Novel Wireless Sensor Network Intrusion Detection Method Based on LightGBM Model

Di Meng, Hong Dai, Qiaochu Sun, Yao Xu, Tianwei Shi

***Abstract*—Wireless sensor networks suffer from a lack of resources. To address the issue, this research provides a wireless sensor network intrusion detection method based on LightGBM. The SMOTE-Tomek approach is used to balance the dataset in the data preparation step to overcome the problem of data imbalance in WSNs intrusion. A feature selection approach is proposed to avoid excessive resource usage and enhance the model's prediction performance. The precise procedure is as follows. Firstly, LightGBM is used to create an iterative lifting tree model and then SHAP is used to determine the relevance of features. Finally, the Recursive Feature Elimination technique is employed in the selection of features. To identify the best-suited settings, the Optuna algorithm with the CMA-ES sampler is utilized to tweak model parameters. The WSN-DS data collection is being used to investigate intrusion detection in wireless sensor networks. In order to assess the model's performance, we compare it to models generated by standard methods. The proposed model outperforms previous models in many predicted performance measures, and the detection rate of each attack type is enhanced to more than 99% according to experimental data. The modeling time is decreased by 46% when compared to the dataset without feature dimension reduction. The suggested approach may efficiently identify network assaults and improve network security.**

***Index Terms*—LightGBM, Optuna, Intrusion Detection, SHAP**

I. Introduction

IN recent years, the Internet of Things has emerged as a hot technology in academic research and industrial manufacture, and it will play a key part in the future Internet world. As an integral component of the Internet of Things, wireless sensor networks (WSNs) are playing an increasingly crucial role in information perception. WSNs are a type of distributed wireless network that consists of several low-power sensor nodes that are installed in the sensing region and connected via wireless communication links. Sensor nodes are a type of tiny computer device with limited store space, low computational performance, and battery power supply. WSNs suffer a range of security vulnerabilities because of the openness of wireless networks and the limitations of sensor nodes.

Intrusion detection technology is a critical technical tool for ensuring security. It is a critical technique for properly identifying various network assaults. Aljawarneh et al. thoroughly investigated several algorithms such as AdaBoost and random forest and coupled numerous models with the concept of integrated learning to improve intrusion detection rates. However, generalization ability was low due to the model's lack of specificity and complexity [1]. As the detecting mechanism of wireless sensor networks, Liu et al. developed the KSOM-PSO [2] neural network technique based on the fusion of kernel self-organizing map and particle swarm optimization clustering algorithm. The detection accuracy is substantially enhanced as compared to other standard methods, and the convergence speed is rapid. The downside is that some neural network settings are random, which has a significant impact on detection outcomes. To address the issue that the AdaBoost method weight allocation is vulnerable to outliers, Dai et al. devised a BOSA-SVM [3] intrusion detection technique. As a consequence of improving the loss function of the self-paced learning model, the detection rate of DoS attacks in wireless sensor networks is enhanced, but the detection rate of other assaults still needs to be improved. Dong et al. suggested an intrusion detection model for distributed WSNs based on random forest and deep forest algorithms [4]. It used lightweight random forests of sensor nodes and cluster heads for anomaly detection and performed misuse detection with deep forest in the base station nod. The method helped in the first and second-order classifiers in the simplest to filter out attack traffic data and distributed detection distracted the energy costs and reduced the communication burden, but the ability to detect multiple classification models needs to be improved further.

Currently, the detection rate of WSN assaults needs to be enhanced. The detection effectiveness is low due to the flexibility of the algorithm and the complexity of traffic characteristics, which cannot match the detection criteria with the spike in network traffic. Furthermore, the present methodology ignores the poor detection rate of a few types of assaults generated by an uneven distribution of regular and anomalous data.

To address the drawbacks of the aforementioned detection approaches at the current time, this research provides a

Manuscript received May 14, 2022; revised September 29, 2022. This work is supported by the Graduate Education Reform and Innovation Entrepreneurship Project, University of Science and Technology Liaoning (2021YJSCX09) and Natural Science Foundation project of Liaoning Province (2021-KF-12-06).

Di Meng is a postgraduate of the College of Computer Science and Software Engineering, University of Science and Technology Liaoning, Anshan, CO 114051, China (e-mail: 997184646@qq.com).

Hong Dai is a professor of the College of Computer Science and Software Engineering, University of Science and Technology Liaoning, Anshan, CO 114051, China (corresponding author: tel+086-18642268599; fax: 0412-5929818; e-mail: dear_red9@163.com).

