Hindawi
Mathematical Problems in Engineering
Volume 2020, Article ID 7219659, 21 pages
https://doi.org/10.1155/2020/7219659

Research Article

Network Traffic Anomaly Detection Based on ML-ESN for Power Metering System

S. T. Zhang,[1] X. B. Lin,[1] L. Wu,[1] Y. Q. Song,[2] N. D. Liao,[2] and Z. H. Liang[3]

[1]*CSG Power, Dispatching Control Center, Guangzhou 510663, China*
[2]*Changsha University of Science and Technology, Changsha 410114, China*
[3]*CSG Power, Digital Grid Research Institute, Guangzhou 510623, China*

Correspondence should be addressed to Y. Q. Song; acl158474361@stu.csust.edu.cn

Received 25 February 2020; Revised 20 June 2020; Accepted 2 July 2020; Published 14 August 2020

Academic Editor: Ivo Petras

Due to the diversity and complexity of power network system platforms, some traditional network traffic detection methods work well for small sample datasets. However, the network data detection of complex power metering system platforms has problems of low accuracy and high false-positive rate. In this paper, through a combination of exploration and feedback, a solution for power network traffic anomaly detection based on multilayer echo state network (ML-ESN) is proposed. This method first relies on the Pearson and Gini coefficient method to calculate the statistical distribution and correlation of network flow characteristics and then uses the ML-ESN method to classify the network attacks abnormally. Because the ML-ESN method abandons the back-propagation mechanism, the nonlinear fitting ability of the model is solved. In order to verify the effectiveness of the proposed method, a simulation test was conducted on the UNSW_NB15 network security dataset. The test results show that the average accuracy of this method is more than 97%, which is significantly better than single-layer echo state network, shallow BP neural network, and some traditional machine learning methods.

1. Introduction

At present, the traditional power grid is developing towards the smart grid. Due to the need to improve efficiency, flexibility, reliability, and loss reduction, power advanced metering infrastructure (AMI) has been rapidly developed. The system integrates smart meters, communication networks, data centers, and software systems [1].

Various application servers are mainly responsible for data collection, business application operations, and system maintenance. Large-scale measurement terminals need to access the measurement automation master station through a virtual private network. These communication processes are very vulnerable to attacks [2]. Therefore, the safe operation of power metering systems must rely on reliable communication networks and security protection, detection, and analysis technologies.

Network security experts have discovered that AMI, as an important infrastructure in modern society, is one of the important targets of cyberattacks launched by hostile organizations. The main attack methods for power networks include malicious attacks, denial of service attacks, data spoofing, and network monitoring [3].

Due to the key information exchanged in AMI communication, AMI needs reliable protection to prevent unauthorized access and malicious attacks. Therefore, when migrating to AMI facilities, we must use security mechanism and intrusion detection technology [3].

At present, intrusion detection methods are divided into host-based intrusion detection and network-based intrusion detection. Host-based intrusion detection mainly solves the collection, forensics, and audit of host intrusion traces; network-based intrusion detection is mainly used to analyze the network flow and judge the network attack behavior in real time.

Among them, researchers at home and abroad have applied network intrusion detection technology to anomaly detection of AMI network flow and proposed a variety of

anomaly detection and analysis models, such as deep neural network [1], Markov [4], density statistics [5], BP neural network [6], attack graph-based information fusion [7], and principle component analysis [8].

In [5], Fathnia and Javidi tried to use OPTICS density-based technology to immediately diagnose AMI anomalies in customer information and intelligent data. In order to improve the efficiency of the method, they used LOF indexing technology. This technology actually detects factors related to data anomalies and judges abnormal behavior based on factor scores.

In [7], an AMI intrusion detection system (AMIDS) was proposed. This system uses information fusion technology to combine sensors and consumption data in smart meters to more accurately detect energy theft.

From most existing research, we find that there are more research studies on the detection of AMI theft behavior anomaly, but fewer research studies on the detection of AMI network traffic anomaly attack.

At present, there are still some problems in the existing research on AMI network traffic anomaly detection; for example, the attack rules in [5] for AMI dynamic network environment must be updated regularly. In [6], the authors established a BP neural network training model based on 6 kinds of simple data of AMI and carried out simulation tests on Matlab software. However, the model is still a long way from real engineering applications.

In this study, we are different from the previous AMI anomaly detection content, focusing on the abnormal situation of AMI platform network flow. By continuously extracting AMI network traffic characteristics, such as protocol type, average packet size, maximum and minimum packet size, packet duration, and other related flow-based characteristics, it is possible to accurately analyze the type of attack anomalies encountered by the AMI platform.

We make the following contributions to AMI network attack anomaly detection by using deep learning methods based on stream feature extraction and multilayer echo state networks:

(1) This paper proposes a deep learning method for AMI network attack anomaly detection based on multilayer echo state networks.

(2) By extracting the statistical features of the collected network data streams, the importance and correlation of the statistical features of the network streams are found, the data input of deep learning is optimized, the model training effect is improved, and the model training time is greatly reduced.

(3) In order to verify the validity and accuracy of the method, we tested it in the UNSW_NB15 public benchmark dataset. Experimental results show that our method can detect AMI anomalous attacks and is superior to other methods.

The rest of the paper is organized as follows: Section 2 describes related research; Section 3 introduces AMI network architecture and security issues; Section 4 proposes security solutions; Section 5 focuses on the application of ML-ESN classification method in AMI; Section 6 completes experiments and comparisons; finally, this paper summarizes the research work and puts forward some problems that need to be solved in the future.

2. Related Work

Smart grid introduces computer and network communication technology and physical facilities to form a complex system, which is essentially a huge cyber-physical system (CPS) [9].

AMI is regarded as one of the most basic implementation technologies of smart grid, but so far, a large number of potential vulnerabilities have been discovered. For example, in the AMI network, smart meters, smart data collectors, and data processing centers have their own storage spaces, and these spaces store a lot of information. However, this information can easily be tampered with due to the placement of malware.

In order to solve the security problems of the AMI system, the AMI Network Engineering Task Force (AMI-SEC) [10] pointed out that intrusion detection systems or related technologies can better monitor the AMI network and analyze and discover different attacks through technical means.

