ingegneria sismica
International Journal of Earthquake Engineering
Anno XLIII - Num. 3 - 2026

# A Machine Learning-Based Anomaly Detection in Digital Power Grid Network Traffic

*Peng Xiao*[1], *Zijie Deng*[2,*] *and Biao Bai*[1]

[1] Information Center of China Southern Power Grid Yunnan Power Grid Co., Ltd., Yunan, China.
[2] China Southern Power Grid Power Grid Group, Co., Ltd., Guangdong Province, China.

**SUMMARY:** *With the in-depth construction of digital power grids, their cyber-physical systems face severe cybersecurity challenges. Malicious encrypted traffic, using HTTPS and SSL/TLS, threatens grid stability, making traditional detection ineffective. This paper focuses on such traffic from mainstream hacking tools, combines network traffic analysis with machine learning, extracts protocol layer features, and constructs decision tree, random forest, and LSTM models. Experiments show their accuracy rates reach 99.85%, 99.93%, and 99.64% respectively, enabling intelligent and accurate detection, providing technical support for grid security.*

***KEYWORDS:*** *Digital Power Grid, Machine Learning, Anomaly Detection, LSTM*

# 1 Introduction

As the core infrastructure supporting the new power system, the security protection system of the digital power grid directly determines the stable operation of the energy internet. In the initial informatization stage of the power system, the communication traffic of power grid monitoring systems and dispatching automation systems was mainly in plaintext. The traditional Deep Packet Inspection (DPI) technology, by parsing the feature codes in data packet payloads, could effectively identify early malicious attacks targeting telecontrol protocols and distribution network terminals, thus building a basic security defense line for key scenarios such as substation automation and centralized control centers.

With the advancement of the "dual-carbon" goals, the digital power grid has entered a stage of comprehensive upgrading. On one hand, the number of ubiquitous power internet of things terminals (such as smart meters and distribution terminals) has exceeded 100 million. The deep integration of power monitoring systems and information networks requires communication links to have millisecond-level real-time performance and high reliability. On the other hand, the transmission of sensitive data in scenarios such as cross-regional power grid dispatching and new energy grid connection has put forward rigid demands for communication encryption. The SSL/TLS protocol has become the standard configuration for power dispatching data networks and new energy power station cloud platforms, while HTTPS encryption is widely used in services such as power marketing and customer data interaction. This trend of widespread encryption not only ensures the security of legitimate communications but also provides opportunities for cyber attacks. Malicious behaviors by attackers targeting the digital power grid, such as forging instructions for the SCADA

*zijiedeng2025@163.com
https://doi.org/10.65102/is20261182 Published: 10 June , 2026

protocol and tampering with distribution terminal firmware, are all transmitted covertly through encrypted traffic.

Traditional DPI technology has gradually shown limitations in the digital power grid environment. Firstly, it cannot parse encrypted GOOSE messages and MMS protocol data, making it difficult to identify malicious traffic disguised as normal control instructions. Secondly, the strict requirements of the power system for communication delay (for example, the delay of relay protection signals must be less than 20ms) mean that the in-depth parsing process of DPI may cause real-time conflicts. Thirdly, there are a large number of custom private protocols in the digital power grid, and the feature library of DPI is difficult to fully cover them, leaving protection blind spots for targeted attacks.

In this context, machine learning technology has become the core direction for detecting malicious encrypted traffic in the digital power grid due to its ability to mine features of encrypted traffic. This method needs to divide units according to the unique traffic attributes of the digital power grid (such as the periodic message characteristics of the SCADA protocol and the session duration distribution of distribution network terminals), extract multi-dimensional features such as traffic packet length sequences, session interaction frequency, and protocol status codes, and achieve accurate identification through the training of classification models. Classic models like decision trees—tree-structured models derived from statistics and computer science, with origins in 1960s research—excel here by recursively partitioning data based on extracted features, making them effective for classifying traffic types using attributes like protocol status codes. More advanced random forests, ensemble models built on multiple decision trees that emerged in the 1990s, enhance detection robustness through random sampling of data and features, reducing overfitting when handling diverse traffic patterns such as session duration distributions in distribution network terminals. For capturing temporal dynamics, LSTMs (Long Short-Term Memory), recurrent neural network variants developed to handle long-term sequence dependencies, prove invaluable: their gated cell structures adeptly process traffic packet length sequences and periodic SCADA message characteristics, critical for identifying stealthy malicious patterns in time-series traffic data. This technical path, leveraging such models, not only breaks through the technical barriers of encryption algorithms, but can also adapt to the dynamic protection needs of the integrated "physical system - information system" scenario of the digital power grid, providing key support for building an active defense system.

The main contributions of the study are as follows:

1. A multi-module collaborative encrypted malicious traffic identification system has been designed and implemented, covering visualization, data storage, model detection, and data processing modules. Each module has a clear division of labor while cooperating with each other, realizing a complete process from traffic data extraction, processing, model training and detection to result display. It supports functions such as real-time detection, user-added models, and report downloading, which enhances the practicality and operability of the system.
2. Three traffic identification models, namely decision tree, random forest, and LSTM, have been proposed and constructed to detect different malicious traffics (trickbot, poshc2, qakbot). After training, their accuracy rates reach 99.85%, 99.92%, and 99.64% respectively. While maintaining a low false alarm rate, they achieve efficient and accurate detection of unknown encrypted malicious traffic, providing an effective technical method for encrypted malicious traffic identification.
3. Combining network traffic analysis and machine learning technologies, the characteristics of various types of encrypted malicious traffics are summarized and analyzed from the network protocol level. Technologies such as traffic session reorganization and statistical feature extraction are used to process the original encrypted traffic. Mutual

information scores are utilized to screen important features, and the SMOTE algorithm is adopted to handle data imbalance issues, which optimizes the model performance and a reference technical path and practical experience for the field of encrypted malicious traffic detection.