Qiaochu Sun is a postgraduate of the College of Computer Science and Software Engineering, University of Science and Technology Liaoning, Anshan, CO 114051, China (e-mail: 912410855@qq.com).

Yao Xu is a postgraduate of the College of Computer Science and Software Engineering, University of Science and Technology Liaoning, Anshan, CO 114051, China (e-mail: 2306387642@qq.com).

Tianwei Shi is an associate professor of the College of Computer Science and Software Engineering, University of Science and Technology Liaoning, Anshan, CO 114051, China (e-mail: tianweiabbcc@163.com).

WSNs intrusion detection framework based on the LightGBM algorithm. Firstly, in order to prevent gradient explosion, the data are numerically processed and normalized. Secondly, the data are resampled using the SMOTE-Tomek approach to ameliorate the imbalance. Thirdly, the SHAP model interpretation approach is used to get the feature importance ranking and the recursive feature elimination method is used to select the features. Finally, the LightGBM method is used to predict multi-classifications and the final model is created using Optuna and the CMA-ES sampler. The test set is utilized for prediction and the model's performance is evaluated based on the findings.

II. Methodology

A. Sampling

The WSN-DS dataset utilized in this work has an abnormally uneven data distribution. The amount of normal data accounted for 90.77% of the overall data, whereas the number of attacks accounted for just 0.88%. Because of the data imbalance, the classifier prefers to forecast samples into most classes when making predictions. Among the challenges investigated in this study, detecting the assault category as a minority class is of particular interest.

In this study, the processing of unbalanced data balance was done using SMOTE-Tomek [5]. This approach is a comprehensive sampling methodology that combines the SMOTE and the Tomek-Links techniques.

Chawla proposed the SMOTE [6] oversampling technique. This approach differs from random sampling in that it is a heuristic method that creates a small number of data categories by conducting functional transformations on the original data. It decreases the danger of overfitting when compared to random sampling. The precise procedure is as follows.

Step1. For each sample x_a in a minority class, the Euclidean distance is used to calculate its distance to all samples in the sample set of other minority classes to obtain its K-nearest neighbor.

Step2. A sampling ratio is set according to the sample imbalance ratio to determine the sampling ratio N. For the minority class sample x_a, multiple samples are randomly selected from their K-nearest neighbors and the selected neighbor is labeled x_b.

Step3. For each neighbor x_b which is randomly selected, a new sample is constructed with the original sample x_a according to the formula (1).

$$c = x_a + rand(0,1) * |x_a - x_b| \quad (1)$$

The SMOTE approach is prone to the issue of sample overlap. To some extent, the method the Tomek-Links method can help to mitigate. In order to achieve a more equitable distribution of the data, the Tomek-Links algorithm proposes eliminating some of the representative samples from the majority class.

The basic principle is to select any minority class sample x_{min} and majority class sample x_{max} and then the distance $d(x_{min}, x_{max})$ between them will be calculated. The algorithm determines whether there is a sample y such that $d(x_{min}, y) < d(x_{min}, x_{max})$ or $d(x_{max}, y) < d(x_{min}, x_{max})$. If no such sample exists, the sample is referred to as a Tomek-Links pair. If the two samples are Tomek-Links pairs, then one of them is noise, or the two samples are spread along the sample boundary. Finally, on the assumption that all types of samples are generally balanced, the majority of class samples in Tomek-Links pairs are eliminated to maintain the classification border as obvious as feasible.

B. LightGBM

The gradient boosting decision tree (GBDT) has been made more efficient by the use of the lightweight gradient boosting model [7] (LightGBM). GBDT uses the CART decision tree as the base classifier. It first trains the base classifier by continuously reducing the residuals and then it weights the base classifier to form a strong classifier. This means that the initial model is left untouched at each time iteration and a new function is added to the model to ensure the predicted values continue to be as close as possible to the actual values.

