Science College, Coimbatore. Ph.D. Research scholar, PG and Research Department of Computer Science, Chikkanna Government Arts College,Tirupur.

Email ID: priyadevadharshini@gmail.com

[2]Assistant Professor, PG and Research Department of Computer Science, Chikkanna Government Arts College,Tirupur.

Email ID: saromaran@gmail.Com

Cite this paper as: Ms. D. Priyadarshini, Dr. K. Sarojini, (2025) Quantile Recursive-Random Outlier Imputation With Wrapper Boruta For Instrusion Detection In WSNs. *Journal of Neonatal Surgery*, 14 (11s), 1027-1050.

ABSTRACT

Wireless Sensor Networks (WSNs) are increasingly vulnerable to variety of security threats, making the detection of network intrusions critical. This study presents a novel approach for Intrusion Detection (ID) in WSNs by combining the advanced data pre-processing and feature selection techniques. Quantile Recursive-Random Outlier Imputing (QR-ROI) algorithm is used for effective data pre-processing, addressing missing or anomalous values in wsnds dataset, which is collected from the Kaggle repository. To further optimize the model's accuracy and efficiency, feature selection is conducted using the Wrapper Boruta Algorithm, enhanced with Light Gradient Boosting Machine (GBM), known for its speed and performance. The integration of these techniques not only improves the quality of dataset but also enhances the overall predictive accuracy to 94.12% of Intrusion Detection System (IDS). Experimental results demonstrate the efficacy of this approach in distinguishing between normal and malicious activities within WSN environments.

Keywords: *Feature selection, Intrusion detection, Light GBM, Quantile recursive, Preprocessing*

1. INTRODUCTION

Sensor nodes in a Wireless Sensor Network (WSN) [1] are small devices which contain sensors, a microprocessor, a wireless transmitter and batteries. Every node that detects a physical event transmits data to Base Station (BS). Data is frequently transmitted from one node to the next until it reaches BS due to the limited communication range of individual nodes (a few tens of meters) and the impossibility to create a direct link between all nodes and BS. WSNs are used in military [2], construction and industrial automation [3], energy management and agriculture [4] along with the applications like ecological [5] and animal monitoring [6].

An Intrusion Detection System (IDS) serves as a crucial component of network security because it helps to stop attacks inside the network. Nodes operating as sensors track only the confined areas. To monitor both local and centralized harmful behavior, ID agents are spread throughout the network. This enables the detection of unavailable network segments. In WSNs, IDS has not used more resources than the requirement to achieve a particular level of detection accuracy. Or else, improved detection accuracy emerges at significant cost because of the restricted resources available in sensor nodes [7].

Sensor nodes of WSNs continuously administers their surroundings as well as send the calculated amount of certain occurrences towards the middle location known as Base Station (BS). As exposed with varied attacks, IDS is recommended to WSN. Unfortunately, majority of these systems make use of the limited resources in sensor nodes, resulting in computational overhead. The resources like memory, CPU, battery, etc., are low for sensor nodes; WSNs consider the development of real-time IDS [8, 9]. Intrusion Detection Systems play the most important role in assuring object security. Therefore, accurate detection of multiple attacks in the network is critical.

However, passive defensive techniques have not provided fully secure solutions for WSNs. This works hold that preventive safety systems [10] have to be installed on board. The use of IDS for active defense construction makes logical [11]. When conventional preventive tactics are ineffective, data-driven approaches found in IDS are utilized in proactive detection of hostile intrusions. Because of the increased traffic amount that a network traverses in real time, IDS finds it harder to assess. Data behavior in WSN is the factor that influences the rate of data processing. As a result, IDS efficiency is increased in WSN [12-14].

High-dimensional data is becoming available for free online. Researchers face considerable struggles while applying Machine Learning (ML) approaches because they are not designed to regulate a wide range of input properties. Without proper data preparation, ML algorithms have not performed to the expected level [15], [16]. Feature selection has evolved into a critical component of ML since it is both regularly used and required data preparation approach. In statistics and ML, it is also known as variable selection, variable subset selection or attribute selection. It is a method for detecting important qualities while removing irrelevant or unnecessary information. This strategy accelerates data mining procedures, improves prediction accuracy and makes forecasts more understandable [17-20].

As the frequency of attacks is increased worldwide, ID is a critical subject for information system security. One of the most serious difficulties with IDS is their overhead, which grow quite large. Analyzing system logs necessitates the operating system retaining knowledge of all the performed activities, which inevitably results in large amounts of data requiring disk space and CPU capabilities. Further, the logs are processed and translated into a usable format before being compared to a database of known misuse and attack patterns to detect potential intrusions [21], [22].

In this paper, Quantile [23] Recursive [24]-Random Outlier Imputing (QR-ROI) [25] algorithm is proposed for data preprocessing. These strategies work together to control data anomalies while maintaining the dataset's consistency and statistical significance. In addition, Wrapper Boruta Algorithm [26] is combined with enhanced Light Gradient Boosting Machine (GBM) [27] for feature selection. In contrast to filter approaches, Wrapper feature selection methods take feature value into account and use a ML algorithm to select the optimal subset for a given model. Wrapper techniques combine feature selection and training procedures for ML algorithms. After the model exploring multiple subsets, algorithm determines the optimal subset of attributes that resulted in the outstanding performance of method [28], [29].

Contribution of this paper: This paper tackles critical concerns such as data abnormalities and feature selection inefficiency, proposes a method for improving ID in WSNs. Quantile Recursive-Random Outlier Imputing approach is proposed for effective data preprocessing and management of missing and anomalous values. To optimize detection performance, Wrapper Boruta Algorithm is enhanced with Light GBM for effective feature selection. The combined strategy significantly improves the predictability and robustness of IDS, as well as the dataset's quality. On a Kaggle dataset wsnds, experimental validation demonstrates the capability of model in contrasting betwixt benign as well as malicious activities.

Motivation of this paper: The sensitivity of WSN is raised to sophisticated security threats, as well as the challenges of handling missing or noisy data and the high dimensionality of features. Existing approaches are either inefficient or unable to generalize over a wide range of intrusion conditions. This study aims to overcome these gaps by using cutting-edge data pre-processing and feature selection approaches for increasing accuracy, efficiency and reliability of IDS in WSN.

State-of-the-art: Modern IDS in WSNs aims in merging cutting-edge ML algorithms with the most effective pre-processing and feature selection procedures. Methods like Light GBM for gradient boosting, Boruta for feature selection and adaptive algorithms for dealing with imbalanced or missing data show significant potential. These strategies reduce processing overhead, increase detection accuracy and provide IDS resilience in dynamic WSN scenarios.

Organization of this paper: In this paper, section 2 discusses about the related works from various authors publications. In section 3, data preprocessing and feature selection methods are clearly explained and in section 4 results and discussions are added based on models performance. Finally, section 5 discusses the conclusion of work.

2. RELATED WORKS

Liu et al. [30] investigated large-scale difficult WSN Intrusion Detection and proposed an IDS based on distributed fuzzy clustering in WSNs and parallel intelligent optimization feature extraction. To improve detection performance, the approach combined the actual requirement for WSN Intrusion Detection with techniques such as feature extraction, intelligent optimization and fuzzy clustering. Simulation experiments further demonstrated the efficacy of proposed strategy from a variety of perspectives.

Wireless Sensor Networks were inherently unsecured and susceptible to a wide range of security threats. To combat such threats, solid security architecture was essential. Although there were several IDS available, their effectiveness had degraded. The suggested effective strategies for feature selection and classification assisted in performance improvement. Thus, Ahmad [31] suggested a Particle Swarm Optimizer (PSO)-based approach for feature selection in WSN Intrusion Detection, selected the optimal subset of features from either the main space or Principle Component Analysis (PCA) space. The author's proposed method was tested against a known benchmark for IDS assessments in NSL KDD dataset. After being confirmed on a Modular Neural Network (MNN), this feature subset based on PSO was compared to another based on Genetic Algorithm (GA).

Aljebreen et al [32] Binary Chimp Optimization Algorithm (BCOA) and ML for Internet of Things Wireless Sensor Network (IoT-WSN) security assignments were provided as the method of accurate ID. Data normalization, feature subset selection based on BCOA, classification using Class-specific Cost Regulation Extreme Learning Machine (CCR-ELM) and parameter tuning related to Sine Cosine Algorithm (SCA) were the strategies presented by the revealed BCOA- Machine Learning

based Intrusion Detection (MLID) methodology for detecting intrusions. Kaggle dataset had been used to explore the experimental results in BCOA-MLID method. Results showed the most significant aspects of procedure with the greatest precision.

Hasan et al. [33] developed Random Forest (RF) model to improve IDS performance by input characterization reduction. In terms of data management and processing performance, fewer features had tended to perform better in real-world scenarios. RF classification with 25 features outperformed RF classification with every 41 features. To complicate matters, RF processing durations for 41 attributes were longer than 25 characteristics. Because of its outstanding performance, RF method had been the subject of extensive research in ID and feature selection.

Liu et al. [34] WSN intelligent IDS was presented using an edge intelligence framework based on evolutionary computing Arithmetic Optimization Algorithm (AOA) and ML's K-Nearest Neighbor (KNN) algorithm. This system primarily detected intrusions when WSN was under a Denial of Service (DoS) attack. Those scientists changed the optimization utilizing the Parallel Lévy (PL) flight technique and parallel approach to boost population communication, hence increasing model accuracy. The proposed PL-AOA approach ensured efficient development of the KNN classifier and performed well in the benchmark function tests.

Modified binary Grey Wolf Optimizer with Support Vector Machine (GWOSVM), had been the improved IDS. It investigated three, five and seven canines to determine the best wolf pack size. Safaldin et al. [35] offered a method for reducing false alarms and characteristics generated by IDS when enhancing ID accuracy and rate in WSN setting. This had resulted in faster processing times for WSN system. Using NSL KDD'99 dataset, the performance of suggested approach was demonstrated and compared against previous solutions. The proposed approaches were evaluated using the parameters such as detection rate, false alarm rate, execution time, feature count and accuracy. The seven-wolf GWOSVM-IDS method outperformed the recommended methodologies and alternatives.

The key hazard regarding WSN was secured, since it is vulnerable to the varied attacks. Fuzzy logic was an updated way for detecting as well as preventing hostile nodes from going into WSN. Fuzzy presented on ID provided recommendations for detecting normal and aberrant nodes, making decision and finishing analysis. Singh et al. [36] presented a fuzzy technique to prevent intrusion. Their proposed strategy to prevent malicious nodes involved three steps: feature selection, membership computation and fuzzy rule applicator.

Brahmam et al. [37] Adaptive Threshold-Based Outlier Detection (ATBOD) paradigm was recently created. This technology reduced errors at node level, enhanced energy efficiency in IoT sensor boards and allowed data collection and defect identification. Their innovative ATBOD algorithm bet existing methods for distinguishing between mistakes and events when possessed a lower false positive rate (ErFPR) and error detection rate (ErDr). Applying ATBOD to real-time data sets with varied degrees of added inaccuracy resulted in astonishing energy savings.

Gebremariam et al. [38] the goal was to develop a hybrid ML-based classification model for enhanced IDS that was used to detect intrusions in WSN. Each sensor node conveyed the current state of its characteristics to the cluster's processing node. After confirmation, the cluster leader transmitted the data to the primary Cluster Head (CH). They suggested hybrid ML models to detect threats after the test and training data was analyzed. The WSN detection and localization accuracy of proposed Intrusion Detection Systems based on Hybrid Machine Learning (IDS-HML) outperformed that of existing systems. The comparison of hypothetical outcomes with previous studies helped in the establishment of believability.