At present, domestic and foreign scholars have conducted a lot of research studies on the security of AMI, mainly focusing on power fraud detection, malicious code detection, and network attack detection [11].

2.1. Power Fraud Detection. In terms of power spoofing, attacks are generally divided into two cases according to the consequences of the attack.

One is to inject the wrong data into the power grid to launch an attack, which causes the power grid to oscillate. Once successful, it will cause a large-scale impact on the power grid and users. The second is to enable attackers to obtain direct economic benefits by stealing electricity.

Jokar et al. in [12] present a new energy theft detector based on consumption patterns. The detector uses the predictability of normal and malicious consumption patterns of users and distribution transformer electricity meters to shortlist areas with a high probability of power theft and identifies suspicious customers by monitoring abnormal conditions in consumption patterns.

The authors in [13] proposed a semisupervised anomaly detection framework to solve the problem of energy theft in the public utility database that leads to changes in user usage patterns. Compared with other methods (such as a class of SVM and automatic encoder), the framework can control the detection intensity through the detection index threshold.

2.2. Malicious Code Detection. Since the smart meter transmits power consumption information to the grid terminal, the detection of malicious code can be extended to the detection of executable code. Once it is confirmed that the data uploaded by the meter contain executable code, the data are likely to be malicious code [14].

In order to achieve the rapid detection of AMI malicious code attacks, the authors in [15] proposed a secure and privacy-protected aggregation scheme based on additive homomorphic encryption and proxy reencryption operations in the Paillier cryptosystem.

In [16], Euijin et al. used a disassembler and statistical analysis method to deal with AMI malicious code detection. The method first looks for the characteristics of each data type, uses a disassembler to study the distribution of instructions in the data, and performs statistical analysis on the data payload to determine whether it is malicious code.

2.3. Network Attack Detection. At present, after a large number of statistical discoveries, the main attack point for hackers against the AMI network is the smart meter (SM).

SM is the key equipment that constitutes the AMI network. It realizes the two-way communication between the power company and the user. On the one hand, the user's consumption data are collected and transmitted to the power company through the AMI network. The company's electricity prices and instructions are presented to users.

The intrusion detection mechanism is an important part of the current smart meter security protection. It will monitor the events that occur in the smart meter and analyze the events. Once an attack occurs or a potential security threat is discovered, the intrusion detection mechanism will issue an alarm, so that the system and managers adopt corresponding response mechanisms.

The current research on AMI network security threats mainly analyzes whether there are abnormalities from the perspective of network security, especially the data and network security modeling for smart meter security. The main reason is that physical attacks against AMI are often strong and the most effective, but they are easier to detect.

The existing AMI network attack detection methods mainly include simulation method [17, 18], k-means clustering [1, 19, 20], data mining [21–23], evaluate prequential [24], and PCA [25].

In [17], the authors investigated the puppet attack mechanism and compared other attack types and evaluated the impact of puppet attack on AMI through simulation experiments.

In [18], authors also use the simulation tool NeSSi to study the impact of large-scale DDoS attacks on the intelligent grid AMI network information communication infrastructure.

In order to be able to more accurately analyze the AMI network anomaly, some researchers start with AMI network traffic and use machine learning methods to determine whether a variety of anomaly attacks have occurred on the network.

In [20], the authors use distributed intrusion detection and sliding window methods to monitor the data flow of AMI components and propose a real-time unsupervised AMI data flow mining detection system (DIDS). The system mainly uses the mini-batch k-means algorithm to perform type clustering on network flows to discover abnormal attack types.

In [22], authors use an artificial immune system to detect AMI network attacks. This method first uses the Pcap network packets obtained by the AMI detection equipment and then classifies the attack types through artificial immune methods.

With the increase of AMI traffic feature dimension and noise data, the traffic anomaly detection method based on traditional machine learning faces the problems of low accuracy and poor robustness of traffic feature extraction, which reduces the performance of traffic attack detection to a certain extent. Therefore, the anomaly detection method based on deep learning has become a hot topic in the current network security research [26–34].

Wang et al. [27] proposed a technique that uses deep learning to complete malicious traffic detection. This technology is mainly divided into two implementation steps: one is to use CNN (convolutional neural network) to learn the spatial characteristics of traffic, and the other is to extract data packets from the data stream and learn the spatiotemporal characteristics through CNN and RNN (recurrent neural network).

Currently, there are three main methods of anomaly detection based on deep learning:

(1) Anomaly detection method based on deep Boltzmann machine [28]: this kind of method can extract its essential features through learning of high-dimensional traffic data, so as to improve the detection rate of traffic attacks. However, this type of method has poor robustness in extracting features. When the input data contain noise, its attack detection performance becomes worse.

(2) Based on stacked autoencoders (SAE) anomaly detection method [29]: this type of method can learn and extract traffic data layer by layer. However, the robustness of the extracted features is poor. When the measured data are destroyed, the detection accuracy of this method decreases.

(3) Anomaly detection method based on CNN [27, 30]: the traffic features extracted by this type of method have strong robustness, and the attack detection performance is high, but the network traffic needs to be converted into an image first, which increases the data processing burden, and the influence of network structure information on the accuracy of feature extraction is not fully considered.

In recent years, the achievements of deep learning in the field of time series prediction have also received more and more attention. When some tasks need to be able to process sequence information, RNN can play the advantages of corresponding time series processing compared to the single-input processing of fully connected neural network and CNN.

As a new type of RNN, echo state network is composed of input layer, hidden layer (i.e., reserve pool), and output layer. One of the advantages of ESN is that the entire network only needs to train the W_{out} layer, so its training process is very fast. In addition, for the processing and

prediction of one-dimensional time series, ESN has a very good advantage [32].

Because ESN has such advantages, it is also used by more and more researchers to analyze and predict network attacks [33, 34].

Saravanakumar and Dharani [33] applied the ESN method to network intrusion detection system, tested the method on KDD standard dataset, and found that the method has faster convergence and better performance in IDS.