The objective function of training is shown in formula (2), where y_i is the actual value of the label, $\hat{y}_i^k$ is the result of the kth learning and c^{k-1} is the sum of the regularization terms of the first k-1 trees, $f_k(x_i)$ is the predicted value of the new model that is going to be added. Ω is the regularization term of the model. The purpose of the goal function is to locate an appropriate tree f_k in such a way that the value of the function is reduced to its smallest possible value.

$$Obj^k = \sum_i L(y_i, y_i^k) + \Omega(f_k) + c^{k-1}$$

$$= \sum_i L(y_i, y_i^{k-1} + f_k(x_i)) + \Omega(f_k) + c^{k-1} \quad (2)$$

Taylor formula is used to expand the objective function as shown in formula (3).

$$f(x+\Delta x) = f(x) + f'(x)\Delta x + \frac{1}{2} f''(x)\Delta x^2 \quad (3)$$

The second-order Taylor expansion results of the loss function are shown in formula (4).

$$\sum_i L(y_i, y_i^{k-1} + f_k(x_i)) = \sum_i [L(y_i, y_i^{k-1}) + L'(y_i, y_i^{k-1}) f_k(x_i)$$

$$+ \frac{1}{2} L''(y_i, y_i^{k-1}) f_k^2(x_i)] \quad (4)$$

The first derivative of the sample objective function of the ith times is called g_i and the first derivative of the sample objective function of the ith times is called h_i. These two variables are obtained from formula (5) and formula (6) respectively

$$g_i = L'(y_i, y_i^{k-1}) \quad (5)$$

$$h_i = L''(y_i, y_i^{k-1}) \quad (6)$$

The simplified objective function can be expressed as formula (7).

$$Obj^k = \sum_i [L(y_i, y_i^{k-1}) + g_i f_k(x_i) + \frac{1}{2} h_i f_k^2(x_i)] + \Omega(f_k) + \mathrm{c} \quad (7)$$

Because the GBDT algorithm uses a pre-sorted iteration mechanism, it must visit the whole dataset multiple times. This defect results in excessive space and time complexity. LightGBM uses the enhanced Histogram technique to partition continuous floating-point eigenvalues into k intervals, as illustrated in Fig.1. The training speed and space consumption efficiency are considerably improved by just picking the ideal segmentation points in the k interval.

Furthermore, the LightGBM method employs the Leaf-Wise strategy instead of the Level-Wise strategy while creating a decision tree. This approach increases the maximum depth limit while guaranteeing high efficiency to avoid over-fitting and reduces the training data. The Level-Wise strategy and the Level-Wise strategy are shown in Fig.2 and Fig.3 respectively.

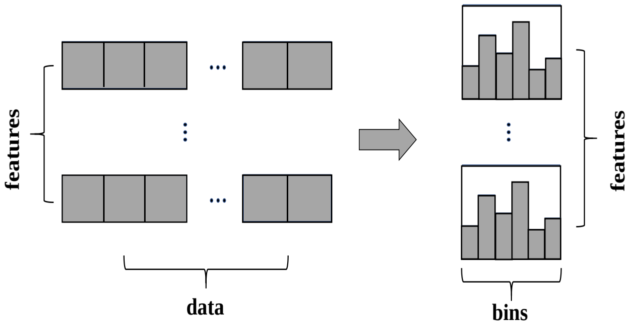

Fig.1. Schematic diagram of histogram algorithm

To provide accurate information gain estimates with a reduced data volume, gradient-based one-side sampling is used to keep examples with greater gradients and random sampling on instances with lesser gradients.

LightGBM uses the Exclusive feature bundling method to bundle mutually Exclusive features at a fixed conflict rate in order to accomplish dimensionality reduction without information loss.

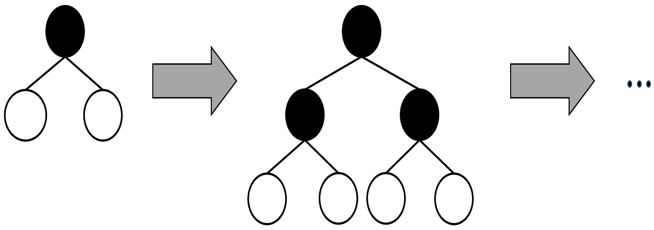

Fig.2. Level-wise strategy

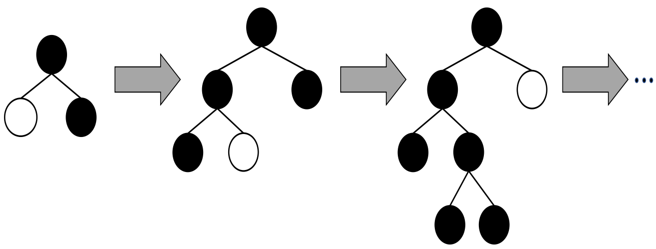

Fig.3. Leaf-wise strategy

C. *Calculation of Feature Importance*

SHAP is a game theory-based strategy that uses the marginal effects of features to explain the output of any machine learning model. When compared to other feature significance measures based on the model's properties, it can explain the contribution to projected outcomes from the perspective of a single sample rather than the global relevance of the entire dataset [8].

