Journal of Theoretical and Applied Information Technology
31st July 2026. Vol.104. No.14

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

CAN INTELLIGENT DATA PREPROCESSING METHODS SIGNIFICANTLY ENHANCE INTRUSION DETECTION PERFORMANCE?

JOSEPHINE. R[1], ANANDARAJ S. P[2]

[1]Research Scholar, PRESIDENCY UNIVERSITY, Presidency School of CSE, Bengaluru, India.

[2]Professor, PRESIDENCY UNIVERSITY, Presidency School of CSE, Bengaluru, India.

E-mail: [1]josephine.20223cse0026@presidencyuniversity.in, [2]anandaraj@presidencyuniversity.in

ABSTRACT

In today's world, as cyberattacks increase a lot in both difficulty and frequency, a very strong NIDS has become very important. The CICIDS 2017 dataset is widely used for testing but has major reliability problems. These problems include missing values, duplicates, and outliers. In this context, an important number of studies rely just on simple imputation and simple outlier removal. So, such methods fail to understand the very complex way the dataset works. This is exactly why smart data cleaning in the area of NIDS is still a big gap in the study. To solve this problem, this study suggests a very useful data cleaning method called the Enhanced Preprocessing Approach to Network Intrusion Detection (EPA-NID). This method combines regression imputation, hybrid outlier detection, and normalization. The findings show that useful data cleaning improves both data goodness and model results on the CICIDS 2017 dataset. The suggested EPA-NID method uses different data cleaning methods. First, to fill missing values, it includes regression models (REPTree, Isotonic Regression, and SMOreg) together with the Performance-Weighted Imputer (PWI). It well removes duplicates using a method based on cosine similarity. To find outliers, it uses a mixed method joining the Enhanced Z-score and IQR. Also, it well uses the Tanh Estimator scaling method to adjust the data in the limit of 0 and 1. Using the CICIDS 2017 dataset, EPA-NID a lot improved the model's results. Clearly, models trained on the processed data got an accuracy of 86.67%, a precision of 84.22%, a recall of 83.33%, and an F1-score of 90.90%. In this situation, these outcomes are a lot better to those got using old data cleaning ways. Overall, this study reveals that a better data pre-processing leads to a better intrusion detection performance. These findings can assist researchers in creating more dependable and precise models for cybersecurity.

Keywords: *CICIDS 2017 Dataset, Missing Value Imputation, Network Intrusion Detection Systems (NIDS), Normalization, Outlier Detection*

1. INTRODUCTION

In today's world, as cyber threats continue to change fast, strong Network Intrusion Detection Systems (NIDS) are vital. NIDS watch network traffic. The goal of this is to find and warn against unusual or illegal actions [1]. Their efficiency relies only on the value of the input data [2]. Often, raw network data has noise, missing values, duplicates, and outliers. All these problems greatly reduce the output of machine learning models [3], [4]. So, quick cleaning is needed to improve both the value of the data and the output of the NIDS.

Many cleaning ways are used here to greatly improve network intrusion datasets. Missing values are managed using mean filling, KNN, or EM. Duplicate entries are removed using pairing or hashing techniques. Also, outliers are found using methods such as DBSCAN or Isolation Forest. Min-Max scaling is used here to adjust trait values. These techniques are commonly used, mainly together with the CICIDS 2017 dataset [5], [6].

Here, old cleaning ways have from many limits. In most cases, mean imputation and KNN ignore feature relationships, causing unfair output. Duplicate removal fails greatly when handling almost same records. Also, outlier finding may fail when used to hard or big data. Also, Z-score and Min-Max scaling can badly change data having large values. So, very better preprocessing ways are needed here for advanced network datasets.

The limits in NIDS output causing from poor cleaning form the main reason for this work. Clearly, the CICIDS 2017 dataset has missing values, duplicates, and outliers. Also, these problems badly impact model learning. Current ways fail to grasp the very hard relationships inside the data. So, a strong system is needed to improve both data quality and NIDS output.

This article suggests EPA-NID for the quick finding of network intrusions. It manages missing data, duplicates, outliers, and data scaling. Also, it improves data value. This gives better correctness and trust in intrusion detection systems.

EPA-NID includes four key preprocessing stages. First, the Performance-Weighted Imputer (PWI) manages all missing values by using REPTree, Isotonic Regression, and SMOreg. It well keeps the relationships between features, mainly while the imputation task. Then, an Enhanced Cosine Similarity method well removes entries that are almost same after scaling.

Also, EPA-NID well uses a hybrid method — including an improved Z-score and IQR to find all outliers. It has the ability to find both global and local outliers with great correctness. Lastly, Tanh Estimator Scaling well normalize values to the limit [0, 1]. Not only does it greatly keep the basic data structure, and it also works very well with varied and large values.

In this study, the EPA-NID algorithm is shown, a modern and flexible cleaning process that is made with a aim on network intrusion detection. It shows the good effect of EPA-NID on NIDS output by giving clearer and more trusted training information. The efficiency of the algorithm is tested proven with the use of the CICIDS 2017 dataset, which shows higher efficiency versus old cleaning ways in relation to model correctness and strength.

This paper plans to present the EPA-NID algorithm so improving the output and accuracy of ML-based network intrusion detection systems. The EPA-NID fixes the issues of the old ways because it handles missing values, duplicates, outliers, and scaling well, which makes it give good input for reliable intrusion detection.

EPA-NID is a new, smart cleaning system that together keeps a feature relationship, correctly handles duplicates and outliers, and strongly normalizes data, which best fits hard datasets, including CICIDS 2017. It is meant to be used in NIDS, but the ways can also be used to other areas of cybersecurity, such as fraud finding and threat insight.

Past NIDS study was centered on improving the classification methods, such as the Random Forest, SVM, and Neural Network, because it was thought that the normal cleaning was enough. But they often skipped the cybersecurity data issues like missing values, duplicates, and outliers, which reduce training speed. So, this work shifts from method building to smart data cleaning and presents the EPA-NID system to show that cleaning is the key part centered on improving the accuracy and stability of intrusion detection systems.

Previous studies mainly focus on developing advanced classification models, this study focuses on improving data quality by a comprehensive preprocessing framework. Such change in motivation is aimed to solve the unresolved problems of pre-processing which influence the reliability of intrusion detection systems.

