ISSN: 1687-1006 p
ISSN: 2682-3640 e

The Egyptian International Journal of Engineering Sciences and Technology

https://eijest.journals.ekb.eg/

Vol. 52 (2025) 105–118

DOI: 10.21608/eijest.2025.350769.1318

Evaluating Feature Selection Methods for Machine Learning Models in Cybersecurity

Anas N. Moursi[a*], Mahmoud Atallah[a], Nesreen I. Ziedan[b]

[a] *Cyber Security, Faculty of Computer Studies, Arab Open University, El-Sorouk, Egypt,*
[b] *Computer and Systems Engineering Department, Faculty of Engineering, Zagazig University, Zagazig, Egypt*

ARTICLE INFO

Article history:

Received 11 January 2025
Received in revised form
02 February 2025
Accepted 02 February 2025
Available online 02
February 2025

Keywords:

Cybersecurity
Machine Learning
Feature Selection
LightGBM

ABSTRACT

Cyber-attack incidents are increasing daily, with the adoption of modern communication technologies, cloud services, and the Internet of things. Providing high accuracy real-time protection for networks against network vulnerabilities is of paramount importance. In machine learning, one of the crucial items, which influence models' performance enhancement in detecting and preventing these threats, is feature selection. This paper evaluates two feature selection methodologies, which are: (1) feature selection using traditional statistical approaches, such as Mutual Information (MI) and correlation-based; and (2) automated feature selection using embedded methods, such as LightGBM. The evaluation is performed on six established cybersecurity datasets which are CIC-DDoS2019, ISCX-IDS2012, UNSW-NB15, CIC-IDS2017, NSL-KDD, and CSE-CIC-IDS2018.

The datasets are used to train and test various models. Each feature selection methodology is applied to get the optimal combination of features. Subsequently, a comparison analysis of multiple metrics, including time cost, is conducted across the models. The findings show that there is a huge variation in model performance, regardless of the dataset or the feature selection methodology. The time cost reduced significantly for the models with LightGBM feature selection method. Some models improved their metrics when using LightGBM. This makes LightGBM a promising choice in cybersecurity applications.

1. Introduction

The evolution of the information technology sector facilitated communication over networks. The attack surface has expanded significantly. The security threats that are facing these systems and devices are increasing massively. This has led to more difficulties to achieve robust network security. Different toolkits and services are being used as the foundation of network security defense. The intrusion detection and prevention systems (IDS/IPS) and the network intrusion and detection systems (NIDS) are part of these tools [1].

The cyber threats landscape is always changing. This adds more challenges to the defense mechanisms, which require advanced and flexible security measures. Machine learning (ML) has become a vital tool for tackling these issues, allowing for automated threat detection, the anomaly recognition, and proactive defense strategies [2]. In Intrusion Detection Systems, ML models can examine large volume of network traffic, system logs, and user behavior data to spot patterns that suggest malicious activities [3]. The effectiveness of

* Corresponding author. Tel.: + 201224633115
E-mail address: anas.naguib@outlook.com

these ML models depends on the quality, relevance, and selection of features used. Feature selection is essential for boosting model performance by reducing the dimensionality, preventing overfitting, and enhancing interpretability. High-dimensional datasets often include irrelevant or redundant features, which can negatively impact model accuracy and training efficiency. Feature selection can help ML models focus on the most informative features, thereby delivering more accurate predictions by concentrating on relevant patterns [4]. Additionally, feature selection can improve the interpretability of ML models in cybersecurity. By highlighting the most influential features that are driving model decisions, the security analysts will understand the reasoning behind threat detection and response activities, building trust and informing better decision-making.

Development of ML models that can handle massive amounts of traffic in modern networks requires that the model should perform the malware identification in real-time. This can be achieved by increasing the processing resources, but it would not be a practical solution as it would increase the hardware required and the cost of the cybersecurity systems. On the other hand, developing robust and effective ML models for cybersecurity applications faces challenges due to the complexities of real-world cyber threat data. Selecting the most relevant features from often high-dimensional datasets poses a critical hurdle.

Cybersecurity datasets have many characteristics that make feature selection difficult, which are:

- High Dimensionality: The datasets may contain hundreds or even thousands of features that represent a variety of network traffic patterns, system logs, user behaviors, and other potential indicators of compromise [5].
- Imbalanced Class Distribution: Most cyberattacks happen in rare cases compared to normal activities. This leads to bias in models toward the majority class due to an imbalance in the dataset, resulting in poor detection of actual threats.
- Feature Redundancy and Noise: Datasets may contain features that are redundant or irrelevant, which can impact model performance negatively and increase training time. Identifying and removing such noise is crucial for effective feature selection [6].

For example, ISCXFlowMeter software [7], more than 80 features can be generated for characterizing data traffic; for the network analyst this is challenging to handle this huge amount of information. In this regard, a very important feature selection pre-processing step is highly valued. Lack of a standardized approach to feature selection in cybersecurity exacerbates these challenges further.

This paper compares two feature selection methodologies with the objective of finding the most effective methodology that would improve the performance of ML models in cyber threats detection and mitigation. The test of these methods on different datasets is performed, including old and modern datasets. Nine widely used machine learning models in cybersecurity tasks have been trained and tested on a wide variety of publicly available datasets. Therefore, the performances of these models have been compared using various methodologies to highlight the influence of feature selection on different model metrics. This paper's contributions are as follows:

- Comparative Analysis: This provides a comparison of how statistical feature selection and LightGBM approaches draw valuable insights from the cybersecurity datasets regarding the various strengths and weaknesses associated with using each method.
- Performance Enhancement: Finding the best feature selection technique for any given dataset and model could significantly improve performance in intrusion detection and other security tasks.
- Practical Guidelines: The study provides guidelines that are useful for researchers and practitioners in the choice of appropriate feature selection methods to optimize machine learning models for cybersecurity applications.

The remainder of this paper is organized as follows. Section 2 provides an overview of related work. Section 3 explains the methodology of the research, describing the process, the used datasets, and the ML models. Section 4 explores the results of each ML model on the various datasets. Section 5 concludes the paper.

2. Literature Review

ML and DL are technologies on which intrusion detection systems depend to enable real-time anomaly and attack detection on networks [3]. AI in cybersecurity [8] is also expected to make advances in the predictive threat hunting and automating the process that analyzes potential threats before they can become actual threats. This predictive capability will be useful in dealing with advanced forms of cyber

threats.