The Shapley value in combinatorial game theory inspires the core notion of SHAP. The Shapley value of a feature is the difference between the old set's contribution and the new feature's contribution. If the shapely value of feature D is computed, the existing feature set is S and $v(S)$ is the existing set's contribution, then the shapely value may be estimated using formula (8).

$$\phi_D = v(S \cup \{D\}) - v(S) \tag{8}$$

Because the original model is too complicated to be utilized as an interpretation model for the integrated model and neural network, an interpretation model with a shorter definition is required to use this model as an approximate interpretation of the original model. The additive feature contribution technique is a linear function of a model-independent binary variable. It is offered as an explanatory approach as indicated in formula (9).

$$g(z') = \phi_0 + \sum_{i=1}^{M} \phi_i z_i' \tag{9}$$

Where g is the explanatory model, M is the collection of features, ϕ_0 is the mean value of the input data used in the model prediction, ϕ_i represents the Shapley value of the ith feature and z_i' takes the value of 0 or 1, denoting the existence or the absence of the feature, respectively. Through formula (8), we can determine the significance of a particular feature. Firstly, we find all possible combinations of features that do not include that feature. Then we compute the Shapley value of the combination that corresponds to that feature. Finally, we weight the value to add it to the total.

$$\phi_j = \sum_{S \subset \{x_1,\dots,x_p\} \setminus \{x_j\}} \frac{|S|!(p-|S|-1)!}{p!} (val(S \cup \{x_j\}) - val(S)) \tag{10}$$

Where S is the subset of features used in the model, x is the vector of eigenvalues of the samples to be interpreted, p is the number of features, $val(S)$ is the model output value in the case of feature combination S, $p!$ Represents the number of combinations in the featured case where the number is p, after selecting a feature *'j'*, the number of combinations remaining is $|S|!(p-|S|-1)!$.

SHAP develops a machine learning model augmentation based on Shapley value, with features serving as participants and model outputs serving as cooperative outcomes. Because of the nature of the machine learning model, the SHAP value is derived not by deleting feature *'j'* from the actual process, but by considering the difference between its actual output and its average output.

The SHAP-based feature significance computation approach is theoretically sound and solves the problem of feature multicollinearity. It investigates not only the impacts of a single variable but also the possible synergistic effect of several variables.

D. *Optuna*

A Japanese deep learning firm proposed Optuna as a parameter optimization system. Optuna achieved the optimal solution throughout the optimization process by continuously calling and evaluating the objective function with varied parameter values [9]. The optimization algorithm's particular flow is as follows.

i. The optimization direction, parameter type, value range, and the maximum number of iterations are all determined during the initialization process.

ii. Enter the cycle;

iii. Choose individuals from the population within the objective function of setting parameter value range;

iv. According to the trimming circumstances, the trimmer automatically ends hopeless people;

v. Calculate the individual objective function value of the unpruned population;

vi. Repeat the above steps until reaching the maximum number of iterations and jumping out of the cycle;

vii. Output the optimal solution and optimal objective functions value.

III. BUILD MODEL

A. Data Preprocessing

The numerical experiments in this research use 10% of the WSN-DS dataset as network intrusion detection dataset to assess the model's classification accuracy. The data collection is created specifically for WSN intrusion detection [10]. It replicates the WSN environment using the NS-2 simulator, gathers data in the perception-layer network environment using the LEACH routing protocol and simulates four attack types: blackhole, grayhole, flooding, and TDMA.

Because there are non-numeric data in the data collection, such as strings, that the model cannot identify, these data are quantized to generate data that the model can utilize regularly.

To increase the model's accuracy and convergence speed, the min-max approach is used to normalize the data and the data are mapped between (0, 1) using linear data operations, as indicated in formula (11).

$$x'' = \frac{x' - min}{max - min} \tag{11}$$

Where *min* and *max* are the minimum and maximum values of *x'* respectively.

Due to the data imbalance in the dataset, the practicability and classification performance of the model will be directly affected and data overfitting will occur. To avoid these issues, the experiment employs SMOTE-Tomek technology to balance the dataset. The pre-processed dataset has met the model's criteria and may be used to assess the model's performance. Table I shows the specifics of data processing.