# 2 Related Work

As an upgraded form of smart grid, the digital power grid is a complex cyber-physical system integrating energy flow, information flow and business flow, whose safe operation relies on digital-adapted anomaly detection mechanisms. The full-domain interconnection enabled by the ubiquitous power Internet of Things, while improving efficiency, expands the attack surface, facing threats such as false data injection, DoS attacks, energy theft and covert hijacking via encrypted traffic. Existing studies, centering on its architecture, have built detection systems from multiple dimensions like physical characteristics, machine learning-based encrypted traffic analysis, edge computing-enabled terminal perception, and cloud-based big data analysis, adapting to new features and supporting the "cloud-edge-terminal" defense. Sahani et al. conducted a comprehensive review of machine learning-based intrusion detection systems in smart grids, systematically categorizing their application scenarios in transmission and distribution networks, dataset construction methods, mainstream machine learning algorithms, and evaluation metrics. They not only summarized lessons learned in practice but also identified future research directions focusing on real-time performance and adaptability, laying a theoretical foundation for subsequent studies. Building on this, deep learning methods have been widely applied due to their ability to model complex patterns. Quraishi et al. proposed a real-time anomaly detection framework integrating autoencoders, LSTM, GANs, SOMs, and transfer learning.[1] By enhancing adaptability to dynamic grid environments through attention mechanisms and transfer learning, its performance in terms of precision, recall, and execution time outperforms traditional methods. Priyadarsini et al. designed a CNN-based detection scheme specifically for false data injection attacks. Through device data matrix denoising, CNN state estimation, and feedback control loops, it achieved low error rates and excellent network performance metrics, verifying the effectiveness of deep learning in specific attack scenarios[2].

Targeting security vulnerabilities in industrial protocols such as Modbus/TCP and DNP3 smart grids, the MENSA method proposed by Siniosoglou et al.[3] integrates autoencoders GANs to construct a unified deep learning architecture. It can both detect operational anomalies and accurately classify 13 types of Modbus/TCP attacks and 5 types of DNP3 attacks. Validation results in real environments such as laboratories and substations show that its accuracy, true positive rate, false positive rate, and F1-score are superior to other machine learning and deep learning methods, highlighting the value of protocol-specific detection schemes. In distributed scenarios, Jithish et al. proposed a federated learning-based anomaly detection scheme. By enabling local model training on smart meters to avoid centralized data sharing, it verified performance on standard datasets and real meter data while ensuring user privacy, providing a privacy-preserving paradigm for large-scale deployment.

Innovations in network infrastructure and communication technologies have also offered new ideas for anomaly detection. Passerini et al.[4] utilized power line modems (PLMs) as sensors to achieve detection, classification, and localization of network anomalies through reflection measurements (input impedance, reflection coefficients) and end-to-end measurements (channel transfer functions) combined with dedicated algorithms. Simulation results confirmed its applicability in medium and low-voltage distribution networks. Jung et

al.[5] explored the advantages of Software-Defined Networking (SDN), designing an entropy-based anomaly detection method based on the global visibility of SDN controllers, which can effectively identify DoS attacks and port scans, with plans for further validation through a test platform containing ONOS controllers and OpenFlow switches. Addressing the specificity of time-series data, Zhang et al.[6] reviewed time-series anomaly detection technologies in smart grids, categorizing anomalies into point anomalies, pattern anomalies, and contextual anomalies. They discussed challenges such as limited data and non-stationarity, emphasizing the advantages of methods like LSTM and autoencoders in capturing temporal dependencies. Fenza et al.[7] proposed a drift-aware model that uses LSTM to monitor prediction errors, distinguishing between real anomalies and legitimate changes in user behavior (e.g., household structure adjustments), enabling near-real-time detection in dynamic environments.

From the perspective of threat modeling and system resilience, Alkuwari et al.[8] reviewed anomaly detection technologies from a cybersecurity perspective, constructing a threat model encompassing attack vectors such as false data injection and energy theft. They analyzed the application value of data sources like state estimation and PMUs, reviewed the role of statistical methods, supervised/unsupervised learning, and deep learning in detection, and clarified research gaps and future directions. Shahinzadeh et al.[9] focused on the resilience of smart grids against false data injection attacks, evaluating transmission line outages, load curtailment, and voltage stability through simulating attack vectors of varying intensities combined with steady-state AC power flow models. Results showed that the target grid exhibits a certain degree of resilience in terms of outage probability and cascading blackouts, but transient voltage stability is vulnerable to attacks, providing a basis for resilience improvement strategies.

In addition, other innovative methods have enriched the detection system: Anwar et al.[10] transformed anomaly detection into a quadratic assignment problem (QAP), using a progressive assignment algorithm for graph matching, achieving a 100% detection rate on IEEE 24-bus, 30-bus, and 118-bus test systems with low time complexity. Chren[11] proposed a multi-layer reliability analysis method based on Stochastic Reward Nets (SRNs) to model failure behaviors in the physical, software, and communication layers of smart grid services. Validated through an Advanced Metering Infrastructure (AMI) scenario, it can output service success probabilities and failure type probabilities, providing quantitative support for system design. Yang et al.[12] proposed a station-level network abnormal traffic detection method based on deep transfer learning for substation scenarios, extracting frequency-domain features and designing a ResNeSt convolutional neural network model. Using transfer learning to address the issue of insufficient labeled samples, it demonstrated high accuracy in experiments on a 110kV substation.