At present, some researchers have found through experiments that there are still some problems with the single-layer echo state network: (1) there are defects in the model training that can only adjust the output weights; (2) training the randomly generated reserve pool has nothing to do with specific problems, and the parameters are difficult to determine; and (3) the degree of coupling between neurons in the reserve pool is high. Therefore, the application of echo network to AMI network traffic anomaly detection needs to be improved and optimized.

From the previous review, we can find that the traditional AMI network attack analysis methods mainly include classification-based, statistics-based, cluster-based, and information theory (entropy). In addition, different deep learning methods are constantly being tried and applied.

The above methods have different advantages and disadvantages for different research objects and purposes. This article focuses on making full use of the advantages of the ESN method and trying to solve the problem that the single-layer ESN network cannot be directly applied to the AMI complex network traffic detection.

3. AMI Network Architecture and Security Issues

The AMI network is generally divided into three network layers from the bottom up: home area network (HAN), neighboring area network (NAN), and wide area network (WAN). The hierarchical structure is shown in Figure 1.

In Figure 1, the HAN is a network formed by the interconnection of all electrical equipment in the home of a grid user, and its gateway is a smart meter. The neighborhood network is formed by multiple home networks through communication interconnection between smart meters or between smart meters and repeaters. And multiple NANs can form a field area network (FAN) through communication interconnections such as wireless mesh networks, WiMAX, and PLC and aggregate data to the FAN's area data concentrator. Many NANs and FANs are interconnected to form a WAN through switches or routers to achieve communication with power company data and control centers.

The reliable deployment and safe operation of the AMI network is the foundation of the smart grid. Because the AMI network is an information-physical-social multidomain converged network, its security requirements include not only the requirements for information and network security but also the security of physical equipment and human security [35].

As FadwaZeyar [20] mention, AMI faces various security threats, such as privacy disclosure, money gain, energy theft, and other malicious activities. Since AMI is directly related to revenue, customer power consumption, and privacy, the most important thing is to protect its infrastructure.

Researchers generally believe that AMI security detection, defense, and control mainly rely on three stages of implementation. The first is prevention, including security protocols, authorization and authentication technologies, and firewalls. The second is detection, including IDS and vulnerability scanning. The third is reduction or recovery, that is, recovery activities after the attack.

4. Proposed Security Solution

At present, a large number of security detection equipment, such as firewalls, IDS, fortresses, and vertical isolation devices, have been deployed in China's power grid enterprises. These devices have provided certain areas with security detection and defense capabilities, but it brings some problems: (1) these devices generally operate independently and do not work with each other; (2) each device generates a large number of log and traffic files, and the file format is not uniform; and (3) no unified traffic analysis platform has been established.

To solve the above problems, this paper proposes the following solutions: first, rely on the traffic probe to collect the AMI network traffic in real time; second, each traffic probe uploads a unified, standard traffic file to the control center; and finally, the network flow anomalies are analyzed in real time to improve the security detection and identification capabilities of AMI.

As shown in Figure 2, we deploy traffic probes on some important network nodes to collect real-time network flow information of all nodes.

Of course, many domestic and foreign power companies have not established a unified information collection and standardization process. In this case, it can also be processed by equipment and area. For example, to collect data from different devices, before data analysis, perform preprocessing such as data cleaning, data filtering, and data completion, and then use the Pearson and Gini coefficient methods mentioned in this article to find important feature correlations, and it is also feasible to use the ML-ESN algorithm to classify network attacks abnormally.

The main reasons for adopting standardized processing are as follows:

(1) Improve the centralized processing and visual display of network flow information

(2) Partly eliminate and overcome the inadequate problem of collecting information due to single or too few devices

(3) Use multiple devices to collect information and standardize the process to improve the ability of information fusion, so as to enhance the accuracy and robustness of classification

For other power companies that have not performed centralized and standardized processing, they can establish corresponding data preprocessing mechanisms and machine

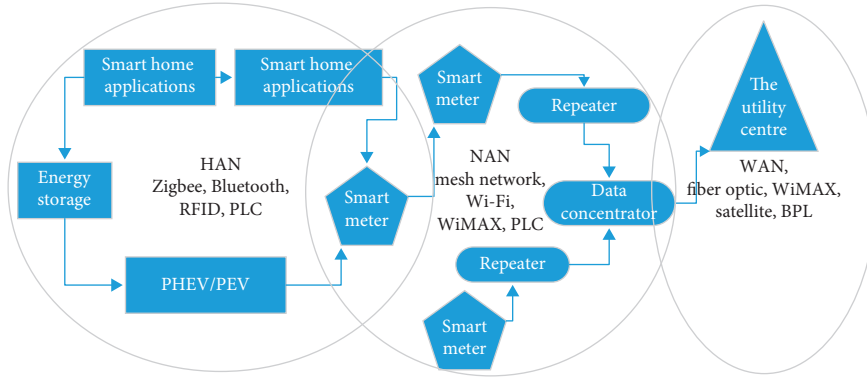

Figure 1: AMI network layered architecture [35].

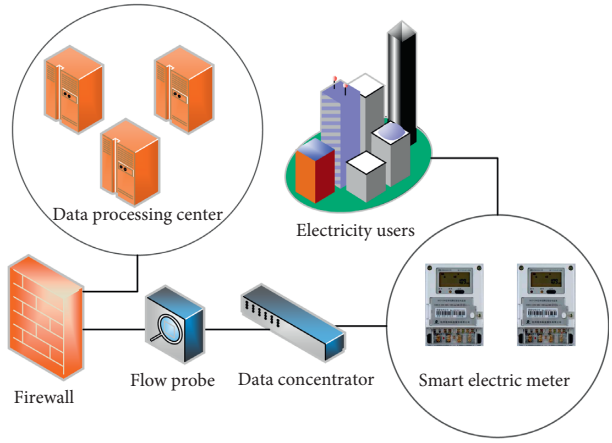

Figure 2: Traffic probe simple deployment diagram.

learning classification algorithms according to their actual conditions.

The goal is the same as this article and is to quickly find abnormal network attacks from a large number of network flow data.