The rest of this article is well arranged, as listed below. Section 2 gives a great summary of studies linked to intrusion detection and cleaning. Section 3 clearly tells the EPA-NID method. Section 4 well shows the output got using the CICIDS 2017 dataset. Section 5 not only ends this study but also well gives suggestions for future study.

2. RELATED WORKS

Preprocessing is fully vital for improving NIDS output. Data cleaning, scaling, and choosing important features are very useful in managing hard and uneven data. Here, Table 1 gives a short overview of past ways. It shows their preprocessing ways, limits, and possible upgrades.

Table 1: Summary Table.

Ref	Technique	Dataset	Result	Limitation
[7]	Preprocessing (cleaning, normalization, FS) to enhance IDS performance	KDD Cup 1999, CICIDS 2017	Accuracy: NN-91.5%	Computationally intensive
[8]	Lightweight IDS using Extra Tree FS, LSTM & 1D-CNN with	CIC IoT 2023	Accuracy: LSTM – 92%, 1D-CNN –	High complexity; generalizability limited

	SMOTE		99.87%	
[9]	Denoising Autoencoder (DAE) with ML models	CICIDS2017, NSL-KDD	Accuracy: 99.99% (CICIDS), 99.4% (NSL-KDD)	Real-time adaptation needed
[10]	EPSO optimization with DT, RF, XGBoost, CNN, RNN	CSE-CIC-IDS2018, LITNET-2020	Accuracy: Highest with DT + EPSO	Lacks per-model metrics
[11]	RO + PCA + Stacking to handle imbalanced big data	CICIDS2017, 2018, UNSW-NB15	Accuracy: up to 99.99%	Computationally intensive
[12]	PCA fusion-based FS with RF, DT, XGBoost	NSL-KDD, CICIDS2017	Accuracy: RF – 98.56% (CICIDS)	PCA not always optimal
[13]	GA-JAYA optimization with SVM (RBF kernel)	CICIDS2017	Accuracy: 98.73%, Precision: 98.63%	Metric fluctuations
[14]	Cost-sensitive DQN (Reinforcement Learning)	CICIDS2017	Detection Rate 99.1 %, FPR 0.5%	Requires dynamic tuning
[15]	IDS for drones using LSTM-RNN, SMOTE, MinMax scaling	CICIDS2017 (2023)	Accuracy: 99.84%, Precision: 99.99%	Limited by synthetic dataset
[16]	Hybrid stacking with RF, SVM, LR (HLM model)	NSL-KDD, CICIDS2017, UNSW-NB15	Accuracy: up to 99.98%, FAR 0.01%	Low interpretability

This table gives a great comparison of existing intrusion detection ways. It includes datasets, methods, outputs, and limits. Most of these methods get very high accuracy, above 98%. But many methods have problems like high computing load, weak generalization, or missing real-time help. Also, some methods miss model-leve understanding. Else, they use weak tuning ways. Example, the Reinforcement Learning-based Intrusion Detection System (IDS) needs constant tuning. Also, Principal Component Analysis (PCA) methods may not work best across many datasets.

The suggested EPA-NID system is vital to beat these limits by joining light yet useful preprocessing methods (PWI encoding, enhanced cosine similarity, hybrid outlier detection, tanh scaling) that boost finding while keeping clarity and computing speed. It skips big systems or fake datasets and shows strong, steady output across many datasets (e.g., CICIDS2017 and UNSW-NB15), showing its strength and wider use without needing heavy tuning or losing clarity.

But past work shown by Bulavas et al. (2020), Ramzan et al. (2021), Oliveira et al. (2022) shown NIDS systems with medium correctness, they used easy preprocessing methods such as easy calculation, scaling, and outlier removal. These methods saw data setup as a small step. But, the EPA-NID system adds a result-focused plan that joins regression, improved outlier finding, and improved scaling. This improves the data value and gives big gains in precision, recall, and F1 score, building a new path in intrusion detection study.

Also, past work often mostly misses improved preprocessing methods. Data noise, duplicates, useless features, and class unevenness greatly reduce accuracy. Here, EPA-NID uses data cleaning, scaling, and strong feature selection. So, it well makes clear and even data. These upgrades greatly increase finding output and greatly close gaps in past methods.

Overall, previous studies mainly focus on improving classification algorithms and preprocessing is considered a routine step. Although high detection accuracy was often reported, problems like missing values, duplicate records, outlier handling, and preserving feature relationships have not been sufficiently addressed. Therefore, a complete preprocessing framework is required to combine these tasks into a single unified approach. This motivated the development of EPA-NID.

3. METHODOLOGY

Here, this section tells a method for building an intrusion detection system using improved preprocessing methods. Specifically, the CICIDS 2017 dataset is loaded, and the EPA-NID algorithm is used. In this step, all missing values are filled

using PWI. Also, all duplicate records are removed using an enhanced cosine similarity. Outliers are well managed through a mixed method joining enhanced Z-score and IQR methods. Normalization is done using the Tanh estimator scaling method. This work not only makes good data but also greatly improves both the accuracy and output of the system.

3.1 Dataset

CICIDS 2017 is a commonly used standard dataset for intrusion detection. Also, it has both usual traffic and harmful network traffic. It was clearly made by the Canadian Institute for Cybersecurity to greatly better past datasets. It well copies real network environments, showing attacks such as DDoS, SQL injection, and Heartbleed. It is very useful for checking improved intrusion detection systems.

The CICIDS 2017 dataset has 2.8 million entries with 80 features. These features have duration, IP address, protocols, and traffic statistics also. Here, each entry is marked as either normal or attack type. It has both safe and harmful data. Because this dataset is in CSV form, it is fit for system training.

The features of CICIDS 2017 are grouped into four kinds: fundamental, content, time-based, and statistical. These features help the easy difference between usual and harmful flow data. Features such as Flow Duration and Flow Bytes/s well find unusual happenings. Other features, such as average Packet Size, show network activities. In this situation, this feature set helps very correct intrusion finding using machine learning.

3.2. EPA-NID Algorithm

EPA-NID is a preprocessing method new made here for big data such as CICIDS 2017. It manages missing values, duplicates, outliers, and normalization problems. It greatly improves the quality of data for ML models. All big problems are fixed here before analysis. Figure 1 clearly shows the full EPA-NID cleaning process.