In summary, existing research covers multiple dimensions from theoretical reviews to technical implementations, from centralized to distributed approaches, and from general detection to scenario-specific solutions, providing rich technical reserves for smart grid anomaly detection. However, there remains room for expansion in areas such as multi-attack collaborative detection, handling extremely imbalanced data, and cross-scenario generalization.

# 3 Materials and Methods

## 3.1 Data preprocessing

In this section, we describe the key components of the methodology system include a multi-module collaborative encrypted malicious traffic identification system (covering visualization, data storage, model detection, and data processing modules), a feature extraction mechanism based on the unique traffic attributes of digital power grids, and the training and detection processes of three types of classification models: decision tree, random forest, and LSTM.

After the system is started, it first reads the given PCAP network packet file and performs traffic splitting through Flowcontainer. Then, it checks whether the traffic uses the TCP/UDP protocol. If yes, it proceeds to the feature acquisition process; if not, it skips this traffic and directly processes the next data packet. After that, it extracts the five-tuple information of each flow, records the start and end timestamps of the flow, and then detects whether there are TLS/SSL encryption features in the flow. If such features exist, it acquires their advanced features, namely SNI, cipher suites, and certificate contents; if not, it stores the basic features in a dictionary. It iterates through all flows until all traffic is processed, thereby generating a raw CSV file containing all flow features.

The collected raw traffic undergoes zero-padding and data cleaning. The corresponding data is then categorically encoded, followed by feature selection. The preprocessed data is stored in the data storage module and input into the model detection module. The flowchart of data extraction is shown in Figure 1.

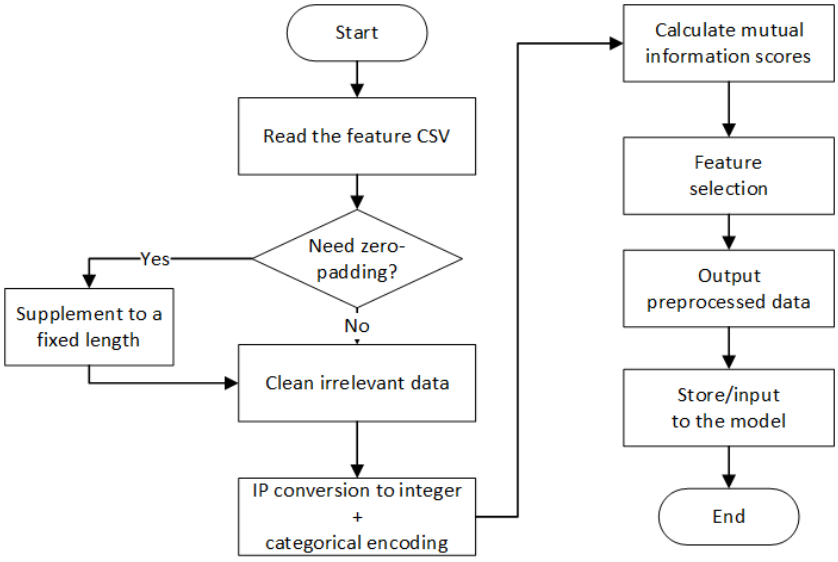


*Figure 1: Data extraction flow chart.*

First, the raw feature CSV file generated after data extraction is read. Then, it checks if the data length is insufficient. If so, it is padded with zeros to a preset length; if not, it proceeds directly to the next step. In the next step, noise data irrelevant to the service is deleted, outliers are handled, string-format IP addresses are converted to integers, and non-numeric features such as protocol types are mapped to integers to complete categorical encoding. After these steps, sklearn.feature_selection is used to calculate mutual information scores, and the features with the highest scores are selected. Finally, the preprocessed data is stored in the database and input into the model detection module for classification.

## 3.2 Model training

To address the issue of detecting malicious encrypted traffic in digital power grids, this study deploys a machine learning - driven anomaly detection framework using three key models. A decision tree, structured with nodes and edges, conditionally splits data for classification or regression tasks. Random Forest, as an ensemble method, trains multiple decision trees on different data subsets and uses voting to generate results, excelling in traffic classification. LSTM, a variant of RNN, resolves the long - term dependency problem through gating mechanisms and is proficient in capturing sequential patterns for traffic analysis. Together, they enable the accurate identification of malicious encrypted traffic to safeguard the operation of digital power grids.

### 3.2.1 Decision tree

A decision tree is structured with nodes and edges, comprising root nodes as the starting point of the dataset, internal nodes indicating judgment conditions on features where each splits based on a certain feature, leaf nodes representing classification results or regression prediction values, and edges connecting nodes to show the path from one decision to the next. Its core principle lies in creating a tree structure through continuous data splitting. In essence, it makes conditional judgments on input features, identifies optimal features for data partitioning, and continues this process until the data meets certain conditions such as high purity. Depending on the type of task, there are two forms of decision trees. Classification trees are used for classification problems with discrete output values. Regression trees are used for regression problems with continuous output values.The process of constructing a decision tree typically adopts the following algorithms:

Iterative Dichotomiser 3 (ID3)

It selects the feature with the maximum information gain for partitioning. The formula for information gain is:

$$\text{Information Gain}(S, A) = \text{Entropy}(S) - \sum_{v \in A} \frac{|S_v|}{|S|} \cdot \text{Entropy}(S_v) \tag{1}$$