4.1. Probe Stream Format Standards and Collection Content. In order to be able to unify the format of the probe stream data, the international IPFIX standard is referenced, and the relevant metadata of the probe stream is defined. The metadata include more than 100 different information units. Among them, the information units with IDs less than or equal to 433 are clearly defined by the IPFIX standard. Others (IDs greater than or equal to 1000) are defined by us. Some important metadata information is shown in Table 1.

Metadata are composed of strings, each information element occupies a fixed position of the string, the strings are separated by ^, and the last string is also terminated by ^. In addition, the definition of an information element that does not exist in metadata is as follows: if there is an information element defined below in a metadata and the corresponding information element position does not need to be filled in, it means that two ^ are adjacent at this time. If the extracted information element has a caret, it needs to be escaped with

Table 1: Some important metadata information.

ID	Name	Type	Length	Description
1	EventID	String	64	Event ID
2	ReceiveTime	Long	8	Receive time
3	OccurTime	Long	8	Occur time
4	RecentTime	Long	8	Recent time
5	ReporterID	Long	8	Reporter ID
6	ReporterIP	IPstring	128	Reporter IP
7	EventSrcIP	IPstring	128	Event source IP
8	EventSrcName	String	128	Event source name
9	EventSrcCategory	String	128	Event source category
10	EventSrcType	String	128	Event source type
11	EventType	Enum	128	Event type
12	EventName	String	1024	Event name
13	EventDigest	String	1024	Event digest
14	EventLevel	Enum	4	Event level
15	SrcIP	IPstring	1024	Source IP
16	SrcPort	String	1024	Source port
17	DestIP	IPstring	1024	Destination IP
18	DestPort	String	1024	Destination port
19	NatSrcIP	IPstring	1024	NAT translated source IP
20	NatSrcPort	String	1024	NAT translated source port
21	NatDestIP	IPstring	1024	NAT translated destination IP
22	NatDestPort	String	1024	NAT translated destination port
23	SrcMac	String	1024	Source MAC address
24	DestMac	String	1024	Destination MAC address
25	Duration	Long	8	Duration (second)
26	UpBytes	Long	8	Up traffic bytes
27	DownBytes	Long	8	Down traffic bytes
28	Protocol	String	128	Protocol
29	AppProtocol	String	1024	Application protocol

the escape string %%%. Part of the real probe stream data is shown in Figure 3.

The first record in Figure 3 is as follows: "*6^69085d3e54323 60300000000^10.107.1.10^10.107.212.41^19341^22^6^40^1^40^ 1^1564365874^15643. . .^^^2019-07-29T03:08:23.969^^^^^^TCP ^^^^^^ 10.107.1.10^ 10.107.212.41^ . . .^^^^*"

Part of the above probe flow is explained as follows according to the metadata standard definition: (1) 6: metadata

6^69085d3e5432360300000000^10.107.1.10^10.107.212.41^19341^22^6^40^1^40^1^1564365874^1564365874^^^^
6^71135d3e5432362900000000^10.107.1.10^10.107.212.41^32365^23^6^40^1^0^0^1564365874^1564365874^^^
6^90855d3e5432365d00000000^10.107.1.10^10.107.212.41^62215^6000^6^40^1^40^1^1564365874^1564365874^
6^c4275d3e5432367800000000^10.107.1.10^10.107.212.41^50504^25^6^40^1^40^1^1564365874^1564365874^^^
6^043b5d3e5432366d00000000^10.107.1.10^10.107.212.41^1909^2048^1^28^1^28^1^1564365874^1564365874^^
6^71125d3e5432362900000000^10.107.1.10^10.107.212.41^46043^443^6^40^1^40^1^1564365874^1564365874^^
6^043b5d3e5432366d00000001^10.107.1.10^10.107.212.41^1909^2048^1^28^1^28^1^1564365874^1564365874^^
6^3ff75d3e5432361600000000^10.107.1.10^10.107.212.41^39230^80^6^80^2^44^1^1564365874^1564365874^^^
6^044a5d3e5432366d00000000^10.107.1.10^10.107.212.41^31730^21^6^40^1^40^1^1564365874^1564365874^^
6^7e645d3e6df9364a00000000^10.107.1.10^10.107.212.41^33380^6005^6^56^1^40^1^1564372473^1564372473^
6^143d5d3e6dfc361500000000^10.107.1.10^10.107.212.41^47439^32776^6^56^1^0^0^1564372476^1564372476^
6^81b75d3e6df8360100000000^10.107.1.10^10.107.212.41^56456^3086^6^56^1^40^1^1564372472^1564372472^
6^e0745d3e6dfc367300000000^10.107.1.10^10.107.212.41^54783^44334^6^56^1^0^0^1564372476^1564372476^

Figure 3: Part of the real probe stream data.

version; (2) 69085d3e5432360300000000: metadata ID; (3) 10.107.1.10: source IP; (4) 10.107.212.41: destination IP; (5) 19341: source port; (6) 22: destination port; and (7) 6: protocol, TCP.

4.2. Proposed Framework. The metadata of the power probe stream used contain hundreds, and it can be seen from the data obtained in Figure 3 that not every stream contains all the metadata content. If these data analyses are used directly, one is that the importance of a single metadata cannot be directly reflected, and the other is that the analysis data dimensions are particularly high, resulting in particularly long calculation time. Therefore, the original probe stream metadata cannot be used directly, but needs further preprocessing and analysis.

In order to detect AMI network attacks, we propose a novel network attack discovery method based on AMI probe traffic and use multilayer echo state networks to classify probe flows to determine the type of network attack. The specific implementation framework is shown in Figure 4.

The framework mainly includes three processing stages, and the three steps are as follows:

Step 1: collect network flow metadata information in real time through network probe flow collection devices deployed in different areas.

Step 2: first, the time series or segmentation of the collected network flow metadata is used to statistically obtain the statistical characteristics of each part of the network flow. Second, the statistically obtained characteristic values are standardized according to certain data standardization guidelines. Finally, in order to be able to quickly find important features and correlations between the features that react to network attack anomalies, the standardized features are further filtered.

Step 3: establish a multilayer echo state network deep learning model, and classify the data after feature extraction, part of which is used as training data and part of which is used as test data. Cross-validation was performed on the two types of data to check the correctness and performance of the proposed model.

