BULGARIAN ACADEMY OF SCIENCES

CYBERNETICS AND INFORMATION TECHNOLOGIES • Volume **24**, No 1
Sofia • 2024 Print ISSN: 1311-9702; Online ISSN: 1314-4081
DOI: 10.2478/cait-2024-0006

Enhancing Intrusion Detection with Explainable AI: A Transparent Approach to Network Security

Seshu Bhavani Mallampati[1], *Hari Seetha*[2]

[1]*School of Computer Science and Engineering, VIT-AP University, Andhra Pradesh, India*
[2]*Centre of Excellence, AI and Robotics, VIT-AP University, Andhra Pradesh, India*
E-mails: bhavani.20phd7017@vitap.ac.in seetha.hari@vitap.ac.in

Abstract**:** *An Intrusion Detection System* (*IDS*) *is essential to identify cyber-attacks and implement appropriate measures for each risk. The efficiency of the Machine Learning* (*ML*) *techniques is compromised in the presence of irrelevant features and class imbalance. In this research, an efficient data pre-processing strategy was proposed to enhance the model's generalizability. The class dissimilarity is addressed using k-Means SMOTE. After this, we furnish a hybrid feature selection method that combines filters and wrappers. Further, a hyperparameter-tuned Light Gradient Boosting Machine* (*LGBM*) *is analyzed by varying the optimal feature subsets. The experiments used the datasets – UNSW-NB15 and CICIDS-2017, yielding an accuracy of 90.71% and 99.98%, respectively. As the transparency and generalizability of the model depend significantly on understanding each component of the prediction, we employed the eXplainable Artificial Intelligence* (*XAI*) *method, SHapley Additive exPlanation* (*SHAP*)*, to improve the comprehension of forecasted results.*

Keywords**:** *Cyber security, Intrusion Detection System* (*IDS*)*, Hybrid feature selection, SMOTE, Light Gradient Boosting Machine* (*LGBM*)*.*

1. Introduction

Cybersecurity is becoming crucial with the massive expansion of networks and the large applications that operate on them [1]. Specifically, 5G transmission, elastic computing, and the Internet of Things are gaining popularity [2]. The attacker seizes control by exploiting the vulnerabilities to insert malicious scripts or other hacking techniques [3]. They utilize multiple techniques to earn money through leveraging fake websites, phishing campaigns, and inserting malware. IBM stated that the overall cost of a data breach worldwide hiked to 4.35 million dollars in 2022 [4]. According to Cybersecurity Outlook published by Statista [5], the rate of cybercrime would rise significantly over the next five years, increasing from \$8.44 trillion in 2022 to \$23.82 trillion by 2027. Hence, governments and global sectors spend huge funds annually on resources to develop antivirus and firewalls to fight against cyber-attacks. IDS, which employs anomaly and signature detection methods, has become a crucial tool for securing cyber networks. Signature detection identifies known attacks by applying pattern matching on the data. However, they cannot identify unknown attacks and must update the database when a new attack pattern is obtained. Anomaly detection determines whether the data exhibits aberrant behavior to make a decision.

In recent years, IDS processes have been challenging due to the proliferation of new types of network assaults and the rise in the volume of network data flow [6]. As a consequence, ML has become popular in the field of IDS. ML models can learn and recognize patterns from complex data by utilizing statistical approaches and highly sophisticated methods. Machine learning-based IDS provides advantages over traditional detection techniques, such as identifying malicious signatures. However, they are more susceptible to the increased rate of false positives when associated with anomaly-based detection solutions. Furthermore, Batchu [7] pointed out that this is primarily due to the impact of large dimensionality, which in turn causes poor performance, with an increase in the time needed to learn the model and an increase in the load placed on computing resources like the Central Processing Unit (CPU) and memory.

In recent times, various ML models have been utilized to identify attacks in the networks. Most of these methods work effectively, but the major limitation is that they have not explained the factors that led to their predictions. Since models are too complex to learn and comprehend due to their black-box form, XAI is used to interpret the output of the models [8].

To surpass the above limitations, this work proposes an XAI framework integrated with hybrid feature selection for detecting unknown attacks. The significant contributions of this paper are as follows.

- To enhance the sample size of the minority class and to achieve a balanced distribution of classes, the resampling technique k-Means SMOTE is used on the datasets UNSW-NB 15 and CICIDS-2017.
- A hybrid feature selection PCIG-SFFS-LGBM is proposed by integrating filter and wrapper-based models. The linearly correlated features are deleted by using Pearson Correlation (PC). Further, the non-linearly associated features are eliminated by using Information Gain (IG).
- A wrapper-based Sequential Forward Feature Selection (SFFS) technique with hyper-tuned LGBM is incorporated to eliminate features that do not impact the actual model performance.
- Further, we employ the XAI method SHAP to explain the effect of specific traits that have been chosen. These explanations assist in a deeper comprehension of the actions and decisions taken by a machine learning model when it generates predictions.

The work is arranged as follows. Section 2 provides a literature survey. Section 3 shows a detailed process of the suggested framework. Section 4 describes the metrics that have been used for evaluation. Section 5 illustrates the experimental setup, its performance, and comparison with state-of-art methods. Finally, Section 6 concludes our work.

2. Literature survey

Over recent years, various intrusion detection algorithms have been proposed to provide security to devices. For example, L. Y. Kim, and H. Kim [9] developed an integrated feature selection by combining a Sequence Forward Selection Decision Tree (SFS-DT) to select important features. Then, the attribute set is trained by LSTM, Recurrent Neural Networks, and Gated Recurrent Unit, proving that LSTM outperformed with an accuracy of 96.90%. However, LSTMs are slower when the data is significant since they require more training time to learn effectively.

Roy, Li, Choi and Bai [10] have proposed an IDS incorporating dimensionality reduction, sampling, and classification to detect attacks in IoT networks. They used SMOTE to balance the data and reduced dimensionality using PCA. The minimized attribute set is then forwarded to the B-Stacking method, which contains Random Forest (RF), K-Nearest Neighbors (KNN), and eXtreme Gradient Boosting (XGBM) as base classifiers and meta classifiers as XGBM. They tested their technique on CIC-IDS 2017 and NSL-KDD datasets and attained an accuracy of 99.11% and 98.5%.

Saha, Priyoti and Sharma [11] provide an ensemble feature selection method by analyzing 15 feature selection models to select the best attributes. The optimal attributes are trained using Unsupervised Learning (UL), Deep Learning (DL), and ML models to recognize attacks. The UNSW-NB 15 dataset is used to test their model, and they attained an accuracy of 87.25% with UL, 76%-ML, and 86.6% with DL. They have not addressed the class imbalance in UNSW-NB 15.

De Souza et al. [12] suggest a detection model for identifying attacks in IoT networks. They used information gain to pick the important attributes. Then, they were trained by a hybrid method containing a deep neural network K-Nearest Neighbor (KNN). They tested their model on CICIDS-2017 and NSL-KDD datasets with an accuracy of 99.85% and 99.77%. However, the computational time required for processing the KNN model is greater.

