A PROJECT REPORT ON

Intrusion detection system in the Smart Distribution Network: A smart feature engineering-based PI-LightGBM approach

A Project report submitted in partial fulfilment of the requirement
for the award of the Degree of

MASTER OF COMPUTER APPLICATIONS

Submitted By

Bhubanesh Maharana -23240002

Under the esteemed guidance of

Mr. J. Chandrakanta Badajena

(School of Computer Sciences)

SCHOOL OF COMPUTER SCIENCES

ODISHA UNIVERSITY OF TECHNOLOGY AND RESEARCH

(Techno Campus, Ghatikia, Bhubaneswar-751003)

Academic Year 2022-2024

ODISHA UNIVERSITY OF TECHNOLOGY AND RESEARCH

School of Computer Sciences

CERTIFICATE

This is to certify that the project report entitled “Intrusion detection system in the Smart Distribution Network: A smart feature engineering-based PI-LightGBM approach” report has been successfully completed by Bhubanesh Maharana bearing registration number 2324002 as part of Degree of Master of Computer Science and Application under the School of Computer Science to Odisha University of Technology and Research project requirement. This is to certify that the report demonstrates a comprehensive understanding of the project, highlighting details and outcomes.

Internal Guide	External Guide	Head of the School
Mr. J. Chandrakanta Badajena		Prof (Dr.) Ranjan Kumar Dash
School of Computer Sciences		School of Computer Sciences

ODISHA UNIVERSITY OF TECHNOLOGY AND RESEARCH

School of Computer Sciences

DECLARATION

I **Bhubanesh Maharana** bearing Registration No: **23240002,** a Bonafede student of **Odisha University of Technology and Research,** would like to declare that the project titled **"Intrusion detection system in the Smart Distribution Network: A smart feature engineering PI-LightGBM approach".** A partial fulfilment of MCA Degree course of Odisha University of Technology and Research is my original work in theyear 2024 Under the guidance of **Mr. J. Chandrakant Badajena,** School of Computer Sciences andit has not previously formed the basis for any degree or diploma or other any similar title submitted to any university.

Date: **Bhubanesh Maharana**

OUTR, BBSR **(23240002)**

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to my advisor, **Mr.J.Chandrakant Badajena**, School of Computer Sciences, His extensive knowledge and invaluable guidance have been instrumental in motivating me to achieve my goals and in the successful completionof this project. His patience, support, and encouragement have been a driving force throughoutmy research and writing process.

I take it as a great privilege to express my heartfelt gratitude to Dr. Ranjan Kumar Dash for his unwavering support and to all the senior faculty members of the MCA department. Their continuous assistance and insightful feedback have greatly contributed to my academic growthand the development of this project. I am also grateful to the programmers and non-teaching staff of the School of Computer Sciences for their indispensable help and cooperation during my course. Their contributions, though often behind the scenes, have been essential in the smooth execution of this project.

I would also like to extend my gratitude to the authors and researchers whose work I have citedand referred to in my project. Their contributions to the field have been a great source of inspiration and knowledge.

Finally, I am profoundly grateful to my parents for their unwavering support and encouragement throughout my life and during this course. I also wish to thank all my friends and well-wishers for their constant support. Their words of encouragement and support have been invaluable in keeping me motivated and focused on my goals.

Date : **Bhubanesh Maharana**

OUTR, BBSR **(23240002)**

ABSTRACT

Intrusion Detection Systems (IDS) are essential for protecting computer networks against cyberattacks because they keep an eye on network traffic and look for indicators of malicious activity. IDS come in two primary flavors: Host-based Intrusion Detection Systems (HIDS), which concentrate on specific devices, and Network-based Intrusion Detection Systems (NIDS), which track all network activity. NIDS use methods including anomaly-based detection, signature-based detection, and real-time analysis to examine data packets and find abnormal patterns. These systems aid in the effective detection and mitigation of threats by offering centralized monitoring of network activity and wide visibility.

Despite their significance, real-time network intrusion detection is still an underexplored field. Traditional approaches, including feature selection, specialized hardware, and the integration of big data technologies with machine learning, have limitations. These methods often compromise detection accuracy, incur high costs, or fail to ensure real-time analysis. Improving real-time performance in IDS is essential to address the growing complexity and frequency of cyber-attacks effectively

1. Introduction:

Have you ever wondered how your computer/network is able to avoid being infected with malware and bad traffic inputs from the internet? The reason why it can detect it so well is because there are systems in place to protect your valuable information held in your computer or networks from malicious traffic and cyber-attacks. These systems that detect malicious traffic inputs are called Intrusion Detection Systems (IDS), which play a crucial role in identifying and mitigating harmful activities within a network.

IDS are specialized systems trained on vast amounts of internet traffic data to recognize patterns indicative of malicious behaviour. They function as vigilant guards, continuously monitoring network traffic for signs of intrusion or abnormal activities that could signify an attack.