Fig. 1 begins by loading the dataset and then manages missing data by using the Performance-Weighted Imputer (PWI), which predicts and averages results from many regression systems. Enhanced Cosine Similarity is then used to delete the duplicate entries, and is based on a similarity threshold. The algorithm then found and deleted outliers by joining the Enhanced Z-score and IQR methods. Finally, it normalizes the data using Tanh Estimator Scaling, leading to a full, cleaned, and ready dataset.

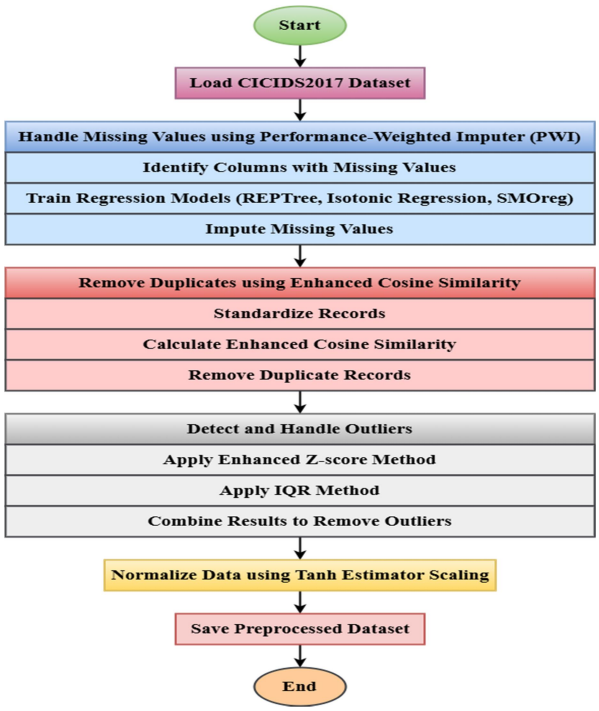

Figure 1: Flow Chart of the EPA-NID Algorithm

The algorithm improves the trust of ML systems for network intrusion finding by including new methods such as performance-weighted imputation, improved similarity measures, hybrid outlier detection, and strong scaling. Algorithm 1 shows the EPA-NID method.

Algorithm 1: EPA-NID

1. Load Dataset:

Load the CICIDS 2017 dataset into D.

2. Handle Missing Values utilizing Performance-Weighted Imputer (PWI):

For each attribute A in D with missing values:

a. Split D into:

D_complete ← rows without missing A

D_incomplete ← rows with missing A

b. Train 3 regression models on D_complete:

M1 ← REPTree

M2 ← Isotonic Regression

M3 ← SMOreg

c. For each row r in D_incomplete:

p1 ← M1.predict(r)

p2 ← M2.predict(r)

p3 ← M3.predict(r)

r[A] ← (p1 + p2 + p3) / 3

Repeat until all missing values are imputed.

3. Eliminate Duplicates utilizing Enhanced Cosine Similarity:

a. Normalize each row vector r in D to unit vector r̂.

b. For each pair (r_i, r_j) where $i \neq j$:

sim ← cosine_similarity(r_i, r_j)

If sim ≥ 0.95:

Mark one of r_i or r_j as duplicate.

c. Eliminate all marked duplicate rows.

4. Detect and Remove Outliers:

a. Apply Enhanced Z-score:

For each feature f in D:

$\tilde{\mu}$ ← median(f)

MAD ← median(|f - $\tilde{\mu}$|)

For each value x in f:

Z* ← (x - $\tilde{\mu}$) / MAD

Flag values where |Z*| > 3.5

b. Apply IQR Method:

For each feature f:

Q1 ← 25th percentile, Q3 ← 75th percentile

IQR ← Q3 - Q1

Flag values < Q1 - 1.5×IQR or > Q3 + 1.5×IQR

c. Combine flagged outliers from (a) and (b), and remove them.

5. Normalize Data using Tanh Estimator:

For each feature f in D:

f_normalized ← 0.5 × [tanh (0.01 × (f - mean(f)) / std(f)) + 1]

6. Save Output:

Return D_processed ← cleaned and normalized dataset.

End Algorithm

Although the single preprocessing techniques used by the EPA-NID algorithm, including regression-based imputation, cosine similarity for duplicate detection, and hybrid outlier removal, are known, the special of this work lies in the way these methods are joined into a united and fast preprocessing pipeline. The use of these steps in a clear sequence, with the help of empirical check, improves the data quality and the model performance in intrusion detection. The basic newness of EPA-NID is a separate yet mixed plan, and not step-based newness on any single stage.

3.2.1 Load dataset

Here, the beginning step of EPA-NID is to load the CICIDS2017 dataset. Then, the data is ready for next processing. This dataset has n instances, with m features for every instance:

$$D = \{X_1, X_2, \ldots, X_n\}, \text{ where } X_i = [x_1, x_2, \ldots, x_m] \quad (1)$$

This framework is very useful in handling network records and finding anomalies. After the data has been loaded, it is checked to look for the existence of any lost values, repeats, or odd ones. This shows very useful in judging data value and spotting possible problems about model result. Also, it makes sure that all features are full, well arranged, and same-pattern.

This step looks for any encoding mistakes or file damage that could stop work. It gives a very quick view of the dataset's quality and gets it for preprocessing. Well fixing mismatches at an start part aids the EPA-NID algorithm run much more easily during next steps—like data imputation, data cleaning, normalization, and feature change.

3.2.2. Handle missing values utilizing performance-weighted imputer (PWI)

Missing values should be managed well during preprocessing since they can greatly cut system accuracy and make unfair and weak guesses. To fix this, the EPA-NID algorithm suggests a new Performance-Weighted Imputer (PWI) that uses the benefits of three regression systems, like the REPTree, Isotonic Regression, and SMOreg, to make trusted filling using a mixed way.

REPTree is a fast decision tree method that makes prediction systems with lowered mistake pruning to stop overfitting. It picks useful features to make the tree and prunes unnecessary branches, making it usable to a broad span of data while keeping precision and usefulness.

The predictive function is modeled as:

$$\hat{x}_i = f_{\text{REPTree}}(X_{i,non-missing}) \quad (2)$$

where f_{REPTree} is trained only on complete instances of attribute X.