Faris et al. [39] brought a new technique for slowing Differential Evolutions (DE's) repetitive development. Another way to put it was that strived to increase Maturity Extension (ME) in order to achieve the most comprehensive global results. Mutation factor was adjusted using KNN to get the desired mutation without complicating the design process. By addressing initial concerns and encouraging better judgments, the proposed approach improved DE operations' efficiency. The DE-ME approach made it easier to identify the strongest features of NSL-KDD dataset that included 41.

Nguyen et al. [40] used metaheuristic optimization to create IDS adapted specifically for WSN. The authors introduced a novel strategy called Genetic Sacrificial Whale Optimization (GSWO) that balanced whale exploration and exploitation by integrating conditional inherited choice with three-population division technique. Both Whale Optimization Algorithm (WHO) and GA had undergone specific alterations. Catboost classifier was upgraded to distinguish between benign and attack patterns by including GSWO and quantization-based optimization approach. Rigid testing on datasets such as CICIDS 2017, NSL-KDD and specialized WSN datasets revealed that the model was usable in real time.

Samadi Bonab et al. [41] presented a novel Fruit Fly Algorithm and Ant Lion Optimizer (FFA-ALO)-based technique for distinguishing between normal and problematic network traffic in IDS. It was applied to IDS datasets via Wrapper-based feature selection and filtered out less significant characteristics. Three well-known datasets; KDD Cup99, NSL-KDD and UNSW-NB15 were used to implement this innovative classification algorithm. Two portions of the FFA-ALO were investigated to ensure that they met the evaluation criteria. Its first stage performance in terms of the sum of within-cluster distance was evaluated on seven popular datasets. Four classifiers such as SVM, KNN, Naïve Bayes (NB) and Decision Trees (DT) were utilized for analyzing the selected features after using the suggested technique in second step. Performance

metrics included elapsed time, accuracy, specificity and sensitivity.

Table 1: Comparison table of various authors work

Authors/Years	Main Focus	Methodology/ Algorithm	Key Contributions	Dataset	Merits
Vijayanand & Devaraj (2020)	ID in WSN	Whale Optimization Algorithm with Genetic Operators	Novel feature selection method that enhances detection accuracy	NSL-KDD	- Effective feature selection improves detection precision. - Innovative use of hybrid optimization techniques.
Harita & Mohammed (2024)	Malicious flow monitoring and prediction	Improved Swarm Optimizer and Boosted Quantile Estimator	Enhanced IDS with predictive capabilities	Simulated Network Traffic	- Capable of handling dynamic network environments. - Incorporates prediction, enhancing future readiness.
Wu & Qu (2023)	Duty cycle scheduling in WSN	Quartile-Directed Adaptive Genetic Algorithm	Optimized additive Increase/Multiplicative Decrease (AIMD) rule-based duty cycle scheduling to save energy	Simulated Sensor Network	- Improves energy efficiency significantly. - Reduces communication delays in sensor networks.
Subbiah et al. (2022)	ID in WSN	Grid Search RF with Boruta Feature Selection	A robust detection model utilizing feature selection for improved results	CICIDS2017	- Enhanced accuracy through Boruta feature selection. - Grid search ensures optimal model performance.
Bhutta et al. (2024)	Real-time IDS in WiFi networks	LightGBM	Lightweight and efficient real-time IDS	Wi-Fi Network Traffic Dataset	- Lightweight model suitable for resource-constrained systems. -Real-time capabilities for ID.
Prajisha & Vasudevan (2022)	IDS in MQTT-IoT networks	Enhanced Chaotic Salp Swarm Algorithm with Light GBM	Combines chaotic optimization with ML for better ID in IoT	MQTT-IoT Specific Dataset	- Highly effective for IoT-specific attacks. - Balances accuracy and computational cost.
Liu et al. (2021)	Network ID	Adaptive Synthetic Oversampling with Light GBM	Improved detection in imbalanced datasets using oversampling techniques	UNSW-NB15, KDD99	- Addresses class imbalance effectively. - Improves model robustness and detection rate.

2.1 Problem Identification

Wireless Sensor Networks provide significant challenges due to resource constraints such as limited processing power and energy along with the security issues like data leaks and unauthorized access. Missing or aberrant data as well as the high complexity of datasets limit efficient IDS, compromising in model accuracy and efficiency. To provide reliable and effective ID in WSN systems, powerful data preprocessing accompanied with feature selection approaches are required.

3. MATERIALS AND METHODS

In this paper, Quantile Recursive-Random Outlier Imputing algorithm is proposed for data preprocessing and Wrapper Boruta with enhanced LightGBM for feature selection. The dataset wsnds is taken from Kaggle datasets. Data pretreatment modifies WSN data by removing outliers, missing values, categorical variables and numerical feature scaling, preparing it for modeling. Feature selection is a technique for improving computational efficiency, reducing overfitting and increasing model performance that selects the most relevant features from wsnds dataset. Figure 1 illustrates the data preprocessing feature long with the feature selection process.

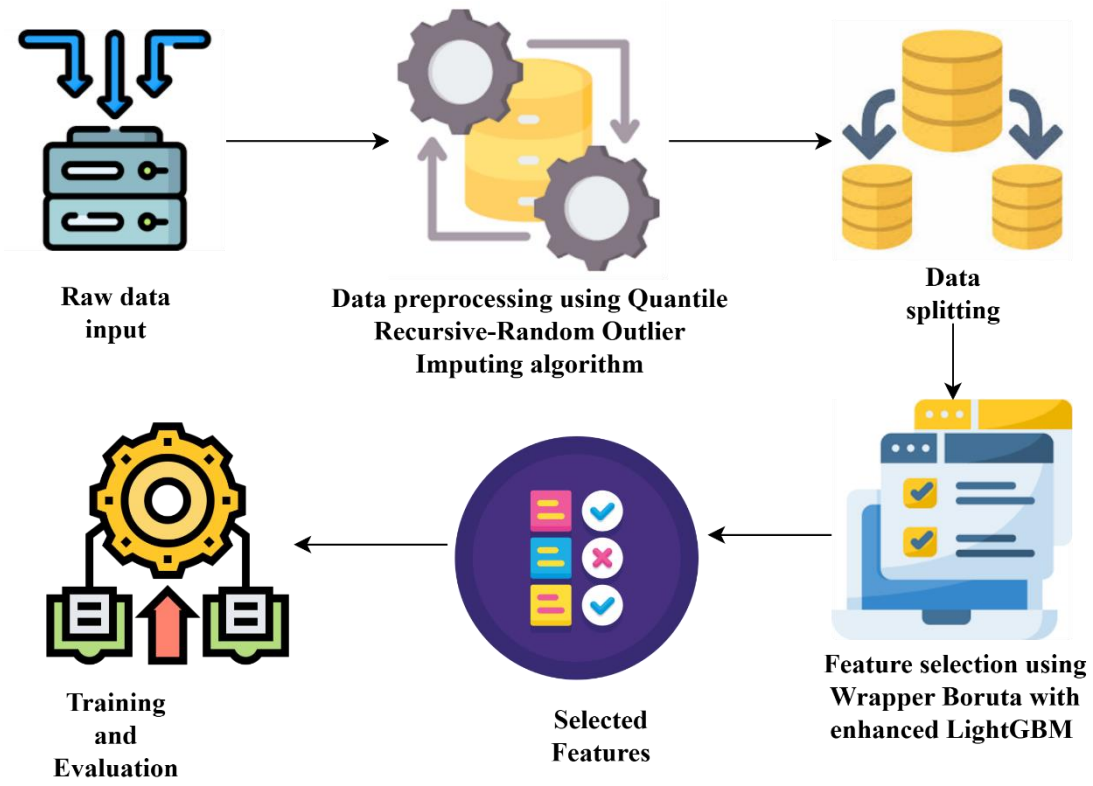

Figure 1: Overall Architecture of preprocessing and feature selection

3.1 Dataset Collection

The wsnds dataset covers environmental parameters like temperature, humidity as well as light intensity using sensor data gathered via WSN. It provides tagged data which is useful for WSN-based applications such as environmental monitoring, energy-efficient networks. More information is accessible from the Kaggle wsnds dataset.

Dataset: - https://www.kaggle.com/datasets/bassamkasasbeh1/wsnds

3.2 DATA PREPROCESSING

A significant data mining activities is pre-processing, which modifies and prepares raw data. Data preparation encompasses a wide range of activities, including feature generation, feature reduction and data purification. Feature reduction includes both extraction and feature selection. Data preparation requires a variety of methodologies for feature extraction, selection and construction. These methods are combined in feature creation followed by feature selection or feature extraction followed by feature selection, depending on the problem at hand.

3.2.1 Quantile method

Statistical divide of data into groups based on percentiles is known as the quantile method. A very common use case with analyzing is distribution of data, outliers and data normalization or scaling on the data before processing it to machine

learning or statistical work. Arellano and Bonhomme [50] presented the Linear Quantile model with sample selection, which is expressed as:

$Y^* = X'\beta(U)$ -------- (1)

$Y = D.\ Y^*$ ---------- (2)

$D = L(V \leq \pi(Z))$ ---------- (3)

When *D* is equal to1, the participants observe the latent result *Y**.The result recorded for those who has not participated is $Y = 0$. The invisible component denoted as *U*, is obtained by scaling the visible components *X*, by the slope parameter β. The instrumental *Z1* and its variables; the invisible variable *V* and the propensity score π have an impact on participation choices. Unobservable *U* and *V* play a significant role in sample selection. Their combined distribution is defined by a copula, $C(u, v, \theta)$; they are uniformly distributed throughout the unit interval without sacrificing generalizability. When the other options have been exhausted, the independence copula serves as a useful paradigm. $G(u, v; \theta) \equiv C(u, v; \theta)/v$ is the only condition under which the copula is true. In terms of estimating and identifying, the most important limitation is this:

$\mathbb{P}(Y^* \leq X'\beta\ \tau\ D=1, Z=z = G(\tau, \pi z; \rho)$ -------- (4)

Formula (4) shows that the conditional distribution of the latent outcome *Y** is influenced by both observed factors $(Z, \pi(z))$ and the dependency structure of unobservable variables (*U, V*) captured by the copula *C*. The function *G* quantifies how the selection process $(\pi(z))$ interacts with the quantile level (τ) and the latent dependencies (ρ) to determine the observed distribution of *Y**.

3.2.2 Recursive

Projects requiring sentiment categorization data typically dedicate 70-80% of their time and resources to data preparation. For better results, utility of recursive data preparation strategy is recommended. The technique's paradigm is built around fundamental data cleaning capabilities. Repeated data cleaning methods improve accuracy and precision. While ML approaches are unable to handle textual data like letters and symbols, they are best suited to numerical data. To convert the input data into numerical values in ML models, the following two conditions are met: predictors that are numerically defined and predictor vectors that have a constant length. Sci-Kit Learn, a Python ML tool has also meet four basic data criteria.

- In individual objects, predictors and target has to be present.
- The above criteria has to be numerical
- Both the criteria has to be numpy arrays
- Their forms could complement one another.

3.2.3 Outlier imputing algorithm

Real-world databases have missing or insufficient data. There are several reasons for missing or incomplete data. These reasons are been described as data collection errors, data entry procedures, wrong measurements, equipment malfunctions, and so on. Missing values in databases lead to variety of challenges throughout the knowledge-seeking process. These shortcomings are summarized as poor data management and analytical challenges. As data is scarce, these missing numbers generate bias in choices. As a result, exact prediction is not possible with this data. Missing values are handled in literature using either data tolerance or data imputation approaches.