Yin et al. [13] provided a hybrid feature selection by using wrapper and filter methods such as RF, Information Gain, and Recursive Feature Elimination with Multi-Layer Perceptron (IGRF-RFE-MLP). They selected 23 optimal attributes out of 43 attributes of the UNSW-NB 15. The optimal attributes are trained by MLP and attained an accuracy of 84.24%. They failed to mention the time required to run the model.

Patil et al. [14] suggested an XAI-IDS framework to identify assaults. To enhance the accuracy and minimize the false positives, an ensemble voting classifier was built by using Random Forests (RF), decision trees, and SVM, and an accuracy of 96.25% was obtained on the CIC-IDS 2017 dataset. For the explainability of the black box model, they use LIME to make IDS reliable.

Kannari, Chowdary and Laxmikanth Biradar [15] presented an IDS that effectively detects, monitors, recognizes, and promptly reacts to network threats. To begin, recursive feature elimination was applied to diminish the high dimensional space, and then an RF classifier was used to identify the attacks. Their suggested model was analyzed on the NSL-KDD and obtained an accuracy of

99.83%. However, the recommended random forest needs a lot of trees, potentially resulting in a decrease in algorithmic efficiency.

Thakkar and Lohiya [16] presented a fusion of feature selection using the difference of mean, median, and standard deviation to identify the most contributed attributes to reduce the feature set. The essential features are recursively added to the feature subset and passed to Deep Neural Networks (DNN). The experiments were done on datasets like UNSW-NB 15, CICIDS-2017, and NSL-KDD and attained an accuracy of 89.03%, 99.80%, and 99.84%, respectively. However, this technique requires more training time, and the class imbalance in the three datasets was not addressed.

Hariharan et al. [17] suggest an XAI-based IDS to provide transparency. They used various methods like SHAP, Permutation Importance (PI), Local Interpretation Model Explainability (LIME), contextual Importance, and Utility algorithms to provide local and global scope for tree-based IDS models like eXtreme Gradient Boosting (XGBM), Random Forest (RF), and LGBM learning. Their results showed that 15 optimal features obtained with PI acquired an accuracy of 92% with the LGBM classifier on the NSL-KDD dataset.

Alani [18] have presented an effective explainable ML model for the industrial Internet of Things. They used Recursive feature elimination and selected 11 optimal features of the WUSTL-IIOT-2021 dataset. Then, they used RF, Logistic Regression, Gaussian Naive Bayes, and Decision Tree (DT) for effective classification. Their outcomes proved that DT works better and recorded an accuracy of 99.97%. Shapley's additive explanations have been used to test for explainability, but the class imbalance was unaddressed.

The following lessons have been learned from studying the literature and are addressed in our work.

- Real-time network traffic contains class dissimilarity. Therefore, IDSs trained using ML approaches on imbalanced datasets perform poorly.
- The issue of data dimensionality influences time consumption, resource utilization, and complexity in data analytics. This can be addressed using feature selection techniques by identifying appropriate attributes in the order of importance.
- Despite the existence of substantial research works aimed at enhancing the explainability and transparency of IDS, there is still scope for explanations and improvements in the field of IDS.

3. Proposed methodology

As illustrated in Fig. 1, the proposed intrusion detection system contains dataset selection, pre-processing, hybrid feature selection, and classification.

3.1. Datasets

The datasets considered for testing the proposed model are CIC-IDS 2017 and UNSW-NB 15 datasets, respectively.

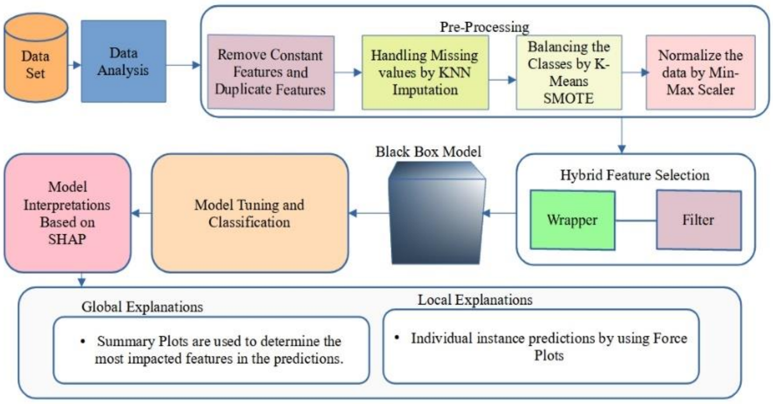
Data Set
Data Analysis
Pre-Processing
Remove Constant Features and Duplicate Features
Handling Missing values by KNN Imputation
Balancing the Classes by K-Means SMOTE
Normalize the data by Min-Max Scaler
Black Box Model
Hybrid Feature Selection
Wrapper
Filter
Model Tuning and Classification
Model Interpretations Based on SHAP
Global Explanations
• Summary Plots are used to determine the most impacted features in the predictions.
Local Explanations
• Individual instance predictions by using Force Plots

Fig. 1. Proposed intrusion detection system framework

3.1.1. CICIDS-2017

The CICIDS dataset was generated by the Canadian Institute of Cybersecurity in 2017. This dataset is considered to be at the forefront of open-source cyber security resources, including the latest instances of cyber-attacks and meeting the necessary criteria for real-world applications in the field of cyber security [19]. The data was gathered for five days, observing 25 users' behaviors that depend on HTTPS, HTTP, email, FTP, and SSH protocols. It contains SQL Injection, Port Scan, DDoS, Botnet, XSS, Infiltration, and Brute Force attacks.

In the suggested study, all categories of attacks are transformed into a single class, assigned the value of one, while the benign class is assigned the value of zero, to facilitate binary categorization. We trained and tested the proposed framework using five CSV files from the CICIDS-2017 dataset, including 79 network flow characteristics and 1405664 records.

3.1.2. UNSW-NB15

It was developed by Moustafa and Slay [20] (2019) in the UNSW cyber security lab. The researchers employed the IXIA PerfectStorm tool to create a combination of authentic, current normal activities and synthetic attack behaviors from network traffic. It has 42 features, three of which are categorical inputs, while the remaining 39 are numerical. In addition, the UNSW-NB15 has separate training and testing files with 175,341 and 82,332 records, respectively. The dataset has nine attack classes and one normal class. Further, all attack classes of the dataset are converted to a single type represented by one and the benign class as zero for binary classification.

3.2. Data pre-processing

After exploratory data analysis on datasets CICIDS-2017 and UNSW-NB 15, we observed that CICIDS-2017 contains noise such as redundant data, NaN (Not a Number), missing values, infinity, negative values, and class imbalance. In addition, UNSW-NB 15 datasets include constant features and class imbalance. The above factors can reduce the classifier's performance. To address the issue, duplicate records and constant features are removed, as shown in Algorithm 1. Furthermore, we imputed negative values with zero and infinity with NaN. Finally, missing or NaN values are imputed by using KNN imputation.