4.3. Feature Extraction. Generally speaking, to realize the classification and identification of network traffic, it is necessary to better reflect the network traffic of different network attack behaviors and statistical behavior characteristics.

Network traffic [36] refers to the collection of all network data packets between two network hosts in a complete network connection. According to the currently recognized standard, it refers to the set of all network data packets with the same quintuple within a limited time, including the sum of the data characteristics carried by the related data on the set.

As you know, some simple network characteristics can be extracted from the network, such as source IP address, destination IP address, source port, destination port, and protocol, and because network traffic is exchanged between source and destination machines, source IP address, destination IP address, source port, and destination port are also interchanged, which reflects the bidirectionality of the flow.

In order to be able to more accurately reflect the characteristics of different types of network attacks, it is necessary to cluster and collect statistical characteristics of network flows.

Firstly, network packets are aggregated into network flows, that is, to distinguish whether each network flow is generated by different network behaviors. Secondly, this paper refers to the methods proposed in [36, 37] to extract the statistical characteristics of network flow.

In [36], 22 statistical features of malicious code attacks are extracted, which mainly includes the following:

Statistical characteristics of data size: forward and backward packets; maximum, minimum, average, and standard deviation; and forward and backward packet ratio

Statistical characteristics of time: duration; forward and backward packet interval; and maximum, minimum, average, and standard deviation

In [37], 249 statistical characteristics of network traffic are summarized and analyzed. The main statistical characteristics in this paper are as follows:

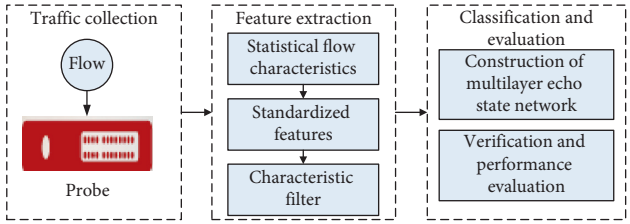

Figure 4: Proposed AMI network traffic detection framework.

Time interval: maximum, minimum, average interval time, standard deviation

Packet size: maximum, minimum, average size, and packet distribution

Number of data packets: out and in

Data amount: input byte amount and output byte amount

Stream duration: duration from start to end

Some of the main features of network traffic extracted in this paper are shown in Table 2.

4.4. Feature Standardization. Because the various attributes of the power probe stream contain different data type values, and the differences between these values are relatively large, it cannot be directly used for data analysis. Therefore, we need to perform data preprocessing operations on statistical features, which mainly include operations such as feature standardization and unbalanced data elimination.

At present, the methods of feature standardization are mainly [38] Z-score, min-max, and decimal scaling, etc.

Because there may be some nondigital data in the standard protocol, such as protocol, IP, and TCP flag, these data cannot be directly processed by standardization, so nondigital data need to be converted to digital processing. For example, change the character "dhcp" to the value "1."

In this paper, Z-score is selected as a standardized method based on the characteristics of uneven data distribution and different values of the power probe stream. Z-score normalization processing is shown in the following formula:

$$x' = \frac{x - \overline{x}}{\delta}, \tag{1}$$

where $\overline{x}$ is the mean value of the original data, δ is the standard deviation of the original data, and $\text{std} = ((x_1 - \overline{x}).^2 + (x_2 - \overline{x})^2 + \ldots/n(\text{number of samples per feature}))$, $\delta = \sqrt{\text{std}}$.

4.5. Feature Filtering. In order to detect attack behavior more comprehensively and accurately, it is necessary to quickly and accurately find the statistical characteristics that characterize network attack behavior, but this is a very difficult problem. The filter method is the currently popular feature filtering method. It regards features as independent objects, evaluates the importance of features according to quality metrics, and selects important features that meet requirements.

Table 2: Some of the main features.

ID	Name	Description
1	SrcIP	Source IP address
2	SrcPort	Source IP port
3	DestIP	Destination IP address
4	DestPort	Destination IP port
5	Proto	Network protocol, mainly TCP, UDP, and ICMP
6	total_fpackets	Total number of forward packets
7	total_fvolume	Total size of forward packets
8	total_bpackets	Total number of backward packets
9	total_bvolume	Total size of backward packets
...	...	...
29	max_biat	Maximum backward packet reach interval
30	std_biat	Time interval standard deviation of backward packets
31	duration	Network flow duration

At present, there are many data correlation methods. The more commonly used methods are chart correlation analysis (line chart and scatter chart), covariance and covariance matrix, correlation coefficient, unary and multiple regression, information entropy, and mutual information, etc.

Because the power probe flow contains more statistical characteristics, the main characteristics of different types of attacks are different. In order to quickly locate the important characteristics of different attacks, this paper is based on the correlation of statistical characteristics data and information gain to filter the network flow characteristics.

Pearson coefficient is used to calculate the correlation of feature data. The main reason is that the calculation of the Pearson coefficient is more efficient, simple, and more suitable for real-time processing of large-scale power probe streams.

Pearson correlation coefficient is mainly used to reflect the linear correlation between two random variables (x, y), and its calculation $\rho_{x,y}$ is shown in the following formula:

$$\rho_{x,y} = \frac{\text{cov}(x, y)}{\sigma_x \sigma_y} = \frac{E\left[\left(x - u_x\right)\left(y - u_y\right)\right]}{\sigma_x \sigma_y}, \tag{2}$$

where $\text{cov}(x, y)$ is the covariance of x, y, σ_x is the standard deviation of x, and σ_y is the standard deviation of y. If you estimate the covariance and standard deviation of the sample, you can get the sample Pearson correlation coefficient, which is usually expressed by r:

$$r = \frac{\sum_{i=1}^{n}\left(x_i - \overline{x}\right)\left(y_i - \overline{y}\right)}{\sqrt{\sum_{i=1}^{n}\left(x_i - \overline{x}\right)^2}\sqrt{\sum_{i=1}^{n}\left(y_i - \overline{y}\right)^2}}, \tag{3}$$

where n is the number of samples, x_i and y_i are the observations at point i corresponding to variables x and y, $\overline{x}$ is the average number of x samples, and $\overline{y}$ is the average number of y samples. The value of r is between −1 and 1. When the value is 1, it indicates that there is a completely positive correlation between the two random variables; when

the value is −1, it indicates that there is a completely negative correlation between the two random variables; when the value is 0, it indicates that the two random variables are linearly independent.

