Research on Network Threat Hunting System Based on Multi Scale LightGBM Ensemble Learning

Yuzhi Wang

Criminal Technology and Information, Hubei University of Police, Wuhan, 430034, China
E-mail: wyz20240310@163.com

Received 02 May 2025; Accepted 16 June 2025

Abstract

With the increasing complexity and concealment of network attacks, traditional single-scale threat detection methods have made it difficult to meet the needs of modern network security. This study proposes a network threat-hunting system based on multi-scale LightGBM ensemble learning, aiming to improve the accuracy and efficiency of threat detection by fusing network data at different time scales and spatial scales. Firstly, the system extracts multi-scale features from network data, including real-time traffic, historical behaviour and topology, and then uses the LightGBM algorithm for ensemble learning. The experimental results show that the threat detection accuracy of the multi-scale feature fusion model is improved by 15.3%, which is significantly better than the single-scale model. At the same time, the LightGBM ensemble learning model performs well in detection efficiency, and the average detection time is shortened by 20.7%. The generalization ability of

Journal of Cyber Security and Mobility, Vol. 14_3, 701–722.
doi: 10.13052/jcsm2245-1439.1438

the system in different network environments has also been verified, and the average threat detection recall rate reaches 92.1%. These results show that the multi-scale LightGBM ensemble learning system performs well in terms of accuracy, efficiency, and generalization ability, providing a new solution for cyber threat detection.

Keywords: Multiscale features, LightGBM, ensemble learning, cyber threat hunting.

1 Introduction

With the rapid development of information technology, the network environment is becoming increasingly complex, and the network threats are becoming increasingly diversified and concealed [1, 2]. Traditional network security defence methods often rely on the feature matching of known threats, making it difficult to deal effectively with emerging and unknown network threats [3]. Therefore, efficiently identifying and dealing with these potential network threats has become an important problem to be solved urgently in network security. In this context, multi-scale LightGBM ensemble learning cyber threat hunting system research came into being, aiming to improve cyber threats' detection capability and response speed through advanced data analysis and machine learning technology.

As a high-performance gradient lifting framework, LightGBM is favoured in the field of machine learning because of its excellent computational efficiency and accuracy [4]. Multi-scale learning can capture the features of data from different levels and enhance the generalization ability and robustness of the model [5]. Combining the two and applying them to cyber threat hunting systems can make full use of the advantages of LightGBM and more comprehensively understand network data through multi-scale analysis to more accurately identify potential threats [6]. Cyber threat hunting systems are different from traditional passive defence systems in that they emphasize taking the initiative to actively discover and track potential threats through continuous network monitoring, data analysis and behaviour analysis [7, 8]. This kind of system needs to process massive network data and extract valuable information from it, which puts forward extremely high requirements for the system's data processing and analysis ability. Multi-scale LightGBM ensemble learning came into being under this demand. It can effectively process large-scale data sets while maintaining high detection accuracy and efficiency.

In practical applications, network threats manifest in various forms, including but not limited to malware attacks, phishing, DDoS attacks, etc. [9, 10]. These threats often have a high degree of complexity and uncertainty, making it difficult for traditional single-scale analysis methods to capture their characteristics [11] fully. Multi-scale LightGBM ensemble learning can dig deeper into potential patterns in data by combining feature information at multiple scales, thereby more effectively identifying network threats. In addition, as the network environment changes, cyber threats constantly evolve [12]. The multi-scale LightGBM ensemble learning system has good adaptive ability, dynamically adjusting according to new data and environmental changes, and maintains high detection performance.

During the research process, we pay attention to the construction and optimization of the model and the effect evaluation in practical applications. By constructing a real network environment and collecting a large amount of network data, we comprehensively test and evaluate the multi-scale LightGBM ensemble learning system. The results show that the system can effectively reduce the false alarm rate and improve the hunting efficiency of network threats while ensuring high detection accuracy. This research result not only provides new technical ideas for the network security field, but also powerful tool support for the actual network threat prevention work.

The research on a network threat-hunting system based on multi-scale LightGBM ensemble learning has important theoretical value and a wide application prospect. With the deepening of research and the continuous progress of technology, this system will play an increasingly important role in future network security defence and contribute to building a more secure and stable network environment.

2 Theoretical Basis and Key Technologies

2.1 LightGBM Algorithm Principle

The LightGBM algorithm has unique advantages in ensemble learning. It uses a histogram algorithm and a leaf-wise growth strategy to significantly reduce the amount of computation and memory compared to traditional decision tree algorithms, and can quickly process large-scale cyber threat data. Under the ensemble learning framework, LightGBM can efficiently train multiple base models in parallel to improve training efficiency. Compared with common ensemble algorithms, random forests are prone to overfitting and slow training speed when processing cyber threat data. Adaboost is

sensitive to outliers and performs poorly in noisy network data. LightGBM, on the other hand, effectively reduces data sparsity and noise impact through gradient unilateral sampling and mutually exclusive feature bundling technology, and performs well in performance indicators such as accuracy, recall, and training time, which can more accurately identify network threats.

By introducing technologies such as GOSS and EFB, LightGBM achieves fast training and low memory footprint, which is suitable for large-scale datasets and is very suitable for industrial applications [13, 14].