3.2.1. KNN imputation

The KNN Imputer module employs the K-Nearest Neighbors (KNN) technique to replace missing values in datasets. Initially, a missing value for a column is considered, and then k-Nearest neighbors of the features next to the missing value are selected using the weighted average Euclidean distance metric. When there are missing coordinates, the Euclidean distance could be computed by disregarding the missing coordinates. Then, the missing value is altered with a mean of the k-Nearest neighbors. The mathematical form of weighted average Euclidean distance is shown in the next equation:

$$\sqrt{\text{Weight}\times((a_j-a_i)^2+(b_j-b_i)^2)}\,, \tag{1}$$

where, a_i, a_j, b_i, b_j are present coordinates, and $\text{Weight}=\dfrac{\text{Total number of coordinates}}{\text{Number of present coordinates}}$.

3.2.2. Data normalisation and transformation

The technique of putting the features on the same scale is known as normalization. In the proposed work, the Min-Max scaler is used, which scales the values to a range between [0,1] by the following equation:

$$A_{ij}=\frac{A_{ij}-\min(A_j)}{\max(A_j)-\min(A_j)}, \tag{2}$$

where max(A_j) is the maximum value min(A_j) is the minimum value of the j-th feature, and A_{ij} is the normalized value. This process eliminates significant variance and bias of features. Moreover, it is observed that data contains symbolic features. The majority of machine learning techniques need the input of numeric values throughout the training process. We used label encoding to transform the category values into numerical values.

3.2.3. Handling imbalance by k-Means SMOTE

The Class imbalance problem is found to influence data when class distributions are significantly unbalanced. Many classification models have low predicted accuracy when data is unevenly distributed. SMOTE is the most popular method used to handle class dissimilarity, but it is highly susceptible to the influence of noise points by which the model's decision boundary will be damaged, and the training results will be poor. Therefore, in the proposed work, we used k-Means SMOTE, which has three steps: 1) clustering, 2) filtering, and 3) oversampling. The process is as follows.

Step 1. The clustering method splits the dataset into K separate clusters by Euclidian distance and then computes the Imbalance Ratio (IR) of a cluster (f) using the mathematical formula shown in equation

$$\text{IR}(f) = \frac{\text{minority count}(f)}{\text{majority count}(f)}. \tag{3}$$

Step 2. The filtering phase determines which cluster to be oversampled and how many artificial instances must be generated for each cluster based on IR.

Step 3. If the imbalance ratio of the cluster is greater than one, then oversample it with SMOTE to generate synthetic samples in the sparse clusters, as it is less vulnerable to noise creation inside minority areas. Table 1 shows the data distribution before and after handling class dissimilarity.

Table 1. Data distribution before and after balancing the datasets

Dataset	Class	Before balancing	After blanching
UNSW-NB 15	1	132,209	1,273,455
	0	1,273,455	1,273,455
CICIDS-2017	1	119,341	119,341
	0	56,000	119,341

3.3. Hybrid feature selection

The process of feature selection has significant importance during the building of a machine learning model aimed at detecting network intrusion. This is because not every attribute included in the dataset can be equally relevant. A model gets too complicated and performs poorly when used with redundant data containing unnecessary features, it will be considered overfit. Hence the proposed work suggests a hybrid feature selection method that contains filter and wrapper techniques like Pearson correlation coefficient, mutual information, and sequential forward floating selection models to select relevant features, as presented in Algorithm 1.

3.3.1. Pearson Correlation Coefficient (PCC*)*

The PCC, also termed “Pearson Product Moment Correlation” [21], is a statistical measurement of linear association between the two variables, and it ranges between [–1, +1]. For two random variables, *M*, *N* the Pearson relation is termed as

$$\rho(M, N) = \frac{C(M, N)}{(\sigma M, \sigma N)}, \tag{4}$$

where $C(M, N)$ is the covariance between variables M and N, σM and σN are the standard deviations. PCC measures the degree to which two variables are correlated with one another. The greater its absolute value, the more significant the relationship is. The PCC value near 0 indicates that the association is not strong. The presence of a positive value indicates the existence of a positive correlation, while the presence of a negative value shows the existence of a negative correlation.

In our experiments, we tailored the PCC threshold with values 0.7, 0.8, and 0.9; however, in the analysis, we set the PCC threshold value to 0.9 as it yielded better results. For instance, when PCC is employed on the UNSW-NB 15 dataset, we observed that features like ct_srv_src and ct_srv_dst show a significant correlation. The ct_srv_src attribute describes the percentage of connections that have the same service and source address as of the most recent instance. The feature ct_srv_dst describes the percentage of connections that have the same service and destination address as of the most recent instance. A strong association exists between the source and destination addresses since they often occur in pairs within the same service and connections. Similarly other features – dloss, ct_dst_sport_ltm, dwin, ackdat, ct_dst_src_ltm, ct_ftp_cmd, ct_src_dport_ltm, dbytes, is_sm_ips_ports, sbytes, sloss, synack – are identified as correlated hence we removed those features. Despite learning linearly correlated characteristics, this approach does not select non-linearly correlated features. Therefore, we employed a mutual information approach to exclude features that were not linearly connected [21].

3.3.2. Mutual Information (MI)

MI of two random variables measures their mutual dependence in Information theory. MI between two random variables *M* and *N* is a ratio of the volume of information on *N* provided by *M.* If *M* and *N* are independent, then their MI is zero. Mutual information *I*(*M*, *N*) is defined as

$$(5)\qquad I(M,N)=H(M)-H\left(\frac{M}{N}\right)=H(M)+H(N)-H(M,N),$$

where *H*(*M*) is the entropy and $H\left(\frac{M}{N}\right)$ is the conditional entropy. Let us assume the set $M = \{d_1, d_2,\dots, d_n\}$ then the entropy H(M) is defined as

$$(6)\qquad H(M)=-\sum\nolimits_{d\in M}P(d).\log P(d),$$

where *P*(*d*) is the probability distribution of *d*. When two random factors are taken into consideration together, the joint entropy measures the uncertainty, and it is defined as

$$(7)\qquad H(M,N)=-\sum\nolimits_{d\in M}\sum\nolimits_{y\in B}P(d,y).\log P(d,y),$$

when the value of *y* is known, the conditional entropy quantifies how much uncertainty the random variable *d* possesses and it is defined as

$$(8)\qquad H\left(\frac{M}{N}\right)=\sum\nolimits_{y\in N}P(y)\left[-\sum\nolimits_{d\in M}P\left(\frac{d}{y}\right).\log P\left(\frac{d}{y}\right)\right].$$

MI is often regarded as a very effective strategy for recognizing the association between two independent random variables, and it is also influential in determining the relationship between characteristics and class. The features with a high probability of predictive capacity have a lot of information. If the features are independent, then MI will be zero. As the value of MI rises, it indicates greater interdependence between the features [22]. Let D^3 be the set of features selected by PCC and *C* be the label. If the attribute d_f belongs to D^3 and provides positive mutual information, then it is selected in the new subset of features D^4.