In today's digital world, the significance of IDS cannot be emphasized. IDS offer an essential barrier of defence against the growing complexity and frequency of cyberattacks, assisting in the identification and fixing of any breaches before they have a substantial negative impact. IDS aids in preserving the availability, confidentiality, and integrity of sensitive data by seeing risks early.

There are primarily two types of IDS: Network-based Intrusion Detection Systems (NIDS) and Host-based Intrusion Detection Systems (HIDS). NIDS monitor the entire network traffic, analysing data packets traveling across the network to identify suspicious activity. In contrast, HIDS are deployed on individual devices or hosts, where they monitor system logs, file integrity, and other local activities to detect anomalies.

This paper sheds light on the strategies and methodologies essential for developing an efficient Network-based Intrusion Detection System (NIDS), with the ultimate goal of creating a robust, attack-free network environment.

NIDS are designed to monitor and analyse network traffic for signs of malicious activities or policy violations. They are typically deployed at strategic points within a network, such as the perimeter (e.g., at the gateway) or critical network segments.

Functionality:

- Traffic Monitoring: NIDS capture and inspect data packets as they traverse the network. By analysing the content and headers of these packets, NIDS can identify suspicious patterns or behaviours.

- Signature-based Detection: Many NIDS use a database of known attack signatures (patterns) to detect malicious activities. When a packet matches a known signature, the NIDS generates an alert.

- Anomaly-based Detection: NIDS can also employ anomaly detection techniques, where they establish a baseline of normal network behaviour. Any deviations from this baseline are flagged as potential threats.

- Real-time Analysis: NIDS operate in real-time, providing immediate detection and alerting of ongoing attacks, allowing for swift response.

Advantages:

- Broad Visibility: NIDS provide a comprehensive view of network activity, making it easier to detect attacks that target multiple hosts or services.

- Centralized Monitoring: By placing NIDS at key network points, administrators can monitor traffic from a centralized location.

The Real-time network intrusion detection system remains a still untapped field in today's digital world, with few research conducted in this area (Habeeb et al., 2019; Salo et al., 2018) [1]. Three categories mainly include conventional research on real-time intrusion detection. The first kind reduces the time complexity of decision-making by using feature selection; still these techniques typically compromise detection performance, such as accuracy. The second type uses specialized hardware to speed up intrusion detection pattern matching, which frequently ends in improved real-time performance. However, hardware-based solutions are typically expensive, rigid, and challenging to use with emerging intrusion detection algorithms. The third kind integrates big data technologies with machine learning techniques. Traffic data is typically stored by these technologies (such as Hadoop-based clusters Fontugne et al., 2014) [2] to a distributed filesystem and process them later, which cannot guarantee a line speed analysis of the network traffic.

The contributions of this paper are summarized as follows:

- I propose and apply two approaches to improve the real-time performance of IDS. One approach is to reduce the time consumption of detection phase by using Intelligent Feature Selection Agent by computing the permutation importance score for feature selection. As In order to create effective machine learning models and data analysis systems, such as Network-based Intrusion Detection Systems (NIDS), feature selection is essential. By improving scope and accuracy, it greatly boosts model performance and enables models to concentrate on the most crucial elements of the data. This emphasis makes sure that models function well on fresh, clean data and reduces the possibility of overfitting. Moreover, feature selection lowers computational costs and streamlines models, making them simpler to read and understanding. This speed up the training and prediction processes.

- The other approach is based on the idea that the task of real-time detection can be decomposed into improving the time efficiency for a single or several phases of each

detection cycle. Specifically, using of LightGBM as the intrusion detection algorithm to reduce the time consumption of data preprocessing and decision-making phase. LightGBM is capable of dealing with categorical features and thus there is no need to carry out numerical transformation (e.g., one-hot encoding) in the data preprocessing phase. What's more, a leaf-wise tree growth strategy is leveraged by LightGBM, which splits at the leaf node with the maximum gain (Ke et al., 2017)[3] . Therefore, the trained model can be composed of fewer trees and leaf nodes while keeping good detection performance. This feature of LightGBM makes the matching in the decision-making phase time-efficient.

The rest of the paper is organized as follows: Section 2 discusses the related researches. Section 3 Proposed Methdology, the smart feature selection agent and the principles of LightGBM mechanism. Section 4 introduces to evaluation metrics and experimental results. Section 5 concludes the paper.

2. Related researches :

Many researchers have proposed several feature selection algorithms for the intrusion detection system. Hui Jiang [4] proposed the PSO-Xgboost model given its overall higher classification accuracy than other alternative models such like Xgboost, Random Forest, Bagging and Adaboost. Firstly, a classification model based on Xgboost is constructed, and then PSO is used to adaptively search for the optimal structure of Xgboost.

The AutoEncoder (AE)-LightGBM describes by Ruizhe Yao [5], emphasizes Borderline-SMOTE to balance data distribution, AE for feature extraction, and LightGBM for intrusion detection. Experiments on the KDDCup99 and NSL-KDD datasets demonstrate that the proposed model outperforms traditional models in accuracy, precision, and F1-score, while significantly enhancing real-time performance.