TABLE I
DATA PROCESSING DETAILS

Attack types	Numerical label	Sample size before sampling	Sample size after sampling
Normal	0	34022	33958
Blackhole	1	1037	33873
Grayhole	2	1482	33872
Flooding	3	281	34022
TDMA	4	644	33959

B. Feature Selection

Redundant and irrelevant features in the dataset will raise the computational load of the model, slow down training speed and potentially lead to over-fitting. Screening these characteristics can decrease excessive resource use while improving the model's prediction performance. The recursive feature elimination approach is used with SHAP in this research for dataset feature selection.

The recursive feature elimination approach sorts the relevance of features using the chosen base model. The features with the lowest weight are deleted in each round of training and the iterative procedure is repeated until the number of features remaining fulfills the requirements.

LightGBM is utilized as the basic classifier in this research, the SHAP approach is employed to determine the significance ranking of features and recursive feature removal is performed on this basis.

Table II shows the feature selection procedure once eight features are chosen.

TABLE II
SHAP-RFE ALGORITHM

Input: Dataset Q=*WSN-DS*, Feature set $F = (f_1, f_2, \ldots, fm)$
Output: The best subset of features

While (F! =0)
- Train model with LightGBM algorithm using F
- Use SHAP to calculate the weight value of each feature
- Sort the feature weight values $New_{rank}=sort\ (Rank)$
- Update the feature sorting list $Update(R)=R+F(New_{rank})$
- Delete the feature contributes the least $Update(F)=F-F(New_{rank})$
- Obtain the evaluation function values of each feature subset
- Determine the best subset of features

End While

The selected features are described in Table III.

TABLE III
INTRODUCTION OF FEATURES

Feature	Description
ADV_S	The number of advertise CH's broadcast messages sent to the nodes
SCH_S	The number of advertise TDMA schedule broadcast messages sent to the nodes.
Is_CH	A flag to distinguish whether the node is CH with value 1 or normal node with value 0.
Data_Sent_To_BS	The number of data packets sent to the BS
ADV_R	The number of advertised CH messages received from CHs
Consumed Energy	The amount of energy consumed in the previous round
Time	The current simulation time of the node
DATA_R	The number of data packets received from CH

C. Optuna with CMA-ES sampler

Optuna is prone to slipping into a local optimum during the solution process, resulting in a large error in the outcome. The CMA-ES algorithm can achieve fast convergence by controlling the step size and improving the algorithm's global search ability. Hansen [11] suggests the CMA-ES algorithm as an optimization technique. It optimizes the problem using the solution space of Gaussian distribution sampling and then uses the better solution to update the Gaussian distribution parameters. The sampling and updating operation are repeated indefinitely until reaching the stop condition.

The specific steps of the CMA-ES algorithm are as follows.

i. The CMA-ES method chooses a solution at random from the problem space and then creates the population using the normal distribution, as indicated in formula (12).

$$x_i^{(g+1)} \sim m^{(g)} + \sigma^{(g)} N(0, C^{(g)}), \quad i=1, 2, \ldots, \alpha \tag{12}$$

Where x_i is the ith individual in the population; α is population size; g is evolutionary algebra; m is the mean of the search distribution; σ is the step size; C is the covariance adaptive matrix of the population and the initial C can be set as an identity matrix of $D \times D$, where D is the dimension of the solution.

ii. Recalculate the mean value. Choose some ideal solutions to represent subpopulations in the created population and the anticipated value of the new population is determined using formulas (13) and (14).

$$m^{(g+1)} = \sum_{i=1}^{\beta} \omega_i x_{i:a}^{(g+1)} \tag{13}$$

$$\sum_{i=1}^{\beta} \omega_i = 1, \omega_1 \geq \omega_2 \geq \cdots \geq \omega_\beta > 0 \tag{14}$$

Where β is the size of the subpopulation; ω_i is a positive weight; $x_{i:\alpha}$ is the *i*th best individual selected from the α *th* individuals in the population.

iii. Update the covariance adaptive matrix. By updating the covariance adaptive matrix, the entire population is directed to find the global optimal solution. The updating formula of the covariance adaptive matrix is computed using formula (15).

$$C^{(g+1)} = (1-c_1-c_\beta)C^{(g)} + c_1 p_c^{(g+1)} (p_c^{(g+1)})^T + c_\beta \sum_{i=1}^{\alpha} \omega_i y_{i:a}^{(g+1)} y_{i:a}^{(g+1)^T} \tag{15}$$