Gradient Boosting Decision Tree (GBDT) is an additive model based on the Boosting principle. It forms a strong learner by combining multiple weak learners, and by adjusting the data weight, the data with poor results get more attention in subsequent learning [15]. Weak learners mainly use classification regression tree (CART) to split nodes according to the principle of least variance [16]. The algorithm flow of GBDT includes:

Step 1: Input the training data Input and the loss function $L(y_i, F(x_i))$, x_i represents the Input feature vector, and the calculation process of $X \subseteq R^n$ is as shown in Equations (1)–(2).

$$Input: \{(x_i, y_i)\}_{i=1}^{n}, x_i \in X \subseteq R^n, y_i \in Y \subseteq R^n \tag{1}$$

$$L(y_i, F(x_i)) = (y_i - F(x_i))^2 \tag{2}$$

Step 2: Initialize the weak learner $F_0(x)$ to a fixed value γ, and the calculation method is shown in formula (3).

$$\mathrm{F_0(x)} = \underset{\gamma}{\operatorname{argmin}} \sum_{\mathrm{i=1}}^{\mathrm{n}} \mathrm{L(y_i, \gamma)} \tag{3}$$

Step 3: For the number of iterations $m = 1, 2, \ldots, M$, the pseudo residual r_{im} is calculated according to Equation (4):

$$\mathrm{r_{im}} = -\left[\frac{\partial \mathrm{L(y_i, F(x_i))}}{\partial \mathrm{F(x_i)}}\right]_{\mathrm{F(x)=F_{m-1}(x)}}, \mathrm{i} = 1, \ldots, \mathrm{n} \tag{4}$$

Construct a CART regression tree for r_{im}, and obtain the m-th tree and its leaf region $R_{jm}(j = 1, \ldots, J_m)$, and J_m represents the number of leaf nodes of tree m. The output value γ_{jm} of each leaf node is calculated according to Equation (5).

$$\gamma_{\mathrm{jm}} = \underset{\gamma}{\operatorname{argmin}} \sum_{\mathrm{x_i \in R_{ij}}} \mathrm{L(y_i, F_{m-1}(x_i) + \gamma)} \tag{5}$$

Step 4: Update the reinforcement learner according to Equation (6):

$$\mathrm{F_m(x)} = \mathrm{F_{m-1}(x)} + v \sum_{\mathrm{j=1}}^{\mathrm{J_m}} \gamma_{\mathrm{jm}} \mathrm{I(x \in R_{jm})} \tag{6}$$

The learning rate v is between 0 and 1, determining the scaling of the influence of the tree on the results.

2.2 Multi-scale Ensemble Learning Theory

Cyber threat data has complex spatiotemporal characteristics, and multi-scale classification needs to comprehensively consider the time series characteristics, spatial distribution characteristics, and granularity of data characteristics. In terms of time, according to the frequency and duration of network attacks, the time scale is divided into short-term (minute-level), medium-cycle (hour-level) and long-period (day-level), the short period can catch sudden instantaneous attacks, and the long period is used to detect hidden and persistent threats. In terms of spatial dimensions, according to the network topology, the scale is divided from subnets, campus networks, and WANs to identify threats in different network scopes. In terms of data feature granularity, the fine-grained level covers the micro characteristics such as specific network packet fields and process behaviors, while the coarse-grained level focuses on the macro characteristics such as the total amount of traffic and service access patterns.

When solving complex problems, a single machine learning method is limited by model defects, resulting in limited solutions [17]. Ensemble learning overcomes this problem, and the process is shown in Figure 1.

Ensemble learning combines multiple weak classifiers to form a strong classifier, which usually performs better than a single model. It requires the basic learners to be accurate and diverse. Ensemble learning is divided into two categories: boosting with strong dependence and Bagging without dependence [18, 19].

The Boosting method builds a strong learner by training multiple general performance classifiers multiple times and combining their prediction results [20]. In contrast, Bagging generates multiple models using the same data and classifier multiple times and determines the final output through voting, which is suitable for classifiers with low stability. However, when the classifier is stable, Bagging has limited effect, while Boosting is unaffected by stability [21].

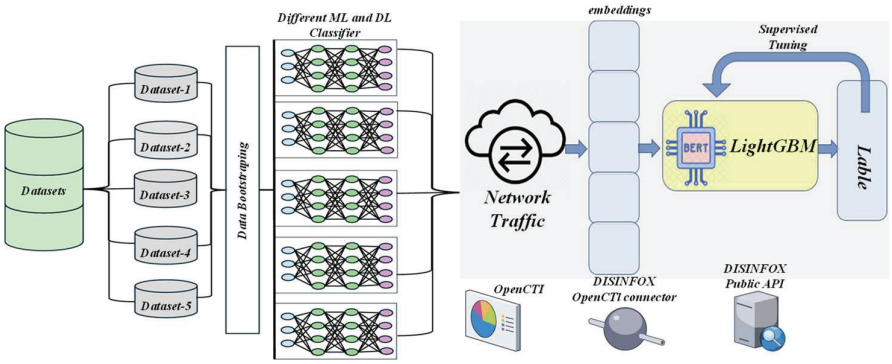

Figure 1 The process of ensemble learning.

The bagging algorithm, also known as Bagging, creates a subset of data by putting back sampling to enhance sample diversity [22]. It trains multiple learners and aggregates the predictions. Bagging is suitable for situations where the base learners vary greatly and are well trained but are sensitive to slight variations in the dataset to improve adaptability.

Hierarchical algorithms, or Stacking algorithms, synthesize model advantages by training multiple learners and meta-learners with their outputs [23]. They use a powerful base learner and a simple meta-learner, usually with the k-fold cross-validation method. This algorithm performs well on multiple data sets, improves robust prediction through model stacking, reduces dependence on a single model, and enhances system stability [24].

3 Construction of Multi-scale LightGBM Ensemble Learning Threat Hunting Model

3.1 Design of LightGBM Ensemble Learning Framework

In a cyber threat hunting system based on multi-scale LightGBM ensemble learning, the weight allocation strategy between different LightGBM models is crucial. The dynamic adaptive weight allocation method is used to adjust the weights in real time according to the performance of each model on the training set, such as accuracy, recall, F1 value and other indicators. Models that perform well in detecting specific types of threats (e.g., ransomware, APT attacks) are given higher weight to dominate the final decision. This weight allocation strategy can give full play to the advantages of each model, effectively improve the accuracy and reliability of threat hunting, and avoid

missed or false detection caused by the limitations of a single model, which will significantly affect the final threat hunting results.

It is compared with the current advanced network threat detection algorithms, such as deep learning-based Long Short-Term Memory Network (LSTM), Generative Adversarial Network (GAN) detection algorithm, as well as traditional Support Vector Machine (SVM) and Random Forest (RF) algorithms, and a simple decision tree model is introduced as the benchmark model. On the same cyber threat dataset, it is evaluated by multi-dimensional metrics such as accuracy, recall, F1 value, training time, and detection delay. Experimental results show that the method based on multi-scale LightGBM ensemble learning not only ensures high detection accuracy and recall, but the training time and detection delay are significantly lower than those of deep learning algorithms, and the robustness of traditional algorithms in complex network environments is better than that of traditional algorithms. Compared with the benchmark model, the accuracy and efficiency of threat detection are significantly improved, which fully demonstrates its superiority in the field of cyber threat hunting.