Additionally, AI will be able to personalize security, user activity countermeasures, and system configurations, strengthening overall security. However, there is a problem with increased usage of AI in cybersecurity, such as the vulnerabilities that the enemies use to launch attacks [8].

Moreover, when it comes to real applications, attackers exploit vulnerabilities to carry out cyber-attacks. Recent attacks have exploited the vulnerabilities of IoT systems in smart cities [9].

Previous research has shown that feature selection is a critical aspect of ML models. [10] presented a computationally efficient filter-based method for feature selection on correlation for data containing high features count. Their study has shown that removing redundant and irrelevant features can significantly improve the learning efficiency and model accuracy. According to the authors, one critical challenge in feature selection involves high-dimensional spaces, which they proclaim is crucial to be resolved to improve machine learning performance. The results here confirm the importance of the step of data preprocessing; indeed, it is the preliminary step in classification task optimization.

Multiple ways can be used to categorize the Feature selection methods. The most common way is to categorize it into filter, wrapper, embedded, and hybrid methods [11].

A well-balanced perspective on feature selection was offered by [5]. It addressed not only established methodologies but also crucial emerging issues such as class imbalance, dataset shift, and scalability. The discussion of real-world case studies and the identification of key open research questions make it a source of great value for practitioners and researchers alike in the field. [4] An overview of feature selection methods from traditional filters to recent hybrid approaches, is presented in [4]. This paper systematically classifies various methods and emphasizes their application across different domains, such as text mining, image processing, and bioinformatics.

Software probes are being used in modern NIDS. These probes are responsible for analyzing network traffic based on some characteristics, including percentages of forward and reverse flows, arrival time distribution, packet size distribution, and the presence of a particular TCP/IP flags. This information is usually instrumental in underpinning anomalous traffic, usually the cause of distributed denial of service (DDoS) attacks [12], [13], covert VoIP sessions [14], threat diffusion [15], and Peer-2-Peer traffic. The conventional signature-based NIDS would allow these different flows to go unobserved in many instances.

Machine learning model developed for cybersecurity improves its performance based on the selected features. [16], in the concluding remarks of their survey on the various supervised feature selection techniques employed in NIDS, pointed out the feasibility of using feature selection in networks for practical benefits. Using recent datasets and a broad spectrum of feature selection algorithms for experimental evaluation, they showcase the ability of feature selection in reducing computational time, often by a factor, while the effects on model performance remain low. The authors have remarked that their assessment framework serves to support network/security management practitioners on the importance of feature selection toward identifying the most key characteristics for accurate, effective, and interpretable models within NIDS. The paper also identified promising future directions involving unsupervised feature selection techniques, streamed data, the analysis of adaptive techniques dealing with the dynamic nature seen in modern network environments to stand as a bedrock in facilitating further innovation in related domains.

The survey conducted by [6] was a good addition to the field of cyber-attack feature detection. It studied the intrinsic details in addition to probing the filter-based feature selection methods. It also explored key technical components such as search algorithms and relevance measures.

3. Methodology

Network flow software, such as Cisco's NetFlow, introduces massive and high-dimensional characteristics. Most datasets in the cybersecurity field are imbalanced. To overcome this challenge, oversampling and under-sampling are used in intrusion detection using machine learning [17]. The process block diagram is described in Fig. [1].

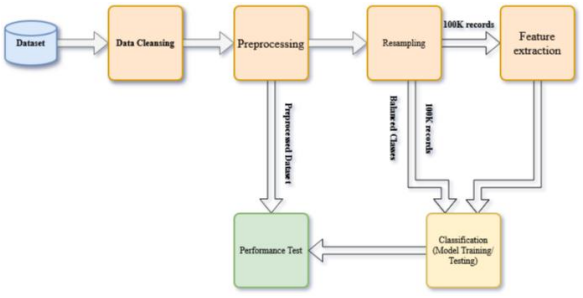

Fig. 1 Block Diagram of the Process

The proposed method includes modules for data cleansing, preprocessing, resampling, feature extraction, classification, and performance testing. In the data cleansing module, each dataset is cleaned to avoid unnecessary features, null values, and duplicates. The data preprocessing module includes normalization, scaling, and encoding of text features. Section 3.1 describes both cleansing and preprocessing modules.

The feature extraction module contains two parallel methods for feature selection. Each methodology is used for feature selection on the resampled data, before applying the training module. The feature selection methodologies are described in section 3.2.

In the classification module, different models, including gradient boosting, XGBoost, and LightGBM, are evaluated.

In the performance test module, the full datasets are used for evaluation on each model. The classification module and performance test module are described in section 3.3.

3.1. Datasets, cleansing and preprocessing.

Multiple datasets have been introduced to the research area over the past years. These datasets vary in their features, classes, and size. This paper evaluates the impact of feature selection on multiple datasets. These datasets are summarized as follows:

NSL-KDD: The NSL-KDD dataset is a filtered version of the original KDD Cup 1999 dataset for network IDS evaluation. It contains labeled traffic data of the network categorized either as "normal" or as one of several types of network attacks. This dataset is a multivariate one, containing features that describe the traffic, such as protocol type, service, and connection status, as well as other derived features like the failed login attempts count and the duration of connections. 51.88% of the records are regular traffic, while non benign traffic makes up 48.12% [18].

UNSW-NB15: It is considered one of the modern network traffic datasets created for the purpose of IDS evaluation. The University of New South Wales generated this dataset, which contains approximately 2.5 million records. Both normal and malicious traffic are characterized by 49 features. Nine attack categories are included. These categories represent a broad range of cyberattacks. In this regard, intrusion detection systems and machine learning models can use this helpful resource to be developed and tested [19].

ISCX-IDS2012: ISCX-IDS2012 is a publicly available dataset widely used in developing and evaluating IDS. The Information Security Centre of Excellence (ISCX) developed it by capturing data from an operating real-world network environment that involved benign and malicious activities. The dataset consists of attack scenarios, such as DoS, DDoS, port scans, and other network intrusions, combined with normal traffic. This is an essential source of packet and flow level features, in detail that can be used to learn ML models and IDS algorithms can. The ISCX-IDS2012 dataset serves as a realistic benchmark for testing the efficiency of network security systems in detecting and preventing cyber-attacks [20].

CIC-IDS2017: The Canadian Institute for Cybersecurity (CIC) generated this dataset, which is a comprehensive collection of network traffic. Real network flow was included with benign and malicious activities contained. It covers many attack types. The dataset is considered high-dimensional, featuring over 70 flow-based and packet-based attributes [21].