Where c_1 is the learning rate in mode *Rank-1-update*, $c_1 \approx 2/D^2, c_\beta \approx min(\mu_{eff}/D^2, 1-c_1)$, c_β is the learning rate in mode *Rank-β -update*, $y_{i:a}^{(g+1)} = x_{i:a}^{(g+1)} - m^{(g)})/\sigma^{(g)}$, μ_{eff} is the effective selection quality of variance, it can be obtained from formula (16), p_c is the evolutionary path of the covariance adaptive matrix.

$$\mu_{eff} = (\sum_{i=1}^{\beta} \omega_i^2)^{-1} \tag{16}$$

The initial p_c is 0 and its update process is shown in formula (17).

$$P_c^{(g+1)} = (1-c_c)P_c^{(g)} + \sqrt{c_c(2-c_c)\mu_{eff}} \bullet [m^{(g+1)} - m^{(g)} / \sigma^{(g)}] \tag{17}$$

Where c_c is the learning rate of evolutionary path p_c.

The updating process of step σ in formula (16) is shown in formula (18).

$$\sigma^{(g+1)} = \sigma^{(g)} \exp[\frac{c_\sigma}{d_\sigma}(\frac{\| p_\sigma^{(g+1)} \|}{E \| N(0,I) \|} - 1)] \tag{18}$$

Where *E ()* is the mathematical expectation function; *I* is the identity matrix; c_σ is the learning rate of step size; d_σ is the damping coefficient of step size update; p_σ is the evolutionary path of step size, initial p_σ is 0 and its updating process is shown in formula (19).

$$p_\sigma^{(g+1)} = (1-c_c)P_\sigma^{(g)} + \sqrt{c_\sigma(2-c_\sigma)\mu_{eff}} \bullet C^{(g-\frac{1}{2})}[m^{(g+1)} - m^{(g)}) / \sigma^{(g)}] \tag{19}$$

CMA-ES algorithm circulates continuously according to the above steps and obtains the optimal solution step by step.

IV. Experiments

A. Feature Selection

Following pre-processing, the original data are modeled using the LightGBM technique and the SHAP value of the data is calculated using the SHAP method. Figure 4 depicts the significance assessment.

SHAP-RFE feature selection is performed after the importance of the features is ranked in succession and the outcome of feature selection is presented in Fig.5. As shown in Fig.5, when the number of selected features is set to 8, the model retains a high level of accuracy while considerably decreasing the number of selected features.

To validate the efficacy of the feature selection method used in this paper, several commonly used feature dimension reduction methods are chosen for comparison with the method in this paper, as shown in Table IV. The table shows that the method proposed in this paper has the highest accuracy when the number of features is the same. The strategy utilized in this study decreases modeling time by 46% and accuracy by just 0.02% when compared to the dataset without feature selection.

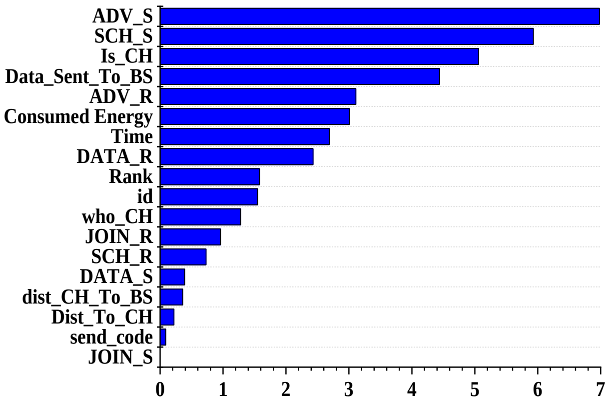

Fig.4. SHAP values of different features

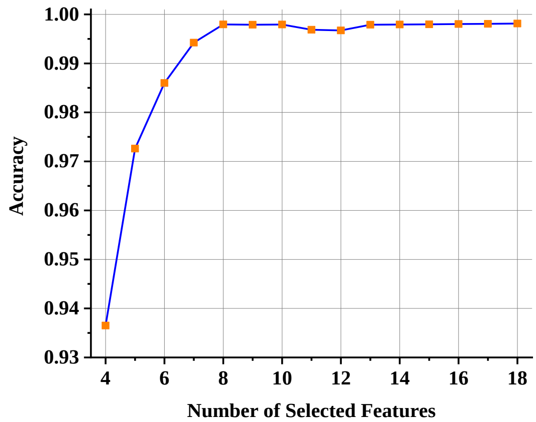

Fig.5. The accuracy of selecting different numbers of features

TABLE IV
COMPARISON OF DIFFERENT FEATURE DIMENSION REDUCTION METHODS

Feature dimension reduction method	Number of features	Accuracy (%)	Time(s)
PCA	8	98.85	4.08
IG	8	98.44	4.23
ICA	8	99.28	4.19
RFE-SVM [12]	8	98.22	4.17
method proposed	8	99.76	4.20
original data	18	99.78	9.11