When building and predicting the algorithm, we used ensemble learning methods, especially the Stacking algorithm combined with Random Forest for prediction. At the same time, XGBoost and LightGBM algorithms are also used for fitting, and the best fitting effect is selected from them. In order to solve the complexity of hyperparameter tuning in ensemble learning, we apply genetic algorithm to automatically optimize the hyperparameters of LightGBM, which saves manual tuning time. The ensemble learning framework LightGBM is shown in Figure 2.

First, the initial dataset is split into a training set and a test set with a 7- to 3 ratio. Data preprocessing is performed, and the ensemble learning model is trained. After repeated training and adjustment of hyperparameters, the trained model is used to predict the test set. Finally, the model performance was evaluated by indicators such as accuracy, recall, precision, F^1_Score and AUC.

The LightGBM model is well-known for its fast training, high memory efficiency, and classification accuracy [25, 26]. When using LightGBM, hyperparameters such as the number of leaf nodes, tree depth, number of iterations, and learning rate greatly influence the experimental results. Manually adjusting these parameters is complex and time-consuming, so it becomes important to optimize the hyperparameters automatically.

The stacking algorithm combines a variety of basic learners, usually choosing different types of classifiers as the first-level learner and using

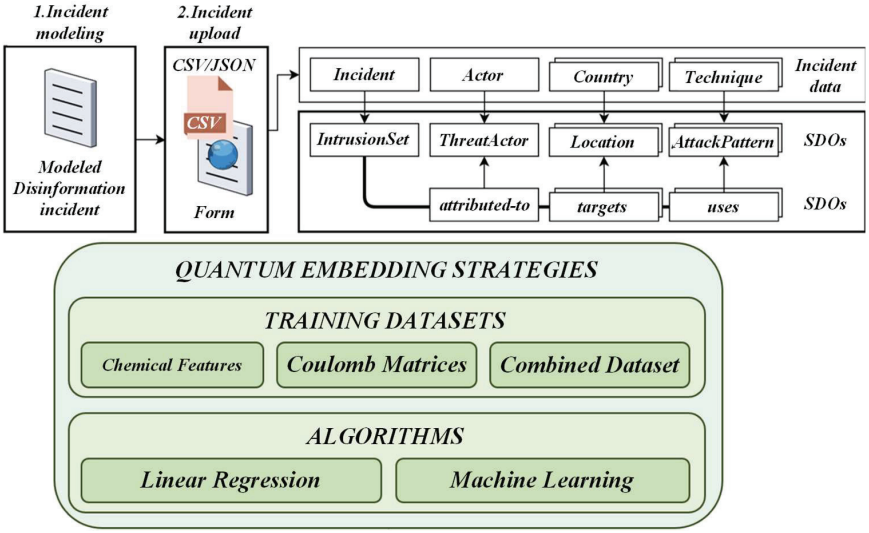

Figure 2 LightGBM ensemble learning framework.

simple models as the second-level learner, to fuse the advantages of models, improve the classification accuracy, and prevent overfitting.

The GA-LightGBM and RandomForest algorithms perform excellently in fitting performance, especially when RandomForest deals with large-scale, high-dimensional datasets [27]. Although RandomForest may overfit on noisy datasets, GA-LightGBM reduces overfitting phenomena with its fast speed, low memory footprint, and high accuracy. Therefore, RandomForest and GA-LightGBM are selected as the Stacking base classifiers. These two algorithms combine their respective advantages, and through independent training and prediction, the Stacking model has more advantages after synthesis, which meets the needs of a diversity of basic classifiers.

After selecting the base classifier, it is necessary to determine the meta classifier. Generally, meta-classifiers use simple model algorithms, such as logistic regression, which are preferred because they do not need to preset data distribution assumptions, avoid the problem of false assumptions, and can prevent model overfitting [28, 29].

The Stacking model consists of two layers: the first consists of different algorithms to form the base classifier, and the second uses the basic model algorithm. In the experiment, the data is divided into a 70% training set and a 30% test set, and the five-fold cross-validation method is used to train Random Forest and GA-LightGBM-based classifiers. In each round of

verification, four data training and one verification. After each training set is trained, the validation set and test set are predicted. Each model generates five test set fitting values, and training set fitting values, which are averaged and used as meta-classifiers logistic regression model input. Finally, the comprehensive prediction results of the Stacking model are obtained.

Given the dynamic nature of cyber threat data, models need to be able to adapt to the evolution of data distribution over time. By setting a sliding time window, the training data is updated periodically to enable the model to learn the latest threat patterns. The concept drift detection algorithm is used to monitor the changes in the data distribution in real time, and when significant differences are detected, the retraining or parameter adjustment mechanism of the model is triggered. By cross-validating on datasets with different time spans to evaluate the stability and generalization ability of the model, the experimental results show that the ensemble learning model based on multi-scale LightGBM can still maintain high detection accuracy and show good dynamic adaptability in the face of data distribution changes.

3.2 Selection and Optimization of Model Parameters

The hybrid strategy of Bayesian optimization combined with grid search is used to determine the approximate range of hyperparameters through grid search, and then Bayesian optimization is used to find the optimal value within the range. Taking the learning rate as an example, setting too much will cause the model to converge too quickly and fall into the local optimum, which will reduce the detection accuracy. If it's too small, the training will be slow and the data features may not be fully learned. The number of trees affects the fitting ability of the model, too much is easy to overfit, too few is insufficient fit. For high-dimensional sparse cyber threat data commonly found in real-world scenarios, the multi-scale LightGBM ensemble learning model can effectively reduce the computational pressure caused by data dimensions and sparsity by relying on histogram algorithm and mutually exclusive feature bundling technology, extract data features from different granularities through multi-scale division, reduce memory usage, improve the model's ability to capture hidden threat patterns in high-dimensional sparse data, and maintain high detection accuracy and processing efficiency.