CIC-DDoS2019: This dataset is aimed to facilitate research on DDoS attacks. It consists of labeled records for both benign and malicious traffic. The dataset was constructed by capturing multiple attack scenarios with varying intensities from real-world network environments. Features consist of packet size, packet duration, flow rate, and protocol type among many others. Its wide range of usage is in testing and training ML models for protection techniques against DDoS. It has been constructed by CIC for research and development in cybersecurity [22].

CSE-CIC-IDS2018: It is a dataset containing network traffic used for the IDS evaluation. It includes normal and attack scenarios, such as DDoS, brute force, and SQL injection, among others, with detailed features. It contains over 80 features per flow, including packet count, flow duration, protocol type, and more [21].

In the cleansing module, each dataset is cleaned of features considered to be metadata features [23]. Duplicate and null data are also removed from the used datasets. Text features are converted to numerical values using categorical encoding for features with a small number of categories and one-hot encoding for features with a large number of categories [24].

To accelerate model training, methods such as data normalization are necessary. In this paper, the

maximum and minimum normalization scaling equation [25] is adopted to normalize the data in the range of [0,1]. The maximum-minimum normalization equation is as follows:

$$MaxMinScaler = \frac{X - X_{min}}{X_{max} - X_{min}} \text{ [26]}, \quad (1)$$

where $\mathbf{X_{max}}$ and $\mathbf{X_{min}}$ represent values of both maximum and minimum data in each column in which the feature **X** is located, respectively.

The target label of each dataset is converted into two classes {0 => Normal, 1 => Anomaly} to prepare for binary classification.

3.2. Feature selection methodologies

Two distinct feature extraction methods are applied in parallel to each dataset. The first method is statistically based, using a correlation matrix heatmap to identify highly correlated features within the dataset. A threshold has been applied: if the two features exhibit a correlation higher than 90%, they are considered highly correlated. The number of highly correlated features are reduced by dropping one feature from each highly correlated pair. The correlation between the remaining features and the target label has been measured to ensure that the selected features have a meaningful relationship with the target label. Next, the MI score between the target label and features has been calculated to obtain and used for further filtering. Features with MI score less than 0.001 have been dropped. The following equation conducts the MI score [27].

$$I(X;Y) = \sum_{x \epsilon X} \sum_{y \epsilon Y} p(x,y) \log\left(\frac{p(x,y)}{p(x)p(y)}\right) \quad (2)$$

where:

- **X** and **Y** are random variables.
- **x** represents the variable **X** specific value.
- **y** represents the variable **Y** specific value.
- **p (x, y)** is **X** and **Y**'s joint probability mass function. It is the likelihood that **X** will take the value **x,** and **Y** will take the value **y**.
- **p (x)** is the marginal probability mass function of **X**. It is the likelihood that **X** will take the value **x**, independent of **Y**'s value.
- **p (y)** is the marginal probability mass function of **Y**. It is the likelihood that **Y** will take the value **x**, independent of **X**'s value.
- **log** is the logarithm (typically base-2 for information in bits, or natural log for nats). You'll often see log base 2 used when thinking about bits, and natural log when dealing with entropy calculations. The difference is just a constant scale factor, so it doesn't really affect what the MI is telling you.

[28] stated that a special case of feature extraction methods is feature selection.

The second method involves using LightGBM to calculate the feature importance for each dataset and then filtering the features with an important score higher than 80. This threshold is determined through trial and error from multiple values including 50, 70, 80, and 100.

3.3. ML Models

In recent years, research on ML-based IDS models have demonstrated reasonable detection rates. have Algorithms including KNN, SVM, Random Forests, Neural Networks, Naive Bayes, XGBoost and LightGBM had been used in previous studies. According to [29] .

Nine different ML models are used in this paper in training and testing two methodologies for feature selection. The models are used with their standard parameters to isolate and evaluate the effect of the feature selection methods under consideration.

Support Vector Machine (SVM): The SVM is a model of ML that can perform classification and regression. It tries to find the best possible hyperplanes that separate classes of data. SVMs work well for high-dimensional data and support multi-type kernel functions. However, there are some disadvantages of SVMs: require accurate tuning, especially in those cases where the number of input dimensions exceeds the number of samples, large-sized datasets, multi-classification problems, and imbalanced datasets result in poor performance of SVMs [30].

Logistic Regression: This is an ML model generally used in binary classification; however, it can also be used in multiclass tasks using the one-vs-rest approach. Using a linear model and the sigmoid function, it produces outputs ranging from 0 to 1, where values closer to 1 indicate a higher likelihood of belonging to a specific class [31].

Decision Tree (DT): This is a supervised ML techniques used for solving classification and regression problems. It has a tree-like structure, making it easy to interpret and visualize using various ML tools [32].

Gaussian Naive Bayes: This is a probabilistic ML classifier based on Bayes' theorem, which assumes that some features follow a Gaussian distribution. It is simple, efficient, and particularly effective for large datasets with independent features [33].

Bernoulli Naive Bayes: It is a probabilistic model

that can be used for both binary and multi-class classification problems. It assumes that features are binary (i.e., either 0 or 1) and uses Bayes' theorem to predict class probabilities. This classifier works well for tasks where the features indicate the presence or absence of a given attribute [34].

Gradient Boosting Classifier: It is a ML model that sequentially constructs a series of decision trees. By focusing on incorrectly classified data points, each tree fixes the mistakes of the one before it. It works well for both classification and regression tasks by optimizing a loss function using gradient descent. Gradient Boosting boasts outstanding accuracy and is widely used in cybersecurity applications [35], but it is prone to overfitting if its parameters are not appropriately set.

Random Forest Classifier: Several decision trees are combined in this ensemble learning model, which increases classification accuracy. It trains multiple trees using random subsets of the data and features to create a 'forest', and then combines their predictions, typically through majority voting. This method reduces overfitting, improve generalization, and robust to noise, making it effective for large and complex datasets [36].

XGBoost Classifier: This model is an ML algorithm and is considered efficient and scalable for supervised learning tasks. It performs well in classification and regression problems. It constructs a group of decision trees in a sequential way, with each tree attempting to perform correction on the errors of the previously built trees. It implements several advanced techniques, such as regularization, feature importance, and parallel processing, which enhance accuracy and speed up its training, improving generalization compared to traditional boosting techniques. It is capable of handling large datasets, feature interactions, and missing values [37].