Al-Safi et al. [6] used SVM as their classification approach and information gain to choose the NSL KDD dataset's useful characteristics. The Artificial Bee Colony and Cuckoo Search algorithms work well together to do classification with Support Vector Machines for hyperparameter tuning. According to the authors, the model performed well in terms of classification accuracy when compared to other approaches.

A modified version of an optimizer inspired by pigeons that is combined with the Tabu Local Search algorithm was proposed by Orieb Abu Alghanam et al. [7]. Here, a one-class strategy is utilized for both feature selection and classification, and the model achieves very good accuracy. An ensembled model coupled with a single class approach is suggested for classification purposes. The framework performed better in terms of FPR and TPR when OC-SVM and OC-IF were employed in an ensemble setting. Comparing the suggested LS-PIO feature selection approach to the Hill climbing PIO algorithm, it performed well significantly.

In [8], Khammassi collaborators used a genetic algorithm, which is primarily based on wrappers, in conjunction with logistic regression to pick features. The UNSWNB15 and KDD CUP 99 datasets are used to test and validate the model. Decision Tree Classifier is used to classify attacks, and the Weka tool is used to visualize the findings. The recommended method improved the detection rate by reducing the characteristics to 18 for KDD CUP 99 and 20 for UNSWNB15. The KDD CUP99 dataset had a detection rate of 99.9%, while the UNSWNB15 dataset had an 81.24% rate. The authors did not, however, mention the execution duration or accuracy.

Thaseen et al [9] devised an intrusion detection framework by utilizing an effective Chi Square feature selection mechanism with a Support Vector Machine algorithm. Apart from these, for parameter optimization, a variance based tuning technique is used and the entire model achieved outstanding performance.

Flexible Mutual Information feature selection was presented by Ambusaidi et al. in [10], and the Least Square Support Vector Machine-based Intrusion Detection System uses the reduced features. Better detection rates and FAR were found in the experimental results on three distinct datasets: Kyoto 2006, KDD CUP 99, and NSL-KDD datasets. [11] Singh et al used rule-based classifiers with a filter-based feature selection technique known as the Information Gain approach in order to achieve classification. 22 characteristics were extracted by Information Gain from the UNSWNB15 dataset, with an accuracy of 84.83%.

The impact of feature selection and SMOTE oversampling on the UNSWNB15 dataset for attack detection was suggested and examined in [12]. To improve the system's performance, a feature selection strategy is applied. Random Forest, Decision Tree, KNN, and LR algorithm are the four primary machine learning classifiers that are subjected to the combined effect of SMOTE and RFE feature selection approach. Using the Random Forest approach, the model's accuracy increased to 84.13%.

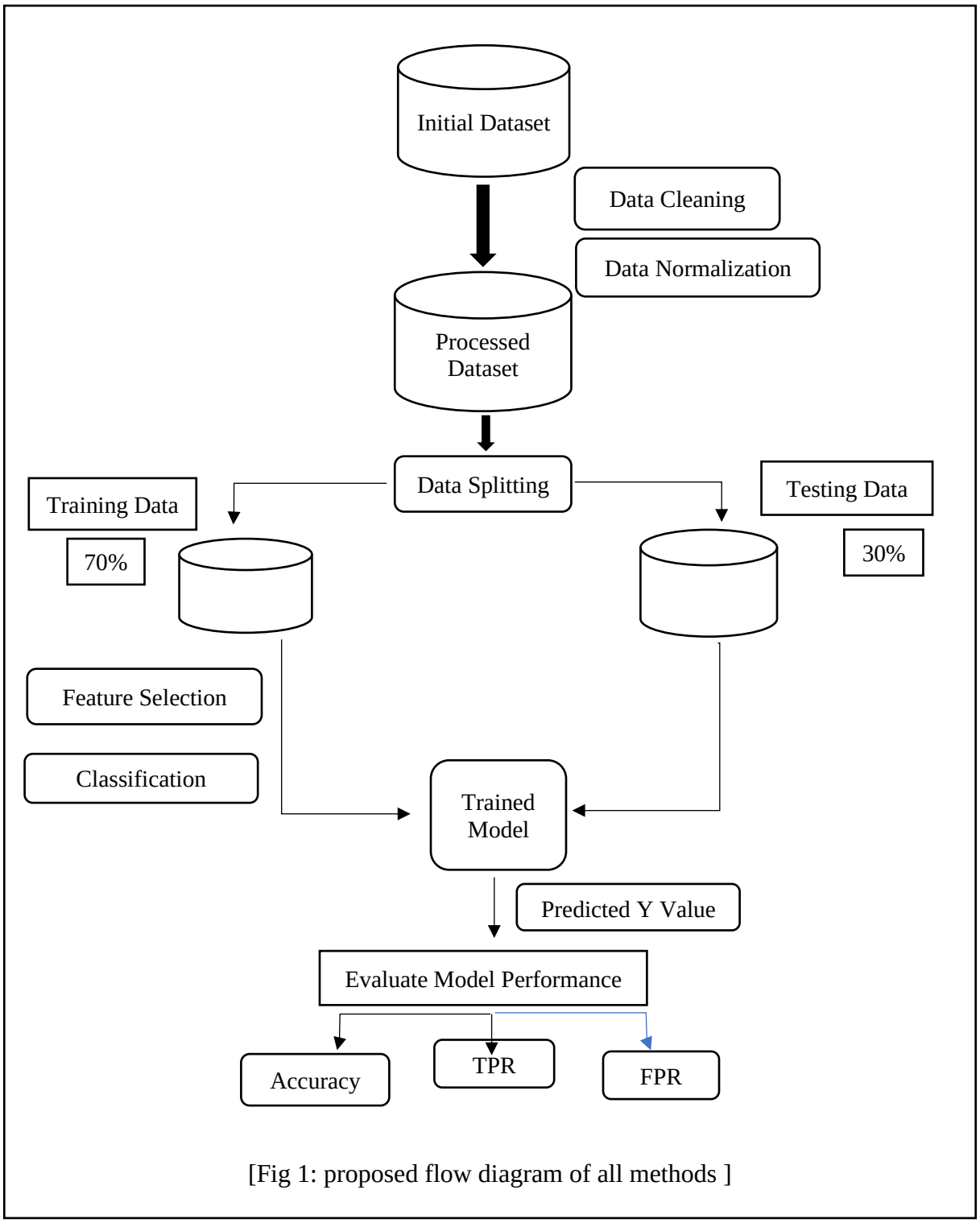
In another work[13] Shalini Subramani et al proposed an intelligent IDS utilizing PSO-based feature selection and enhanced multiclass SVM based classification mechanism for intrusion detection. The entire machine learning model is validated on two datasets namely the KDD'99 Cup data set and the CIDD data set. It enhances the performance in terms of accuracy and balances False Positive Rate.