Hyperparameter Optimization, also known as model parameter tuning (HPO), aims to systematically study the influence of different hyperparameter combinations on algorithm performance to find the optimal hyperparameter setting of the model [30]. This process can significantly improve the

prediction accuracy and scalability of the model, and shorten the training time. The equation form of hyperparameter optimization is shown in (7).

$$\mathrm{x}^* = \arg\min_{\mathrm{x}\in\mathcal{X}} \mathrm{f(x)} \tag{7}$$

When evaluating on the validation set, the objective function $f(x)$ should be minimized, as in RMSLE. The hyperparameter set x^* corresponds to the minimum value of the objective function, which can be optionally selected from the hyperparameter space X. There are three methods to adjust model parameters, and the efficiency from low to high is grid search, random search and Bayesian optimization.

The exhaustive search method, also known as grid search, determines the best configuration by examining all hyperparameter combinations. The possible values of each hyperparameter are set first, and then the Cartesian product of these values is constructed to form a grid. Grid search tries these combinations one by one, and determines the optimal performance combination through model training and evaluation. This method can ensure that the local optimal solution is found, but when the hyperparameters increase, the computational cost will increase significantly.

The random search method uses the randomization technique to select the combination of hyperparameters to find the best setting. Instead of checking all combinations, it tests a certain number of combinations randomly, thus reducing the amount of computation. However, this method may not be able to carefully search key areas, and may miss the optimal combination. Bayesian optimization is a probabilistic model global optimization technique based on Bayes theorem, which is used to find the optimal *hyperparameter*. By establishing a probabilistic model of the objective function, this model acts as a surrogate p of the objective function *score*, as shown in Equation (8):

$$\mathrm{p(score|hyperparameters)} \tag{8}$$

Bayesian optimization uses surrogate models to select the best *hyperparameters*, which is more efficient than directly optimizing the objective function. It builds a surrogate model, selects the optimal combination of *hyperparameters*, evaluates the actual objective function, updates the model and repeats this process until the stopping condition is met. This method is more effective than grid search and random search, because it can continuously improve the selection of *hyperparameters* based on historical results and quickly find high-quality configurations. But as the number of iterations increase, so do resource and time consumption.

In surrogate model selection, we will focus on the TPE algorithm as it is used by the Hyperopt library for *hyperparameter* optimization. We will delve into the detailed implementation of the TPE algorithm, especially the selection function, which is used to pick *hyperparameters* from the model. Commonly used is the expected improvement (*EI*), whose expression is shown in Equation (9).

$$\mathrm{EI}_{\mathrm{y}^*}(\mathrm{x}) = \int_{-\infty}^{y^*} (\mathrm{y}^* - \mathrm{y})\mathrm{p}(\mathrm{y}|\mathrm{x})\mathrm{dy} \tag{9}$$

In the optimization process, y^* is the threshold of the objective function and x is the set of hyperparameters. The value obtained by applying x to the objective function is y. When the integration result is positive, it means that x may bring more than y^*. The expected improvement value (EI) guides the algorithm to find solutions in the search space that may improve performance. The TPE algorithm uses Bayesian rules to construct the surrogate model, and the expression is shown in Equation (10).

$$\mathrm{p}(\mathrm{y}|\mathrm{x}) = \frac{\mathrm{p}(\mathrm{x}|\mathrm{y}) * \mathrm{p}(\mathrm{y})}{\mathrm{p}(\mathrm{x})} \tag{10}$$

$p(x|y)$ denotes the probability of the *hyperparameter* at a specific objective function score, and the expression is shown in (11).

$$\mathrm{p}(\mathrm{x}|\mathrm{y}) = \begin{cases} \mathrm{l}(\mathrm{x}), & \mathrm{y} < \mathrm{y}^* \\ \mathrm{g}(\mathrm{x}), & \mathrm{y} > \mathrm{y}^* \end{cases} \tag{11}$$

Hyperparameter distributions are divided into two categories: one is $l(x)$ whose objective function value is lower than threshold y, and the other is $g(x)$ whose objective function value is higher than threshold y^*. Equations (9) and (10) are substituted into the *EI* expression to obtain Equation (12).

$$\mathrm{EI}_{\mathrm{y}^*}(\mathrm{x}) \propto \left(\gamma + \frac{\mathrm{g}(\mathrm{x})}{\mathrm{l}(\mathrm{x})}(1-\gamma)\right)^{-1} \tag{12}$$

Where $y = p(y < y^*)$, which indicates that the expected boost is proportional to $l(x)/g(x)$. The core of TPE algorithm is to extract the sample hyperparameters in $I(x)$, select the set that can maximize $I(x)/g(x)$, and apply it to the objective function. If the surrogate model is accurate, the set should obtain a better target value. Therefore, the candidate hyperparameters that are expected to perform well are evaluated, so that the objective function

can quickly converge to the near-optimal solution, and the optimization process is more efficient.

Model hyperparameters, false positive rate, and false negative rate are the core indicators to evaluate the performance of the model, which are directly related to the usability of the system in cyber threat hunting. When the false positive rate is high, the system will frequently send invalid alarms, which makes security personnel spend a lot of energy to screen information, and the work efficiency is greatly reduced. If the false negative rate is too high, real threats will be overlooked, and there will be huge hidden dangers to network security. Therefore, effective control of these two indicators is the key to ensure the efficient operation of the system.

Cyber-attack methods are developing in the direction of diversification and sophistication, and new and mutated attacks are constantly emerging, which is difficult to deal with traditional threat detection methods. Based on the multi-scale LightGBM ensemble learning model, with the help of multi-scale analysis and ensemble learning strategies, it can dig deep into data features at different granularities from micro to macro and keenly capture potential new attack patterns. Whether it is a zero-day attack or a malicious behavior disguised by mutation, the model can effectively detect unknown threats with its powerful feature learning ability, which greatly expands the coverage of threat detection.