LightGBM: LightGBM is a gradient boosting framework that uses tree-based learning algorithms. It is a fast, distributed, high performance gradient boosting (GBT, GBDT, GBRT, GBM or MART) framework based on decision tree algorithms, used for ranking, classification and many other machine learning tasks [38].

Each ML model is trained with each dataset using two distinct features selection methods. The mode used with standard parameters to ensure that the observed results are primarily driven by the feature selection process itself, without interference from model-specific parameter tuning. Additionally, the models' performance measured using different familiar metrics and the prediction time cost.

4. Experiments and Results

ML models' performance of different binary classification, on multiple datasets after pre-processing, and data resampling, is presented in this section. To address class imbalance, each dataset has been resampled to include 100,000 records for each class (benign and anomaly). Then, 80% and 20% of the 200,000 records are split into training and testing, respectively. The performance of each model has been evaluated on the entire dataset. As an exception, for large dataset, CSE-CIC-IDS2018, only the leading five million records have been used. The experiments were conducted on a Dell host with 64GB DDR4 memory, an Intel i7-7920HQ 3.1 GHz processor, and Python 3.9 as the programming language. The scikit-learn, XGBoost, and LightGBM libraries have been used to build the models.

4.1. Evaluation Metrics

Four metrics — precision, accuracy, F1 score, and recall — are used to assess model performance during training and testing. Prediction time was included as a fifth metric in the models' performance evaluation to measure how long each model takes to make a prediction on a given dataset. The F1 score, recall, accuracy, and precision are calculated using the following formulas [39]:

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \quad (3)$$

$$Precision = \frac{TP}{TP+FP} \quad (4)$$

$$Recall = \frac{TP}{TP+FN} \quad (5)$$

$$F1\ Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (6)$$

Where: TP = True Positives, TN = True Negatives, FP = False Positives, and FN = False Negatives.

4.2. Results and Analysis

CIC-DDOS2019, CSE-CIC-IDS2018, CIC-IDS2017, NSL-KDD, ISCX-IDS2012 and UNSW-NB15 are used for binary classification experiments. The features in these datasets are selected through preprocessing. Table 1 and Fig. 2 compare the selected number of features for each dataset.

Table 1 Comparison of features count

Dataset	Statistical	LightGBM
CIC-DDOS2019	57	12
CSE-CIC-IDS2018	61	13
CIC-IDS2017	68	12
UNSW-NB15	30	12
NSL-KDD	30	15
ISCX-IDS2012	16	6

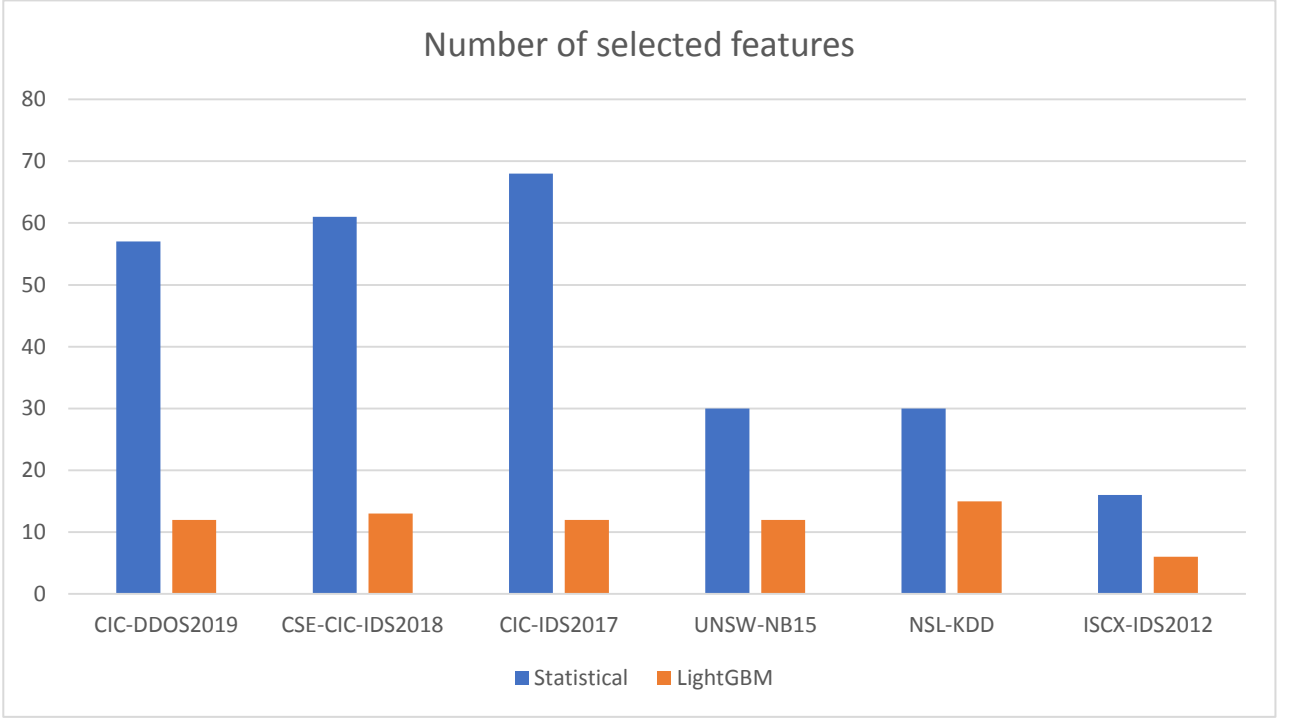

Fig. 2 Number of selected features for each dataset

The number of features selected using LightGBM is significantly lower than the count of statistically selected features. This observed variation highlights that the various feature selection methods can adapt to different characteristics in datasets. Further, this variation indicates that selection should be guided by the peculiar features of the data, including, but not limited to, its dimensionality, feature redundancy, and sample size.

Table 2 shows that the Logistic Regression model achieves similar metrics on dataset of CIC-DDOS2019. However, the prediction time improves significantly, with a reduction of 68.53% when using features selected by LightGBM. On other dataset, the CSE-CIC-IDS2018, the model time cost improves by 69.98%, and higher improvement in metrics is achieved. On the CIC-IDS2017, the model achieves a significant time cost improvement of 80.29% with

the LightGBM methodology. On other datasets, a slight reduction in model metrics is observed, but there is a noticeable improvement in time cost when LightGBM is used.