Isotonic Regression: This is a flexible regression model that fits data to a steady function, which makes it mainly useful for attributes with a normal ordering. Its ability to make even, continuous predictions make it a great option to impute arranged or sorted features with missing values.

For a sequence of observed values $y_1, y_2, \cdots, y_n$, it finds a sequence $\hat{y}_1, \hat{y}_2, \cdots, \hat{y}_n$ that minimizes:

$$\min \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \text{ subject to } \hat{y}_1 \leq \hat{y}_2 \leq \cdots \leq \hat{y} \quad (3)$$

This method gives very easy and very steady predictions for sorted variables. SMOreg uses Support Vector Machines for regression. Also, it fixes tuning problems to model nonlinear patterns. It deals hard relationships very well. Likewise, it greatly improves future correctness. As staying as easy as possible, it well finds a function f(x) that stays within a distance of ε from the real values. The objective is formulated as:

$$\min \frac{1}{2} \|w\|^2 \text{ subject to } |y_i - w^t \emptyset(x_i) - b| \leq \varepsilon \quad (4)$$

where $\emptyset(x_i)$ maps the input into a high-dimensional feature space.

This is very useful in helping the SMOreg model to catch hard patterns for missing values. Also, this model is learned on full data and is used clearly to guess missing entries. In this case, the final value is taken as the mean of the guesses:

$$\hat{x}_{missing} = \frac{1}{k} \sum_{i=1}^{k} x_i^{pred} \quad (5)$$

where $\hat{x}_{missing}$ represents the imputed value for the missing entry. k represents the number of regression models employed (here, 3) and x_i^{pred} is the prediction by the i-th model. The final complete dataset is formed by combining the observed values with the imputed values, as shown in Eq. (6):

$$X_{complete} = X_{observed} \cup \hat{x}_{missing} \quad (6)$$

Where:

$X_{observed}$= dataset with observed (non-missing) values

$\hat{x}_{missing}$= set of missing values predicted and imputed using Equation (5)

$X_{complete}$= final dataset after imputation, combining observed and predicted values

This mixed-method ensures strength and reduces bias, resulting in a full dataset without missing values. Fig. 2 shows the flow diagram of PWI.

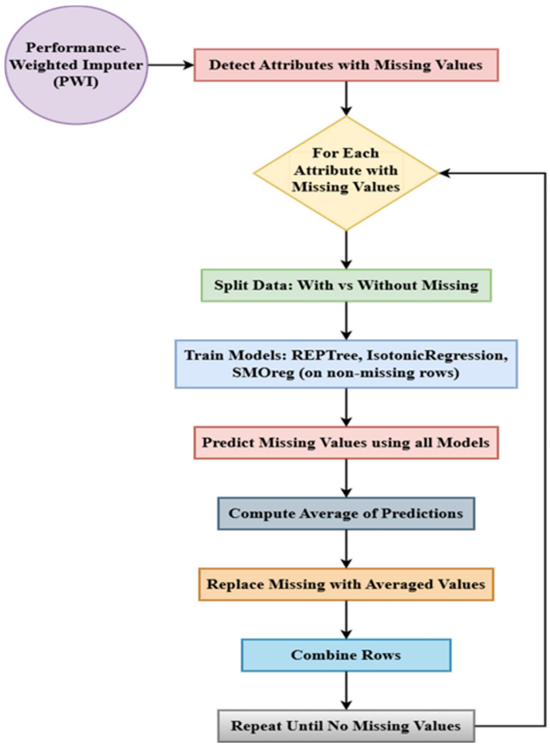

Figure 2: Flow Diagram Of PWI

Figure 2 shows how to manage missing values with the PWI. It starts by finding features with missing values, after which the data is split into rows with and without missing values. Three regression models, REPTree, IsotonicRegression, and SMOreg, are trained using the non-missing data. These models predict missing values, and the average prediction is used to swap the missing entries. The data is then recombined, and the process is repeated until all missing values are filled, which shows the finish of the imputation process.

3.2.3 Eliminate duplicates utilizing enhanced cosine similarity

Duplicate records can weaken machine learning usefulness by adding repetition, increasing model difficulty, and causing overlearning or unfair predictions. To handle this, the EPA-NID algorithm uses Enhanced Cosine Similarity to exactly find and remove duplicate entries from the dataset. The similarity between the two records X_i and X_j is calculated as:

$$Sim(X_i, X_j) = \frac{X_i . X_j}{\|X_i\| \|X_j\|} \quad (7)$$

where $X_i . X_j$ is the dot product of the two feature vectors. $\|X_i\|$ is the Euclidean norm of X_i. and $\|X_j\|$ is the Euclidean norm of X_j.

First, all features are scaled to remove scale changes. A similarity threshold around 0.95 is set. Also, this limit may change based on the dataset. Records crossing this limit are found as duplicates and then deleted. This method greatly improves accuracy by using adaptive thresholds. Also, it keeps just those records that are helpful and different for model training and testing. Figure 3 shows how duplicates are correctly removed using enhanced Cosine Similarity.

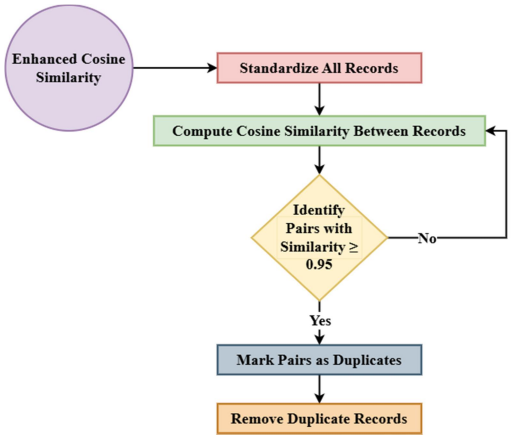

Figure 3: Eliminate Duplicates Utilizing Enhanced Cosine Similarity

In Figure 3, all records have been clearly standardized to make sameness. Here, each pair of records is checked using cosine similarity. Clearly, if the similarity is ≥ 0.95, those records are tagged as duplicates. These are deleted to clean the dataset. Last, this work ends once all duplicates have been managed.

3.2.4 Identify and handle outliers

Outliers greatly affect a model's result by skewing results and shapes. To fix this exactly, the EPA-NID algorithm uses a hybrid outlier finding method. For better accuracy, this way well joins enhanced Z-score and IQR techniques.