In our experiments, we used the “mutual_info_classif” of the “sklearn” library to calculate the mutual information of the variables. On the analysis, we observed that the features Fwd Avg Bytes/Bulk, Bwd Avg Bulk Rate, Bwd Avg Bytes/Bulk, Fwd Avg Packets/Bulk, Bwd Avg Packets/Bulk, and Fwd URG Flags have MI score as 0. We removed those features as they contribute less towards the prediction. Although the aforementioned filter-based feature selection approaches are resistant to overfitting, they fail to identify the appropriate feature subset for classification [23]. To address this, Sequential Forward Floating Selection, a wrapper-based model, is applied to determine relevant features from the feature set D^4.

3.3.3. Sequential Forward Floating Selection (SFFS)

SFFS is a variant of the Sequential Feature Selection (SFS) method. SFS uses a greedy search technique to consolidate *m*-dimensional feature space into an *l*-dimensional feature subspace, where *m* is higher than *l*. This technique initiates with an empty subset feature vector and chooses the first feature in the subsequent phase. Following that, the feature vector is updated with the unused features that provide the highest classification rate. This process is repeated until an appropriate feature subset is produced with minor errors and the highest level of accuracy. The limitation of SFS is that the features cannot be updated once it is included in the subset. To overcome this limitation, a floating variant of SFS, SFFS, is defined by Pudil, Novovičová and Kittler [24]. In SFFS, the attributes added can be discarded at any stage if the feature is least significant. This method selects more accurate features than the filter method [25]. This process contains two phases: 1) inclusion, and 2) conditional exclusion.

In the inclusion phase, the most contributing features are selected by using SFS and added to Z_r. In the conditional exclusion phase, the features in Z_r will be excluded at any point if the features are contributing less, as shown in Step 5 of Algorithm 1. The two stages will be repeated until the required features are selected. The attributes from the initial attribute set will be reduced as a result.

Algorithm 1. Proposed Hybrid Feature Selection

```
Input: Feature set $D = \{d_1, d_2, d_3, ..., d_q\}$
Step 1. Remove duplicate attributes
    if $(d_a == d_b)$ where $a, b = \{1, 2, 3, ..., q\}$; $a \neq b$
    $D^1 = D - d_a$;  $D^1 = \{d_1, d_2, d_3, ..., d_p\}$ , $p \leq q$
  Repeat Step 1 until duplicate attributes are removed
Step 2. Remove constant valued attributes
    Initialize threshold variance $\varnothing = 0.01$
    If $(d_c == \varnothing)$ where $c = \{1, 2, 3, ..., p\}$
    $D^2 = D^1 - d_c$;  $D^2 = \{d_1, d_2, ..., d_o\}$  $o \leq p$
Step 3. Remove linearly correlated features by Pearson correlation
    Initialize threshold $\alpha = 0.9$
    If ( $\rho(d_d, d_e) \geq \alpha$; where $d, e = \{1, 2, 3, ..., o\}$, $d_d \neq d_e$,
```

$$\rho = \frac{\sum_{d=1}^{o}(d_d - \overline{d})(y_d - \overline{y})}{\sqrt{\sum_{d=1}^{o}(d_d - \overline{d})^2 \sum_{d=1}^{o}(y_d - \overline{y})^2}}.$$

$D^3 = D^2 - d_d$ where $D^3 = \{d_1, d_2, ..., d_n\}; \quad n \leq 0.$

Step 4. Remove non-linearly correlated features by calculating the mutual information of each feature

```
    Initialize $\text{corr}_{\text{pos}} = \{\}$
    MI= $\arg\max(I(C; d_f)$ where $f = \{1, 2, 3, ..., n\}, C$ is the class label.
    If (MI $(d_f) > 0)$ then $\text{corr}_{\text{pos}} = d_f$
    Repeat Step 4 for all the features.
    $D^4 = D^3 \cap \text{corr}_{\text{pos}}$ where $D^4 = \{d_1, d_2, d_3, ..., d_m\},\ m \leq n$
```

Step 5. The obtained features from phase 4 are passed to wrapper-based SFFS to select optimal features

```
        Initialize $Z_r = \phi$, op=number of optimal features
        Initialize $r$ = 0
    while (op)
    {
        # Inclusion phase
        $F = \arg\max_{g \in D^4} \varpi(Z_r \bigcup g)$; where $\varpi$ is LGBM accuracy
        $Z_{r+1} = Z_r \bigcup \{g\}; r = r + 1;$ where $r = \{1, 2, 3, ..., m\}$
# Exclusion Phase
        while ( $\varpi(Z_{r+1} - h) > \varpi(Z_r)$) //where $h \in Z_{r+1}$)
        {
           $Z_r = Z_{r+1} - \{h\}; r = r - 1$
        /* repeat inclusion and exclusion until ‘op’ features are selected */
        }
    }
```

Output: Optimal feature set $Z = \{d_1, d_2, ..., d_l\}$ where $l < m$.

The optimal attributes obtained from the proposed feature selection are depicted in Table 2. These features are then passed to classifiers such as DT, LGBM, RF, and Extra Tree (ET). Every model learns parameters automatically during training, while some parameters must be tuned to improve the classifier’s performance. Hence, the Random search CV tuning method is used in the proposed work to select appropriate parameters, as depicted in Table 3.

Table 2. Optimal features

Data set	Number of features	Optimal features selected
CICIDS-2017	5	Fwd IAT Min, Total Length of Fwd Packets, Destination Port, Init_Win_bytes_backward, Init_Win_bytes_forward
UNSW-NB 15	7	proto, service, rate, dttl, smean, dmean, ct_flw_http_mthd

Table 3. List of best hyperparameters obtained using random search

Hyperparameters for CICIDS-2017	
DT	min_samples_leaf = 2, max_depth = 13, min_samples_split = 2, criterion = entropy
LGBM	reg_alpha = 0, num_leaves = 100, min_child_samples = 10, learning_rate = 0.2, max_depth = 10.
RF	n_estimators = 48, min_samples_split = 7, max_depth = 20, min_samples_leaf = 2, criterion = 'entropy'
Extra tree	min_samples_leaf = 5, n_estimators = 20, max_depth = 40, min_samples_split = 20, criterion = 'entropy'
Hyperparameters for UNSW-NB 15	
DT	max_depth=10, min_samples_split=2, criterion=entropy
LGBM	reg_alpha=0.01, max_depth=4, min_child_samples=10, num_leaves=80, learning_rate=0.2,.
RF	min_samples_leaf=4, n_estimators=15, max_depth=20, min_samples_split=2, criterion='gini'
Extra tree	n_estimators = 50, min_samples_split = 5, criterion = entropy, min_samples_leaf = 5, max_depth = 40

3.3.4. Light Gradient Boosting Model (LGBM)

LGBM enhances the performance of Gradient Boosted Decision tree methods by reducing processing time and memory usage while maintaining accuracy [26]. Furthermore, the LGBM uses a histogram technique to limit the effects of accelerated high-dimensional data processing and avoid overfitting. It differs from XGB because it uses a DT methodology. Furthermore, it builds considerably more sophisticated trees by employing a leaf-wise split strategy rather than a level-wise split method, contributing to improved accuracy.