3.2.4 Quantile Recursive-Random Outlier Imputing algorithm

Because of their dispersed nature and limited resources, WSNs are vulnerable to several attacks. To handle WSNs, IDS has to preprocess missing, noisy or inconsistent data. Quantile Recursive-Random Outlier Imputation approach, which effectively resolves outliers and missing data, is a novel preprocessing tool designed for maximizing IDS accuracy and reliability in WSNs. It is a preprocessing strategy for dealing with outliers and missing values in datasets that combines statistical quantile techniques with recursive imputation. It ensures randomization to retain data variety and reduce overfitting by regularly replacing statistically robust estimates based on quantile values for outliers or missing data points. The cyclical nature of the technique ensures delayed convergence to a stable and optimum setup. Dataset is divided into quartiles and for each feature, find the first (*I1*) and third (*I3*). Quantiles help determine outlier thresholds. Abnormalities are identified using Inter Quartile Range (IQR):

$I = I3 - I1$ ---------- (5)

Outlier boundaries are defined as:

Lower Bound $= I1 - k \times I$ ---------- (6)

Equation (6) defines the lower threshold for detecting outliers. *k* is a scaling factor, controlling the sensitivity of outlier detection. Any data point below this bound is considered an outlier.

Upper Bound = $I3+k\times I$ --------- (7)

k is the same scaling factor as in the lower bound. Any data point above this bound is also considered an outlier. Together, the lower and upper bounds create a range for acceptable data values, helps to identify anomalies. Replace outliers and missing data with estimates derived from their nearest neighbors. Use the median of neighbors as the imputed value to minimize skewness:

Xi = Median (Neighbors within bounds) ------- (8)

Xi is a Neighbours within bounds. The median is used instead of the mean to minimize the impact of outliers and maintain robustness. To maintain data diversity and prevent overfitting in models, add small random noise to the imputed value:

$x'=x+\epsilon, \epsilon \sim N(0, \sigma)$ ---------- (9)

x is the imputed value. ϵ is a small random noise, drawn from a normal distribution N(0,σ) with mean 0 and standard deviation σ. Adding noise prevents models from being overly reliant on patterns in the training data, improving generalization.

Continue recursive imputation until changes between iterations are negligible:

$\Delta = \frac{1}{N}\sum_{i=1}^{N} |x_i^{t+1} - x_i^t| < \epsilon$---------- (10)

x_i^t is a value of the *i*-th data point at iteration *t*. *N* is a total number of data points. Δ *is a* Average absolute change between consecutive iterations. ϵ is a small tolerance threshold for convergence. Iterative imputation refines the estimated values until the data stabilizes.

Normalize the preprocessed data to a uniform scale to enhance compatibility with IDS. Figure 2 represent the data processing and analysis pipeline. Data collection, preprocessing, modeling, result storage, data analysis, report generation and visualization are the steps depicted in this figure.

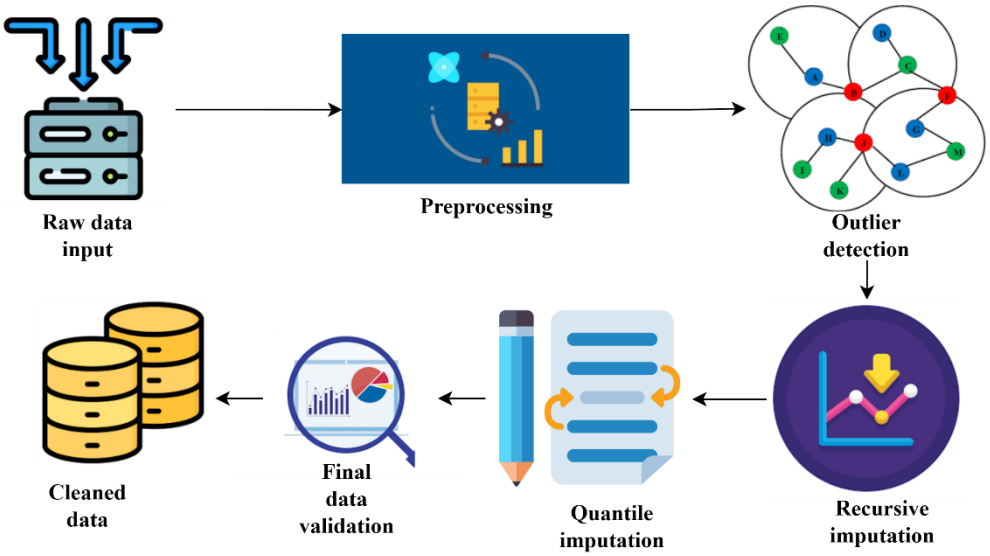

Figure 2: Data preprocessing process

Algorithm 1: Quantile Recursive-Random Outlier Imputing algorithm

Input: Dataset X = {x1, x2, ..., xN}, Threshold k, Noise σ, Convergence ϵ

1. Compute IQR: I = I3 - I1

Lower Bound = I1 - k × I

Upper Bound = I3 + k × I

2. Repeat until convergence:

a. Identify outliers: For each xi in X: If xi < Lower Bound or xi > Upper Bound: Mark xi as outlier

b. Replace outliers: For each marked xi: Neighbors = {xj | Lower Bound ≤ xj ≤ Upper Bound} xi = Median(Neighbors)

c. Add random noise: xi = xi + ε, where ε ~ N (0, σ) d. Check convergence: $\Delta = \frac{1}{N}\sum_{i=1}^{N} |x_i^{t+1} - x_i^t| < \epsilon$

Break

3. Normalize dataset: X (clean) = Normalize(X)

Output: Cleaned and normalized dataset X (clean)

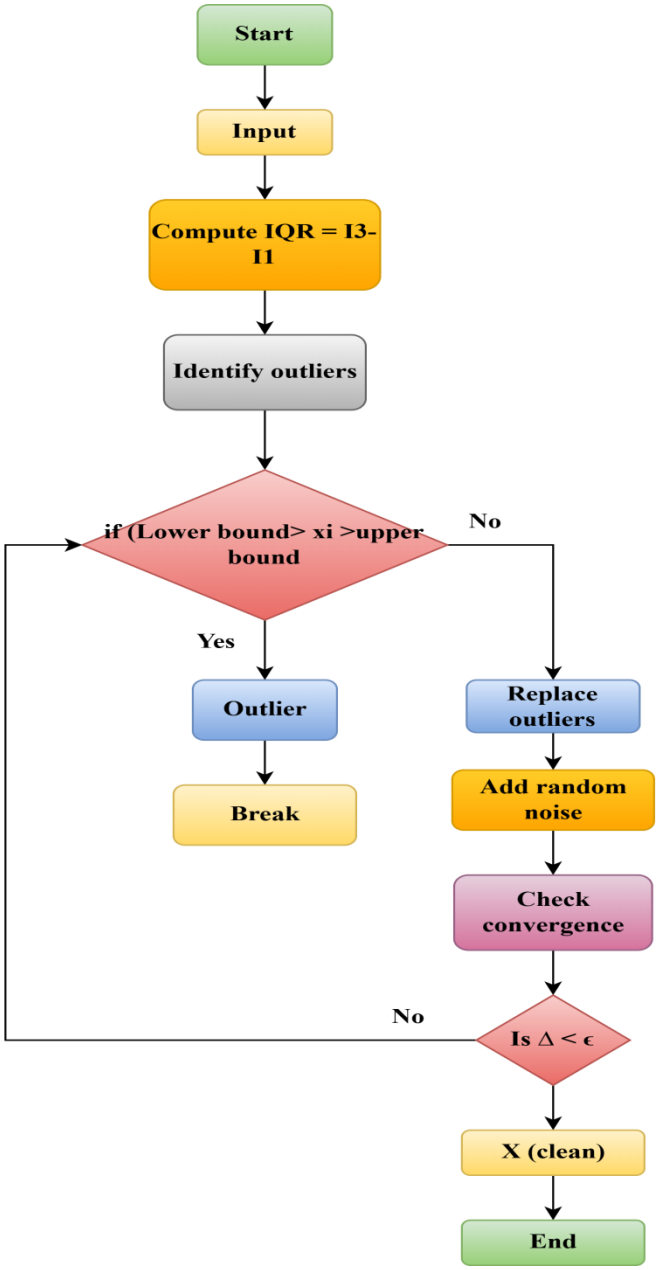

Figure 3: Flowchart for QR-ROI algorithm

Algorithm 1 and figure 3 illustrates a Quantile Recursive Random Outlier Imputing algorithm is aimed at the outliers in datasets, which discovers and replaces the outliers with its neighbors values in an iterative process. It starts with calculation of InterQuartile Range (IQR) to identify the upper and lower bounds in order to identify the outliers. Those values that are beyond these bounds are known as outliers. Outliers once identified, are replaced with the median of the neighboring values that are within bounds. With this step, I make sure that the outliers are replaced by reasonable values. Furthermore, random noise is added so that the replacement values are not too close to each other, thus we avoid losing variability. It will keep iterating until convergence that is change in the dataset between iterations is less than some specified threshold (ϵ). Lastly, the dataset is normalized to be on similar scale so that all features are comparable and are ready for additional analysis or modeling.

3.3 FEATURE SELECTION

3.3.1 Overview of Feature Selection

Feature selection (FS) aims to maximize the feature space, which is defined as an n-dimensional space with points corresponding to each sample (Di Mauro et al [18]). Working with extensive feature spaces make data analysis to begin with representative features, more resource and time intensive. Thus, it is critical to create proper FS approaches in order to remove features that are redundant, heavily reliant on other qualities and irrelevant, which have not contributed to the construction of an ideal feature subset. Feature selection is one way to evaluate feature extraction approaches. The second one provides numerous methods for generating a new feature space from the preceding one, avoiding the dreadful disease of dimensionality. These include plotted center analysis, linear discriminate analysis and single value decomposition. One has to be mindful that feature extraction possess a high risk of yielding an altered feature space lacking its original physical significance.

3.3.2 Wrapper Method

Wrapper techniques analyze feature subsets using a computationally intensive classification procedure. The features are selected based on the utilized classification strategies. These approaches uncover multiple ways to combine risk variables. Even though it is sluggish, it consistently produces superior feature selection.

3.3.3 Boruta Model

Boruta approach uses feature selection to identify significant variables in datasets. It creates shadow or duplicates features, shuffles their values and ranks their importance in comparison to the original features. Characteristics which consistently outperform the shadow features are considered noteworthy. Typically, this technique assesses feature relevance using Random Forests (RF). This approach involves the following steps:

- It starts by making duplicates and altering the value order in each column. It is called as shadow characteristics. It uses a RF method trained on dataset to estimate the importance of Mean Decrease Accuracy and Mean Decrease Impurity.
- Once that is done, this method prioritizes the actual qualities based on their significance.
- This is only true when feature's Z-score surpasses the highest Z-score possible for its shadow feature.
- At each cycle, this approach compares their Z-scores to determine whether the shuffled copies outperform the original features. If yes, classify the features as major.

3.3.4 Light GBM

The integration of many weak base learners helps the Gradient Boosting Machine (GBM) in reaching the desired classification result as an iterative ensemble model. To train Decision Trees (DT), GBM fits the negative gradient, also known as residuals. One technique to define GBM model *f(x)* is to combine several DT.

$f(x) = \sum_{m=1}^{M} \gamma_m D(x; \theta_m)$--------- (11)

D (x; θ_m*)* represents the learning rate (γ_m), tree counts model (M) and the tree parameter (θ_m). This iterative process allows the GBM to combine the strengths of individual weak learners to make accurate predictions. Lowering the loss function *L* as a function of tree parameters, θ *m* allows the m^{th} tree to learn and predict.

$\theta_m = \arg\min \theta_m \sum_{i=1}^{N} l(y_i, f_{m-1}(x_i) + \gamma_m D(x_i; \theta_m)$ --------- *(12)*

The training sample count is *N*, and the target variable isy_i. Forecast for the former tree is$f_{m-1}(x_i)$. This equation describes the optimization process for training the m-th decision tree. The goal is to minimize the loss function $l(y_i, f_{m-1}(x_i)$ by finding the optimal parameters $\theta_m$for the m-th tree. Gradient descent uses tree parameters to compute the gradient of loss function, which is then used to complete the optimization. Overfitting issues exist in classic Gradient Boosting Decision Trees (GBDT) and their processing complexity is mostly due to the design. It requires looking at each data point to determine a feature, while using as a split point in maximizing the gathered information. This is a time-consuming technique.