For intrusion detection systems, it is evident that numerous researchers have put forth different feature selection techniques. While some of these techniques employ ensembled models using a one-class approach, others make use of well-known machine learning algorithms like SVM, Decision Tree, and KNN. On a variety of datasets, including NSL KDD, UNSWNB15, KDD CUP 99, and Kyoto 2006, the majority of the approaches demonstrated improved detection rates and excellent accuracy.

My proposed work, "An Agent for Feature Selection in Network Intrusion Detection Systems Using a PI Intelligent Techniques", aims to improve the performance of intrusion detection systems by using a PI algorithm for feature selection. This approach of Permutation Importance (PI) enhances the accuracy and reduce the number of features required for the detection process. Connect it to LightGBM, an enhanced version of the Gradient Boosting Decision Tree (GBDT) method, as well. to improve network intrusion detection systems' functionality.

3. Proposed Methodology:

My proposed methodology comprises three components: Data reprocessing, a Feature selection agent, and a classification module. The flow of +the process are described in the figure 1.



[Fig 1: proposed flow diagram of all methods]

3.1 Data Preprocessing :-

Preparing data for analysis and consistency is a crucial step in machine learning, as it sets the stage for model training. The process requires a number of actions to ensure that the data is in a state that machine learning models can use by cleaning, transforming, and standardizing it. The most important factor of data preprocessing is data scaling [15], which is especially important for forecasting output and minimizing errors. For instance, the ranges of the many features in the NSL-KDD dataset vary, which may have an impact on the accuracy of machine learning models.

Standard Scaler Normalization (Z-score Normalization) :

Standard Scaler normalization is another widely used method to scale data. This method transforms the data such that it has a mean of 0 and a standard deviation of 1. It is particularly useful when the data follows a Gaussian distribution.

For every feature:

- The mean value of that feature gets transformed into a 0.
- Values above the mean become positive.
- Values below the mean become negative.

The general formula for a standard scaler normalization is given as:

$$x' = (x - u) / s \qquad (1)$$

where u is the mean of the training samples or zero and s is the standard.

Equation (1) represents a transformation of a variable x to a standard normal distribution. This transformation is accomplished by subtracting the mean of the distribution, μ, from each data point and then dividing by the standard deviation of the distribution, S This transformation is commonly used in statistics to normalize data, making it easier to compare data sets with different means and standard deviations. The resulting transformed data points have a mean of 0 and a standard deviation of 1.

3.2 Intelligent feature selection with Permutation Improtance :

Permutation importance is a method for feature importance calculation in machine learning. It measures the decrease in model performance when the values of a feature are randomly permuted, while holding all other features constant. The feature importance score is then calculated as the difference between the original performance and the performance with permuted features. This score represents the contribution of the feature to the model's accuracy.