C4.5 Algorithm

Based on ID3, it uses information gain ratio instead of information gain to avoid favoring features with more values. The formula for information gain ratio is:

$$\text{Gain Ratio}(S, A) = \frac{\text{Information Gain}(S,A)}{\text{Split Information}(A)} \tag{2}$$

Classification and Regression Tree (CART)

A feature selection method based on Gini coefficient for classification and regression tasks:

$$\text{Gini}(S) = 1 - \sum_{i=1}^{C} p_i{}^2 \tag{3}$$

Decision trees can be applied in various scenarios, mainly including classification tasks such as spam identification, disease diagnosis, and customer classification; regression tasks such as house price prediction and stock price prediction; feature importance analysis, as the splitting process of decision trees can reflect the importance of each feature; and rule extraction, as decision trees can visualize complex decision-making processes and generate easy-to-understand decision rules.

### 3.2.2 Random Forest model

The Random Forest model falls within the realm of ensemble learning. Rooted in the decision tree algorithm, it generates a large number of decision trees and employs a voting mechanism. This algorithm finds extensive application in diverse classification and regression tasks, particularly in the domain of network security for the effective detection of malicious traffic. Comprising multiple decision trees, each tree in the Random Forest is trained on a distinct subset of the dataset. Based on the Bagging (Bootstrap Aggregating) principle, it randomly samples subsets of the dataset for training and introduces randomness during feature selection.

Given the entire feature set X extracted from the traffic and the entire training data Y containing both benign and malicious traffic, after determining the number of decision trees N, the features and training data are sampled with replacement. For each sampling result, a decision tree is constructed.

$$Z = \{(X1, Y1), (X2, Y2), (X3, X3) \ldots\ldots (Xn, Yn)\} \tag{4}$$

For each dataset constructed in (4), construct a decision tree as follows: take the training subset as the root node, and recursively select features from the training subset at each split node in a top-down manner. Calculate the information gain ratio for each feature, select the feature with the highest gain as the splitting attribute for splitting, until all samples reach the leaf nodes. Construct such a decision tree for each training subset in (4).

$$H(S) = -\sum_{i=1}^{m}(P_i \log_2 P_i) \tag{5}$$

The final result of the model is generated by the combined voting of multiple decision trees, where n is the number of decision trees, qi is the weight of a tree, and yi is the voting result of a tree. RF represents the voting result of the random forest.

$$RF = \sum_{i=1}^{n} yi * qi \tag{6}$$

Through the above steps, a random forest classifier can be obtained. This classifier can classify various input encrypted data flows into poshc2 and other types of traffic.

### 3.2.3 Long Short-Term Memory

Long Short-Term Memory (LSTM) is a variant of Recurrent Neural Networks (RNNs) designed to address the issues of gradient vanishing and explosion that traditional RNNs face when handling long-term dependencies. By effectively memorizing long-sequence data, it has been widely applied in fields such as time series prediction, speech recognition, and natural language processing. The basic unit of an LSTM network consists of three gating mechanisms: the Forget Gate determines how much of the state from the previous time step should be retained; the Input Gate controls the extent to which the current input information is updated into the cell state; and the Output Gate regulates the output value of the unit. These gates enable long-term memory and short-term forgetting of information by adjusting the flow of input, output, and state information. An LSTM network is composed of multiple units, each containing three core gates with specific functions: the Forget Gate decides which information in the current unit should be discarded; the Input Gate determines how much of the current input information should be retained; and the Output Gate governs the output of the current unit.The core of LSTM lies in the calculation of these gating mechanisms:

Formula for the forget gate:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{7}$$

Formula for the input gate:

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{8}$$

Formula for the output gate:

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{9}$$

where $h_{t-1}$ is the hidden state at the previous time step, $x_t$ is the input at the current time step, and $W_o$ and $b_o$ are learnable weights and biases, respectively.

At each time step, the LSTM updates its cell state and hidden state:

Updating the cell state:

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t \tag{10}$$

Outputting the hidden state:

$$h_t = o_t \cdot \tanh(C_t) \tag{11}$$

LSTM has been extensively applied in time series prediction, natural language processing, speech recognition, and video analysis. It excels at capturing long-term dependencies in time series data. In the field of network security, for instance, LSTM can be used to detect malicious traffic such as that from the Qakbot family by analyzing historical traffic data to identify patterns in future traffic.

Divide the preprocessed dataset into a training set and a test set according to a certain proportion (80% as the training set and 20% as the test set), and then adjust the format to ensure that the data format meets the input requirements of the selected model. Then, according to the selected model, set the initial parameters of the model. For example, for a decision tree model, it is necessary to set parameters such as the maximum depth of the tree and the minimum number of samples for splitting. Input the training set data into the model for multiple iterative training. Finally, use the test set data to evaluate the trained model, calculate the evaluation indicators of the model, such as accuracy, recall, F1-score and precision, and then adjust and optimize the model according to the evaluation results. The training process is shown in Figure 2.