Table 2 Logistic Regression Performance Metrics

Dataset	Feature Selection Method	Prediction Time (s)	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)	Time Cost Improvement (%)
CIC-DDOS2019	Statistical	1.098	99.84	99.86	99.84	99.85	
	LightGBM	0.346	99.84	99.86	99.84	99.85	68.53
CSE-CIC-IDS2018	Statistical	0.573	92.48	97.90	92.48	94.46	
	LightGBM	0.172	**95.23**	**98.19**	**95.23**	**96.25**	69.98
CIC-IDS2017	Statistical	0.305	98.18	98.24	98.18	98.20	
	LightGBM	0.060	95.66	95.97	95.66	95.76	**80.29**
ISCX-IDS2012	Statistical	0.025	96.88	98.96	96.88	97.70	
	LightGBM	0.017	95.79	98.87	95.79	97.05	34.38
NSL-KDD	Statistical	0.015	91.69	91.80	91.69	91.68	
	LightGBM	0.008	91.39	91.54	91.39	91.37	48.26
UNSW-NB15	Statistical	0.104	97.85	98.54	97.85	98.07	
	LightGBM	0.066	97.80	98.43	97.80	98.01	36.48

The results in Table 3 show that with the LightGBM methodology, the Support Vector Machine model improves in both metrics and time cost for the CSE-CIC-IDS2018 and UNSW-NB15. The largest improvement in time cost for the model is recorded for UNSW-NB15, at 49%. For the other datasets, a slight reduction is observed in the model metrics, but there is still a noticeable improvement in time cost when LightGBM is used.

Table 3 Support Vector Machine Performance Metrics

Dataset	Feature Selection Method	Prediction Time (s)	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)	Time Cost Improvement (%)
CIC-DDOS2019	Statistical	430.58	99.85	99.87	99.85	99.86	
	LightGBM	286.08	99.82	99.85	99.82	99.83	34
CSE-CIC-IDS2018	Statistical	3359.73	95.29	98.21	95.29	96.30	
	LightGBM	1890.82	**95.53**	**98.25**	**95.53**	**96.46**	44
CIC-IDS2017	Statistical	501.88	99.29	99.30	99.29	99.30	
	LightGBM	476.80	98.74	98.74	98.74	98.74	5
ISCX-IDS2012	Statistical	144.85	98.55	99.31	98.55	98.81	
	LightGBM	112.68	97.55	99.16	97.55	98.15	22
NSL-KDD	Statistical	81.80	95.78	95.90	95.78	95.78	
	LightGBM	54.84	95.44	95.58	95.44	95.44	33
UNSW-NB15	Statistical	1747.48	96.72	98.30	96.72	97.22	
	LightGBM	885.25	**97.97**	**98.68**	**97.97**	**98.18**	**49**

The results in Table 4 show that the Bernoulli Naive Bayes model improves in both metrics and time cost for the UNSW-NB15. The model's time cost improves by 76.66% for the CIC-IDS2017 when using the LightGBM methodology for feature selection. Time cost improvement is observed on the other datasets, though there is a slight reduction in the model metrics when using the LightGBM methodology.

Table 4 Bernoulli Naive Bayes Performance Metrics

Dataset	Feature Selection Method	Prediction Time (s)	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)	Time Cost Improvement (%)
CIC-DDOS2019	Statistical	6.505	93.35	99.12	93.35	95.81	
	LightGBM	1.946	93.31	99.16	93.31	95.79	70.09
CSE-CIC-IDS2018	Statistical	3.283	67.40	97.35	67.40	77.98	
	LightGBM	0.959	57.60	97.02	57.60	70.40	70.80
CIC-IDS2017	Statistical	1.586	70.02	87.51	70.02	73.75	
	LightGBM	0.370	69.42	87.40	69.42	73.23	**76.66**
ISCX-IDS2012	Statistical	0.190	96.67	98.91	96.67	97.57	
	LightGBM	0.099	92.02	98.76	92.02	94.90	47.62
NSL-KDD	Statistical	0.092	85.29	85.71	85.29	85.23	
	LightGBM	0.036	85.04	85.06	85.04	85.04	60.79
UNSW-NB15	Statistical	0.746	84.71	95.75	84.71	89.18	
	LightGBM	0.295	**96.57**	**95.38**	**96.57**	**95.63**	60.52

The results of the Gaussian Naive Bayes in Table 5 show that the model achieves better metrics and time cost on the datasets of CIC-DDOS2019, CIC-IDS2017, ISCX-IDS2012, and NSL-KDD when using the LightGBM methodology. The model scored a significant improvement in time cost of 81.53% for the CIC-IDS2017 dataset. For the CSE-CIC-IDS2018 and UNSW-NB15 datasets, time cost is improved when the LightGBM methodology is used.

Table 5 Gaussian Naive Bayes Performance Metrics

Dataset	Feature Selection Method	Prediction Time (s)	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)	Time Cost Improvement (%)
CIC-DDOS2019	Statistical	11.109	96.57	99.24	96.57	97.64	
	LightGBM	2.920	**99.67**	**99.75**	**99.67**	**99.69**	73.72
CSE-CIC-IDS2018	Statistical	5.799	93.50	98.01	93.50	95.12	
	LightGBM	1.400	69.66	97.38	69.66	79.61	75.86
CIC-IDS2017	Statistical	2.562	90.68	93.35	90.68	91.33	
	LightGBM	0.473	**93.66**	**94.56**	**93.66**	**93.91**	**81.53**
ISCX-IDS2012	Statistical	0.169	54.85	98.58	54.85	69.66	
	LightGBM	0.123	**96.28**	**98.99**	**96.28**	**97.36**	27.27
NSL-KDD	Statistical	0.063	84.72	85.13	84.72	84.69	
	LightGBM	0.040	**87.84**	**87.93**	**87.84**	**87.83**	36.82
UNSW-NB15	Statistical	1.182	94.76	96.29	94.76	95.41	
	LightGBM	0.517	94.35	96.84	94.35	95.32	56.28

In Table 6, the Random Forest model results show some improvement in time cost. It reaches 15.71% for the UNSW-NB15 dataset. Despite the small time cost improvement, an improvement in the model metrics is noticed for CIC-DDOS2019 and NSL-KDD datasets when the LightGBM methodology is

used. The same results are achieved for the model metrics on the CSE-CIC-IDS2018 with time cost improvement of 4.16% in favor of the LightGBM methodology.