The mathematical expression for importance can be represented as follows:

Original performance: L

Feature importance score:

$$PIi = L - Li \quad (2)$$

where Li is the performance with the feature i permuted

Normalized feature importance score:

$$NPIi = PIi / sum(PI) \quad (3)$$

where sum (PI) is the sum of all feature importance cores

3.3.1 Compute Permutation Importance Score :

Algorithm 1: Calculate Normalized Feature Importance Scores:
Input:
- Dataset D
- Machine learning model M
- Feature importance method F
- Normalization method N area

Output:
- Normalized feature importance scores

Step 1. Train the machine learning model M on the dataset D.
Step 2. Calculate feature importance scores using the selected feature importance method F.
Step 3. Normalize the feature importance scores using the chosen normalization method N:
Step a. Apply the normalization method to the feature importance scores obtained in step 2.
Step b. Scale the importance scores to a common range, such as [0, 1] or [-1, 1].
Step 4. Output the normalized feature importance scores.

$piscores$<− Compute permutation importance scores for all features in $featureset$. $selectedfeatures$ <− Select features whose permutation importance scores are greater than $pithreshold$. The algorithm 1 show the process for deriving permutation importance score.

3.3.2 Select Top Features with set the PI_Threshold :-

Compute permutation importance scores for all features in X. Select the top features whose permutation importance scores are greater than $pi\_threshold$ to create a set of selected features S.in my case the pi_threshold value is 0.001.

3.3 Model Training with LightGBM:

The Gradient Boosting Decision Tree (GBDT) algorithm has been enhanced with the introduction of LightGBM. To provide a final forecast that is well-generalized, it integrates the predictions of several decision trees. Integrating several "weak" learners into one "strong" learner is the key principle of LightGBM. This concept serves as the foundation for two types of machine learning algorithms. Initially, "weak" learners are simple to pick up. Second, the generalization performance of a group of learners is usually higher than that of a single student. LightGBM has been shown to be superior in a number of machine learning tasks, including tumor classification (Wang et al., 2017)[16] and air quality prediction (Zhang et al., 2019)[17].

Algorithm 2: The training of LightGBM

Require: Input: Training set $\{(x_i, y_i)\}_{i=1}^{N}$
Ensure: Output: LightGBM model $\hat{y}_i^{(t)}$

Step 1. Initialize the first tree as a constant:

$$\hat{y}_i^{(0)} = f_0 = 0$$

Step 2. Train the next tree by minimizing the loss function:

$$f_t(x_i) = \arg\min_{f_t} L_{(t)} = \arg\min_{f_t} L\left(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)\right)$$

Step 3. Get the next model in an additive manner:

$$\hat{y}_i^{(t)} = \hat{y}_i^{(t-1)} + f_t(x_i)$$

Step 4. Repeat the Step2 and Step 3 until the model reaches the stop condition.

Step 5. Obtain and return the final model:

$$\hat{y}_i^{(t)} = \sum_{t=0}^{M-1} f_t(x_i)$$

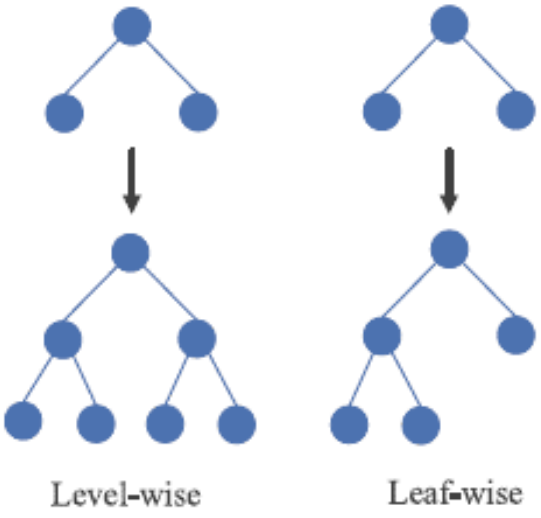
To clearly illustrate the training process of LightGBM, we take a model consisting of M trees as an example described in Algorithm 2 .

A few clarifications of the procedure mentioned above are provided here. The prediction for the i-th case at the t-th iteration is ˆ y (t) i. The learning function of the t-th decision tree is represented by the symbol f t (x i). To punish the model's complexity, a regularization term can be included. L (t) is the loss function that measures the difference between the prediction ˆy (t) i and the actual y i. When the M-th iteration of the training process is complete, that is the stop condition. It is important to note that M can be substituted as the stop condition with a reasonable loss value. The training process ends when the model's training loss is less than the predefined loss value.

In contrast to previous GBDT techniques, LightGBM maintains its efficiency even with large and highly dimensioned data sets. The two exclusive techniques, Exclusive Feature Bundling (EFB) and Gradient-based One-Side Sampling (GOSS), are the reason for this. One type of downsampling method is GOSS. Higher gradient examples will add more to the information gained during the model training phase. As a result, GOSS retains data instances with big gradients and drops those with minor gradients at random while down sampling the data instances. For data instances with minor gradients, GOSS includes a constant multiplier to offset the influence on the data distribution. GOSS is able to maintain excellent accuracy while reducing the amount of data by doing this. It is suggested that EFB effectively reduces the amount of characteristics. Since high-dimensional data are typically quite sparse, it is possible to construct a nearly loss-less method for reducing the number of features.