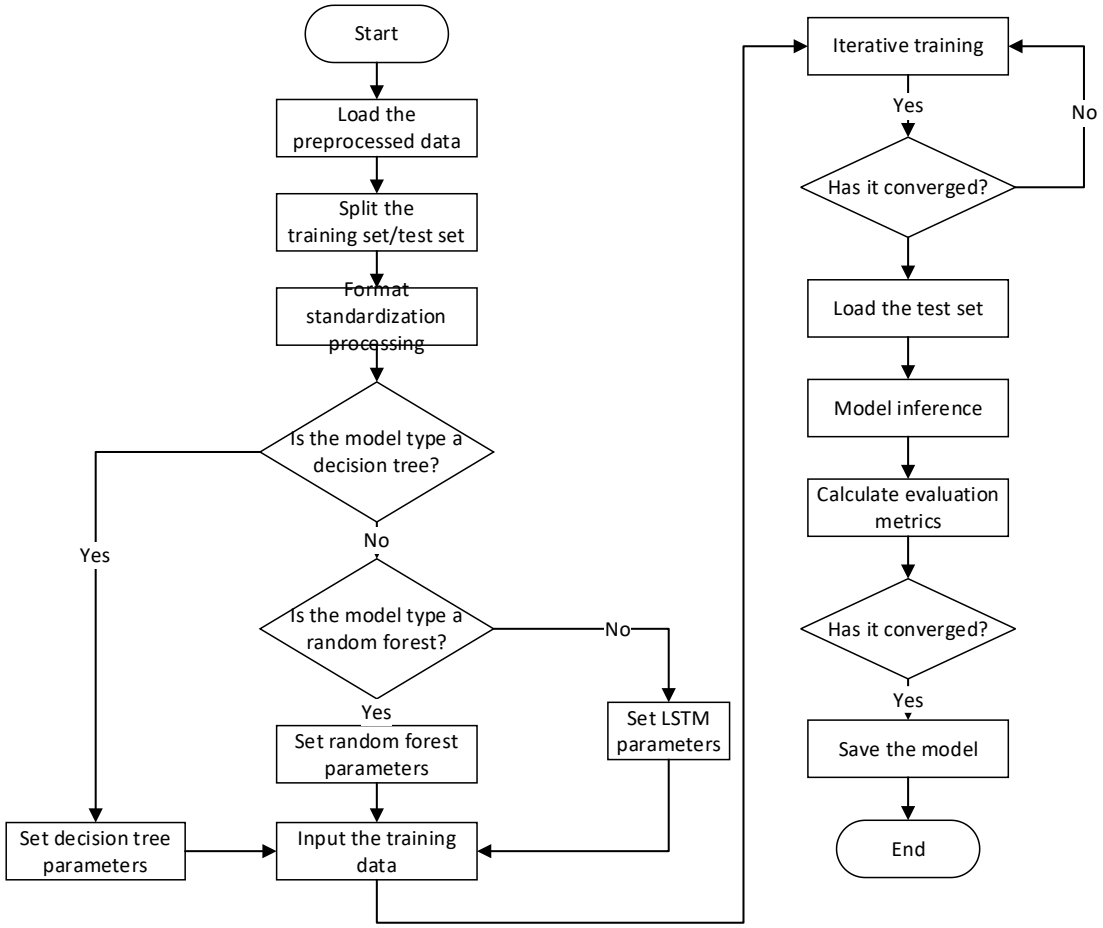


*Figure 2: Model training flow chart*

First, split the input preprocessed standardized data in proportion: 80% training set + 20% test set, then perform format conversion, that is, convert it to Tensor (for LSTM) or DataFrame (for tree models) according to the model type, and then set the parameters. During iterative training, EarlyStopping is used to prevent overfitting, and then the test set data is used to evaluate the trained model, calculating the model's evaluation indicators such as accuracy, recall, F1 - score and precision. The calculation methods of the evaluation indicators are as follows:

Accuracy:

The proportion of correctly predicted samples to the total number of samples.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{12}$$

The numerator is the total number of samples correctly predicted by the model (including true positives (TP) and true negatives (TN)), and the denominator is the total number of samples in the dataset (the sum of correctly predicted and incorrectly predicted samples, where incorrectly predicted samples include false positives (FP) and false negatives (FN)).

Precision:

The proportion of samples predicted as positive that are actually positive.

$$Precision = \frac{TP}{TP+FP} \tag{13}$$

The numerator is the number of samples correctly predicted as positive (TP), and the denominator is the total number of samples predicted as positive by the model (including correctly predicted TP and incorrectly predicted FP).

Recall:

The proportion of actually positive samples that are correctly predicted as positive.

$$Recall = \frac{TP}{TP+FN} \tag{14}$$

The numerator is the number of samples correctly predicted as positive (TP), and the denominator is the total number of actually positive samples in the dataset (including correctly predicted TP and incorrectly predicted negative FN).

F1-Score:

The harmonic mean of precision and recall, which integrates the performance of both to avoid the one-sidedness of a single indicator.

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{15}$$

It combines precision and recall into a comprehensive indicator through harmonic mean. Its value ranges from 0 to 1, and the closer it is to 1, the better the balanced performance of the model in terms of precision and recall. Compared with the arithmetic mean, the harmonic mean focuses more on indicators with smaller values, thus effectively reflecting the overall advantages and disadvantages of the two indicators.

## 3.3 Model detection

The model detection module receives standardized data from the preprocessing module, then loads the model to perform the same standardization operations on real-time data as during training, processes the dataset for result labeling, and finally stores the detection results as a CSV file in the database. The model detection process is described as follows:

Receive the standardized data sent from the preprocessing module and perform data verification. Check whether the data contains necessary fields, ensure that numeric fields do not contain illegal characters, and confirm that categorical fields have been encoded into integers. If any field is missing or formatted incorrectly, immediately throw a ValueError and stop the process.After completing these steps, load the model, which will automatically record the model version number and the timestamp during training. Then, perform the same standardization operation on the real-time data as during training, restore integers to their original labels, and convert numeric IP addresses into IP address format.