B. Parameter Optimization

In the classification process, parameters [13] need to be adjusted to take full advantage of the LightGBM algorithm. During the parameter optimization process, accuracy is employed as the objective function and the dataset should be separated into training and test sets. The experimental findings reveal that the model performs best when the training set accounts for 70% and the test set accounts for 30%. The training set sample has 118,778 pieces of data and the test set sample contains 50,906 pieces of data. The training set sample to the test set sample ratio is 7 to 3. In the experiment, the number of iterations of the Optuna optimization method was set at 100. The parameter range and the final optimal value are presented in Table V. The curve of the iterative process is displayed in Fig.6.

When compared to the default parameters, the parameters chosen by the parameter optimization method enhance the overall accuracy of the model by 0.04%.

TABLE V
SPECIFIC VALUE OF PARAMETER OPTIMIZATION

Parameter name	Parameter range	Parameter values
max_depth	(5,15)	7
learning_rate	(0.1,1.0)	0.141
feature_fraction	(0.6,1)	0.809
min_split_gain	(0,0.1)	0.010
bagging_fraction	(0.6,1)	0.789
n_estimators	(100,350)	291
num_leaves	(150,300)	250
max_bin	(100,256)	200

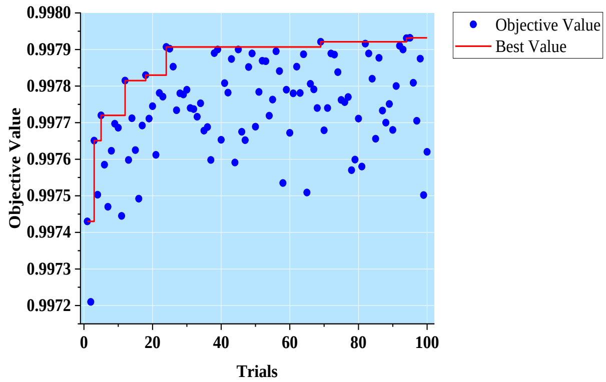

Fig.6. Optuna optimized iterative process

C. Result comparisons and discussion

Some performance markers for comparison are required to assure the experiment's correctness.

The four types of performance indicators used to evaluate the classification model are as follows: accuracy, precision, recall and F measure [14]. Indicators of performance are specified as shown in formula (20)-formula (23). *TP* is a positive sample that has been judged to be positive, *TN* is a negative sample that has been judged to be negative, *FP* is a negative sample that has been judged to be positive and *FN* is a positive sample that has been judged to be negative.

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

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

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

$$F\_measure = \frac{2 * Precision * Recall}{(Precision + Recall)} \tag{23}$$

The confusion matrix obtained in the final experiment is shown in Table VI below.

TABLE VI
CONFUSION MATRIX

Predicted class \ Concrete class	Normal	Blackhole	Grayhole	Flooding	TDMA
Normal	9995	0	8	11	10
Blackhole	0	10118	33	0	0
Grayhole	0	7	10166	0	1
Flooding	0	0	0	10290	0
TDMA	35	0	0	0	10232

In order to provide a more in-depth analysis of the efficacy of the model that is presented in this article, it is contrasted with several algorithms that are typically used, as shown in Table VII.

In addition, we compared the modeling time of different algorithms, as shown in Table VIII.

The experimental results show that the method provided in this study is the most effective in all indicators for processing the same type of data. The results of the experiments demonstrate that the suggested algorithm is better than other algorithms when it comes to the identification of intrusions using wireless sensors.