As numerous features in a sparse feature space are mutually exclusive, EFB can safely combine disparate features into a single feature. From feature bundles, EFB constructs feature histograms that are identical to those from individual features. In this method, the histograms' complexity decreased, greatly accelerating the GBDT training process without compromising accuracy.

LightGBM also has the benefit of supporting optimal split of category features. This implies that category features in data can be directly handled by LightGBM. Many other machine learning techniques, on the other hand, require that the numerical transformation (such as label encoding or one-hot encoding) be performed first because they are unable to interpret category features. The data is transformed into a format that machine learning algorithms can handle, but it also makes the data sparser. According to Fisher (1958)[18], Light-GBM facilitates the best division of category traits through grouping techniques.



[**fig 4:** Comparison of level-wise and leaf-wise tree generation strategy.]

Additionally, when growing a leaf, a leaf-wise tree creation approach is used, which can cut losses more than the conventional level-wise strategy. The leaf-wise and level-wise strategies are contrasted in Fig. 4. Less decision trees and fewer leaves per decision tree are usually present in the final LightGBM model. LightGBM is time-efficient during the decision-making process because to this model feature.

4. Experiments And Results:

Description of datasets :

Among the publicly available datasets utilized for intrusion detection evaluation, the KDD99 and NSL-KDD datasets stand out as widely recognized benchmarks. These datasets have been widely used to evaluate intrusion detection system (IDS) performance. These three datasets were specifically picked by me to provide a thorough and reliable comparison with state-of-the-art techniques today. This choice provides a comprehensive analysis, offering insights into the effectiveness and adaptability of the suggested strategy in many real-world scenarios.

KDD99 And NSL-KDD Dataset :-

The KDD99 dataset is one of the most widely used datasets for evaluation. The original KDD99 dataset consists of 5,000,000 labeled records with 41 attributes. Traffic belongs to either the "Normal" class or an "Attack" class. The "Attack" class is further divided into four categories: DDOS (Distributed denial of service), Probing, R2L (Remote to Local), and U2R (User to Root) (Tavallaee et al., 2009) [14]. Within the data set exists 4 different classes of attacks: Denial of Service (DoS), Probe, User to Root(U2R), and Remote to Local (R2L). A brief description of each attack can be seen below:

Table 1
The distribution of records in KDD99 and NSL-KDD datasets.

Traffic type	KDD99		NSL-KDD	
	Size	Distribution (%)	Size	Distribution (%)
Dos	54,572	37.48	45,927	36.46
Probe	2131	1.46	11,656	9.25
R2L	999	0.69	995	0.79
U2R	52	0.04	52	0.04

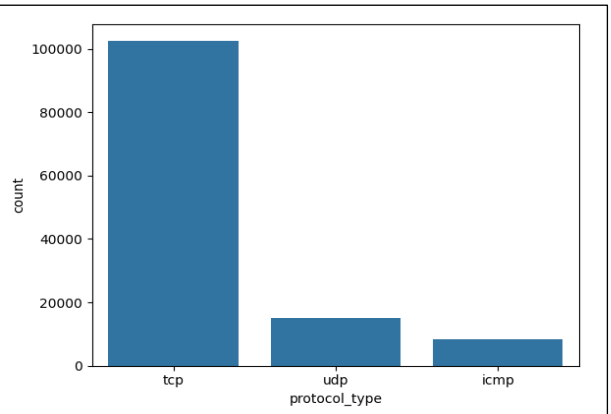
- DoS is an attack that tries to shut down traffic flow to and from the target system. The IDS is flooded with an abnormal amount of traffic, which the system can't handle, and shuts down to protect itself. This prevents normal traffic from visiting a network. An example of this could be an online retailer getting flooded with online orders on a day with a big sale, and because the network can't handle all the requests, it will shut down preventing paying customers to purchase anything. This is the most common attack in the data set.

- Probe or surveillance is an attack that tries to get information from a network. The goal here is to act like a thief and steal important information, whether it be personal information about clients or banking information.

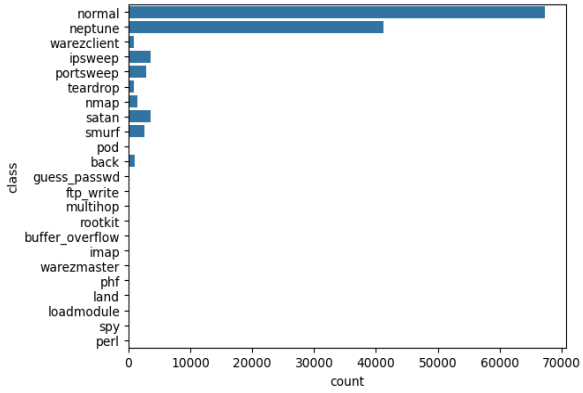
- U2R is an attack that starts off with a normal user account and tries to gain access to the system or network, as a super-user (root). The attacker attempts to exploit the vulnerabilities in a system to gain root privileges/access.