Next, either process the entire dataset at once (suitable for analyzing historical data) or process it in a streaming manner one by one to generate labels with confidence levels. Finally, mark the results to distinguish between malicious traffic and normal traffic. For example, in the decision tree model, malicious traffic is marked as 'trickbot'; only high-confidence malicious traffic is saved.Integrate the prediction results with the original traffic metadata, add tracking information such as timestamps and processing node IDs, and then store the detection results in the database in CSV format.

# 4 Results and Experimentation

## 4.1 Model Training

Randomly select 80% of the data from the dataset as the training set and 20% as the test set. Use X_train, X_test, y_train, y_test = train_test_split(X_train, y_train, test_size=0.20, random_state=42) to ensure the consistent class distribution between the training set and the test set, so as to avoid model bias.Mutual information is used to measure the correlation between each feature in the dataset and the target variable, so as to select important features. The core purpose is to preprocess data and select features, thereby providing high-quality data input for subsequent machine learning tasks.

The results of the decision tree are shown in Figure 3:

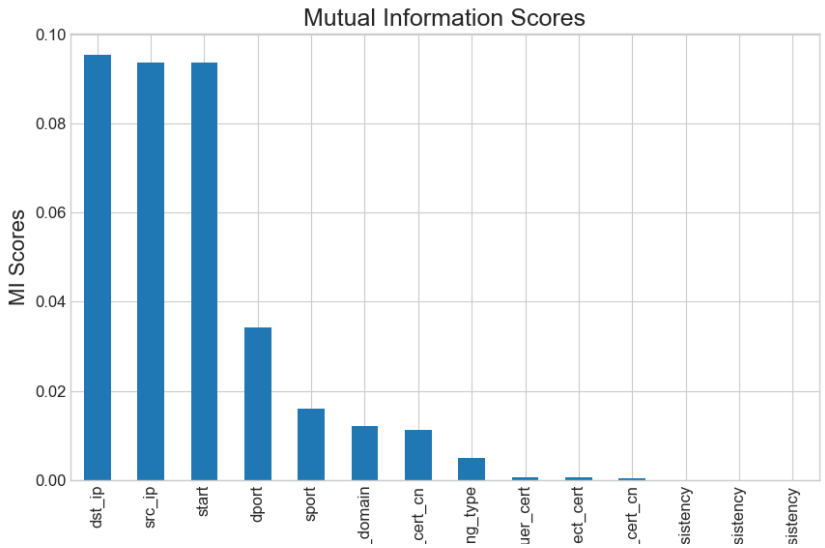


*Figure 3: Mutual information between features and the target in the decision tree model*

The results of the random forest are shown in Figure 4:

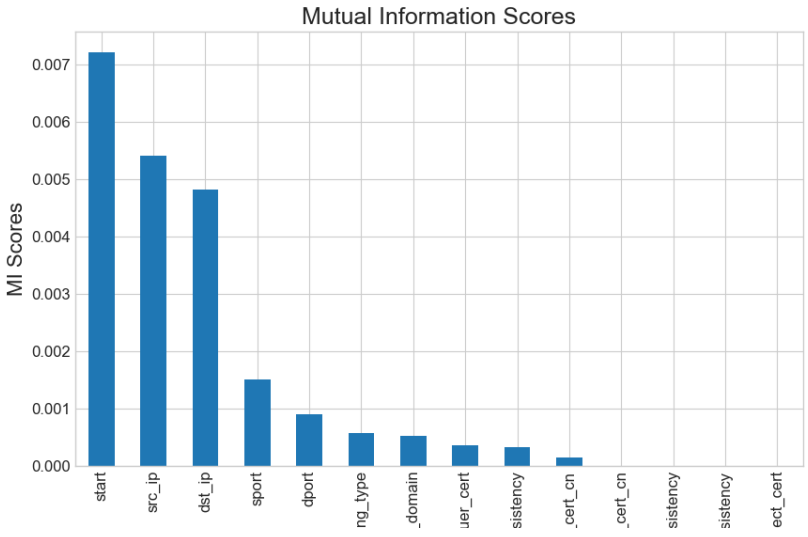


*Figure 4: Mutual information between features and the target in the random forest model*

The results of the LSTM model are shown in Figure 5:

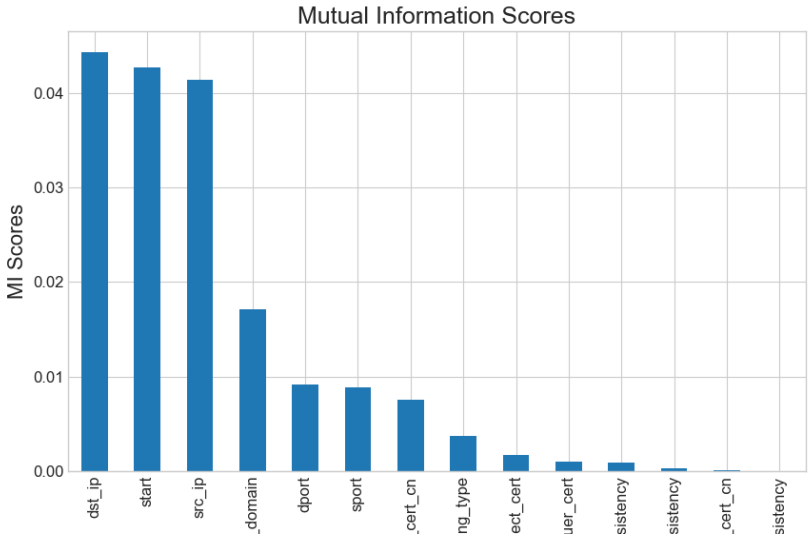


*Figure 5: Mutual information between features and the target in the LSTM model.*

The SMOTE method is used to handle the class imbalance problem in the data, preventing the model from ignoring minority classes during training. The grid search method is adopted to adjust hyperparameters. Cross-validation is used to evaluate the performance of different hyperparameter combinations and select the optimal parameters. After training, the model is saved as a file for subsequent loading and prediction.