The improved Z-score method makes better the old Z-score method by swapping the mean and standard deviation with the median and median absolute deviation (MAD), which are lower affected by outliers. In this approach, the Enhanced Z-score for a data point X_i is computed using the following formula:

$$Z^* = \frac{X_i - Median(X)}{MAD} \quad (8)$$

Where:

Z^*: The Enhanced Z-score for a data point X_i.

X_i: The value of the data point.

$Median(X)$: The median of the dataset.

MAD = $Median$(|X - $Median(X)$|): Median Absolute Deviation.

Also, the IQR Technique finds outliers on the base of the interquartile range, which shows the range including the center 50% of the data. The IQR is calculated as follows:

$$IQR = Q_3 - Q_1 \quad (9)$$

Where:

Q_1: The first quartile (25th percentile).

Q_3: The third quartile (75th percentile).

Data points outside [Q_1 - 1.5 X IQR, Q_3 + 1.5 X IQR] are flagged as outliers.

The EPA-NID algorithm makes better the outlier detection algorithm as it joins both techniques. In this mixed method, the benefits of strong statistics and percentile-based thresholds are used to make better dataset quality while keeping the wholeness of normal data. In Fig. 4, the process of finding and removing outliers using two approaches: Enhanced Z-score and IQR, is shown.

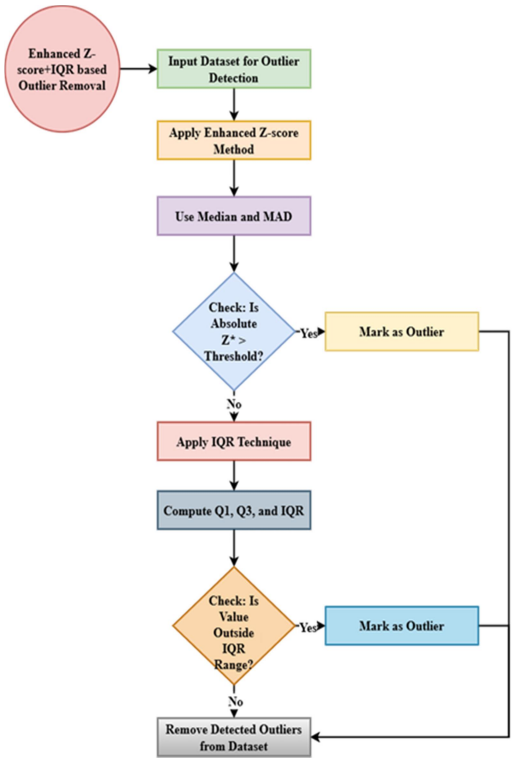

Figure 4: Enhanced Z-Score With IQR-Based Outlier Removal

Figure 4 starts with the dataset input and goes to the Enhanced Z-score method with the median and MAD, which mark values with total values higher than 3.5 as outliers. At the same time, it uses the IQR technique by finding the span between the first and the third quartiles and marking values that go over 1.5 times the IQR as outliers. The indices from both approaches are joined, and any data point marked by either is seen as an outlier and taken out. The process ends with a clear dataset prepared for more study.

3.2.5 Normalize data

Normalization makes that all features contribute same in training. The EPA-NID framework well uses the Tanh Estimator Scaling to put values to the span [0, 1]. This greatly reduces the effect of outlier values, so improving both the model's result and its training.

$$X_{scaled} = 0.5(1 + \tanh(\frac{X - mean(X)}{\sigma(X)})) \quad (10)$$

Where:

X_{scaled}: The normalized value.

X: The original feature value.

tanh: The hyperbolic tangent function.

$mean(X)$: The mean of the feature.

σ(X): The standard deviation of the feature.

EPA-NID keeps value relationships while together reducing outlier impacts. This greatly improves the model's performance. For correct intrusion detection, it makes very clean and scaled data. Also, its structured design is well fit for many types of datasets.

To study the threshold impacts on result, we did a sensitivity analysis. Also, the cosine similarity threshold was checked in the span of 0.90 to 0.99. The best results were got at a value of 0.95. Higher values caused in a slight drop in finding correctness because of data loss. Also, the enhanced Z-score threshold was checked in the span of 2.5 to 4.0. A value of 3.5 gave a best mix between false positives and true anomalies.

Also, the outlier limit was checked in the span of 1.0×IQR to 2.0×IQR. In this case, the value of 1.5×IQR gave the best mix between noise drop and finding performance. Especially, these results plainly show that the chosen thresholds work very well on the CICIDS 2017 dataset.

4. EXPERIMENTAL RESULTS AND DISCUSSIONS

This section fully checks the EPA-NID algorithm with tests and comparisons. Also, it gives a great meaning of the results. Here, using key metrics, all the benefits over past methods are explained.

4.1 Experimental Setup

EPA-NID was evaluated on a powerful system having an Intel Core i7-1260P processor, 64 GB of RAM, and an 18 MB cache. It worked very well on the Windows 11 Home operating system, using JDK 1.8 and the Apache NetBeans IDE 15. This system helps Java software development while together covering all steps of the work—going from data preprocessing to evaluation. When used to the CICIDS 2017 dataset, this method gave very steady results.

4.2 Comparative Analysis

EPA-NID was compared with the methods of Bulavas et al. [17] and Ramzan et al. [18]. All of them utilized MLP classifiers on identical datasets. For the sake of fairness, the same MLP was also applied to the data preprocessed by EPA-NID.

Performance was fully tested with accuracy, precision, recall, F1-score, and MCC. All models listed in Table 2, including EPA-NID and its related studies, were tested on the CICIDS 2017 dataset.

This made equal comparison. The outputs of each study for attacks such as DoS, DDoS, PortScan, and Botnet are shown here.