- R2L is an attack that tries to gain local access to a remote machine. An attacker does not have local access to the system/network, and tries to "hack" their way into the network.

To resolve the problem of redundancy in KDD99, the NSL-KDD data set was brought into existence, as a revised, cleaned-up version of the KDD'99. The distribution of the modified KDD99 and NSL-KDD datasets I used are presented in Table 1 and the figures 2 and 3.



[fig 2: NSL-KDD protocol type distribution]

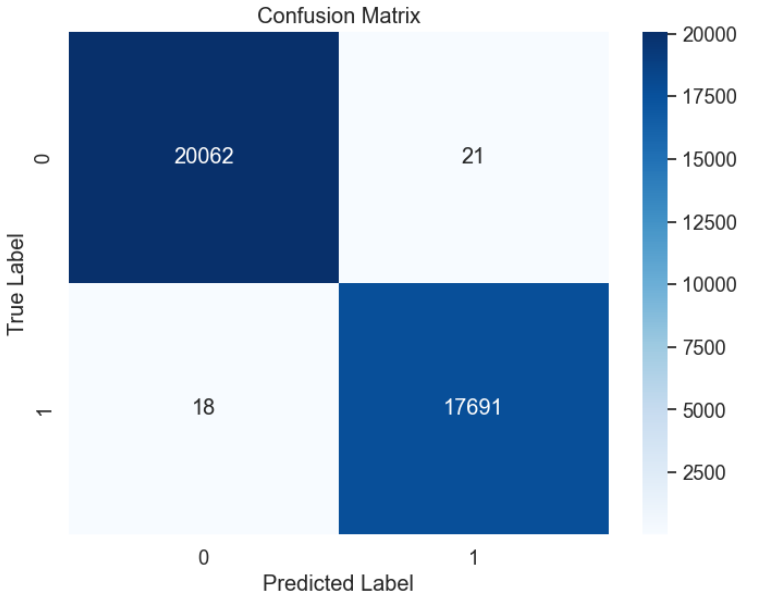


[fig 3: NSL-KDD attack class distribution]

This chapter presents the findings of the intrusion detection system using the PI-LightGBM technique on the NSL-KDD dataset. The detection performance is assessed in this work using a variety of metrics, including accuracy, false positive rate, F-measure, and true positive rate (also known as detection rate). This review highlights the framework's efficacy in producing accurate predictions for detecting threats in network security and analyzes the challenges encountered as well as possible areas for growth. We also compare the model's performance with that of current IDS to help contextualize the gains made in

This paragraph lists the definitions of each: False Positive (FP) is the number of actual normal records classified as attacks; True Positive (TP) is the number of actual attacks classified as attacks; and False Negative (FN) is the number of actual attacks classified as normal ones.

The confusion matrix of my proposed frame work is given below in figure 5.



[Fig 5 : Confusion Matrix of the experiment]

Accuracy (ACC) is a measure of the system's overall performance.

$$\text{Accuray} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FN} + \text{FP}} \qquad (4)$$

True Positive Rate (TPR) is used to evaluate the system's perfor- mance with respect to its malware traffic detection. This metrics is also well known as Detection Rate (DR) or Recall.

$$\text{True Positive Rate} = \frac{\text{TP}}{\text{TP} + \text{FN}} \qquad (5)$$

False Positive Rate (FPR) is used to evaluate the misclassifications of normal traffic.

$$\text{False Positive Rate} = \frac{\text{FP}}{\text{TN} + \text{FP}} \qquad (6)$$

I've examined my suggested framework with alternative algorithms. The binary classification based on the NSL-KDD dataset is displayed in Table 2. The TPR, FPR, and Accuracy (ACC) that each classifier produced are shown in this table. My framework yields the best results, as can be seen, with 99.89% TPR, 0.10% FPR, and 99.63% accuracy.

Table 2			
Method	**TPR**	**FPR**	**ACC**
SwiftIDS(Dongzi Jin et al., 2020)[19]	98.93	0.28	99.4
LSSVM-IDS+FMIFS(Ambusaidi et al., 2016)[20]	98.88	1.12	96.55
PI-LightGBM	**99.89**	**0.10**	**99.63**

5. Conclusion:

To sum up, the main objective of intrusion detection systems (IDS) is to detect and stop attacks before they have a substantial negative impact. Nonetheless, the growing amount of traffic data in high-speed networks presents serious difficulties for IDS solutions now in use. For real-time intrusion detection systems (IDSs) to function well in real-time settings and retain high detection performance, several issues must be resolved. The use of PI-LightGBM is suggested in this study to improve the accuracy and speed of traffic analysis. This method seeks to greatly enhance the real-time capabilities of IDS by combining Intelligent Feature Selection with the LightGBM algorithm, guaranteeing strong and effective network protection against dynamic cyber threats.