TABLE VII
CLASSIFICATION PERFORMANCE

Model	Performance Index	Model Performance (%)				
		Normal	Blackhole	Grayhole	Flooding	TDMA
Our Model	Accuracy			99.79		
	Precision	99.65	99.93	99.59	99.89	99.89
	Recall	99.71	99.67	99.92	100	99.65
	F_measure	99.68	99.80	99.76	99.94	99.77
CatBoost	Accuracy			99.72		
	Precision	99.45	99.85	99.48	99.89	99.91
	Recall	99.70	99.61	99.85	99.98	99.47
	F_measure	99.58	99.73	99.66	99.40	99.69
Random Forests	Accuracy			98.82		
	Precision	96.84	98.35	99.63	99.89	99.48
	Recall	99.28	99.71	99.41	98.91	96.79
	F_measure	98.04	99.03	99.52	99.39	98.12
SVM	Accuracy			89.13		
	Precision	97.52	72.35	92.32	97.91	92.16
	Recall	97.30	97.69	55.50	97.42	97.12
	F_measure	97.41	83.13	69.33	97.67	94.58

TABLE VIII
MODELING TIME OF THE ALGORITHMS

Model	Time(s)
Our Model	4.20
CatBoost	5.79
Random Forests	6.02
SVM	13.55

V. CONCLUSION

In this study, a method called feature recursive elimination is employed for the screening of features and as the result, new feature vectors are generated. The LightGBM algorithm is enhanced using the Optuna parameter optimization approach with the CMA-ES sampler. Then, the improved LightGBM model is used for the classification and assessment of wireless sensor network intrusion. The results of the experiments reveal that the suggested model has a higher level of performance than the conventional techniques of machine learning. Furthermore, it has the potential to effectively increase the accuracy of intrusion detection as well as the capacity to recognize numerous types of attacks.

REFERENCES

[1] Aljawarneh S., Aldwairi M., and Yasin M.B., “Anomaly-based intrusion detection system through feature selection analysis and building hybrid efficient model,” Journal of Computational Science, vol.22, no.5, pp628-637, 2017.
[2] Liu S., Shi F., Wang L.J., Qin J.W., and Guo Y., “Research on intrusion detection in wireless sensor networks based on KSOM-PSO algorithm,” China Science Paper, vol.12, no.2, pp148-153, 2017.
[3] Dai J.J., Tao Y., and Zhang S.Q., “Intrusion detection integration algorithm based on self-paced learning in wireless sensor networks,” Computer Engineering and Design, vol.40, no.5, pp1249-1253, 2019.
[4] Dong R.H., Run H.H., Zhang Q.Y., and Li X.Y., “Distributed WSN intrusion detection model based on deep forest algorithm,” Journal of Lanzhou University of Technology, vol.46, no.4, pp103-109, 2020.
[5] Wei Z.Q., Zhang H., and Chen L., “Web Anomaly Detection Model Using SmoteTomek and LightGBM Algorithm,” Journal of Chinese Computer Systems, vol.41, no.3, pp587-592, 2020.
[6] Chawla N.V., Bowyer K. W., Hall L. O., and Kegelmeyer W. P., “Smote: synthetic minority over-sampling technique,” Journal of Artificial Intelligence Research, vol.16, no.1, pp321-357, 2002.
[7] Gu T., Xu G.L., and Li W.L., “Intelligent House Price Evaluation Model based on Ensemble LightGBM and Bayesian Optimization Strategy,” Journal of Computer Applications, vol.361, no.9, pp290-295, 2020.
[8] Lundberg S.M, and Lee S.I., “A unified approach to interpreting model predictions,” Proceedings of the 31st International Conference on Neural Information Processing Systems, 4-9 December, 2017, Anchorage, United States, pp4765-4774.
[9] Akiba T., Sano S., Yanase T., Ohta T., and Koyama M., “Optuna: A next-generation hyperparameter optimization framework,” Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 4-8 August, 2019, Long Beach, United States, pp2623-2631.
[10] Almomani I., Al-Kasasbeh B., and Al-Akhras M., “WSN-DS: a dataset for intrusion detection systems in wireless sensor networks,” Journal of Sensors, vol.12, no.6, pp158-175, 2016.
[11] Hansen N., Müller Sibylle D., and Koumoutsakos P. “Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (cma-es),” Evolutionary Computation, vol.11, no.1, pp1-18, 2014.
[12] Wu L., “Rule Extraction Method Based on SVM-RFE Feature Selection,” Microcomputer Application, vol.37, no.9, pp150-154, 2021.
[13] Wang Y.B., Li Y.J., Wu H.Y., and Duan Y. “Video Flame Detection Based on Multi-feature Fusion and Double-layer XGBoost,” Engineering Letters, vol. 30, no. 2, pp904-911, 2022.
[14] Arwatchananukul S., Saengrayap R., Chaiwong S., and Aunsri N., “Fast and Efficient Cavendish Banana Grade Classification using Random Forest Classifier with Synthetic Minority Oversampling Technique,” IAENG International Journal of Computer Science, vol.49, no.1, pp46-54, 2022.