Because the Pearson method can only detect the linear relationship between features and classification categories, this will cause the loss of the nonlinear relationship between the two. In order to further find the nonlinear relationship between the characteristics of the probe flow, this paper calculates the information entropy of the characteristics and uses the Gini index to measure the nonlinear relationship between the selected characteristics and the network attack behavior from the data distribution level.

In the classification problem, assuming that there are k classes, and the probability that the sample points belong to the i classes is P_i, the Gini index of the probability distribution is defined as follows [39]:

$$\text{Gini}(P) = \sum_{i=1}^{K} P_i(1-P) = 1 - \sum_{i=1}^{K} p_i^2. \tag{4}$$

Given the sample set D, the Gini coefficient is expressed as follows:

$$\text{Gini}(P) = 1 - \sum_{i=1}^{K} \left(\frac{|C_k|}{D}\right)^2, \tag{5}$$

where C_k is a subset of samples belonging to the Kth class in D and k is the number of classes.

5. ML-ESN Classification Method

ESN is a new type of recurrent neural network proposed by Jaeger in 2001 and has been widely used in various fields, including dynamic pattern classification, robot control, object tracking, nuclear moving target detection, and event monitoring [32]. In particular, it has made outstanding contributions to the problem of time series prediction. The basic ESN network model is shown in Figure 5.

In this model, the network has 3 layers: input layer, hidden layer (reservoir), and output layer. Among them, at time t, assuming that the input layer includes k nodes, the reservoir contains N nodes, and the output layer includes L nodes, then

$$\begin{aligned} U(t) &= [u_1(t), u_2(t), \ldots, u_k(t)]^T, \\ x(t) &= [x_1(t), x_2(t), \ldots, x_N(t)]^T, \\ y(t) &= [y_1(t), y_2(t), \ldots, y_L(t)]^T. \end{aligned} \tag{6}$$

$W_{in}(N * K)$ represents the connection weight of the input layer to the reservoir. $W(N * N)$ represents the connection weight from $x(t-1)$ to $x(t)$. W_{out} $(L * (K + N + L))$ represents the weight of the connection from the reservoir to the output layer. $W_{\text{back}}(N * L)$ represents the connection weight of $y(t-1)$ to $x(t)$, and this value is optional.

When $u(t)$ is input, the updated state equation of the reservoir is given by

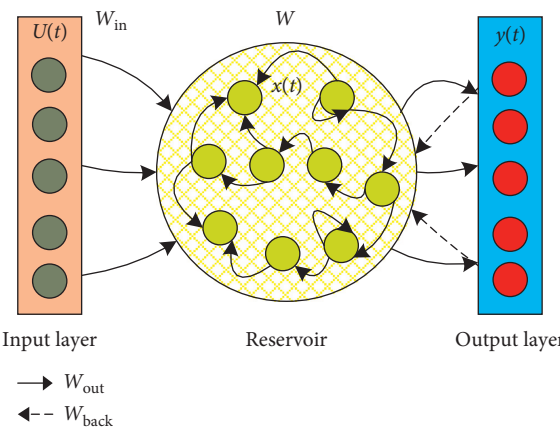

Figure 5: ESN basic model.

$$x(t+1) = f\left(w_{\text{in}} * u(t+1) + w_{\text{back}} * x(t)\right), \tag{7}$$

where f is the selected activation function and $f\prime$ is the activation function of the output layer. Then, the output state equation of ESN is given by

$$y(t+1) = f'\left(w_{out} * ([u(t+1); x(t+1)])\right), \tag{8}$$

Researchers have found through experiments that the traditional echo state network reserve pool is randomly generated, with strong coupling between neurons and limited predictive power.

In order to overcome the existing problems of ESN, some improved multilayer ESN (ML-ESN) networks are proposed in the literature [40, 41]. The basic model of the ML-ESN is shown in Figure 6.

The difference between the two architectures is the number of layers in the hidden layer. There is only one reservoir in a single layer and more than one in multiple layers. The updated state equation of ML-ESN is given by [41]

$$\begin{aligned} x_1(n+1) &= f\left(w_{\text{in}} u(n+1) + w_1 x_1(n)\right), \\ x_k(n+1) &= f\left(w_{\text{inter}(k-1)} x_k(n+1) + w_k x_k(n)\right), \\ &\vdots \\ x_M(n+1) &= f\left(w_{\text{inter}(M-1)} x_k(n+1) + w_M x_M(n)\right). \end{aligned} \tag{9}$$

Calculate the output ML-ESN result according to formula (9):

$$y(n+1) = f_{\text{out}}\left(W_{\text{out}} x_M(n+1)\right). \tag{10}$$

5.1. ML-ESN Classification Algorithm. In general, when the AMI system is operating normally and securely, the statistical entropy of the network traffic characteristics within a period of time will not change much. However, when the network system is attacked abnormally, the statistical characteristic entropy value will be abnormal within a certain time range, and even large fluctuations will occur.

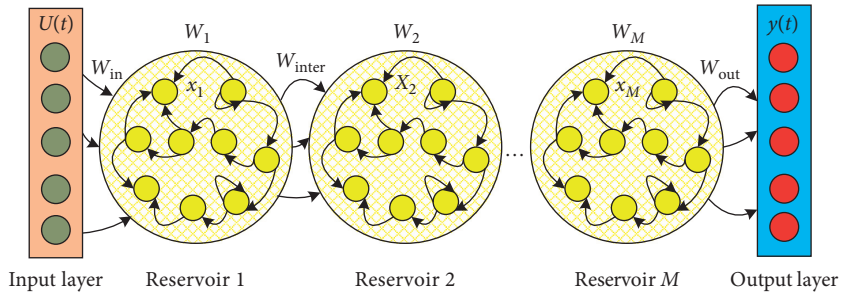

Figure 6: ML-ESN basic model.