3.3.5 Wrapper Boruta Algorithm, with enhanced LightGBM

Wrapper Boruta Algorithm, a robust feature selection approach, identifies relevant features by comparing their relevance features to randomized shadow features. This technique ensures only the selection of qualities with real predictive potential. Boruta investigates the relevance of each feature by repeatedly training the ML method like Light GBM in dataset that includes shadow properties. While those consistently fail are discarded, characteristics which are significantly superior to best shadow feature are deemed desirable.

Light GBM enhances the Boruta approach by using efficient tree-based learning for high-dimensional and large-scale datasets. It improves computing efficiency and model accuracy by using histogram-based Decision Tree learning and leaf-

wise evolution. While using with Boruta, Light GBM speeds feature evaluation in maintaining robustness.

The revised Light GBM technique prioritizes important features via hyperparameter tuning and feature weighting. It also supports parallel processing, which reduces Boruta's operational costs. This combination reduces overfitting and improves model interpretability by enabling precise feature selection. This method is suitable for large and complex datasets since it improves model accuracy and processing efficiency by retaining just required attributes.

Imputation techniques like mean, median for numerical data and mode are used for categorical data.

Xi = Median *(*Neighbors*(*x*))* for missing values -------- (13)

This imputation enables missing values to be treated statistically robustly so as to remove any biases brought by the missing data.

$X' = \frac{x - \min(x)}{\max(x) - \min(x)}$ ---------- (14)

This formula normalizes feature values to a range of [0, 1]. x is a original feature value. This transformation allows the model to gracefully deal with features of different scales, and facilitates numerical stability during training.

Generate random shadow features to serve as a baseline for comparison. For each feature F Fi, create a shadow feature Si by permuting the values of Fi.

Si = Random Permute (Fi) ---------- (15)

To evaluate the importance of a feature Fi, a shadow feature Si is created by randomly permuting Fi 's values. Feature importance is based on the shadow feature. A feature is considered significant if its shadow counterpart has lower importance compared to the real feature. Train the Light GBM model with the dataset containing both original and shadow features. It uses gradient-boosted trees to compute the importance based on reduction in error:

Li = Splits in Fi·ΔLoss -------- (16)

Splits in Fi are the number of splits in the decision trees where Fi is used. ΔLoss is the reduction in error or loss achieved by splitting on Fi. These measures how much a feature contributes to reducing prediction errors in the model.

Importance scores for all features (original and shadow) are extracted from the trained Light GBM model. Identify important and unimportant features by comparing their features with shadow features.

Score (Fi)) > max(Score(Si)), ∀ i ∈ shadow features --------- (17)

A feature this is considered important only if its score is higher than the average score of all of its shadow features. Si this ensures that with respect to random noise, the feature has significantly greater predictive power.

Score (Fi) < min(Score(Si)),∀ i ∈ shadow features --------- (18)

A feature Fi If the importance score of such shadow features is lower than the lower setpoint of its corresponding shadow features, then it is considered unimportant, and therefore less relevant (than its shadow features). Si the model probably doesn't care about these features and can ignore them.

Convergence: Δ Score <ϵ ---------- (19)

Features are refined iteratively until the importance scores converge. Refine the feature set by iteratively removing unimportant features. Enhanced Light GBM paired with the Wrapper Boruta Algorithm provides a powerful solution for feature selection in difficult data sets. This strategy increases model performance by using Light GBM's ability to manage large volumes of data and extending Boruta's iterative feature selection technique, resulting in decreased computational overhead. The process illustrated in figure 4 is feature selection, in which data preprocessing, selection of significant features, method training are carried out along with the fine- training and evaluation of the model.

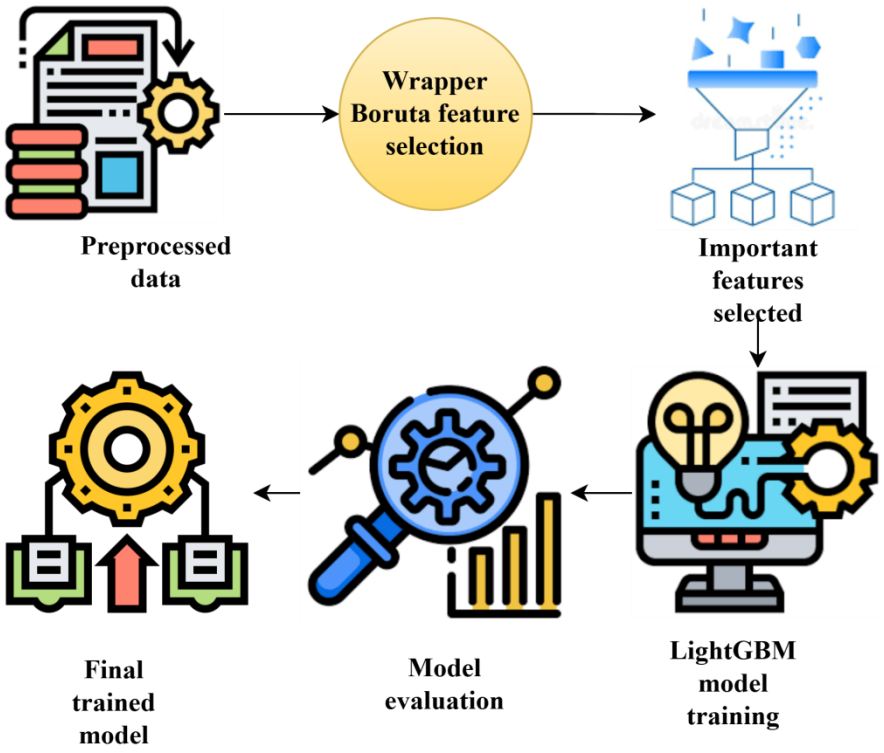

Figure 4: Feature selection process

Algorithm 2: Wrapper Boruta Algorithm, with enhanced Light GBM

Input: The dataset is trained using high-dimensional characteristics *xi* and *yi* in conjugation with training data. In every iteration,

P % indicates the retained features in percentage.

Θ refers to the loss threshold

k represents the low feature count

Initialization: Performance of RMSE for all features in validation set R0; iteration count t = 0; features selected Feature = x_{all}

Steps:

t = t + 1

The best feature combination *best = Feature*.

Light GBM method is trained on the current feature set.

By estimating the usage of data split in every feature, feature importance is calculated along with the feature extraction.

Calculate informative gain L (f) of feature f using variance as the splitting criterion

$L(f)=\sum_{c\in L}|c|\ (Var(base)-\frac{Var(c)}{|c|}$

Where *Var* is the variance of target Y and C is the set of child nodes after splitting on f

Compute the feature importance I *(f)* for each feature f as the number of splits involving f.

Sort the features in descending order of *I (f)*

For acquiring the result *R*, RMSE is utilized for assessing the performance in validation set.

Apply the formula to find the most important characteristic from the top p percent.

Feature = tp, where *tp* represents the top p percent feature count.

We need new training sets.

If either *Len(tp) < k or R - R0 > θ*

Output: Aggregation of best optimal features.

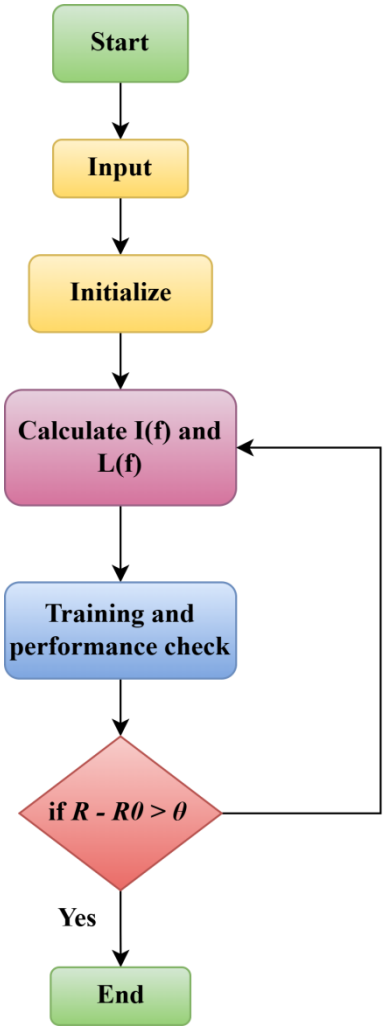

Figure 5: Flowchart of Wrapper Boruta Algorithm, with enhanced Light GBM

Algorithm 2 and figure 5 illustrates Wrapper Boruta with enhanced Light GBM, a method that attempts to identify the most informative features in high dimension data. We start calculating a performance metric, say RMSE, for all features using a validation set. Then it iteratively picks the best feature combinations and training Light GBM model on this subsets. The contribution to data splits is used for measuring feature importance, and only the most relevant part of feature variance is conserved during each iteration. The algorithm is terminated when the performance (RMSE after subtraction with R0) increases less than certain value (e.g. performance reduction > threshold Θ), or when the number of chosen features is lower than a given amount k. It gives a set of the optimal features that contribute maximum to model performance.

4. RESULTS AND DISCUSSIONS

In this paper, python is used for implementation. This paper proposes IDS for WSN to improve performance using data preprocessing and feature selection. Indeed, QR-ROI naturally imputes missing values as well as outliers by handling handles the similar data points. It is combined with enhanced Light GBM by using Wrapper Boruta for recognizing the related features and performs dimensional reduction as well as increases classification accuracy. Based on experimental data, it outperforms the existing methods in terms of recall, F1-score and precision, even though the systematic value is lower. Following feature selection and data preparation, the model obtains an accuracy of 94.12%. Integration of these techniques allows the creation of system that maintains robust IDS with efficiency and scalability for WSN real world deployments.

Wireless Sensor Network pattern of functioning has been captured in dataset with 19 attributes focusing on the different aspects of network. Here, each record corresponds to node or activity within the network. These include id, Time and Is_CH, who CH, Dist_To_CH, ADV_S/SCH_R, JOIN_S/SCH_R, DATA_S/Sent To_BS. ; Very interestingly, while Rank

corresponds to node ranking where node 1 has the highest rank, dist_CH_To_BS points towards the distance between Clusters Heads (CH) to BS trans_Send Code seems to be the particular code of messages transmission and Expaned Energy depicts energy consumption. Attack type column classifies the attacks; its existence indicates the state of protection in a certain network. The dataset covers a rich set of parameters in network traffic flows, energy consumption and possible intrusions.

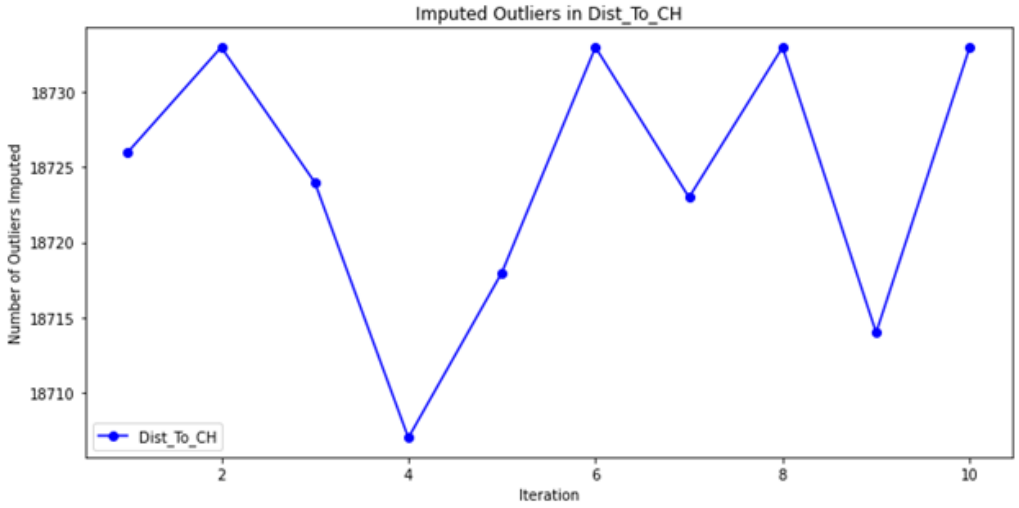

Figure 6: Imputed outliers in Dist_To_CH flowchart

Over 10 iterations, the figure 6 depicts the count of outlier imputed using "Dist_To_CH" feature. With the goal of improving data consistency and stability, the versions demonstrate the alteration and imputation of outliers by QR-ROI. Recurring changes emphasis on the iterative improvement process.