4. Evaluation metrics

We employed performance measures, such as Accuracy (Ac), F1-score, Precision (Pre), area under the ROC Curve, and Recall (Rec) metrics, to test the model’s quality, as depicted in the next equations:

(9) $$\mathrm{Ac}=\frac{(\mathrm{TP}+\mathrm{TN})}{(\mathrm{TP}+\mathrm{TN}+\mathrm{FP}+\mathrm{FN})},$$

$$\text{Re}\,\text{call}=\frac{\text{TP}}{\text{TP+FN}}, \tag{10}$$

$$\text{Precision}=\frac{\text{TP}}{\text{TP+FP}}, \tag{11}$$

$$\text{F1-score}=\frac{(2\times\text{Precision}\times\text{Recall})}{(\text{Precision+Recall})}. \tag{12}$$

The Area Under the ROC Curve (AUC) assesses a model's ability to distinguish between assaults and normal classes and is applied to evaluate the ROC curve. When AUC is higher, the model is more accurate. Where TP (True Positive) and TN (True Negative) denote accurately identified values, whereas FP (False Positive) and FN (False Negative) denote misclassified occurrences, respectively.

5. Results and model interpretations

The experiments of the proposed model were conducted with a workstation having 64 GB RAM Intel Xeon CPU E-3 1271, 3.6 GHz clock speed and 64-bit Windows operating system. By Exploratory Data Analysis (EDA), we observed that datasets are imbalanced, as shown in Table 1. Hence, we conducted experiments with balanced and unbalanced datasets. We used four classifiers to analyse the performance of hybrid feature selection: LGBM, DT, RF, and ET. The evaluation metrics used include F1-score, recall, accuracy, precision, and training time. We analysed the effectiveness of the learning model in two scenarios:

- Case 1: With all features, without class balancing and parameter tuning,
- Case 2: With optimal features, class balancing and parameter tuning.

5.1. Performance analysis of the suggested model on the UNSW-NB 15 dataset

Table 4 shows the outcomes of the classifiers with all features on the UNSW-NB 15 and without balancing the data. The classifiers do not detect attacks accurately without feature selection and class balancing. Out of all models, ET performs better, with an accuracy of 79.11% compared to DT, LGBM, and RF classifiers. Moreover, it is observed that the outcomes of the learning models were not good in terms of all evaluation metrics due to class imbalance and irrelevant features. The F1-score obtained is 47.51% for DT, 48.18% for LGBM, 48.01% for RF and 82.01% for ET. With a training time of 14.82 s, ET achieves a better F1 score than all other models.

Table 4. Performance of learning classifiers with all attributes of the UNSW-NB 15 dataset

Case 1. Without feature selection, class balancing, and hyperparameter tuning						
Classifier	Ac	Pre	Rec	F1-score	AUC	Time, s
DT	52.35	60.37	39.17	47.51	53.83	1.5
LGBM	52.84	60.99	39.81	48.18	82.98	1.04
RF	52.61	60.63	39.74	48.01	54.06	20.04
ET	79.11	77.99	86.46	82.01	78.28	14.82

In the case of imbalanced data, the models do not exhibit better performance. The data is balanced to address this problem, and the suggested feature selection is applied to minimise the attribute space and improve efficiency. From Table 5 it is observed that the efficacy of the model improved on applying the suggested feature selection on balanced data. The performance of the classifiers is tested by varying the attribute subsets (FS-1 to FS-5), having 14, 11, 9, 7, and 5 features, respectively. Our experimental analysis showed that the proposed model works better with seven optimal features with the LGBM classifier.

Table 5. Performance of classifiers with hybrid feature selection and class balancing with parameter tuning on UNSW-NB 15 dataset

Case 2. Proposed model with parameter tunning								
Feature Subset (FS)	Number of features	Model	Ac	Pre	Rec	F1-score	AUC	Time, s
FS-1	14	DT	87.01	83.55	95.13	88.97	86.07	0.142
		LGBM	87.01	82.2	97.52	89.21	85.83	0.81
		RF	87.65	83.37	96.88	89.62	86.61	3.74
		ET	87.11	82.58	97.06	89.24	85.99	2.08
FS-2	11	DT	87.99	85.03	94.89	89.69	87.22	0.15
		LGBM	88.36	84.87	95.96	90.08	87.5	0.9
		RF	88.68	85.65	95.43	90.28	87.92	7.63
		ET	87.03	82.03	82.89	96.32	89.1	9.75
FS-3	9	DT	87.91	85.11	94.6	89.6	87.16	0.16
		LGBM	88.36	84.81	96.05	88.85	87.49	0.945
		RF	88.18	84.96	95.42	89.89	87.37	8.04
		ET	89.34	87.69	93.18	90.65	88.84	7.69
FS-4	7	DT	89.8	90.12	90.81	79.35	89.61	0.13
		LGBM	90.71	90.85	92.43	91.64	90.52	0.438
		RF	90.16	89.5	93.05	91.24	89.84	10.96
		ET	89.81	88.99	93.01	90.59	89.45	6.74
FS-5	5	DT	88.25	92.18	85.95	88.96	88.51	0.18
		LGBM	89.74	88.94	92.92	90.88	89.38	0.85
		RF	89.13	87.34	93.87	90.48	88.6	21
		ET	89.62	88.27	93.57	90.85	89.17	7.085

To further enhance the effectiveness of the recommended system, classifier hyperparameters are adjusted using a Random search CV. It is analysed that when the parameters are tuned, the suggested model performance is boosted in the case of all feature subsets. With a training time of 0.438 s, the LGBM achieves an accuracy of 90.71%, precision of 90.85%, recall of 92.43%, F1-score of 91.64%, and AUC of 90.52%. Comparing all feature subsets, subset FS-4 LGBM is superior.

5.2. Experiment analysis on the CICIDS-2017 dataset

We also have evaluated the proposed model on CICIDS-2017 data to check model generalisation. Tables 6 and 7 show the experimental results of the CICIDS-2017. It is observed from Table 6 that LGBM performs better when compared to DT, RF, and ET by an accuracy of 99.21% as it is less sensitive towards imbalance data. Though DT, RF, and ET exhibit accuracy of around 99%, the other metrics like Pre, Rec and F1-score are not up to the mark as these models are sensitive towards class dissimilarity.

Table 6. Performance of learning models with all features of the CICIDS-2017 dataset

Case 1. Without feature selection, class balancing, and hyperparameter tuning						
Model	A_c	Pre	Rec	F1-score	AUC	Time, s
DT	99.02	98.65	92.73	95.00	98.63	2.72
LGBM	99.21	98.12	98.0	98.14	98.11	5.54
RF	99.11	96.25	93.21	94.70	96.55	352
ET	99.12	97.6	92.10	94.91	96.24	130.8

In the case of imbalanced data, the F1-score is an important metric to consider. DT, LGBM, RF, and ET methods have attained an F1-score of 95%, 98.14%, 94.70%, and 94.91% which is not up to the mark. Hence, to improve the performance of all metrics, Case 2 is applied, and outcomes are provided in Table 7.