## 4.2 Model Calling

Obtain network traffic data from local files or databases, then check for missing values in the data. If there are any, perform imputation operations. Divide the data into feature sets (such as traffic size, protocol type, etc.) and label sets (using 0 to represent normal traffic and 1 to represent abnormal traffic), and further split the data into training sets and test sets.The classification algorithm is selected as the classification model, and a set of candidate hyperparameters are set. The parameters are shown in Table 1.Train on the training set using the optimized hyperparameters, and select the ideal model based on the conclusions from cross-validation. During training, many decision trees will be synthesized according to the given parameters, and then binary classification will be performed through voting. Evaluate the model's performance on the test set.

*Table 1: Model Selection and Candidate Hyperparameters.*

| Model Selection | Candidate Hyperparameters |
|---|---|
| Decision Tree | max_depth: Maximum depth of the tree<br>min_samples_split: Minimum number of samples required to split a node<br>min_samples_leaf: Minimum number of samples required for a leaf node |
| Random Forest | n_estimators: Number of decision trees in the random forest<br>max_features: Maximum number of features available for splitting at each tree node<br>max_depth: Maximum depth of the tree<br>min_samples_split: Minimum number of samples required to split a node<br>min_samples_leaf: Minimum number of samples required for a leaf node |
| LSTM | max_depth: Maximum depth of the tree min_samples_split: Minimum number of samples required to split a node min_samples_leaf: Minimum number of samples required for a leaf node |

Apply the trained model to predict new traffic data. If the prediction result of a piece of traffic is 1, it is judged as malicious traffic; otherwise, it is normal traffic. This step can be applied to traffic monitoring in actual production environments. If the prediction performance of the model is unsatisfactory, it can be further optimized by adjusting the range of hyperparameters, increasing or cleaning training data, or trying other feature engineering methods.

## 4.3 Model Prediction Results

### 4.3.1 Prediction Results of the Decision Tree Model

The accuracy of the model on the test set is 99.85%, indicating that the model can well distinguish between malicious traffic and normal traffic. The predicted confusion matrix is shown in Figure 6:

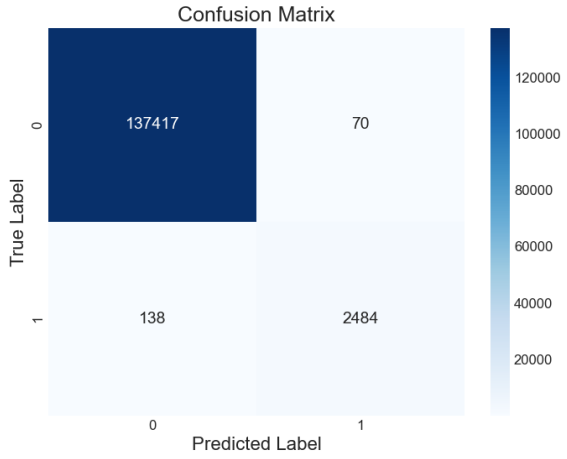


*Figure 6: Confusion matrix of decision tree model prediction*

### 4.3.2 Prediction Results of the Random Forest Model

The accuracy of the model on the test set is 99.92%, indicating that the model can well distinguish between poshc2 traffic and normal traffic. The predicted confusion matrix is shown in Figure 7:

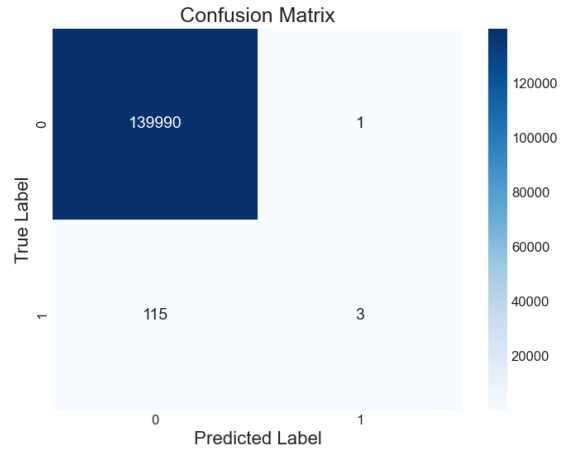


*Figure 7: Confusion matrix of random forest model prediction*

### 4.3.3 Prediction Results of the LSTM Model

The accuracy of the model on the test set is 99.64%, indicating that the model can well distinguish between Qakbot traffic and normal traffic. The predicted confusion matrix is shown in Figure 8:

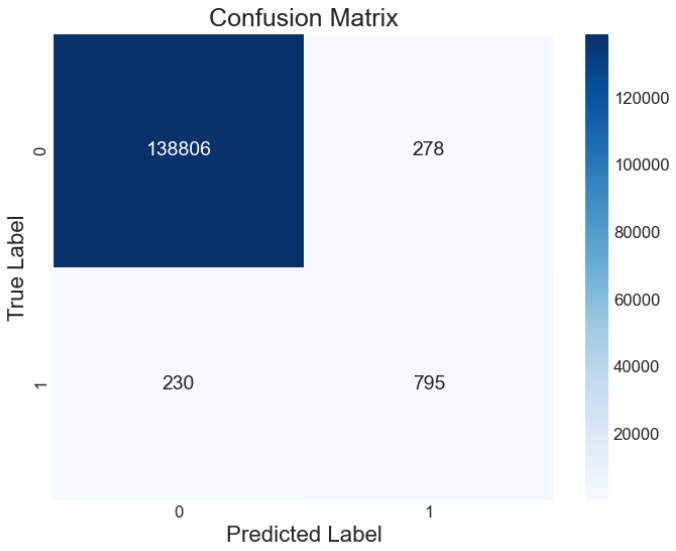


*Figure 8: Confusion matrix of LSTM model prediction*

## 4.4 Model Testing Status

Four commonly used evaluation indicators in deep learning—Accuracy, Precision, Recall, and F1-score—are selected to evaluate and verify the classification effects of different models.The calculation results of the evaluation metrics are shown in Table 1:

*Table 2: Accuracy, Precision, Recall, and F1-score of the three models*

| Model | Accuracy(%) | Precision(%) | Recall (%) | F1-Score(%) |
|---|---|---|---|---|
| decision tree | 99.85 | 99.95 | 99.90 | 99.92 |
| Random Forest | 99.93 | 99.99 | 99.92 | 99.95 |
| LSTM | 99.64 | 99.80 | 99.83 | 99.81 |

Through the calculation and analysis of the four selected indicators, it is known that the three models proposed in this paper have good classification performance in encrypted traffic classification tasks and can well distinguish malicious traffic.

# 5 Conclusions

This study focuses on tackling the tough challenge of detecting malicious encrypted traffic in digital power grids. By integrating network traffic analysis with machine learning technologies, it goes through data preprocessing steps like traffic segmentation, feature extraction, and feature selection, then constructs decision tree, random forest, and LSTM models. Experimental results demonstrate that these three models achieve impressive accuracy rates of 99.85%, 99.93%, and 99.64% respectively, along with excellent performance in precision, recall, and F1-score. Such outcomes enable efficient and accurate identification of malicious traffic, providing robust technical support for digital power grids to

fend off encrypted attacks and holding great significance for safeguarding the security and stable operation of power systems.

# Author's Profile

Peng Xiao is a manager in Information Center of China Southern Power Grid Yunnan Power Grid Co., Ltd., Yunan, China. His main research direction is digital technology and Cyber Security.

Zijie Deng obtained a M.S.E. degree from the South China University of Technology, Guangzhou, China in 2018. He is an engineer in China Southern Power Grid Power Grid Group, Co., Ltd., Guangdong Province, China. His main research direction is digital technology and Cyber Security.

Biao Bai is a general manager in Information Center of China Southern Power Grid Yunnan Power Grid Co., Ltd., Yunan, China. His main research direction is digital technology and Cyber Security.

# References

[1] Quraishi, Aadam, et al. "Employing deep neural networks for real-time anomaly detection and mitigation in IoT-based smart grid cybersecurity systems." 2024 Third International Conference on Distributed Computing and Electrical Circuits and Electronics (ICDCECE). IEEE, 2024.

[2] Priyadarsini, Madhukrishna, and Nitin Sonekar. "A CNN-based approach for anomaly detection in smart grid systems." Electric Power Systems Research 238 (2025): 111077.

[3] Siniosoglou, Ilias, et al. "A unified deep learning anomaly detection and classification approach for smart grid environments." IEEE Transactions on Network and Service Management 18.2 (2021): 1137-1151.

[4] Passerini, Federico, and Andrea M. Tonello. "Smart grid monitoring using power line modems: Anomaly detection and localization." IEEE Transactions on Smart Grid 10.6 (2019): 6178-6186.

[5] Jung, Oliver, et al. "Anomaly Detection in Smart Grids based on Software Defined Networks." SMARTGREENS. 2019.

[6] Zhang, Jiuqi Elise, Di Wu, and Benoit Boulet. "Time series anomaly detection for smart grids: A survey." 2021 IEEE electrical power and energy conference (EPEC). IEEE, 2021.

[7] Fenza, Giuseppe, Mariacristina Gallo, and Vincenzo Loia. "Drift-aware methodology for anomaly detection in smart grid." IEEE Access 7 (2019): 9645-9657.

[8] Alkuwari, Ahmad N., Saif Al-Kuwari, and Marwa Qaraqe. "Anomaly detection in smart grids: A survey from cybersecurity perspective." 2022 3rd International Conference on Smart Grid and Renewable Energy (SGRE). IEEE, 2022.

[9] Shahinzadeh, Hossein, et al. "Anomaly detection and resilience-oriented

countermeasures against cyberattacks in smart grids." 2021 7th International Conference on Signal Processing and Intelligent Systems (ICSPIS). IEEE, 2021.

[10] Anwar, Adnan, and Abdun Naser Mahmood. "Anomaly detection in electric network database of smart grid: Graph matching approach." Electric Power Systems Research 133 (2016): 51-62.

[11] Chren, Stanislav. "Multi-layered Reliability Analysis in Smart Grids." Masarykova univerzita Fakulta informatiky, Brno (2017).

[12] Yang, Ting, et al. "WPD-ResNeSt: Substation station level network anomaly traffic detection based on deep transfer learning." CSEE Journal of Power and Energy Systems 10.6 (2021): 2610-2620.