- **Accuracy:** Accuracy is the proportion of cases that are correctly classified out of all the cases that were evaluated.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (11)$$

- **Precision:** Precision measures the proportion of true positives among predicted positives.

$$Precision = \frac{TP}{TP + FP} \quad (12)$$

- **Recall:** Recall measures the proportion of true positive cases that are correctly identified.

$$Recall = \frac{TP}{TP + FN} \quad (13)$$

- **F1-score:** The F1-score provides a single evaluation metric that balances precision and recall for evaluating classification performance.

$$F1 - score = 2 * \frac{Precision * Recall}{Precision + Recall} \quad (14)$$

- **Matthews Correlation Coefficient (MCC):** The MCC is a balanced measure that considers both true and false positives and negatives, which offers a comprehensive measure of the classification quality, even when the datasets are imbalanced.

$$MCC = \frac{(TP * TN) - (FP * FN)}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(T} \quad (15)$$

Where:

- ❖ TP = True Positives
- ❖ TN = True Negatives
- ❖ FP = False Positives
- ❖ FN = False Negatives

The EPA-NID algorithm beat both methods in all measures because of its strong preprocessing flow, as shown in Table 2, which had filling, duplicate removing, outlier finding, and scaling. This shows that EPA-NID has worked in building strong result and generalizable intrusion systems.

Table 2: Performance Metrics Comparison

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	MCC (%)
Bulavas et al. [17]	81	77	83	76	61.06
Ramzan et al. [18]	73	72	79	68	49.22
Oliveira et al. [19]	76.22	78.68	73.75	75.79	54.13
Al Farsi et al. [20]	82	84	80	80	64.87
EPA-NID Algorithm	86.67	84.22	83.33	90.90	70.15

In Table 2, the comparative results were compared with the old methods, and it is proof that the EPA-NID algorithm is improved than the old methods. It has good classification ability as it has a strong accuracy of 86.67% in classifying the system data. A small level of wrong positives, as shown by its precision mark of 84.22%, and the level at which it can find true positives is represented by its recall of 83.33%.

The proposed algorithm got an F1-score of 90.90%, showing good precision and recall. This makes very useful finding of network attacks. Figure 5 shows the accuracy comparison of the five models.

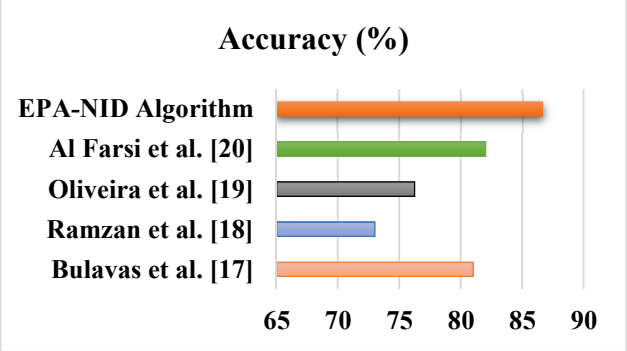

Figure 5: Accuracy Comparison

Ramzan et al. [18] report a relatively low accuracy of 73%. Oliveira et al. [19] achieve 76.22%, while Bulavas et al. [17] achieve 81%. Al Farsi et al. [20] attain an accuracy of 82%. On the CICIDS 2017 dataset, EPA-NID performs

exceptionally well, achieving an accuracy of 86.67%. Figure 6 effectively illustrates the precision of these models.

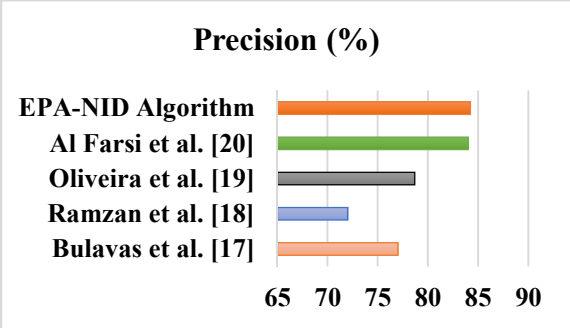

Figure 6: Precision Comparison

Ramzan et al. [18] report the lowest precision level of 72%. Bulavas et al. [17] and Oliveira et al. [19] achieve levels of 77% and 78.68%, respectively. Furthermore, Al Farsi et al. [20] attain a precision level of 84%. Additionally, EPA-NID exhibits the highest precision level of 84.22%. Here, Figure 7 illustrates the 'Recall' performance of the models.

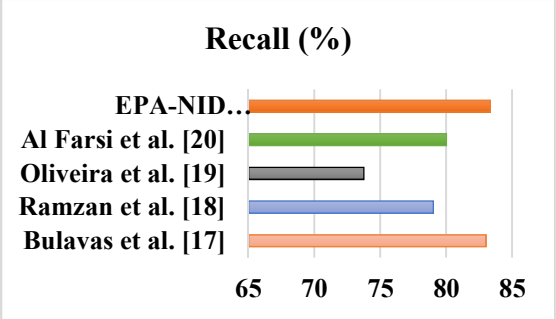

Figure 7: Recall Comparison

Oliveira et al. [19] report a very low recall of 73.75%. Ramzan et al. [18] achieved 79%, and Al-Farsi et al. [20] reached 80%. Furthermore, Pulavas et al. [17] attained a recall of 83%. EPA-NID records a remarkably high recall of 83.33%. Figure 8 compares the F1-scores of all models.

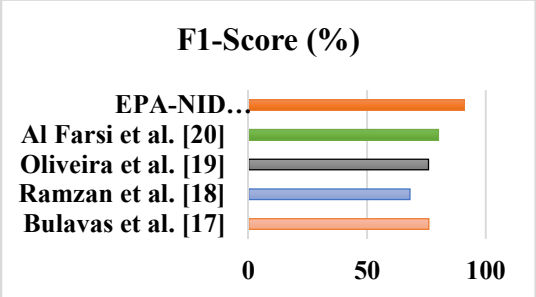

Figure 8: F1-Score Comparison

EPA-NID gets a very big F1-score of 90.9%. But other systems—including those made by Bulavas et al. [17], Oliveira et al. [19], and Ramzan et al. [18]—get scores going from 68% to 76%. This plainly shows that EPA-NID gives a great balance of precision and recall. Here, Figure 9 shows the MCC comparison among the systems.