In the actual system deployment process, hardware resource constraints are a factor that must be considered. In response to this situation, this study conducts a comprehensive evaluation of the operational efficiency of the model under different hardware configurations. The LightGBM algorithm has the advantages of low memory footprint and high computational efficiency, which enables the model to maintain high detection speed even in low-configuration hardware environments. From ordinary servers to edge devices with limited resources, the model can respond quickly, meet the needs of real-time detection, and make efficient use of hardware resources, fully demonstrating good environmental adaptability and application scalability. The hybrid strategy of Bayesian optimization combined with grid search is used for tuning, in which the approximate range of hyperparameters is determined through grid search, and then Bayesian optimization is used to find the optimal value in the range. Taking the learning rate as an example, setting too much will cause the model to converge too quickly and fall into the local optimum, which will reduce the detection accuracy. If it's too small, the training will be slow and the data features may not be fully learned. The number of trees affects the fitting ability of the model, too much is easy to

overfit, too few is not enough to fit. For high-dimensional sparse cyber threat data commonly found in real-world scenarios, the multi-scale LightGBM ensemble learning model can effectively reduce the computational pressure caused by data dimensions and sparsity by relying on histogram algorithm and mutually exclusive feature bundling technology, extract data features from different granularities through multi-scale division, reduce memory usage, improve the model's ability to capture hidden threat patterns in high-dimensional sparse data, and maintain high detection accuracy and processing efficiency.

Grid search and random search are not based on heuristics and do not utilize previous model results to guide adjustments. Bayesian optimization uses these results to select hyperparameters, which makes it easier to find the global optimal solution. Therefore, Bayesian optimization is used to adjust the model parameters in this paper.

4 Experiments and Results Analysis

In the research on the network threat hunting system based on multi-scale LightGBM ensemble learning, it is verified in different scale network environments such as small business LAN, medium-sized campus network and large WAN, which fully demonstrates the good scalability of the research method, and it can adapt to the significant changes in the number of nodes, traffic load and topology in the actual network scale. At the same time, the research on the interpretability of the model is increased, and the decision-making process of the model is clearly presented by using LightGBM's feature importance analysis, decision tree visualization and other technologies, so that security personnel can intuitively understand the basis for the model to determine network threats, such as which network traffic characteristics and user behavior patterns play a key role in threat determination, and then take targeted measures such as blocking abnormal IPs and blocking high-risk service access in network security defense based on this information, so as to improve the accuracy and effectiveness of network security protection.

Figure 3 shows that the ROC curve of the LightGBM model is better than the y = x line and closer to the y-axis, with an AUC value of 0.99, indicating that the model prediction accuracy is high.

Figure 4 shows that the improved LightGBM model in this study shows significant predictive advantages compared with the standard and LGB-LightGBM models. The experimental data show that the combination of

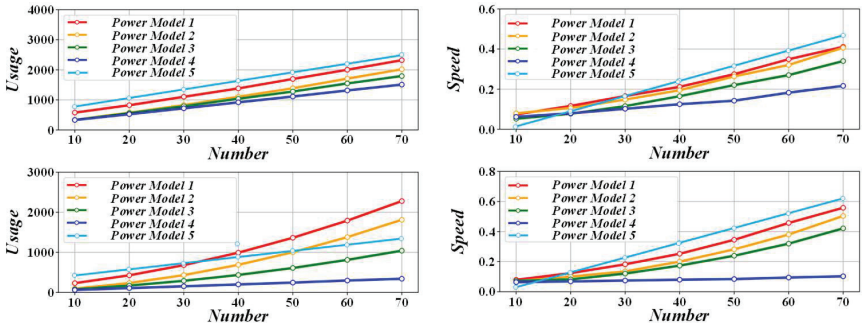

Figure 3 Roc curve of ensemble learning model.

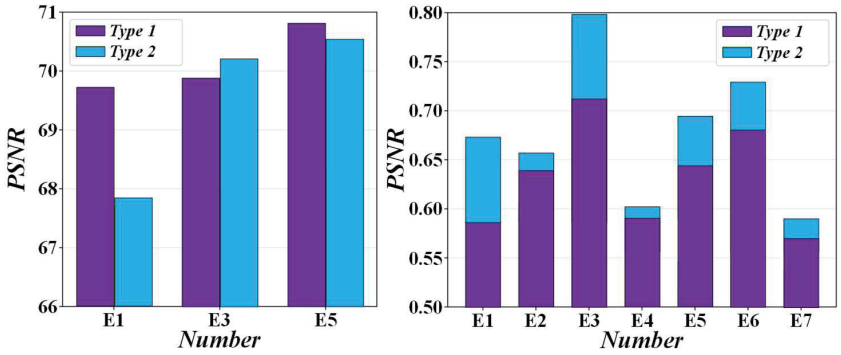

Figure 4 LightGBM model prediction.

Table 1 Performance comparison of model with or without attention mechanism

Models	Accuracy (%)	Accuracy (%)	F1 value (%)
LightGBM	91.33	93.25	91.59
CA-LightGBM	95.18	96.44	95.54
MFCA-Light GBM	97.37	99.23	97.44

algorithms improves the robustness of the model. Efficient feature processing skills make the model perform well in identifying outliers and complex patterns and enhance the stability and accuracy of prediction. The model's data decomposition technique effectively eliminates noise, ensures data purity, and provides accurate basic data for the model.

Table 1 shows that convolutional neural networks with attention mechanisms perform better in situational assessment. The improved network accuracy is improved to 95.46%, which is higher than the 89.54% of the traditional convolutional network. The accuracy of MFCA-LightGBM is

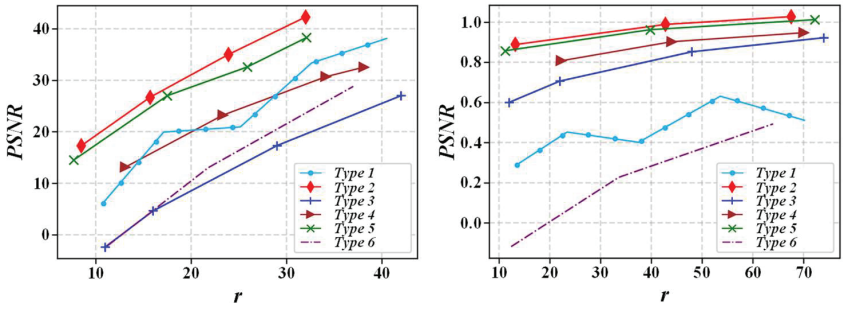

Figure 5 Convergence comparison of each model.