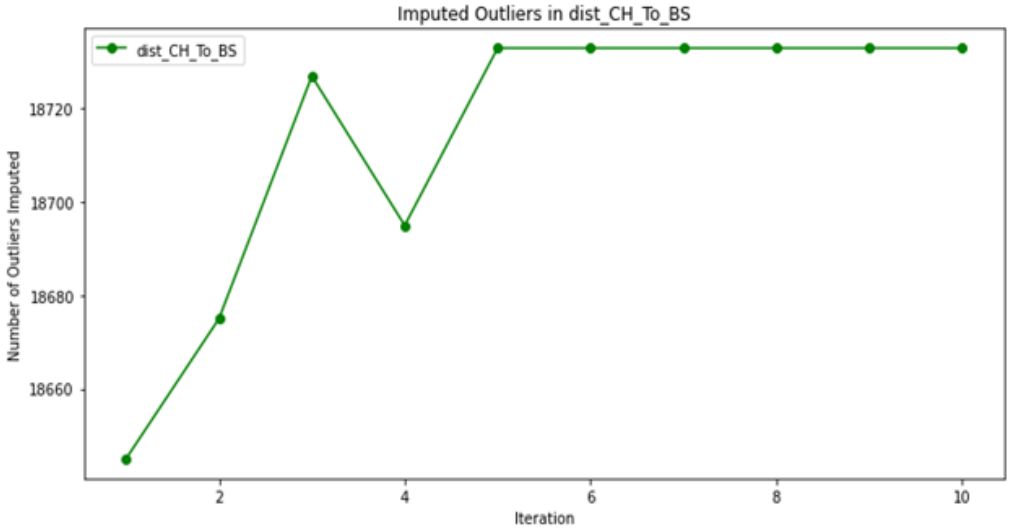

Figure 7: Imputed outliers in Dist_CH_To_BS

Across 10 iterations, the figure 7 depicts the number of outliers imputed in the "dist_CH_To_BS" feature. Outlier imputations first grow and settle after the fourth iteration. This demonstrates that QR-ROI approach converges, allowing for consistent data refinement and stability in subsequent rounds.

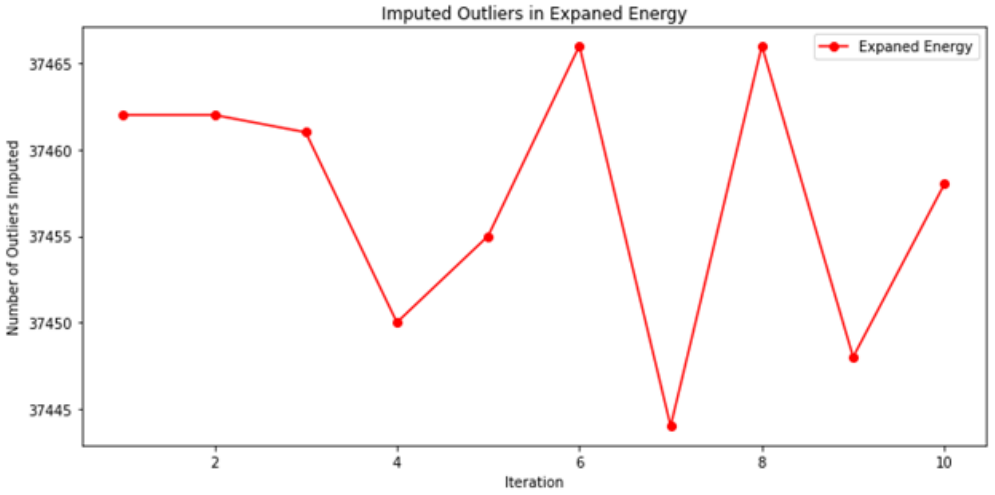

Figure 8: Imputed outliers in expected energy chart

In Figure 8, the number of outliers included in the "Expanded Energy" feature is being imputed over 10 iterations. Fluctuations are the indication of QR-ROI algorithm in iteratively correcting outliers and stabilizing the data quality. The dynamic nature of imputation is highlighted with the periodic variations.

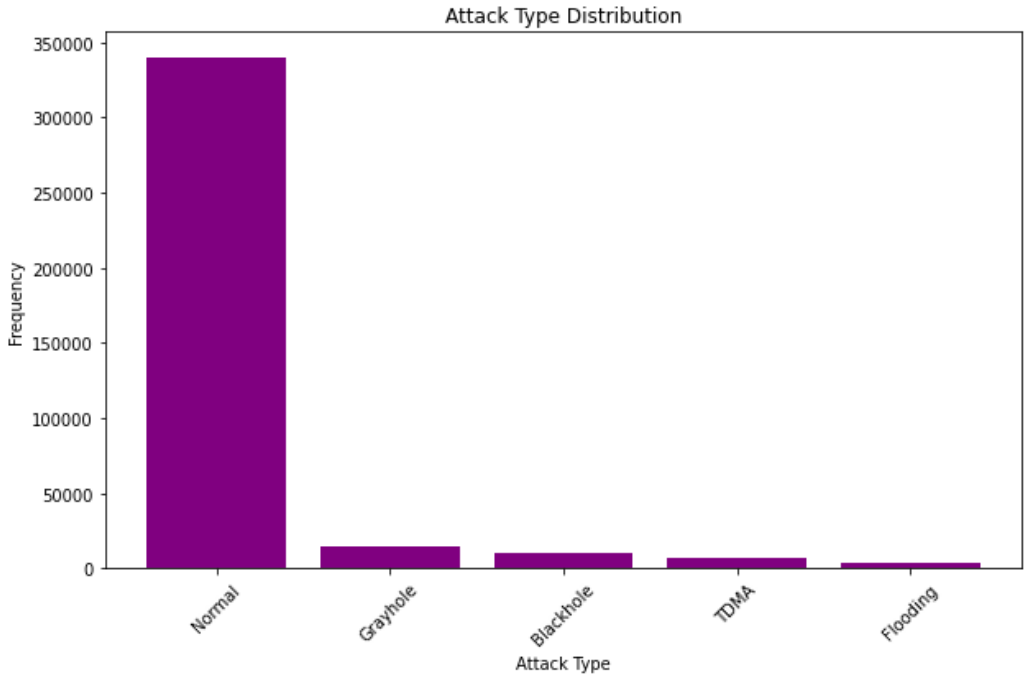

Figure 9: Attack type distribution chart

Distribution of attack types in the dataset is shown in the figure 9. The majority of these packets are observed in the Normal category, where other attack types such as Grayhole, Blackhole, Time Division Multiple Access (TDMA) and Flooding have the lowest frequency. In case of narrowing, this explains IDS requirement of specialized techniques to deal with skewed datasets.

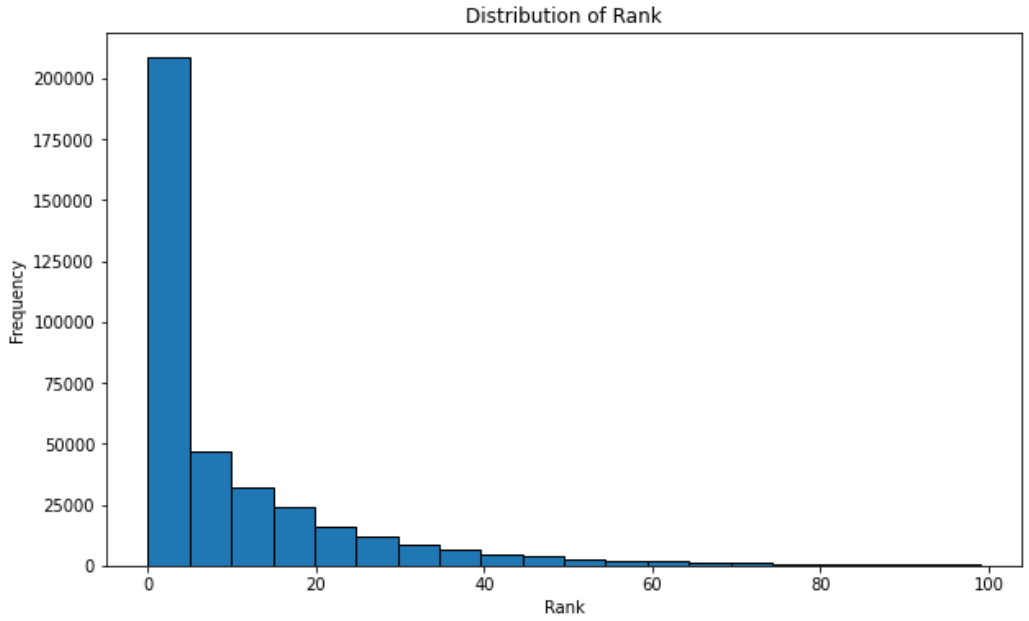

Figure 10: Distribution of rank chart

Histogram of the rank values is shown in figure 10; most of the values are near zero, drops very sharply with the rank. Thus, it is highly skewed with lower ranks occurring often. Hence, there are few most important features or entities overwhelming the data set.

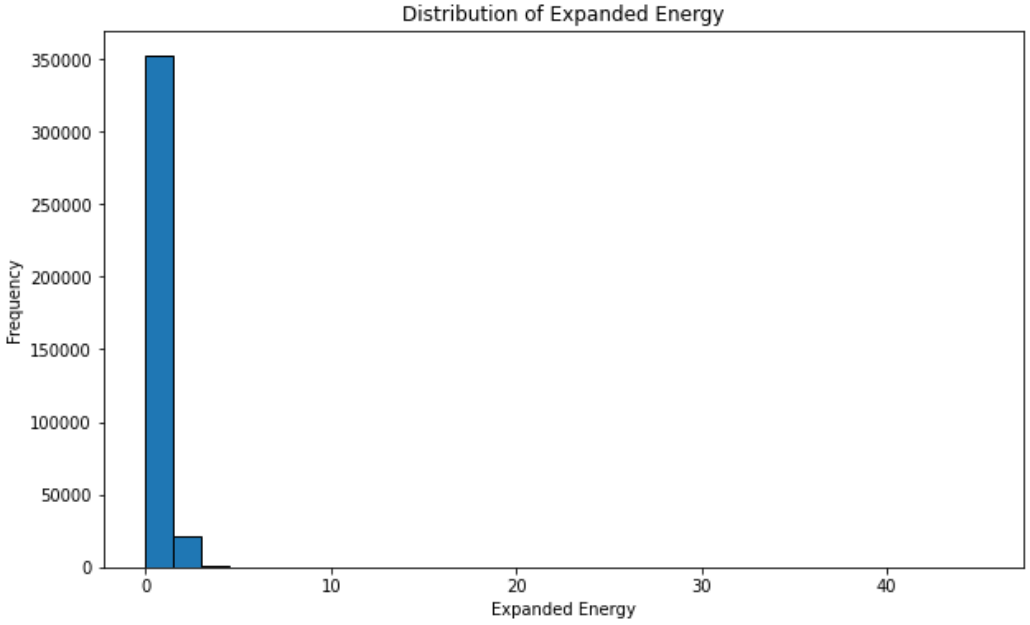

Figure 11: Distribution of energy comparison histogram chart

This histogram figure 11 expresses a distribution of Expanded Energy", most of which are near zero. It reveals that there are high frequencies of low energy values and long process towards higher energy levels, which implies our heavily right skewed data.