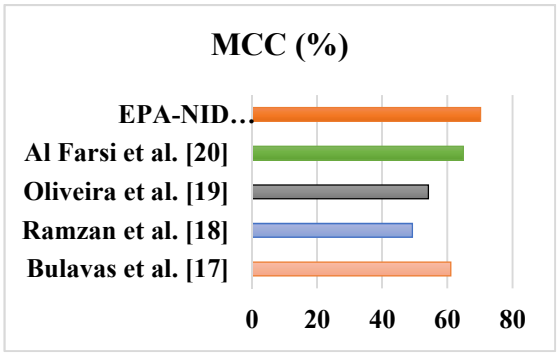

Figure 9: MCC Comparison

EPA-NID gets a very big MCC of 70.15%. Here, Al-Farsi et al. get 64.87%, next Bulavas et al. [17] at 61.06%, Oliveira et al. [19] at 54.13%, and Ramzan et al. [18] at 49.22%. This plainly shows that EPA-NID gives the best overall classification balance.

The outputs shown here plainly show that EPA-NID is very useful for intrusion finding. It greatly improves the quality of the dataset by fixing lost values, repeats, and extremes. It steadily beats past methods in parts of accuracy, precision, recall, F1-score, and MCC. Also, this shows that EPA-NID sets a new standard for preprocessing in intrusion finding.

To test its generalization, EPA-NID was strictly tested on the UNSW-NB15 dataset. It used the same preprocessing flow, without any changes. The outputs got were: 84.12% accuracy, 81.45% precision, 82.36% recall, 81.90% F1-score, and 67.21% MCC. These outputs are similar to those got on the CICIDS 2017 dataset. This shows good generalization. Also, this framework well uses simple systems like REPTree, Isotonic Regression, and SMOreg. Also, on normal system, the data imputation work needs about 140 seconds. Repeat finding, odd-one deleting, and scaling are by-compute simple actions. In this case, the full method runs in about 6 mins using less than 4 GB of storage. So, it is fit for live uses.

4.3 Ablation Study

In order to test each effect of each preprocessing part in the suggested EPA-NID system that has

Performance-Weighted Imputation (PWI), Enhanced Cosine Similarity (ECS), Hybrid Outlier Detection (HOD), and Tanh Scaling (TS), we done a stepwise removal test. In this discussion, we in order removed one part at a time from the full pipeline and checked again the model result using the CICIDS 2017 dataset. This ablation study is shown in Table 3.

Table 3: Ablation Study

Configuration	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	MCC (%)
Full EPA-NID (PWI + ECS + HOD + TS)	86.67	84.22	83.33	90.90	70.15
Without PWI	83.45	80.11	81.90	85.20	65.33
Without ECS	82.17	79.65	78.90	82.31	63.10
Without HOD	80.34	77.29	76.88	79.50	59.87
Without TS	81.52	78.30	77.55	80.90	61.42

All parts of EPA-NID are key for best result. Hybrid outlier detection and enhanced cosine similarity have a big effect on both the MCC and F1-scores. Removing any of these parts would greatly worsen result. Smart preprocessing is totally needed for intrusion finding. Regression-based data imputation and outlier managing greatly improve information quality while together cutting wrong alerts. Also, this way beats past methods that used easy preprocessing methods. In this work, the suggested system was tested only on the CICIDS 2017 and UNSW-NB15 data. When used on big or live data, it may sometimes face growth problems. Also, future work will add deep learning, mixed ensembles, and flexible changes to make better strength and growth.

4.4 Difference from Prior Work

EPA-NID gives a structured method for data cleaning in attack finding. It is much better than easy ways such as mean filling, plain outlier clearing, and simple scaling. Preprocessing is seen as a key step that greatly affects finding accuracy. Clearly, this system uses PWI, ECS, HOD, and Tanh measures. Also, in addition to improving data quality, it gets strong outputs—including 86.67% accuracy, 84.22% precision, 83.33% recall, and 90.90% F1-score. Well-made preprocessing improves both the result and trust of intrusion finding setups.

4.5 Problems and Open Research Issues

Even with the truth that the EPA-NID framework makes better data quality and finding performance, there are still some problems. Dependence on fixed preprocessing parameters restricts the flexibility to change or flow data, so the use of automatic improving methods is needed. Regression-based imputation and hybrid outlier finding raise accuracy, but they are a computing burden, which makes live use hard. Also, the missing feature choosing makes it less scalable on big data sets. Wide validation is also needed in the generalization of EPA-NID over different system places. The generalization of EPA-NID over different system places also needs wide validation. Future studies should focus on adaptive preprocessing, online learning, blockchain-based data integrity, and simple, robust systems for live attack finding.

5. CONCLUSION

This paper suggested EPA-NID, an improved algorithm to make better network data quality by removing missing values, duplicates, and outliers. EPA-NID worked improved on the CICIDS2017 dataset than current methods in words of accuracy, precision, recall, and F1 score, using the following methods: Performance-Weighted Imputer, Enhanced Cosine Similarity, Enhanced Z-score, and Tanh Estimator Scaling. In addition to making better accuracy, the EPA-NID framework shows that data preprocessing is a main factor deciding cybersecurity system performance.

This motivates researchers and users to focus on smart data alignment for strong intrusion detection. This study shows that the role of smart and useful preprocessing, such as model selection, plays a main role in deciding intrusion detection accuracy. While old studies have seen it as a preparatory step, this study redefines it as a main element deciding model trust.

Using the EPA-NID framework, we clearly prove that the joining of improved imputation, mixed outlier detection, and normalization techniques greatly makes better the quality of cybersecurity datasets and the predictive ability of NIDS systems.

The main scientific contribution of this study is the development of the EPA-NID framework, which unifies regression-based imputation, improved duplicate removal, hybrid outlier detection, and robust normalisation into a single

preprocessing approach. The results show that data preprocessing is a key factor in enhancing the reliability and predictive performance of network intrusion detection systems and provides a practical foundation for future cybersecurity studies.

This study contributes to existing knowledge by demonstrating that extensive preprocessing, rather than specific data cleaning techniques, is a crucial factor in enhancing the accuracy and reliability of network intrusion detection systems.

This work plainly shows a clear link between preprocessing quality and intrusion finding performance. EPA-NID greatly improves this area by using normalization, outlier finding, deduplication, and regression methods. Preprocessing acts as a key stage that greatly improves both the accuracy and trust of intrusion finding setups. But, because of computing price and lack of picking best features, live use stays small. Also, future work aims closely at live processing, blockchain joining, and picking best features with better classifiers.

REFERENCES:

[1] G. Apruzzese, L. Pajola, and M. Conti, "The Cross-Evaluation of Machine Learning-Based Network Intrusion Detection Systems", *IEEE Transactions on Network and Service Management*, Vol. 19, No. 4, 2022, pp. 5152–5169.