Table 6 Random Forest Performance Metrics

Dataset	Feature Selection Method	Prediction Time (s)	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)	Time Cost Improvement (%)
CIC-DDOS2019	Statistical	25.568	99.37	99.62	99.37	99.45	
	LightGBM	25.532	**99.87**	**99.88**	**99.87**	**99.87**	0.14
CSE-CIC-IDS2018	Statistical	18.211	99.99	99.99	99.99	99.99	
	LightGBM	17.372	**99.99**	**99.99**	**99.99**	**99.99**	4.61
CIC-IDS2017	Statistical	7.288	99.75	99.75	99.75	99.75	
	LightGBM	6.530	99.39	99.40	99.39	99.40	10.39
ISCX-IDS2012	Statistical	2.383	99.46	99.59	99.46	99.50	
	LightGBM	2.172	99.27	99.53	99.27	99.35	8.85
NSL-KDD	Statistical	0.490	95.80	95.96	95.80	95.80	
	LightGBM	0.488	**95.97**	**96.08**	**95.97**	**95.97**	0.42
UNSW-NB15	Statistical	6.473	98.49	98.95	98.49	98.62	
	LightGBM	5.456	98.47	98.93	98.47	98.60	**15.71**

The Decision Tree Classifier results in Table 7 show that improvements in the model metrics are achieved on the UNSW-NB15, CIC-DDOS2019, and CSE-CIC-IDS2018 when the LightGBM methodology is applied. Time cost improved for all datasets, with a 58.35% improvement is achieved on CIC-IDS2017 when the LightGBM methodology is used.

Table 7 Decision Tree Classifier Performance Metrics

Dataset	Feature Selection Method	Prediction Time (s)	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)	Time Cost Improvement (%)
CIC-DDOS2019	Statistical	1.602	99.95	99.95	99.95	99.95	
	LightGBM	0.688	**99.96**	**99.96**	**99.96**	**99.96**	57.04
CSE-CIC-IDS2018	Statistical	1.168	99.83	99.84	99.83	99.83	
	LightGBM	0.511	**99.88**	**99.88**	**99.88**	**99.88**	56.27
CIC-IDS2017	Statistical	0.432	99.90	99.90	99.90	99.90	
	LightGBM	0.180	99.81	99.82	99.81	99.82	**58.35**
ISCX-IDS2012	Statistical	0.071	99.41	99.60	99.41	99.47	
	LightGBM	0.069	99.39	99.59	99.39	99.45	3.21
NSL-KDD	Statistical	0.020	99.25	99.25	99.25	99.25	
	LightGBM	0.016	99.15	99.15	99.15	99.15	22.95
UNSW-NB15	Statistical	0.436	98.69	99.03	98.69	98.78	
	LightGBM	0.300	**98.70**	**99.04**	**98.70**	**98.80**	31.29

In Table 8, Gradient Boosting model is improved in metrics and time cost on the CSE-CIC-IDS2018 when the LightGBM methodology is used. A very slight or no difference in metrics is noticed for the remaining datasets. The time cost improvement for the model by 57.18% is achieved on the CIC-

IDS2017 in favor of the LightGBM methodology.

Table 8 Gradient Boosting Performance Metrics

Dataset	Feature Selection Method	Prediction Time (s)	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)	Time Cost Improvement (%)
CIC-DDOS2019	Statistical	18.231	99.92	99.93	99.92	99.92	
	LightGBM	8.414	99.91	99.92	99.91	99.91	53.85
CSE-CIC-IDS2018	Statistical	11.570	99.90	99.9	99.90	99.90	
	LightGBM	5.548	**99.91**	**99.92**	**99.91**	**99.91**	52.05
CIC-IDS2017	Statistical	4.310	99.89	99.89	99.89	99.89	
	LightGBM	1.845	99.89	99.89	99.89	99.89	**57.18**
ISCX-IDS2012	Statistical	0.623	98.78	99.38	98.78	98.98	
	LightGBM	0.562	98.68	99.35	98.68	98.91	9.88
NSL-KDD	Statistical	0.197	97.16	97.17	97.16	97.16	
	LightGBM	0.165	96.97	96.97	96.97	96.97	16.21
UNSW-NB15	Statistical	2.683	98.51	98.96	98.51	98.64	
	LightGBM	1.863	98.45	98.93	98.45	98.59	30.56

XGBoost model results in Table 9 show 66.14% improvement in the time cost on the CIC-DDOS2019 dataset with the LightGBM methodology is applied. The model metrics have a very narrow changes for the metrics on other datasets, while time cost improvements is noticeable when the LightGBM methodology is used.

Table 9 XGBoost Performance Metrics

Dataset	Feature Selection Method	Prediction Time (s)	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)	Time Cost Improvement (%)
CIC-DDOS2019	Statistical	12.061	99.96	99.96	99.96	99.96	
	LightGBM	4.051	**99.96**	**99.96**	**99.96**	**99.96**	**66.41**
CSE-CIC-IDS2018	Statistical	8.179	99.96	99.96	99.96	99.96	
	LightGBM	3.306	**99.96**	**99.96**	**99.96**	**99.96**	59.58
CIC-IDS2017	Statistical	4.230	99.98	99.98	99.98	99.98	
	LightGBM	2.400	99.91	99.91	99.91	99.91	43.26
ISCX-IDS2012	Statistical	2.115	99.25	99.53	99.25	99.34	
	LightGBM	2.005	99.24	99.53	99.24	99.33	5.22
NSL-KDD	Statistical	1.899	99.23	99.23	99.23	99.23	
	LightGBM	1.757	99.07	99.07	99.07	99.07	7.47
UNSW-NB15	Statistical	2.839	98.58	99.00	98.58	98.70	
	LightGBM	2.418	**98.58**	**98.99**	**98.58**	98.69	14.81

For LightGBM classification results in Table 10, the metrics and time cost improve when the LightGBM methodology is used for all the datasets except NSL-KDD there is a slight reduction in the model metrics. On the other hand, NSL-KDD achieves the highest time cost improvement by 41.18% when the LightGBM methodology is used.