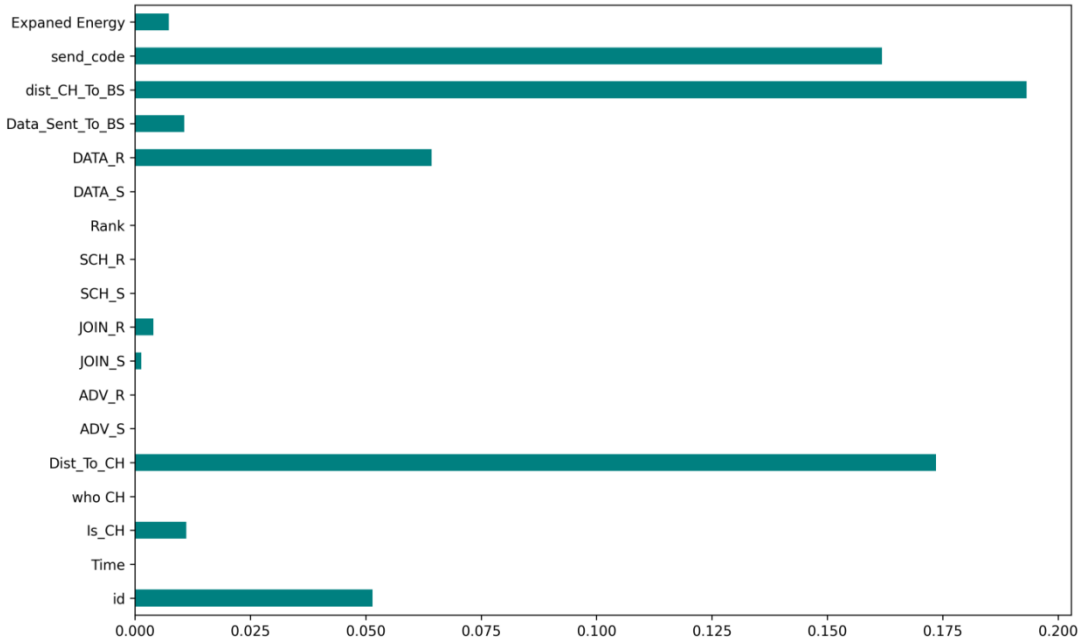

Figure 12: Pattern of function with 19 attributes

Figure 12 represented as bar chart shows which feature is important or the contribution with 19 attributes. Two of which having the highest values in this bar chart is dist_To_CH and dist_CH_To_B. Features such as "DATA_R" and "send_code" also play an important role, whereas the other features including "JOIN_R" and "ADV_S" contribute little.

4.1 Performance metrics

4.1.1 Accuracy

In predictive modeling, accuracy is the measure of how close the model's projections are to real-world outcomes. Making predictions and judgments in a variety of circumstances relies on the model's reliability and accuracy. Thus, these characteristics are assessed. T is true; F is false; P is positive; N is negative.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \text{ ------- } (20)$$

4.1.2 Precision

In predictive modeling, accuracy is the proportion of total expected positive observations to correctly forecast positive observations. It displays the model's effectiveness in lowering false positives, ensuring the genuine accuracy and reliability of the positive predictions it generates like the qualities necessary for decision-making. As a result, error is reduced in many other domains.

$$Precision = \frac{TP}{TP+FP} \text{ ------ } (21)$$

4.1.3 Recall

Recall in predictive modeling is the fraction of real positive instances the model properly detected. In sectors like medical diagnosis or fraud detection, identifying the positives is critical since it shows the efficiency of model in detecting the relevant instances of particular class.

$$Recall = \frac{TP}{TP+FN} \text{------ } (22)$$

4.1.4 F-measure

It determines the average of recall and accuracy, which is a strong all-around measurement of effective model performance essential to prevent both false positives and false negatives.

$$F-measure = 2 \times \frac{Precision \times recall}{precision+recall} \text{------- } (23)$$

Table 2: Accuracy comparison table before and after preprocessing

	Algorithms	Accuracy
Before pre-processing	KNN [51]	93.19
	Quantile [43]	93.23
	Recursive [24]	93.56
	Random Outlier Imputing [37]	93.82
After pre-processing	KNN [51]	93.21
	Quantile [43]	93.40
	Recursive [24]	93.71
	Random Outlier Imputing [37]	93.86
	QR-ROI	93.92

Table 2 compares the accuracy of algorithms such as KNN, Quantile, Recursive, Random Outlier Imputing and the proposed QR-ROI method. Before preprocessing, accuracy of algorithms is less compared after preprocessing. Compared to other methods after preprocessing, the proposed method outperforms with 93.92 % accuracy.

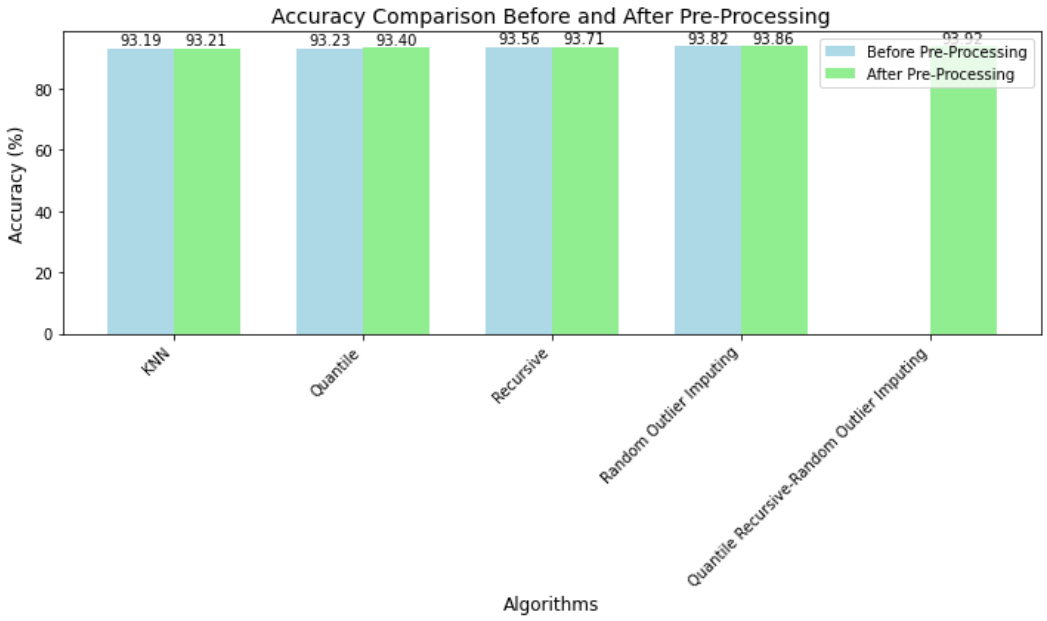

Figure 13: Accuracy comparison before and after preprocessing

Figure 13 shows the accuracy comparison chart of various algorithms before and after preprocessing. The proposed QR-ROI outperforms after preprocessing. X-axis depicts the existing and proposed methods, whereas the accuracy values are represented on Y-axis in percentage.

Table 3: Before and after feature selection comparison table

	Algorithms	Accuracy	Precision	Recall	F-measure
	RF [33]	93.00	92.89	92.98	92.68

Before Feature Selection	Wrapper [41]	93.45	93.11	93.23	93.00
	Boruta [26]	93.67	93.24	93.46	93.10
	Light GBM [46]	93.96	93.67	93.85	93.42
After Feature Selection	RF [33]	92.95	92.59	92.87	92.32
	Wrapper [41]	93.01	92.84	92.96	92.76
	Boruta [26]	93.25	93.05	93.11	93.00
	Light GBM [46]	93.94	93.75	93.82	93.58
	Wrapper Boruta Algorithm, with enhanced Light GBM	94.02	93.87	93.94	93.76

Table 3 illustrates the performance metrics of algorithms like RF, Wrapper, Boruta, Light GBM and proposed Wrapper Boruta with enhanced Light GBM before and after feature selection. The proposed algorithm outperforms with 94.02 % after feature selection.

Figure 14 and figure 15 shows the comparison of accuracy as well as the precision values of various algorithms before and after feature selection. Proposed Wrapper Boruta with enhanced Light GBM outperforms than the other algorithms. The utilized algorithms and its percentage of the metrics are indicated on X—axis and Y- axis.

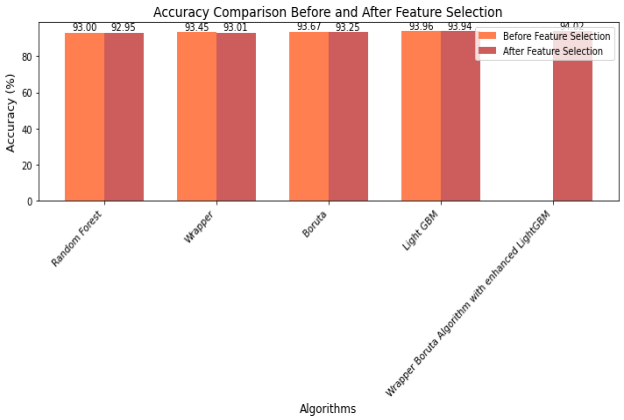

Figure 14: Before and after feature selection accuracy comparison

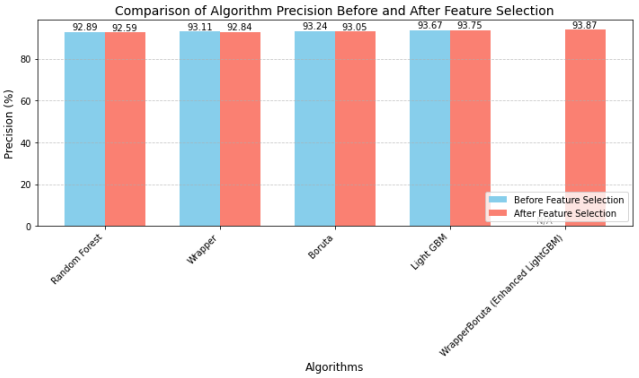

Figure 15: Before and after feature selection Precision comparison

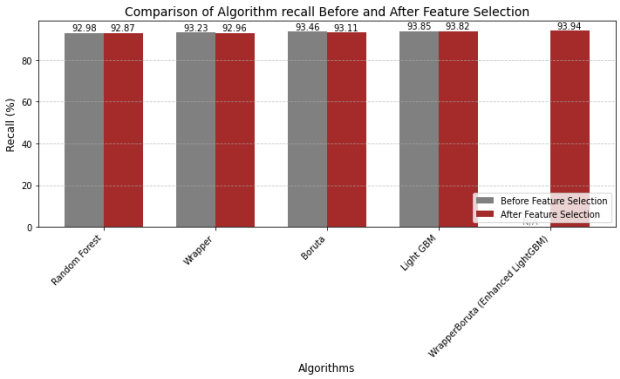

Figure 16: Before and after feature selection recall comparison

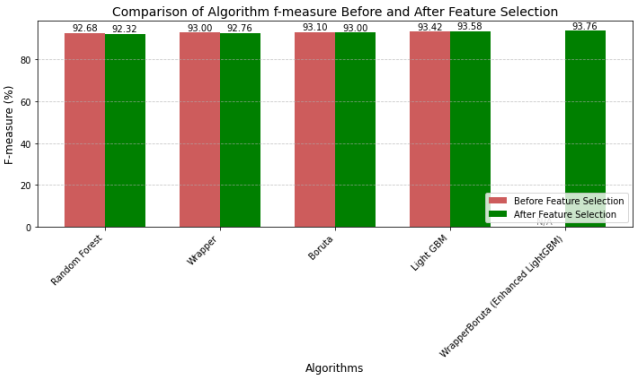

Figure 17: Before and after feature selection F-measure comparison

Recall values are displayed in figure 16, whereas figure 17 shows the f-measure values of various algorithms before and after feature selection. Compared to other algorithms, the proposed Wrapper Boruta with enhanced Light GBM has outperformed. The inscription on X-axis and Y-axis points out the exploited algorithms and metrics percentage.