References :

[1]
Habeeb, R.A .A . , Nasaruddin, F. , Gani, A . , Hashem, I.A .T. , Ahmed, E. , Imran, M. , 2019. Real-time big data processing

[2]
Hadoop,. http://hadoop.apache.org/ .
Huang, G.B. , Zhou, H. , Ding, X. , Zhang, R. , 2012. Extreme learning machine for regression and multiclass classification. IEEE Trans. Syst. Man Cybern. B Cybern. 42 (2), 513–529 .

[3]
Ke, G. , Meng, Q. , Finley, T. , Wang, T. , Chen, W. , Ma, W. , Ye, Q. , Liu, T.Y. , 2017. Light- GBM: a highly efficient gradient boosting decision tree. Adv. Neural Inf. Process Syst. 3149–3157 .

[4]
HUI JIANG 1, ZHENG HE 2,3, GANG YE 2,3, AND HUYIN ZHANG: Network Intrusion Detection Based on PSO-Xgboost Model:2020

[5]
Ruizhe Yao, Ning Wang∗, Zhihui Liu, Peng Chen, Di Ma, Xianjun Sheng, Intrusion detection system in the Smart Distribution Network: A feature engineering based AE-LightGBM approach

[6] A.Rebekah Johnson, N.Parashuram .S.Prem Kumar, (2014). Organizing of Multipath Routing For International Journal of Intelligent Systems and Applications in Engineering IJISAE, 2023, 11(7s), 718–731 | 731

[7] Al-Safi, A. H. S., Hani, Z. I. R., & Zahra, M. A. (2021). Using a hybrid algorithm and feature selection for network anomaly intrusion detection. J Mech Eng Res Dev, 44(4), 253-262.

[8]
Alghanam, O. A., Almobaideen, W., Saadeh, M., & Adwan, O. (2023). An improved PIO feature selection algorithm for IoT network intrusion detection system based on ensemble learning. Expert Systems with Applications, 213, 118745.

[9]
Thaseen, I. S., & Kumar, C. A. (2017). Intrusion detection model using fusion of chi-square feature selection and multi class SVM. Journal of King Saud University-Computer and Information Sciences, 29(4), 462-472.

[10]
Ambusaidi, M.A. , He, X.J. , Nanda, P. , Tan, Z.Y. , 2016. Building an intrusion detection system using a filter-based feature selection algorithm. IEEE Trans. Comput. 65 (10), 2986–2998 .

[11]
Singh, P., & Tiwari, A. (2015, May). An efficient approach for intrusion detection in reduced features of KDD99 using ID3 and classification with KNNGA. In 2015 second international conference on advances in computing and communication engineering (pp. 445-452). IEEE.

[12]
Barkah, A. S., Selamat, S. R., Abidin, Z. Z., & Wahyudi, R. (2023). Impact of Data Balancing and Feature Selection on Machine Learning-based Network Intrusion Detection. JOIV: International Journal on Informatics Visualization, 7(1).

[13]
Subramani, S., & Selvi, M. (2023). Multi-objective PSO based feature selection for intrusion detection in IoT based wireless sensor networks. Optik, 273, 170419.

[14]
Ambusaidi, M.A. , He, X.J. , Nanda, P. , Tan, Z.Y. , 2016. Building an intrusion detection system using a filter- based feature selection algorithm. IEEE Trans. Comput. 65 (10), 2986–2998 .

[15]
V. N. G. Raju, K. P. Lakshmi, V. M. Jain, A. Kalidindi and V. Padma, "Study the Influence of Normalization/Transformation process on the Accuracy of Supervised Classification," 2020 Third International Conference on Smart Systems and Inventive Technology (ICSSIT), Tirunelveli, India, 2020

[16]
Wang, D. , Zhang, Y. , Zhao, Y. , 2017. LightGBM: an effective miRNA classification method in breast cancer patients. In: Proc. Int. Conf. Comput. Biol. Bioinf. (IC- CBB), pp. 7–11 .

[17]
Zhang, Y. , Wang, Y. , Gao, M. , Ma, Q. , Zhao, J. , Zhang, R. , Wang, Q. , Huang, L. , 2019. A predictive data feature exploration-based air quality prediction approach. IEEE Access 7, 30732–30743 .

[18]
Fisher, W.D. , 1958. On grouping for maximum homogeneity. Amer. Statist. J. 789–798 .

[19]
Dongzi Jin a , ∗, Yiqin Lu a , b , 1 , Jiancheng Qin a , Zhe Cheng b , Zhongshu Mao :
SwiftIDS: Real-time intrusion detection system based on LightGBM and parallel intrusion detection mechanism

[20]
Ambusaidi, M.A. , He, X.J. , Nanda, P. , Tan, Z.Y. , 2016. Building an intrusion detection system using a filter-based feature selection algorithm. IEEE Trans. Comput. 65 (10), 2986–2998 .