Table 10 LightGBM Performance Metrics

Dataset	Feature Selection Method	Prediction Time (s)	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)	Time Cost Improvement (%)
CIC-DDOS2019	Statistical	10.071	99.96	99.97	99.96	99.96	
	LightGBM	8.203	99.96	99.96	99.96	99.96	18.55
CSE-CIC-IDS2018	Statistical	8.839	99.97	99.97	99.97	99.97	
	LightGBM	5.948	**99.98**	**99.98**	**99.98**	**99.98**	32.71
CIC-IDS2017	Statistical	3.285	99.98	99.98	99.98	99.98	
	LightGBM	2.675	99.91	99.91	99.91	99.91	18.56
ISCX-IDS2012	Statistical	1.052	99.20	99.51	99.20	99.30	
	LightGBM	0.934	**99.26**	**99.53**	**99.26**	**99.34**	11.26
NSL-KDD	Statistical	0.340	98.96	98.96	98.96	98.96	
	LightGBM	0.200	98.80	98.80	98.80	98.80	**41.18**
UNSW-NB15	Statistical	2.455	98.55	98.98	98.55	98.67	
	LightGBM	2.077	**98.56**	**98.99**	**98.56**	**98.68**	15.38

5. Summary and Conclusion

This paper investigates feature selection methodologies for machine learning models in cybersecurity, and compares statistical techniques with LightGBM on several intrusion detection datasets.

In most cases, there is a big improvement in time cost for all models, except some cases where there is a slight improvement when using LightGBM selected features over statistically selected ones. The time cost improves by 80.29% for Logistic Regression, 76.66% for the Bernoulli Naive Bayes classifier, 81.53% for Gaussian Naive Bayes, 58.35% for the Decision Tree Classifier, and 57.15% for Gradient Boosting when using LightGBM-selected features on the CIC-IDS2017 dataset. The Support Vector Machine achieves a 49.34% reduction in time cost, and Random Forest Classifier achieves a 15.71% reduction in time cost on UNSW-NB15 dataset. The XGBoost achieves a 66.41% reduction in time cost on the CIC-DDOS2019 dataset. The LightGBM classifier achieves a 41.18% improvement in time cost on the NSL-KDD dataset.

The results indicate that LightGBM consistently selects significantly fewer features, as shown in Table 1, which reduces data dimensionality without substantial loss in predictive performance. These features, as selected by LightGBM, perform comparably and, in some metrics, even slightly better than those selected using statistical methods. This balance between reduced computational cost and consistent model performance underlines the efficiency of LightGBM as a feature selection method in cybersecurity applications.

While this paper focuses on binary classification, extending such methods to multiclass classification could enable addressing more challenging tasks related to cybersecurity, including the recognition of attack types. Other methodologies, such as unsupervised and semi-supervised feature selection techniques and deep learning-based are out of the scope of this paper. . Most challenges in cybersecurity relate to discrimination among several types of attacks, rather than simple binary classifications like benign versus malicious traffic. The performance assessment of LightGBM-selected features in a multiclass setting could provide a comprehensive understanding of their effectiveness. Expanding into multi-class classification could help in addressing real-world cybersecurity needs and advance feature selection studies.

References

[1] Zhong C., Lin T., Liu P., Yen J., and Chen K., 2018, “A Cyber Security Data Triage Operation Retrieval System,” Comput Secur, 76 , pp. 12–31. https://doi.org/https://doi.org/10.1016/j.cose.2018.02.011 .

[2] Apruzzese G., Laskov P., de Oca E., Mallouli W., Brdalo Rapa L., Grammatopoulos A. V., and Di Franco F., 2023, "The Role of Machine Learning in Cybersecurity," Digital Threats: Research and Practice, 4 (1), pp. 1–38. https://doi.org/10.1145/3545574.

[3] Khraisat A., Gondal I., Vamplew P., and Kamruzzaman J., 2019, "Survey of Intrusion Detection Systems: Techniques, Datasets and Challenges," Cybersecurity, 2 (1), p. 20. https://doi.org/10.1186/s42400-019-0038-7.

[4] Jovic A., Brkić K., and Bogunovic N., 2015, *A Review of Feature Selection Methods with Applications*. https://doi.org/10.1109/MIPRO.2015.7160458.

[5] Bolón-Canedo V., Sánchez-Maroño N., and Alonso-Betanzos A., 2016, "Feature Selection for High-Dimensional Data," Progress in Artificial Intelligence, 5 , pp. 65–75.

[6] Lyu Y., Feng Y., and Sakurai K., 2023, "A Survey on Feature Selection Techniques Based on Filtering Methods for Cyber Attack Detection," Information, 14 (3). https://doi.org/10.3390/info14030191.

[7] Gil G. D., Lashkari A. H., Mamun M., and Ghorbani A. A., 2016, "Characterization of Encrypted and VPN Traffic Using Time-Related Features," *Proceedings of the 2nd International Conference on Information Systems Security and Privacy (ICISSP 2016)*, pp. 407–414.

[8] Mamidi S., 2024, "Future Trends in AI Driven Cyber Security."

[9] Liu X., Qian C., Hatcher W., Xu H., Liao W., and Yu W., 2019, "Secure Internet of Things (IoT)-Based Smart-World Critical Infrastructures: Survey, Case Study and Research Opportunities," IEEE Access, PP , p. 1. https://doi.org/10.1109/ACCESS.2019.2920763.

[10] Yu L., and Liu H., 2003, "Feature Selection for High-Dimensional Data: A Fast Correlation-Based Filter Solution," *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, pp. 856–863.

[11] Hoque N., Bhattacharyya D. K., and Kalita J. K., 2014, "MIFS-ND: A Mutual Information-Based Feature Selection Method," Expert Syst Appl, 41 (14), pp. 6371–6385. https://doi.org/https://doi.org/10.1016/j.eswa.2014.04.019.

[12] Matta V., Di Mauro M., and Longo M., 2016, "Botnet Identification in Randomized DDoS Attacks," *2016 24th European Signal Processing Conference (EUSIPCO)*, pp. 2260–2264.

[13] Matta V., Di Mauro M., and Longo M., 2017, "Botnet Identification in Multi-Clustered DDoS Attacks," *2017 25th European Signal Processing Conference (EUSIPCO)*, pp. 2171–2175.

[14] Addesso P., Cirillo M., Di Mauro M., and Matta V., 2019, "ADVoIP: Adversarial Detection of Encrypted and Concealed VoIP," IEEE Transactions on Information Forensics and Security, 15 , pp. 943–958.

[15] Matta V., Di Mauro M., Longo M., and Farina A., 2018, "Cyber-Threat Mitigation Exploiting the Birth–Death–Immigration Model," IEEE Transactions on Information Forensics and Security, 13 (12), pp. 3137–3152.

[16] Di Mauro M., Galatro G., Fortino G., and Liotta A., 2021, "Supervised Feature Selection Techniques in Network Intrusion Detection: A Critical Review," Eng Appl Artif Intell, 101 , p. 104216. https://doi.org/https://doi.org/10.1016/j.engappai.2021.104216.