Table 4: After preprocessing and feature selection comparison table

	Algorithms	Accuracy	Precision	Recall	F-measure
After Pre-processing and Feature Selection	KNN [51]	92.00	91.78	91.93	91.75
	Quantile [43]	92.13	91.99	92.02	91.91
	Recursive [24]	92.51	92.20	92.36	92.13
	Random Outlier Imputing [37]	92.79	92.46	92.64	92.31
	RF [33]	93.01	92.91	92.98	92.78
	Wrapper [41]	93.18	93.00	93.04	92.97
	Boruta [26]	93.56	93.19	93.32	93.05
	Light GBM [46]	93.97	93.45	93.78	93.39
	QR-ROI	94.10	93.89	93.95	93.87
	Wrapper Boruta Algorithm, with enhanced Light GBM	94.12	94.00	94.04	93.98

Table 4 illustrates the comparison of metrics among the existing and proposed algorithms. After preprocessing and feature selection, the proposed Wrapper Boruta with enhanced Light GBM outperforms with 94.12% accuracy.

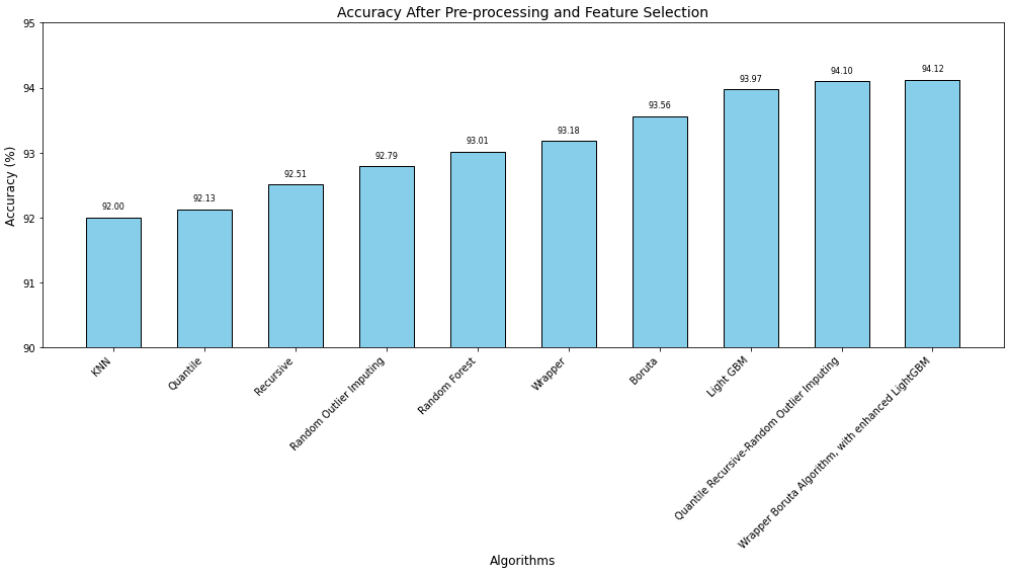

Figure 18: After preprocessing and feature selection accuracy

Figure 18 shows the preprocessing and feature selection accuracy of various algorithms. Compared to other methods, the proposed method performs well. In this chart, X and Y-axis displays the algorithms used in this research and accuracy values.

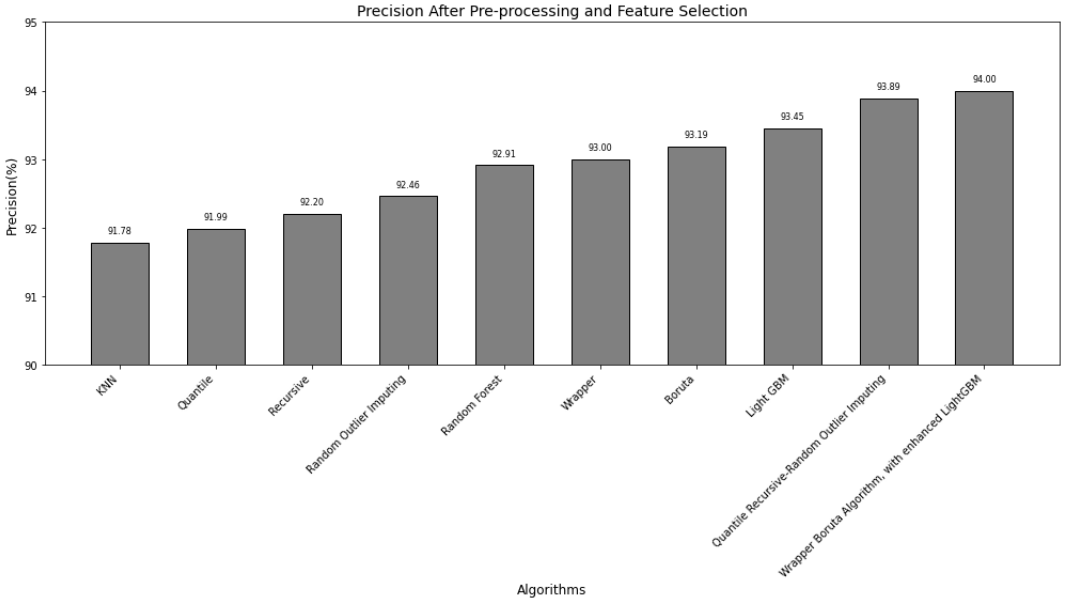

Figure 19: Precision values after preprocessing and feature selection

Figure 19 shows the precision comparison of various algorithms after preprocessing and feature selection process. The proposed method performs well compared to other methods. While the X-axis represents different algorithms, percentage of precision values is pointed on the Y-axis.

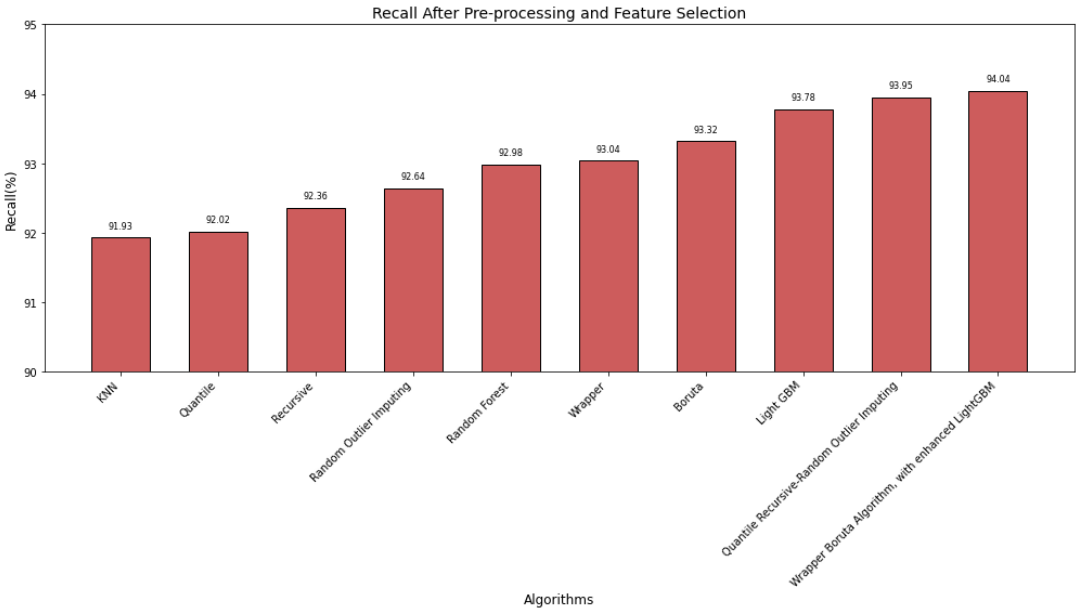

Figure 20: After preprocessing and feature selection recall

Figure 20 shows the recall comparison of various algorithms after preprocessing and feature selection processes. The proposed method performs well compared to other methods. The x-axis shows various algorithms, while the y-axis shows the recall values represented in percentages.

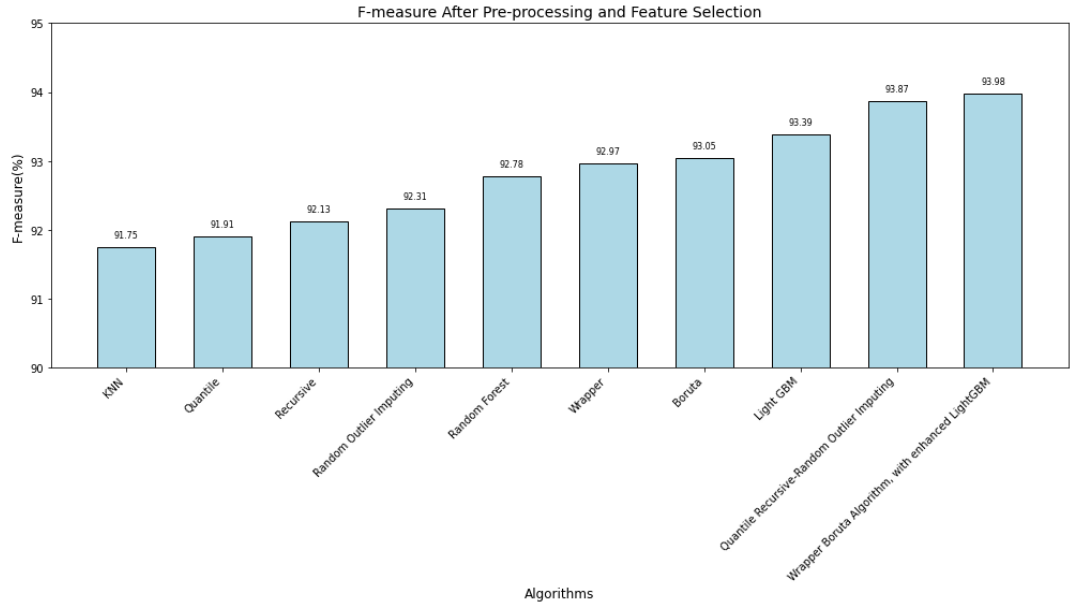

Figure 21: After preprocessing and feature selection f-measure

Figure 21 shows the f-measure comparison of various algorithms after preprocessing and feature selection processes. The proposed method performs well compared to other methods. In this chart, x-axis shows the various algorithms and y-axis shows the f-measure values in %.

5. CONCLUSION

This paper proposes an effective solution for ID in WSN with data preprocessing and feature selection. By performing QR-ROI, it assesses the effect of missing values and outliers to guarantee the dataset up to analytical standard. Preprocessed data is consistent for further analysis. The proposed algorithm Wrapper Boruta with Enhanced Light GBM is designed for feature selection, increase the methods efficiency regarding the choice of appropriate features among the entire characteristic that potentially affects the outcome. The application of Boruta helps to filter only crucial characteristics and decrease

dimensionality to avoid obtaining an overfitting model. Light GBM extended with improved hyperparameter tuning, improves classification performance and decreases computational cost. After preprocessing and feature selection, this model outperforms with 94.12% accuracy. Along with Wrapper Boruta with enhanced Light GBM, QR-ROI helps in the development of better features for increasing IDS efficiency and reliability within WSNs. Future advances focus on integrating Deep Learning models with adaptive feature selection approaches to improve scalability and detection accuracy. Real-time implementation on WSN nodes with restricted resources has been researched to enhance realistic deployment.