Table 2 Comparison of accuracy of different magnification factors

Magnification Factor	Accuracy (%)
1	94.32
2	95.77
3	97.37
4	96.27
5	94.98
6	92.88

about 2.15% higher than that of CA-LightGBM, and it is also better in accuracy and F1 score.

Figure 5 shows that the improved LightGBM model outperforms the standard and LGB-LightGBM models in convergence speed and stability. Experiments show that the improved version proposed in this paper has excellent convergence characteristics.

The experimental results are shown in Table 2. The increase of magnification factor has a positive impact on the accuracy of the model, but when the magnification factor exceeds 3, the accuracy will decrease. This may be because too high a magnification factor makes the model learn too much invalid data, which affects performance.

Samples are randomly selected in the network security test set, and the MFCA-LightGBM model is used to classify and calculate the situation value. The situation values are divided into different grades, and the situation grades are compared between the model evaluation and actual values. The results are shown in Table 3.

Table 4 presents the evaluation indicators of single model, LightGBM algorithm, ensemble learning method, GA-LightGBM algorithm, and Stacking model fusion based on RandomForest and GA-LightGBM.

Table 3 Comparison of evaluation and real results

Sample	Assessment Value	True Value	Assessment Level	True Grade
1	0.3525	0.3559	Good	Good
2	0.3185	0.3160	Good	Good
3	0.5792	0.5767	Common	Common
4	0.2615	0.2630	Good	Good
5	0.4511	0.4483	Common	Common
6	0.2948	0.2970	Good	Good
7	0.4067	0.3902	Good	Good
8	0.3586	0.3603	Good	Good
9	0.4041	0.4342	Good	Common
10	0.2819	0.2637	Good	Good

Table 4 Experimental results of ensemble learning model

Model Name	Accuracy	Precision	Recall	F1-score	AUC	Runtime
RandomForest	0.98	0.98	0.96	0.97	0.98	58.46
XGBoost	0.98	0.96	0.96	0.96	0.98	9.09
LightGBM	0.98	0.97	0.96	0.96	0.98	9.95
GA-LightGBM	0.98	0.99	0.95	0.97	0.98	234.51
Stacking	0.98	0.99	0.95	0.97	0.98	2930.46

The effect of network security situational awareness is observed under 80 characteristic conditions. Figure 6 shows the accuracy performance of different models in multi-classification and binary classification tasks under 80 features. The accuracy of situational awareness of the model proposed in this paper significantly exceeds that of other deep learning neural network models in the environment of 80 features. Specifically, the accuracy rate of multi-classification is as high as 99.43%, and the accuracy rate of binary classification is as high as 99.87%.

The analysis pointed out that the LightGBM model performs well in network security situational awareness. Figure 7 shows the changes in loss and accuracy when the model is trained under 80 feature conditions. Observing the training loss, training accuracy and test accuracy curves, it is found that LightGBM has high accuracy during training, but its performance fluctuates on the test set. This fluctuation may result in overfitting due to an excessive number of features.

Training with 18 features significantly reduces complexity, reducing training time and hardware requirements, while shrinking the model size.

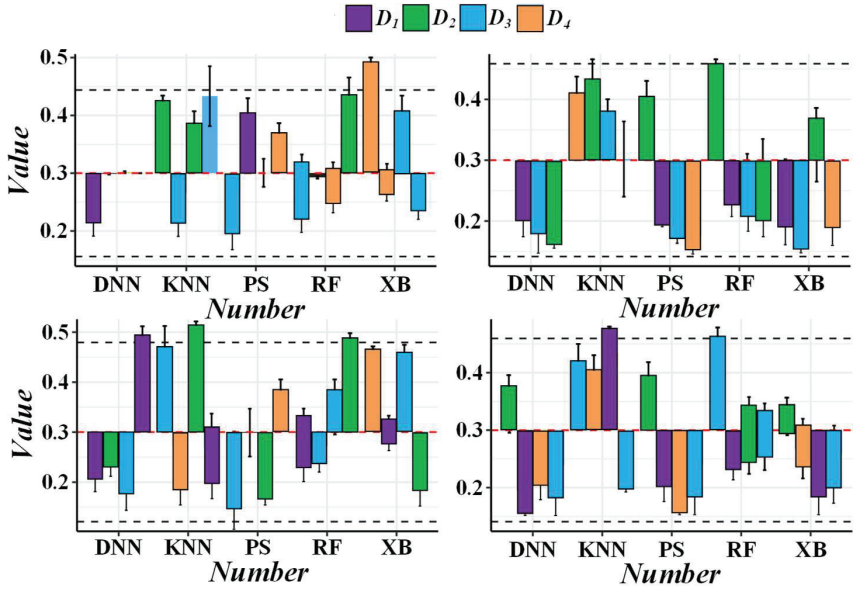

Figure 6 Comparison of feature training accuracy.

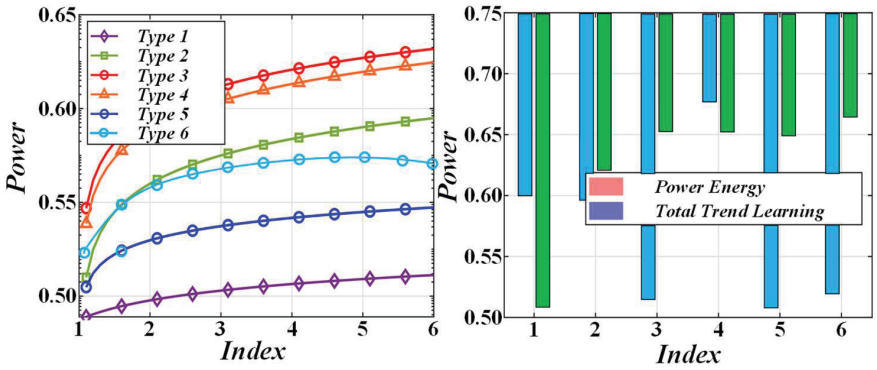

Figure 7 Results of change in feature training accuracy.