Table 7 depicts that, following data balance and the use of the suggested feature selection with hyperparameter adjustment, the performance of all models has increased. It is observed that RF takes more time to train with all sets of features, whereas DT takes less training time. By varying the features as 15, 11, 9, 7 and 5, LGBM has attained an accuracy of 99.91% with 15 features. Further, it was enhanced to 99.98% when five optimal features were selected. When data is balanced, the F1-score is improved for all classifiers with all sub-sets of features (FS-1 to FS-5). With five optimal features, the F1-score of DT was improved by 4.97%, LGBM by 1.87%, RF by 5.25%, and ET by 5.05%. Even though DT takes less time (1.55 s) to train with five features, the proposed model performance is good with five optimal features on LGBM (FS-5). Moreover, the LGBM performance is efficient when compared with Table 6 and Table 7 (FS-1 to FS-4) by an AC of 99.98%, Rec of 99.98%, Pre of 99.97%, AUC of 99.98%, and F1-score of 99.98% with a training time of 3.22 s.

5.3. Model Explainability

The capacity to comprehend and provide an explanation of, how a machine learning model generates predictions is referred to as model explainability. It is a crucial component in applications where the results of model mistakes or biases have substantial implications, such as in healthcare, finance, and IDS. Additionally, it stops classifiers from operating in a “black box” and guarantees that the high accuracy of the classifier achieves transparent facts. In this study, we employ SHAP to provide global and local explanations of our trained model.

Table 7. Performance of classifiers with hybrid feature selection and class balancing with parameter tuning on the CICIDS-2017 dataset

Case2. Proposed model with parameter tunning								
Feature Subset (FS)	Number of features	Model	Ac	Pre	Rec	F1-score	AUC	Time, s
FS-1	15	DT	99.9	99.83	99.9	99.81	99.9	2.6
		LGBM	99.91	99.94	99.91	99.91	99.91	4.77
		RF	99.95	99.95	99.95	99.95	99.95	138.02
		ET	99.92	99.98	99.87	99.92	99.92	70.3
FS-2	11	DT	99.95	99.94	99.96	99.95	99.95	2.76
		LGBM	99.93	99.88	99.99	99.93	99.93	3.57
		RF	99.93	99.93	99.92	99.93	99.92	126.65
		ET	99.95	99.94	99.94	99.94	99.94	118.25
FS-3	9	DT	99.94	99.94	99.94	99.94	99.94	2.66
		LGBM	99.96	99.96	99.97	99.96	99.96	3.3
		RF	99.96	99.98	99.94	99.96	99.96	164
		ET	99.96	99.96	99.96	99.96	99.96	130
FS-4	7	DT	99.94	99.94	99.92	99.94	99.94	2.29
		LGBM	99.96	99.95	99.96	99.95	99.95	3.45
		RF	99.95	99.94	99.95	99.94	99.95	136.81
		ET	99.94	99.97	99.9	99.94	99.94	18.8
FS-5	5	DT	99.95	99.95	99.95	99.95	99.95	1.55
		LGBM	99.98	99.97	99.98	99.98	99.98	3.22
		RF	99.95	99.96	99.96	99.96	99.95	120.17
		ET	99.96	99.96	99.96	99.96	99.96	29.97

5.3.1. Global explanation

SHapley Additive exPlanations (SHAP) were introduced by Lundberg, Allen and Lee [27]. The fact that it is a model-agnostic explainer has the potential to generate general explanations regardless of the type of classifier that was employed. We use SHAP values to quantify the influence of each attribute, which aids in understanding the impact of the attribute on the inference output more clearly. The proposed model employs Tree SHAP, a variant of the SHAP model.

Figs 2-3 depict the summary plots of UNSW-NB 15 and CICIDS-2017 datasets. It provides global explanations of the optimal features for the LGBM classifier and shows a graphical representation of the connection among the Shapley value of an attribute and its impact on the outcome. Each point on the summary plot denotes the Shapley value associated with an attribute and each occurrence of that feature. The SHAP values are shown along the *x*-axis. On the *y*-axis, every attribute is graded based on its level of significance. The attribute at the very top is the one that makes the most significant contribution to the forecasts, while the one at the very bottom makes less contribution. No contribution is made when the value on the *x*-axis is equal to zero, and the magnitude of contributions exhibits an upward trend when the SHAP value deviates from zero.

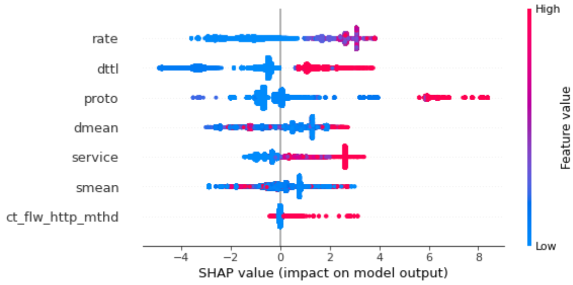

Fig. 2. Summary plot of UNSW-NB 15

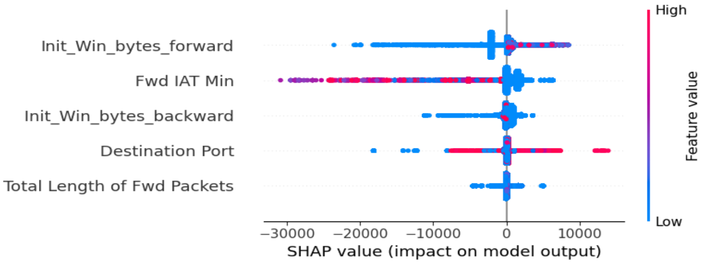

Fig. 3. Summary plot of CIC-IDS 2017

The attribute values represented by the dots on the left side of the axis pull the prediction value downwards, towards “normal,” while the values located on the right side of the axis exert an upward influence on the forecast value, causing it to fall into the “attack” category. The red dots indicate a high value for the attribute, while the blue dots indicate a low value.

5.3.2. Local explanations

It helps users understand how the model arrives at a specific prediction for a given data point by decomposing the prediction into contributions from each feature. The visual representation illustrates the impact of each feature on the model’s output when they diverge from their respective base values. The base value is determined by taking the mean forecast of the dataset, which is based on every feature. It acts as a reference point around which the contributions from individual features are measured. In the plot, red denotes attributes that had a positive impact on the model score, while blue denotes features that hurt the score. Scores that are higher cause the model to predict an attack, whereas scores that are lower cause the model to predict a non-attack.

Fig. 4 shows the force plot of instance 289 of the UNSW-NB 15 test sample which is labelled as non-attack. The score assigned to this instance by the model is shown in bold, i.e., –7.90, which is less than the base value. It is observed that blue-coloured features protocol, domain and rate, etc., are forcing the classification towards lower, i.e., non-attack while service and dttl drive towards higher to make the classification as an attack.