It can be seen from Figure 5 that ESN is an improved model for training RNNs. The steps are to use a large-scale random sparse network (reservoir) composed of neurons as the processing medium for data information, and then the input feature value set is mapped from the low-dimensional input space to the high-dimensional state space. Finally, the network is trained and learned by using linear regression and other methods on the high-dimensional state space.

However, in the ESN network, the value of the number of neurons in the reserve pool is difficult to balance. If the number of neurons is relatively large, the fitting effect is weakened. If the number of neurons is relatively small, the generalization ability cannot be guaranteed. Therefore, it is not suitable for directly classifying AMI network traffic anomalies.

On the contrary, the ML-ESN network model can satisfy the internal training network of the echo state by adding multiple reservoirs when the size of a single reservoir is small, thereby improving the overall training performance of the model.

This paper selects the ML-ESN model as the AMI network traffic anomaly classification learning algorithm. The specific implementation is shown in Algorithm 1.

6. Simulation Test and Result Analysis

In order to verify the effectiveness of the proposed method, this paper selects the UNSW_NB15 dataset for simulation testing. The test defines multiple classification indicators, such as accuracy rate, false-positive rate, and $F1$-score. In addition, the performance of multiple methods in the same experimental set is also analyzed.

6.1. UNSW_NB15 Dataset. Currently, one of the main research challenges in the field of network security attack inspection is the lack of comprehensive network-based datasets that can reflect modern network traffic conditions, a wide variety of low-footprint intrusions, and deep structured information about network traffic [42].

Compared with the KDD98, KDDCUP99, and NSLKDD benchmark datasets that have been generated internationally more than a decade ago, the UNSW_NB15 dataset appeared late and can more accurately reflect the characteristics of complex network attacks.

The UNSW_NB15 dataset can be downloaded directly from the network and contains nine types of attack data, namely, Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode, and Worms [43].

In these experiments, two CSV-formatted datasets (training and testing) were selected, and each dataset contained 47 statistical features. The statistics of the training dataset are shown in Table 3.

Because of the original dataset, the format of each eigenvalue is not uniform. For example, most of the data are of numerical type, but some features contain character type and special symbol: "-," so it cannot be directly used for data processing. Before data processing, the data are standardized, and some of the processed feature results are shown in Figure 7.

6.2. Evaluation Indicators. In order to objectively evaluate the performance of this method, this article mainly uses three indicators: accuracy (correct rate), FPR (false-positive rate), and $F-$score (balance score) to evaluate the experimental results. Their calculation formulas are as follows:

$$\begin{aligned}
\text{accuracy} &= \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}},\\
\text{FPR} &= \frac{\text{FP}}{\text{FP} + \text{FN}},\\
\text{TPR} &= \frac{\text{TP}}{\text{FN} + \text{TP}},\\
\text{precision} &= \frac{\text{TP}}{\text{TP} + \text{FP}},\\
\text{recall} &= \frac{\text{TP}}{\text{FN} + \text{TP}},\\
F-\text{score} &= \frac{2 * \text{precision} * \text{recall}}{\text{precision} + \text{recall}}.
\end{aligned} \tag{11}$$

The specific meanings of TP, TN, FP, and FN used in the above formulas are as follows:

TP (true positive): the number of abnormal network traffic successfully detected

TN (true negative): the number of successfully detected normal network traffic

(1) **Input**:
(2) $D1$: training dataset
(3) $D2$: test dataset
(4) $U(t)$: input feature value set
(5) N: the number of neurons in the reservoir
(6) R_i: the number of reservoirs
(7) α: interconnection weight spectrum radius
(8) **Output**
(9) Training and testing classification results
(10) **Steps**:
(11) (1) Initially set the parameters of ML-ESN, and determine the corresponding number of input and output units according to the dataset:

(i) Set training data length: *trainLen*
(ii) Set test data length: *testLen*
(iii) Set the number of reservoirs: R_i
(iv) Set the number of neurons in the reservoir:N
(v) Set the speed value of reservoir update: α
(vi) Set $x_i(0) = 0$, $(1 \le i \le M)$

(12) (2) Initialize the input connection weight matrix w_{in}, internal connection weight of the cistern $w_i\,(1 \le i \le M)$, and weight of external connections between reservoirs w_{inter}:

(i) Randomly initialize the values of w_{in}, w_i, and w_{inter}.
(ii) Through statistical normalization and spectral radius calculation, w_{inter} and w_i are bunched to meet the requirements of sparsity. The calculation formula is as follows: $w_i = \alpha(w_i/|\lambda_{\text{in}}|)$, $w_{\text{inter}} = \alpha(w_{\text{inter}}/\lambda_{\text{inter}})$, and λ_{in} andλ_{inter} are the spectral radii of w_i and w_{inter} matrices, respectively.

(13) (3) Input training samples into initialized ML-ESN, collect state variables by using equation (9), and input them to the activation function of the processing unit of the reservoir to obtain the final state variables:

(i) For t from 1 to T, compute $x_1(t)$
(a) Calculate $x_1(t)$ according to equation (7)
(b) For i from 2 to M, compute $x_i(t)$
(i) Calculate $x_i(t)$ according to equations (7) and (9)
(c) Get matrix H, $H = [x(t+1); u(t+1)]$

(14) (4) Use the following to solve the weight matrix W_{out} from reservoir to output layer to get the trained ML-ESN network structure:

(i) $W_{\text{out}} = \text{DH}^T(\text{HH}^T + \beta I)^{-1}$, where β is the ridge regression parameter, I matrix is the identity matrix, and $D = [e(t)]$ and $H = [x(t+1); u(t+1)]$ are the expected output matrix and the state collection matrix.

(15) (5) Calculate the output ML-ESN result according to formula (10).

(i) Select the SoftMax activation function and calculate the output f_{out} value.

(16) (6) The data in $D2$ are input into the trained ML-ESN network, the corresponding category identifier is obtained, and the classification error rate is calculated.

Algorithm 1: AMI network traffic classification.

Table 3: The statistics of the training dataset.