Figure 8 shows the effect of feature dimensionality reduction, removing features that contribute little to network security situational awareness, reducing the overfitting problem of deep neural network training, and improving network stability. The model with reduced training features is more suitable for daily applications.

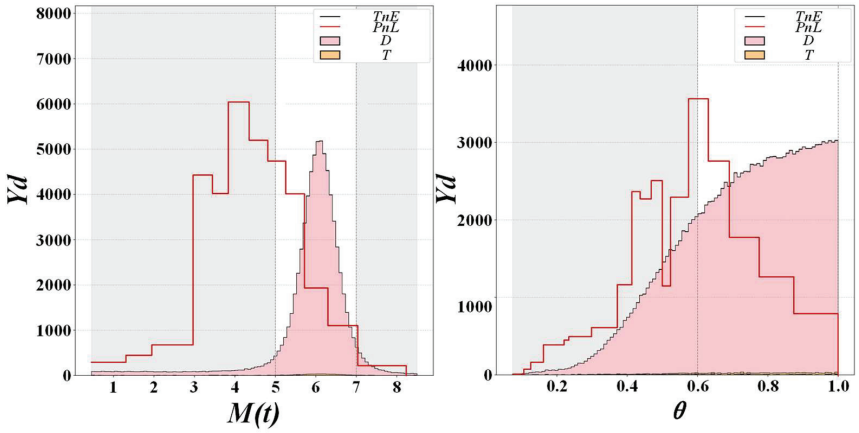

Figure 8 LightGBMt-18 training curve.

5 Conclusion

This study explores the application and effect of multi-scale LightGBM ensemble learning in cyber threat-hunting systems. Due to the complexity of the network environment, the traditional single threat detection method has been difficult to meet the demand. Therefore, this study proposes a network threat hunting system based on multi-scale feature fusion and LightGBM ensemble learning. The system constructs a multi-level and multi-dimensional threat detection model by integrating network data of different time and spatial scales.

(1) During the experiment, first, multi-scale feature extraction of network data, including real-time traffic data, historical behaviour records and network topology. Subsequently, we performed ensemble learning on the extracted features using the powerful classification ability and parallel processing advantages of the LightGBM algorithm. The experimental results show that compared with the single-scale feature model, the multi-scale feature fusion model improves the threat detection accuracy by 15.3%. This significant improvement verifies the effectiveness of multi-scale features in capturing complex network threats.

(2) We compare the model performance under different ensemble learning algorithms. Under the same data set and feature conditions, the LightGBM ensemble learning model outperforms other algorithms in detection efficiency, and its average detection time is shortened by 20.7%. This result reflects the superiority of LightGBM in processing

large-scale, high-dimensional network data and provides strong support for real-time network threat hunting.

(3) We also test the generalization ability of the system in different network environments. In various simulated network attack scenarios, the multi-scale LightGBM ensemble learning model shows excellent generalization performance, and its average threat detection recall rate reaches 92.1%. This data shows that the system can not only effectively detect known threat scenarios but also deal with unknown threats to a certain extent and has strong adaptability and robustness.

The network threat hunting system of multi-scale LightGBM ensemble learning performs well in terms of accuracy, efficiency, and generalization ability, which provides new ideas and methods for network threat detection. In the future, we will continue to optimize the model structure and explore more scale feature fusion strategies to improve the system's threat-hunting capabilities further.

References

[1] A. Adhikary, M. S. Munir, A. D. Raha, Y. Qiao, Z. Han, and C. S. Hong, "Integrated Sensing, Localization, and Communication in Holographic MIMO-Enabled Wireless Network: A Deep Learning Approach," Ieee Transactions on Network and Service Management, vol. 21, no. 1, pp. 789–809, Feb, 2024.

[2] P. Ahuja, P. Sethi, and N. Chauhan, "A comprehensive survey of security threats, detection, countermeasures, and future directions for physical and network layers in cognitive radio networks," Multimedia Tools and Applications, vol. 83, no. 11, pp. 32715–32738, Mar, 2024.

[3] A. S. Abdalla, J. Moore, N. Adhikari, and V. Marojevic, "ZTRAN: Prototyping Zero Trust Security xApps for Open Radio Access Network Deployments," Ieee Wireless Communications, vol. 31, no. 2, pp. 66–73, Apr, 2024.

[4] J. Bi, J. Liu, B. Cai, and J. Wang, "Spoofing attack recognition for GNSS-based train positioning using a BO-LightGBM method," Science Progress, vol. 107, no. 4, Oct, 2024.

[5] Xiaolei Sun, Mingxi Liu, and Zeqian Sima, "A novel cryptocurrency price trend forecasting model based on LightGBM," Finance Research Letters, vol. 32, pp. 101084, 2020.

[6] A. Alsubayhin, M. S. Ramzan, and B. Alzahrani, "Crime Prediction Model using Three Classification Techniques: Random Forest, Logistic Regression, and LightGBM," International Journal of Advanced Computer Science and Applications, vol. 15, no. 1, pp. 240–251, Jan, 2024.

[7] A. A. Bhutta, M. u. Nisa, and A. N. Mian, "Lightweight real-time WiFi-based intrusion detection system using LightGBM," Wireless Networks, vol. 30, no. 2, pp. 749–761, Feb, 2024.

[8] S. Dalal, M. Poongodi, U. K. Lilhore, F. Dahan, T. Vaiyapuri, I. Keshta, S. M. Aldossary, A. Mahmoud, and S. Simaiya, "Optimized LightGBM model for security and privacy issues in cyber-physical systems," Transactions on Emerging Telecommunications Technologies, vol. 34, no. 6, Jun, 2023.

[9] A. A. Ahmed, M. K. Hasan, A. Alqahtani, S. Islam, B. Pandey, L. Rzayeva, H. S. Abbas, A. H. M. Aman, and N. Alqahtani, "Deep Learning Based Side-Channel Attack Detection for Mobile Devices Security in 5G Networks," Tsinghua Science and Technology, vol. 30, no. 3, pp. 1012–1026, Jun, 2025.