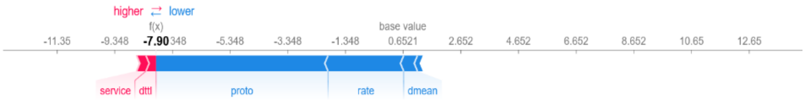

Fig.4. Force Plot of an instance of UNSW-NB 15

Similarly, by observing Fig. 5, the model scores –11.14 for the instance 103356 of the CIC-IDS 2017 dataset, the features Fwd IAT Min, Init_Win_backward, and Total Length of Fwd Packets are the most contributed features, which makes the instance to classify as non-attack.

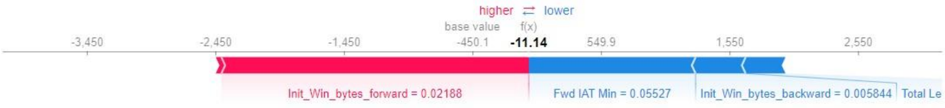

Fig. 5. Force plot of an instance CICIDS-2017

5.4. Discussion

This study highlights the use of both filter and wrapper approaches that contribute to an enhanced feature selection process that is more suitable for interpretation. Filter techniques are used to get an early comprehension of the importance of features, while wrapper approaches give insights into the precise influence of features on the selected learning process. Further, the explanations that are provided in this work help interpret the decisions that are made using the technique. The limitation of the proposed method is that the wrapper-based SFFS algorithm may incur significant computing costs, particularly when used to high dimensional datasets.

5.5. Comparison of the proposed model with existing methods

Table 8 presents a comparative analysis between the proposed model and other existing models. To establish assessment metrics about existing models, an analysis is conducted on two well-recognized IDS datasets that are publicly accessible. Various feature selection techniques, such as information gain [12], Fusion of Feature Selection (FSI) [16], Entropy + SFS [28], and CFS-FP [29], are employed in the existing works on the CICIDS-2017 dataset that generate 38, 64, 7, and 30 optimal features, respectively. Similarly, IGRF-RFE [13], DNN[16], XGBM [31], ET [32], RF [33] and GA-RF [34] are used to obtain 23, 21, 17, 22, 20, and 16 optimal features of the UNSW-NB 15 dataset. With the suggested feature selection, we have attained 5 and 7 optimal features of CICIDS-2017 and UNSW-NB 15 data sets, respectively, which are less when compared with existing feature selection methods. The outcomes show that the proposed model is effective with less optimal features.

The efficacy of the suggested methodology has been shown via the assessment of Pre, Recall, and F1-score evaluation measures. In the context of IDS classification, the recall evaluation metric pertains to the accurate identification of attack data samples as attacks.

The precision evaluation metric relates to the fraction of properly recognized attack data samples out of the total anticipated attacks. Therefore, when evaluating the precision and recall metrics, it is desirable for IDS to accurately detect intrusions while paring down the occurrence of false positives. This suggests that a low recall signifies a significant number of undetected assaults, whereas a low Pre indicates a substantial number of wrongly anticipated attacks.

Table 8. Comparison between the suggested approach and existing approaches

Dataset	Model	Feature selection	Number of features	Ac	Pre	Rec	F1-score	AUC	XAI	Time, s
CICIDS-2017	DNN-KNN [12]	Information gain	38	99.85	99.87	99.87	99.87	NA	NA	NA
	DNN[16]	FSI	64	99.8	99.85	99.94	99.89	NA	NA	27,719
	Weighted k-Means [28]	Entopy +SFS	7	NA	NA	98.86	NA	NA	NA	1
	EL [29]	CFS-FP	30	99	NA	NA	NA	NA	NA	5
	ResNet-18 [30]	Feature fusion	NA	99.78	99.82	99.79	99.8	NA	NA	106
	Proposed model	PCIG-SFFS-LGBM	5	99.98	99.97	99.98	99.98	99.98	Yes	3.22
UNSW-NB 15	MLP [13]	IGRF-RFE	23	84.24	83.6	84.24	82.85	NA	NA	NA
	DNN[16]	FSI	21	89.03	95	98.95	96.93	NA	NA	13,913
	Simple RNN [31]	XGBM	17	88.13	NA	NA	99.58	NA	NA	225.46
	FFDNN [32]	ET	22	87.48	NA	NA	NA	NA	NA	NA
	DNN [33]	RF	20	82	NA	NA	NA	NA	NA	0.089
	RF [34]	GA-RF	16	87.61	NA	NA	NA	98	NA	2.2
	Proposed model	PCIG-SFFS-LGBM	7	90.71	90.85	92.43	91.64	90.52	Yes	0.43

*NA – Not Applicable

Based on the findings presented in our study, it can be deduced that the suggested approach demonstrates effectiveness in terms of Pre and recall evaluation metrics. The proposed approach exhibits high Pre, Recall, and F1-score values across the CICIDS 2017 dataset. While considering the UNSW-NB 15 dataset, Thakkar et al. [15] got higher Pre, Recall, and F1-score values than our work. This is because the authors have not addressed the issue of class imbalance in the UNSW-NB. In the case of imbalanced data when the majority class has significant influence within the dataset, a model can get a high accuracy by simply predicting all cases belonging to the majority class. Nevertheless, the metrics Pre, Recall, and the F1-score exhibit more sensitivity towards the performance of the minority class, resulting in potentially higher values when the model effectively identifies instances belonging to the minority class.

6. Conclusion and future scope

High-dimensional data and class imbalance are serious problems in network intrusion detection systems. These problems could lead to low detection accuracy. The proposed work employs feature selection and data resampling techniques to address this issue. Initially, the data was pre-processed, and then k-Means SMOTE was used to balance the minority samples. A hybrid feature selection-based detection model is proposed, and it results in 7 and 5 optimal features of UNSW-NB 15 and CICIDS-2017 datasets. Then, the features are trained using LGBM to assess the performance of the classifier. Further, to enhance interpretability and develop trust, our work employs global and local interpretations using SHAP; it gives essential insights into the interpretability of the proposed IDS. This work can be further expanded using a meta-heuristic approach to choose the most useful characteristics for multi-class classification and detecting attacks in IoT and SDNs.