[17] Bulavas V., Marcinkevičius V., and Ruminski J., 2021, "Study of Multi-Class Classification Algorithms' Performance on Highly Imbalanced Network Intrusion Datasets," Informatica, 32 , pp. 1–35. https://doi.org/10.15388/21-INFOR457.

[18] Alshaibi A., Al-Ani M., Al-Azzawi A., Konev A., and Shelupanov A., 2022, "The Comparison of Cybersecurity Datasets," Data (Basel), 7 (2). https://doi.org/10.3390/data7020022.

[19] Moustafa N., and Slay J., 2015, "UNSW-NB15: A Comprehensive Data Set for Network Intrusion Detection Systems (UNSW-NB15 Network Data Set)," *2015 Military Communications and Information Systems Conference (MilCIS)*, pp. 1–6. https://doi.org/10.1109/MilCIS.2015.7348942.

[20] Shiravi A., Shiravi H., Tavallaee M., and Ghorbani A. A., 2012, "Toward Developing a Systematic Approach to Generate Benchmark Datasets for Intrusion Detection," Comput Secur, 31 (3), pp. 357–374. https://doi.org/https://doi.org/10.1016/j.cose.2011.12.012.

[21] Sharafaldin I., Habibi Lashkari A., and Ghorbani A., 2018, *Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization*. https://doi.org/10.5220/0006639801080116.

[22] Sharafaldin I., Lashkari A. H., Hakak S., and Ghorbani A. A., 2019, "Developing Realistic Distributed Denial of Service (DDoS) Attack Dataset and Taxonomy," *2019 International Carnahan Conference on Security Technology (ICCST)*, pp. 1–8. https://doi.org/10.1109/CCST.2019.8888419.

[23] D'hooge L., Verkerken M., Volckaert B., Wauters T., and De Turck F., 2022, "Establishing the Contaminating Effect of Metadata Feature Inclusion in Machine-Learned Network Intrusion Detection Models," *Detection of Intrusions and Malware, and Vulnerability Assessment*, L. Cavallaro, D. Gruss, G. Pellegrino, and G. Giacinto, eds., Springer International Publishing, Cham, pp. 23–41.

[24] Bolikulov F., Nasimov R., Rashidov A., Akhmedov F., and Young-Im C., 2024, "Effective Methods of Categorical Data Encoding for Artificial Intelligence Algorithms," Mathematics, 12 (16), p. 2553.

[25] Sharma V., 2022, "A Study on Data Scaling Methods for Machine Learning," International Journal for Global Academic & Scientific Research, 1 (1), pp. 31–42.

[26] developers S., 2025, "MinMaxScaler — Scikit-Learn 1.5.0 Documentation." [Online]. Available: https://scikit-learn.org/1.5/modules/generated/sklearn.preprocessing.MinMaxScaler.html.

[27] MacKay D. J. C., 2003, *Information Theory, Inference and Learning Algorithms*, Cambridge university press.

[28] Camastra F., and Vinciarelli A., 2008, "Feature Extraction Methods and Manifold Learning Methods," Machine Learning for Audio, Image and Video Analysis: Theory and Applications, pp. 305–341.

[29] Buczak A. L., and Guven E., 2016, "A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection," IEEE Communications Surveys & Tutorials, 18 (2), pp. 1153–1176. https://doi.org/10.1109/COMST.2015.2494502.

[30] Cervantes J., Garcia-Lamont F., Rodríguez-Mazahua L., and Lopez A., 2020, "A Comprehensive Survey on Support Vector Machine Classification: Applications, Challenges and Trends," Neurocomputing, 408 , pp. 189–215. https://doi.org/https://doi.org/10.1016/j.neucom.2019.10.118.

[31] Dreiseitl S., and Ohno-Machado L., 2002, "Logistic Regression and Artificial Neural Network Classification Models: A Methodology Review," J Biomed Inform, 35 (5), pp. 352–359.

https://doi.org/https://doi.org/10.1016/S1532-0464(03)00034-0.

[32] Safavian S. R., and Landgrebe D., 1991, "A Survey of Decision Tree Classifier Methodology," IEEE Trans Syst Man Cybern, 21 (3), pp. 660–674. https://doi.org/10.1109/21.97458.

[33] Bi Z., Han Y., Huang C., and Wang M., 2019, "Gaussian Naive Bayesian Data Classification Model Based on Clustering Algorithm," *Proceedings of the 2019 International Conference on Modeling, Analysis, Simulation Technologies and Applications (MASTA 2019)*, Atlantis Press, pp. 396–400. https://doi.org/10.2991/masta-19.2019.67.

[34] Sayfullina L., Eirola E., Komashinsky D., Palumbo P., Miche Y., Lendasse A., and Karhunen J., 2015, "Efficient Detection of Zero-Day Android Malware Using Normalized Bernoulli Naive Bayes," *2015 IEEE Trustcom/BigDataSE/ISPA*, pp. 198–205. https://doi.org/10.1109/Trustcom.2015.375.

[35] Omari K., 2023, "Phishing Detection Using Gradient Boosting Classifier," Procedia Comput Sci, 230 , pp. 120–127. https://doi.org/https://doi.org/10.1016/j.procs.2023.12.067.

[36] Parmar A., Katariya R., and Patel V., 2019, "A Review on Random Forest: An Ensemble Classifier," *International Conference on Intelligent Data Communication Technologies and Internet of Things (ICICI) 2018*, J. Hemanth, X. Fernando, P. Lafata, and Z. Baig, eds., Springer International Publishing, Cham, pp. 758–763.

[37] Chen Z., Jiang F., Cheng Y., Gu X., Liu W., and Peng J., 2018, "XGBoost Classifier for DDoS Attack Detection and Analysis in SDN-Based Cloud," *2018 IEEE International Conference on Big Data and Smart Computing (BigComp)*, pp. 251–256. https://doi.org/10.1109/BigComp.2018.00044.

[38] Ke G., Meng Q., Finley T., Wang T., Chen W., Ma W., Ye Q., and Liu T.-Y., 2017, "LightGBM: A Highly Efficient Gradient Boosting Decision Tree," *Advances in Neural Information Processing Systems*, I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, eds., Curran Associates, Inc. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2017/file/6449f44a102fde848669bdd9eb6b76fa-Paper.pdf.

[39] developers S., 2025, "Model Evaluation — Scikit-Learn 1.5.0 Documentation." [Online]. Available: https://scikit-learn.org/1.5/modules/model_evaluation.html.