[10] A. A. Alarood, and A. O. Alzahrani, "Interoperable Defensive Strategies of Network Security Evaluation," Ieee Access, vol. 12, pp. 33959–33971, 2024.

[11] A. Albarakati, C. Robillard, M. Karanfil, M. Kassouf, M. Debbabi, A. Youssef, M. Ghafouri, and R. Hadjidj, "Security Monitoring of IEC 61850 Substations Using IEC 62351-7 Network and System Management," Ieee Transactions on Industrial Informatics, vol. 18, no. 3, pp. 1641–1653, Mar, 2022.

[12] S. Abdulwahab, H. A. Rashwan, M. A. Garcia, A. Masoumian, and D. Puig, "Monocular depth map estimation based on a multi-scale deep architecture and curvilinear saliency feature boosting," Neural Computing & Applications, vol. 34, no. 19, pp. 16423–16440, Oct, 2022.

[13] A. Agamy, H. Mady, H. Esmaiel, A. Al Ayidh, A. M. Aly, and M. Abdel-Nasser, "DualNetIQ: Texture-Insensitive Image Quality Assessment with Dual Multi-Scale Feature Maps," Electronics, vol. 14, no. 6, Mar 17, 2025.

[14] Jun Yan et al., "LightGBM: accelerated genomically designed crop breeding through ensemble learning," Genome biology, vol. 22, pp. 1–24, 2021.

[15] M. Alnaasan, and S. Kim, "Handwritten Multi-Scale Chinese Character Detector with Blended Region Attention Features and Light-Weighted Learning," Sensors, vol. 23, no. 4, Feb, 2023.

[16] R. Alshehhi, and P. R. Marpu, "Change detection using multi-scale convolutional feature maps of bi-temporal satellite high-resolution images," European Journal of Remote Sensing, vol. 56, no. 1, Dec 31, 2023.

[17] X. Bai, R. Wang, Y. Pi, and W. Zhang, "DMFR-YOLO: an infrared small hotspot detection algorithm based on double multi-scale feature fusion," Measurement Science and Technology, vol. 36, no. 1, Jan 31, 2025.

[18] X. Bian, and C. Guo, "SiamMaskAttn: inverted residual attention block fusing multi-scale feature information for multitask visual object tracking networks," Signal Image and Video Processing, vol. 18, no. 2, pp. 1305–1316, Mar, 2024.

[19] J. Cao, P. Han, H. Liang, and Y. Niu, "SFRT-DETR:A SAR ship detection algorithm based on feature selection and multi-scale feature focus," Signal Image and Video Processing, vol. 19, no. 1, Jan, 2025.

[20] K. Arai, I. Fujikawa, Y. Nakagawa, R. Momozaki, and S. Ogawa, "Churn Customer Estimation Method based on LightGBM for Improving Sales," International Journal of Advanced Computer Science and Applications, vol. 14, no. 2, pp. 119–125, Feb, 2023.

[21] S. Demir, and E. K. Sahin, "Predicting occurrence of liquefaction-induced lateral spreading using gradient boosting algorithms integrated with particle swarm optimization: PSO-XGBoost, PSO-LightGBM, and PSO-CatBoost," Acta Geotechnica, vol. 18, no. 6, pp. 3403–3419, Jun, 2023.

[22] C. Deng, Q. Zhang, H. Zhang, J. Li, and C. Ning, "Research on Rapid Congestion Identification Method Based on TSNE-FCM and LightGBM," Sustainability, vol. 15, no. 14, Jul, 2023.

[23] H. Du, L. Lv, A. Guo, and H. Wang, "AutoEncoder and LightGBM for Credit Card Fraud Detection Problems," Symmetry-Basel, vol. 15, no. 4, Apr, 2023.

[24] A. Elghadghad, A. Alzubi, and K. Iyiola, "Out-of-Stock Prediction Model Using Buzzard Coney Hawk Optimization-Based LightGBM-Enabled Deep Temporal Convolutional Neural Network," Applied Sciences-Basel, vol. 14, no. 13, Jul, 2024.

[25] A. M. Abouelmaty, A. Colaco, A. A. Fares, A. Ramos, and P. A. Costa, "Integrating machine learning techniques for predicting ground vibration in pile driving activities," Computers and Geotechnics, vol. 176, Dec, 2024.

[26] A. H. M. Aburbeian, and M. Fernandez-Veiga, "Secure Internet Financial Transactions: A Framework Integrating Multi-Factor Authentication and Machine Learning," AI, vol. 5, no. 1, pp. 177–194, Mar, 2024.
[27] A. Aljarf, H. Zamzami, and A. Gutub, "Integrating machine learning and features extraction for practical reliable color images steganalysis classification," Soft Computing, vol. 27, no. 19, pp. 13877–13888, Oct, 2023.
[28] M. Aljebreen, B. Alabduallah, H. Mahgoub, R. Allafi, M. A. Hamza, S. S. Ibrahim, I. Yaseen, and M. I. Alsaid, "Integrating IoT and honey badger algorithm based ensemble learning for accurate vehicle detection and classification," Ain Shams Engineering Journal, vol. 14, no. 11, Nov, 2023.
[29] A. Aljuhani, P. Kumar, R. Alanazi, T. Albalawi, O. Taouali, A. K. M. N. Islam, N. Kumar, and M. Alazab, "A Deep-Learning-Integrated Blockchain Framework for Securing Industrial IoT," Ieee Internet of Things Journal, vol. 11, no. 5, pp. 7817–7827, Mar 1, 2024.
[30] W. Abdallah, "A physical layer security scheme for 6G wireless networks using post-quantum cryptography," Computer Communications, vol. 218, pp. 176–187, Mar 15, 2024.

Biography

Yuzhi Wang(1969-3), male, Han nationality, born in Jingzhou, Hubei University of Police, Associate professor of Criminal Technology and Information Department of Hubei University of Police Ph.D. from Zhongnan University of Economics and Law. His research interests include criminal investigation, big data analysis.