References

1. Udas, M., E. Karim, K. S. Ro. SPIDER: A Shallow PCA-Based Network Intrusion Detection System with Enhanced Recurrent Neural Networks. – Journal of King Saud University – Computer and Information Sciences, Vol. **34**, 2022, No 10, pp. 10246-10272.
2. Wang, K. Zheng, Y. Yang, X. Wang. An Explainable Machine Learning Framework for Intrusion Detection Systems. – IEEE Access, Vol. **8**, 2020, pp. 73127-73141.
3. Premkumar, T., V. P. Sundararajan. DLDM: Deep Learning-Based Defense Mechanism for Denial of Service Attacks in Wireless Sensor Networks. – Microprocess. Microsystems, Vol. **79**, 2020, No August, 103278.
4. IBM Security Cost of a Data Breach Report 2022. 2022.
5. Fleck, A. Inflation Becomes the Leading Global Concern in 2022. – Statista, 2022 (Accessed 22 June 2023).
   https://www.statista.com/chart/28878/expected-cost-of-cybercrime-until-2027/
6. Alhenawi, H., R. Alazzam, O. Al-Sayyed, Abualghanam, O. Adwan. Hybrid Feature Selection Method for Intrusion Detection Systems Based on an Improved Intelligent Water Drop Algorithm. – Cybernetics and Information Technologies, Vol. **22**, 2022, No 4, pp. 73-90.
7. Batchu, H. Seetha. An Integrated Approach Explaining the Detection of Distributed Denial of Service Attacks. – Computer Networks, 2022, 109269.
8. Mallampati, H. Seetha. A Review on Recent Approaches of Machine Learning, Deep Learning, and Explainable Artificial Intelligence in Intrusion Detection Systems. – Majelisi Journal of Electrical Engineering, Vol. **17**, 2023, No 1, pp. 29-54.
9. Kim, L. Y., H. Kim. Network Intrusion Detection Based on Novel Feature Selection Model and Various Recurrent Neural Networks. – Applied Sciences, Vol. **9**, 2019, No 7.
10. Roy, J., B. Li, Choi, Y. Bai. A Lightweight Supervised Intrusion Detection Mechanism for IoT Networks. – Futurre Genereration Computer Systems, Vol. **127**, 2022, pp. 276-285.
11. Saha, A., T. Priyoti, A. Sharma. Towards an Optimised Ensemble Feature Selection for DDoS Detection Using Both Supervised and Unsupervised Method. – In: Proc. of 19th Annual Consumer Communications and Networking Conference (CCNC’22), Las Vegas, N. V., USA, 2022.
12. De Souza, C., B. Westphall, R. B. Machado, J. B. M. Sobral, G. dos S. Vieira. Hybrid Approach to Intrusion Detection in Fog-Based IoT Environments. – Computer Networks. Vol. **180**, 2020.
13. Yin, Y., et al. IGRF-RFE: A Hybrid Feature Selection Method for MLP-Based Network Intrusion Detection on UNSW-NB15 Dataset. – Journal of Big Data, Vol. **10**, 2023, No 1.
14. Patil, S., et al. Explainable Artificial Intelligence for Intrusion Detection System. – Electronics, Vol. **11**, 2022, No 19.
15. Kannari, N., S. Chowdary, R. Laxmikanth Biradar. An Anomaly-Based Intrusion Detection System Using Recursive Feature Elimination Technique for Improved Attack Detection. – In: Theory of Compututer Science. Vol. **931**. 2022, pp. 56-64.
16. Thakkar, A., R. Lohiya. Fusion of Statistical Importance for Feature Selection in Deep Neural Network-Based Intrusion Detection System. – Information Fusion, Vol. **90**, 2023, No February, pp. 353-363.
17. Hariharan, R., R. Rejimol Robinson, R. R. Prasad, C. Thomas, N. Balakrishnan. XAI for Intrusion Detection System: Comparing Explanations Based on Global and Local Scope. – Journal of Computer Viroogy. Hacking Technologies, Vol. **19**, 2023, No 2, pp. 217-239.

18. Alani, M. M. An Explainable Efficient Flow-Based Industrial IoT Intrusion Detection System. – Computers Electrctriacal Engginering, Vol. **108**, 2023, No April, 108732.

19. Sharafaldin, A., H. Lashkari, A. A. Ghorbani. Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization. – In: Proc. of Int. Conf. on Systems Security and Privacy, 2018, No Cic, pp. 108-116.

20. Moustafa, N., J. Slay. UNSW-NB15: A Comprehensive Data Set for Network Intrusion Detection Systems (UNSW-NB15 Network Data Set). – In: Proc of Mil. Commun. Inf. Syst. Conf. (MilCIS'15), 2015, No November.

21. Veera Brahmam, M., S. Gopikrishnan, K. Raja Sravan Kumar, M. Seshu Bhavani. Pearson Correlation Based Outlier Detection in Spatial-Temporal Data of IoT Networks. – In: Proc. of Innov. Data Commun. Technol. Appl. Proc. ICIDCA 2021, Singapore, Springer, Nature, Singapore, Vol. **96**, 2022, pp. 1019-1028).

22. Silva, N., R. de Oliveira, D. S. V. Medeiros, M. A. Lopez, D. M. F. Mattos. A Statistical Analysis of Intrinsic Bias of Network Security Datasets for Training Machine Learning Mechanisms. – Annals of Telecommunications,Vol. **77**, 2022, pp. 555-571

23. Vergara, P., A. Estévez. A Review of Feature Selection Methods Based on Mutual Information. – Neural Computer Applications, Vol. **24**, 2014, No 1, pp. 175-186.

24. Pudil, J., Novovičová, J. Kittler. Floating Search Methods in Feature Selection. – Pattern Recognit. Lett., Vol. **15**, 1994, No 11, pp. 1119-1125.

25. Shirbani, F., H. Soltanian-Zadeh. Fast SFFS-Based Algorithm for Feature Selection in Biomedical Datasets. – Amirkabir Journal of Science and Technology, Vol. **45**, 2013, No 2, pp. 43-56.

26. Ke, G., et al. LightGBM: A Highly Efficient Gradient Boosting Decision Tree. – In: Proc of Conference on Neural Information Processing Systems, Vol. **2017**, 2017, No December, pp. 3147-3155.

27. Lundberg, P., G. Allen, S.-I. Lee. A Unified Approach to Interpreting Model Predictions (online).
https://github.com/slundberg/shap

28. Gu, K., Li, Z. Guo, Y. Wang. Semi-Supervised k-Means DDOS Detection Method Using Hybrid Feature Selection Algorithm. – IEEE Access, Vol. **7**, 2019, pp. 64351-64365.

29. Mhawi, A., Aldallal, S. Hassan. Advanced Feature-Selection-Based Hybrid Ensemble Learning Algorithms for Network Intrusion Detection Systems. – Symmetry (Basel), Vol. **14**, 2022, No 7.

30. Fu, J., X. Lan Zhang. Gradient Importance Enhancement Based Feature f Fusion Intrusion Detection Technique. – Computer Networks, Vol. **214**, 2022 No May, 109180.

31. Kasongo, Sydney Mambwe. A Deep Learning Technique for Intrusion Detection System Using a Recurrent Neural Networks Based Framework. – Computer Communications, Vol. **199**, 2023, pp. 113-125.

32. Kasongo, Y. Sun. A Deep Learning Method with Wrapper Based Feature Extraction for Wireless Intrusion Detection System. – Computers Security, Vol. **92**, 2020.

33. Eunice, Q., M. Gao, Y. Zhu, Z. Chen, N. Lv. Network Anomaly Detection Technology Based on Deep Learning. – In: Proc. of 3rd Int. IEEE Conf. Front. Technol. Inf. Comput. ICFTIC 2021, pp. 6-9.

34. Kasongo. An Advanced Intrusion Detection System for IIoT Based on GA and Tree Based Algorithms. – IEEE Access, Vol. **9**, 2021, pp. 113199-113212.

Received: 27.09.2023; Second Version: 25.11.2023; Accepted: 14.12.2023