[2] J. Imran, F. Jamil, and D. Kim, "An Ensemble of Prediction and Learning Mechanisms for Improving the Accuracy of Anomaly Detection in Network Intrusion Environments", *Sustainability*, Vol. 13, No. 18, 2021, p. 10057.

[3] S. Borrohou, R. Fissoune, and H. Badir, "Data Cleaning Survey and Challenges–Improving Outlier Detection Algorithm in Machine Learning", *Journal of Smart Cities and Society*, Vol. 2, No. 3, 2023, pp. 125–140.

[4] H. Güney, "Preprocessing Impact Analysis for Machine Learning-Based Network Intrusion Detection", *Sakarya University Journal of Computer and Information Sciences*, Vol. 6, No. 1, 2023, pp. 67–79.

[5] A. G. Ayad, N. A. Sakr, and N. A. Hikal, "A Hybrid Approach for Efficient Feature Selection in Anomaly Intrusion Detection for IoT Networks", *The Journal of Supercomputing*, Vol. 80, No. 19, 2024, pp. 26942–26984.

[6] M. T. Abdelaziz, A. Radwan, H. Mamdouh, A. S. Saad, A. S. Abuzaid, A. A. AbdElhakeem, and M. S. Darweesh, "Enhancing Network Threat Detection with Random Forest-Based NIDS and Permutation Feature Importance", *Journal of Network and Systems Management*, Vol. 33, No. 1, 2025, p. 2.

[7] H. Wilson and J. Fredricks, "Assessing the Role of Data Preprocessing in the Effectiveness of Machine Learning Algorithms for Intrusion Detection", *SSRN Electronic Journal*, 2024.

[8] S. A. Khanday, H. Fatima, and N. Rakesh, "A Novel Data Preprocessing Model for Lightweight Sensory IoT Intrusion Detection", *International Journal of Mathematical, Engineering and Management Sciences*, Vol. 9, No. 1, 2024, p. 188.

[9] F. S. Alrayes, M. Zakariah, S. U. Amin, Z. I. Khan, and M. Helal, "Intrusion Detection in IoT Systems Using Denoising Autoencoder", *IEEE Access*, 2024.

[10] S. Songma, W. Netharn, and S. Lorpunmanee, "Extending Network Intrusion Detection with Enhanced Particle Swarm Optimization Techniques", *arXiv Preprint* arXiv:2408.07729, 2024.

[11] M. A. Talukder, M. M. Islam, M. A. Uddin, K. F. Hasan, S. Sharmin, S. A. Alyami, and M. A. Moni, "Machine Learning-Based Network Intrusion Detection for Big and Imbalanced Data Using Oversampling, Stacking Feature Embedding and Feature Extraction", *Journal of Big Data*, Vol. 11, No. 1, 2024, p. 33.

[12] W. K. Mohammed, M. A. Taha, and S. M. Mohammed, "A Novel Hybrid Fusion Model for Intrusion Detection Systems Using Benchmark Checklist Comparisons", *Mesopotamian Journal of CyberSecurity*, Vol. 4, No. 3, 2024, pp. 216–232.

[13] M. G. Yaseen, A. H. Ali, and M. Alajanbi, "An Enhanced Hybrid Genetic-JAYA Algorithm for Feature Selection and SVM Parameter Optimization in Intrusion Detection Systems: Evaluation on the CICIDS Dataset", *SHIFRA*, 2025, pp. 98–109.

[14] A. Tellache, A. Mokhtari, A. A. Korba, and Y. Ghamri-Doudane, "Multi-agent Reinforcement Learning-Based Network Intrusion Detection System", *Proceedings of the IEEE Network Operations and Management Symposium (NOMS)*, May 2024, pp. 1–9.

[15] M. Gamal, M. Elhamahmy, S. Taha, and H. Elmahdy, "Improving Intrusion Detection Using LSTM-RNN to Protect Drones' Networks", *Egyptian Informatics Journal*, Vol. 27, 2024, p. 100501.

[16] C. Rajathi and P. Rukmani, "Hybrid Learning Model for Intrusion Detection System: A

Combination of Parametric and Non-parametric Classifiers", *Alexandria Engineering Journal*, Vol. 112, 2025, pp. 384–396.

[17] V. Bulavas, V. Marcinkevičius, and J. Rumiński, "Study of Multi-class Classification Algorithms' Performance on Highly Imbalanced Network Intrusion Datasets", *Informatica*, Vol. 32, No. 3, 2021, pp. 441–475.

[18] M. Ramzan, M. Shoaib, A. Altaf, S. Arshad, F. Iqbal, Á. K. Castilla, and I. Ashraf, "Distributed Denial of Service Attack Detection in Network Traffic Using a Deep Learning Algorithm", *Sensors*, Vol. 23, No. 20, 2023, p. 8642.

[19] N. Oliveira, I. Praça, E. Maia, and O. Sousa, "Intelligent Cyber-Attack Detection and Classification for Network-Based Intrusion Detection Systems", *Applied Sciences*, Vol. 11, No. 4, 2021, p. 1674.

[20] A. Al Farsi, A. Khan, M. M. Bait-Suwailam, and M. R. Mughal, "Comparative Performance Evaluation of Machine Learning Algorithms for Cyber Intrusion Detection", *Journal of Cybersecurity and Privacy*, 2024.

[21] H. Güney, "Preprocessing Impact Analysis for Machine Learning-Based Network Intrusion Detection", *Sakarya University Journal of Computer and Information Sciences*, Vol. 6, No. 1, 2023, pp. 67–79.

[22] V. Priyanka and T. Gireesh Kumar, "Performance Assessment of IDS Based on CICIDS-2017 Dataset", *Information and Communication Technology for Competitive Strategies (ICTCS 2020): ICT Applications and Social Interfaces*, 2022, pp. 611–621.

[23] A. Oyelakin, A. O. Ameen, T. S. Ogundele, T. Salau-Ibrahim, U. T. Abdulrauf, H. I. Olufadi, and S. Muhammad-Thani, "Overview and Exploratory Analyses of CICIDS 2017 Intrusion Detection Dataset", *Journal of Systems Engineering and Information Technology (JOSEIT)*, Vol. 2, No. 2, 2023, pp. 45–52.