REFERENCES

[1] Almomani, I., Al-Kasasbeh, B., & Al-Akhras, M. (2016). WSN-DS: a dataset for intrusion detection systems in wireless sensor networks. *Journal of Sensors*, *2016*(1), 4731953.

[2] Ghosh, K., Neogy, S., Das, P. K., & Mehta, M. (2018). Intrusion detection at international borders and large military barracks with multi-sink wireless sensor networks: An energy efficient solution. *Wireless Personal Communications*, *98*, 1083-1101.

[3] Bagwari, A., Logeshwaran, J., Usha, K., Kannadasan, R., Alsharif, M. H., Uthansakul, P., & Uthansakul, M. (2023). An Enhanced Energy Optimization Model for Industrial Wireless Sensor Networks Using Machine Learning. *IEEE Access*.

[4] Rao, A. K., Nagwanshi, K. K., & Shukla, M. K. (2024). An optimized secure cluster-based routing protocol for IoT-based WSN structures in smart agriculture with blockchain-based integrity checking. *Peer-to-Peer Networking and Applications*, *17*(5), 3159-3181.

[5] Karthikeyan, M., Manimegalai, D., & RajaGopal, K. (2024). Firefly algorithm based WSN-IoT security enhancement with machine learning for intrusion detection. *Scientific Reports*, *14*(1), 231.

[6] Bapat, V., Kale, P., Shinde, V., Deshpande, N., & Shaligram, A. (2017). WSN application for crop protection to divert animal intrusions in the agricultural land. *Computers and electronics in agriculture*, *133*, 88-96.

[7] Stehlik, M., Saleh, A., Stetsko, A., & Matyas, V. (2013, September). Multi-objective optimization of intrusion detection systems for wireless sensor networks. In *Artificial Life Conference Proceedings* (pp. 569-576). One Rogers Street, Cambridge, MA 02142-1209, USA journals-info@ mit. edu: MIT Press.

[8] Elsaid, S. A., & Albatati, N. S. (2020). An optimized collaborative intrusion detection system for wireless sensor networks. *Soft Computing*, *24*(16), 12553-12567.

[9] Stetsko, A., Smolka, T., Matyáš, V., & Stehlík, M. (2014). Improving intrusion detection systems for wireless sensor networks. In *Applied Cryptography and Network Security: 12th International Conference, ACNS 2014, Lausanne, Switzerland, June 10-13, 2014. Proceedings 12* (pp. 343-360). Springer International Publishing.

[10] Batra, I., Verma, S., Kavita, & Alazab, M. (2020). A lightweight IoT-based security framework for inventory automation using wireless sensor network. *International journal of communication systems*, *33*(4), e4228.

[11] Reddy, S. V. V., Manonmani, S. P., Anitha, C., Jaganathan, D., Reena, R., & Suresh, M. (2024, March). MLIDS: Revolutionizing of IoT based Digital Security Mechanism with Machine Learning Assisted Intrusion Detection System. In *2024 International Conference on Automation and Computation (AUTOCOM)* (pp. 277-282). IEEE.

[12] Kangethe, L., Wimmer, H., & Rebman Jr, C. M. (2024). Network Intrusion Detection System with Machine Learning as a Service. *Journal of Information Systems Applied Research*, *17*(3).

[13] Ahmed, O. (2024). Enhancing Intrusion Detection in Wireless Sensor Networks through Machine Learning Techniques and Context Awareness Integration. *International Journal of Mathematics, Statistics, and Computer Science*, *2*, 244-258.

[14] Saleh, H. M., Marouane, H., & Fakhfakh, A. (2024). Stochastic gradient descent intrusions detection for wireless sensor network attack detection system using machine learning. *IEEE Access*.

[15] Gnanaprasanambikai, L., & Munusamy, N. (2018). Data pre-processing and classification for traffic anomaly intrusion detection using NSLKDD dataset. *Cybernetics and Information Technologies*, *18*(3), 111-119.

[16] Durairaj. M and D. Radhika, Pre-Processing Techniques for Enhancing the Performance of Intrusion Detection System. International Journal of Electrical Engineering and Technology, 11(5), 2020, pp. 191-206.

[17] Grando, F., Granville, L. Z., & Lamb, L. C. (2018). Machine learning in network centrality measures: Tutorial and outlook. *ACM Computing Surveys (CSUR)*, *51*(5), 1-32.

[18] Di Mauro, M., Galatro, G., Fortino, G., & Liotta, A. (2021). Supervised feature selection techniques in network intrusion detection: A critical review. *Engineering Applications of Artificial Intelligence*, *101*, 104216.

[19] Alazab, A., Hobbs, M., Abawajy, J., & Alazab, M. (2012, October). Using feature selection for intrusion detection system. In *2012 international symposium on communications and information technologies*

(ISCIT) (pp. 296-301). IEEE.

[20] Abdullah, M., Alshannaq, A., Balamash, A., & Almabdy, S. (2018). Enhanced intrusion detection system using feature selection method and ensemble learning algorithms. *International Journal of Computer Science and Information Security (IJCSIS)*, *16*(2), 48-55.

[21] Aburomman, A. A., & Reaz, M. B. I. (2017). A survey of intrusion detection systems based on ensemble and hybrid classifiers. *Computers & security*, *65*, 135-152.

[22] Sharma, N. V., & Yadav, N. S. (2021). An optimal intrusion detection system using recursive feature elimination and ensemble of classifiers. *Microprocessors and Microsystems*, *85*, 104293.

[23] Pereda-Fernández, S. (2024). Fast Algorithms for Quantile Regression with Selection. *arXiv preprint arXiv:2402.16693*.

[24] Sabitha, E., & Durgadevi, M. (2022). Improving the diabetes Diagnosis prediction rate using data preprocessing, data augmentation and recursive feature elimination method. *International Journal of Advanced Computer Science and Applications*, *13*(9).

[25] Karrar, A. E. (2022). The effect of using data pre-processing by imputations in handling missing values. *Indonesian Journal of Electrical Engineering and Informatics (IJEEI)*, *10*(2), 375-384.

[26] Kursa, M. B., Jankowski, A., & Rudnicki, W. R. (2010). Boruta–a system for feature selection. *Fundamenta Informaticae*, *101*(4), 271-285.

[27] LI, Z. S., YAO, X., LIU, Z. G., & ZHANG, J. C. (2021). Feature selection algorithm based on LightGBM. *Journal of Northeastern University (Natural Science)*, *42*(12), 1688.

[28] Alsaffar, A. M., Nouri-Baygi, M., & Zolbanin, H. M. (2024). Shielding networks: enhancing intrusion detection with hybrid feature selection and stack ensemble learning. *Journal of Big Data*, *11*(1), 133.

[29] Maldonado, J., Riff, M. C., & Neveu, B. (2022). A review of recent approaches on wrapper feature selection for intrusion detection. *Expert Systems with Applications*, *198*, 116822.

[30] Liu, Z., Wei, W., Wang, H., Zhang, Y., Zhang, Q., & Li, S. (2018). Intrusion detection based on parallel intelligent optimization feature extraction and distributed fuzzy clustering in WSNs. *IEEE Access*, *6*, 72201-72211.

[31] Ahmad, I. (2015). Feature selection using particle swarm optimization in intrusion detection. *International Journal of Distributed Sensor Networks*, *11*(10), 806954.

[32] Aljebreen, M., Alohali, M. A., Saeed, M. K., Mohsen, H., Al Duhayyim, M., Abdelmageed, A. A., ... & Abdelbagi, S. (2023). Binary chimp optimization algorithm with ML based intrusion detection for secure IoT-assisted wireless sensor networks. *Sensors*, *23*(8), 4073.

[33] Hasan, M. A. M., Nasser, M., Ahmad, S., & Molla, K. I. (2016). Feature selection for intrusion detection using random forest. *Journal of information security*, *7*(3), 129-140.

[34] Liu, G., Zhao, H., Fan, F., Liu, G., Xu, Q., & Nazir, S. (2022). An enhanced intrusion detection model based on improved kNN in WSNs. *Sensors*, *22*(4), 1407.

[35] Safaldin, M., Otair, M., & Abualigah, L. (2021). Improved binary gray wolf optimizer and SVM for intrusion detection system in wireless sensor networks. *Journal of ambient intelligence and humanized computing*, *12*, 1559-1576.

[36] Singh, N., Virmani, D., & Gao, X. Z. (2020). A fuzzy logic-based method to avert intrusions in wireless sensor networks using WSN-DS dataset. *International Journal of Computational Intelligence and Applications*, *19*(03), 2050018.

[37] Brahmam, M. V., & Gopikrishnan, S. (2024). Adaptive threshold based outlier detection on IoT sensor data: A node-level perspective. *Alexandria Engineering Journal*, *106*, 675-690.

[38] Gebremariam, G. G., Panda, J., & Indu, S. (2023). Design of advanced intrusion detection systems based on hybrid machine learning techniques in hierarchically wireless sensor networks. *Connection Science*, *35*(1), 2246703.

[39] Faris, M., Mahmud, M. N., Salleh, M. F. M., & Alsharaa, B. (2023). A differential evolution-based algorithm with maturity extension for feature selection in intrusion detection system. *Alexandria Engineering Journal*, *81*, 178-192.

[40] Nguyen, T. M., Vo, H. H. P., & Yoo, M. (2024). Enhancing Intrusion Detection in Wireless Sensor Networks Using a GSWO-CatBoost Approach. *Sensors*, *24*(11), 3339.

[41] Samadi Bonab, M., Ghaffari, A., Soleimanian Gharehchopogh, F., & Alemi, P. (2020). A wrapper-based feature

selection for improving performance of intrusion detection systems. *International Journal of Communication Systems*, *33*(12), e4434.

[42] Vijayanand, R., & Devaraj, D. (2020). A novel feature selection method using whale optimization algorithm and genetic operators for intrusion detection system in wireless mesh network. *IEEE Access*, *8*, 56847-56854.

[43] Harita, U., & Mohammed, M. (2024). Modelling an Improved Swarm Optimizer and Boosted Quantile Estimator For Malicious Flow Monitoring And Prediction In Network. *Journal of Cybersecurity & Information Management*, *13*(2).

[44] Wu, L., & Qu, J. (2023). AIMD rule-based duty cycle scheduling in wireless sensor networks using quartile-directed adaptive genetic algorithm. *IEEE Sensors Journal*, *23*(5), 4905-4921.

[45] Subbiah, S., Anbananthen, K. S. M., Thangaraj, S., Kannan, S., & Chelliah, D. (2022). Intrusion detection technique in wireless sensor network using grid search random forest with Boruta feature selection algorithm. *Journal of Communications and Networks*, *24*(2), 264-273.

[46] Bhutta, A. A., Nisa, M. U., & Mian, A. N. (2024). Lightweight real-time WiFi-based intrusion detection system using LightGBM. *Wireless Networks*, *30*(2), 749-761.

[47] Prajisha, C., & Vasudevan, A. R. (2022). An efficient intrusion detection system for MQTT-IoT using enhanced chaotic salp swarm algorithm and LightGBM. *International Journal of Information Security*, *21*(6), 1263-1282.

[48] Liu, J., Gao, Y., & Hu, F. (2021). A fast network intrusion detection system using adaptive synthetic oversampling and LightGBM. *Computers & Security*, *106*, 102289.

[49] Meng, D., Dai, H., Sun, Q., Xu, Y., & Shi, T. (2022). Novel Wireless Sensor Network Intrusion Detection Method Based on LightGBM Model. *IAENG International Journal of Applied Mathematics*, *52*(4).

[50] Arellano, M., & Bonhomme, S. (2017). Quantile selection models with an application to understanding changes in wage inequality. *Econometrica*, *85*(1), 1-28.

[51] Shapoorifard, H., & Shamsinejad, P. (2017). Intrusion detection using a novel hybrid method incorporating an improved KNN. *Int. J. Comput. Appl*, *173*(1), 5-9.