ID	Type	Number of packets	Size (MB)
1	Normal	56000	3.63
2	Analysis	1560	0.108
3	Backdoors	1746	0.36
4	DoS	12264	2.42
5	Exploits	33393	8.31
6	Fuzzers	18184	4.62
7	Generic	40000	6.69
8	Reconnaissance	10491	2.42
9	Shellcode	1133	0.28
10	Worms	130	0.044

FP (false positive): the number of normal network traffic that is identified as abnormal network traffic

FN (false negative): the number of abnormal network traffic that is identified as normal network traffic

6.3. Simulation Experiment Steps and Results

Step 1. In a real AMI network environment, first collect the AMI probe stream metadata in real time, and these metadata are as shown in Figure 3; but in the UNSW_NB15 dataset, this step is directly omitted.

	dur	proto	service	state	spkts	dpkts	sbytes	dbytes
0	−0.19102881	0.151809388	−0.70230738	−0.40921807	−0.10445581	−0.1357688	−0.04913362	−0.10272556
1	−0.10948479	0.151809388	−0.70230738	−0.40921807	−0.04601353	0.172598967	−0.04640996	0.188544124
2	0.040699218	0.151809388	−0.70230738	−0.40921807	−0.08984524	−0.02693312	−0.04852709	−0.01213277
3	0.049728681	0.151809388	0.599129702	−0.40921807	−0.0606241	−0.06321168	−0.04701649	−0.09856278
4	−0.14041703	0.151809388	−0.70230738	−0.40921807	−0.07523467	−0.11762952	−0.04755436	−0.10205729
5	−0.15105199	0.151809388	−0.70230738	−0.40921807	−0.07523467	−0.11762952	−0.04755436	−0.10205729
6	−0.11145895	0.151809388	−0.70230738	−0.40921807	−0.07523467	−0.09949024	−0.04755436	−0.10145863
7	−0.12928625	0.151809388	−0.70230738	−0.40921807	−0.07523467	−0.09949024	−0.04755436	−0.10145863
8	−0.12599609	0.151809388	−0.70230738	−0.40921807	−0.07523467	−0.09949024	−0.04755436	−0.10145863

Figure 7: Partial feature data after standardized.

Step 2. Perform data preprocessing on the AMI metadata or UNSW_NB15 CSV format data, which mainly include operations such as data cleaning, data deduplication, data completion, and data normalization to obtain normalized and standardized data, and standardized data are as shown in Figure 7, and normalized data distribution is as shown in Figure 8.

As can be seen from Figure 8, after normalizing the data, most of the attack type data are concentrated between 0.4 and 0.6, but Generic attack type data are concentrated between 0.7 and 0.9, and normal type data are concentrated between 0.1 and 0.3.

Step 3. Calculate the Pearson coefficient value and the Gini index for the standardized data. In the experiment, the Pearson coefficient value and the Gini index for the UNSW_NB15 standardized data are as shown in Figures 9 and 10, respectively.

It can be observed from Figure 9 that the Pearson coefficients between features are quite different, for example, the correlation between spkts (source to destination packet count) and sloss (source packets retransmitted or dropped) is relatively large, reaching a value of 0.97. However, the correlation between spkts and ct_srv_src (no. of connections that contain the same service and source address in 100 connections according to the last time.) is the smallest, only −0.069.

In the experiment, in order not to discard a large number of valuable features at the beginning, but to retain the distribution of the original data as much as possible, the initial value of the Pearson correlation coefficient is set to 0.5. Features with a Pearson value greater than 0.5 will be discarded, and features less than 0.5 will be retained.

Therefore, it can be seen from Figure 9 that the correlations between spkts and sloss, dpkts (destination to source packet count), and dbytes (destination to source transaction bytes), tcprtt and ackdat (TCP connection setup time, the time between the SYN_ACK and the ACK packets) all exceed 0.9, and there is a long positive correlation. On the contrary, the correlation between spkts and state, dbytes, and tcprtt is less than 0.1, and the correlation is very small.

In order to further examine the importance of the extracted statistical features in the dataset, the Gini coefficient values are calculated for the extracted features, and these values are shown in Figure 10.

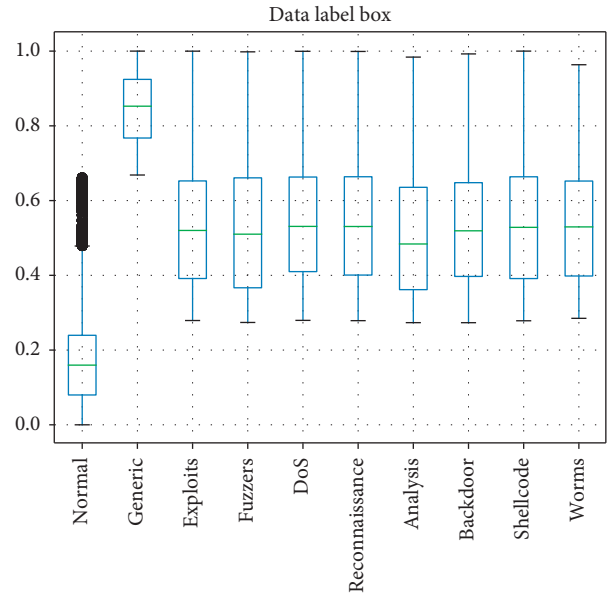

Figure 8: Normalized data distribution.

As can be seen from Figure 10, the selected Gini values of dpkts, dbytes, loss, and tcprtt features are all less than 0.6, while the Gini values of several features such as state and service are equal to 1. From the principle of Gini coefficients, it can be known that the smaller the Gini coefficient value of a feature, the lower the impureness of the feature in the dataset, and the better the training effect of the feature.

Based on the results of Pearson and Gini coefficients for feature selection, in the UNSW_NB15 dataset, this paper finally selected five important features as model classification features, and these five features are rate, sload (source bits per second), dload (destination bits per second), sjit (source jitter (mSec)), and dtcpb (destination TCP base sequence number).

Step 4. Perform attack classification on the extracted feature data according to Algorithm 1. Relevant parameters were initially set in the experiment, and the specific parameters are shown in Table 4.

In Table 4, the input dimension is determined according to the number of feature selections. For example, in the

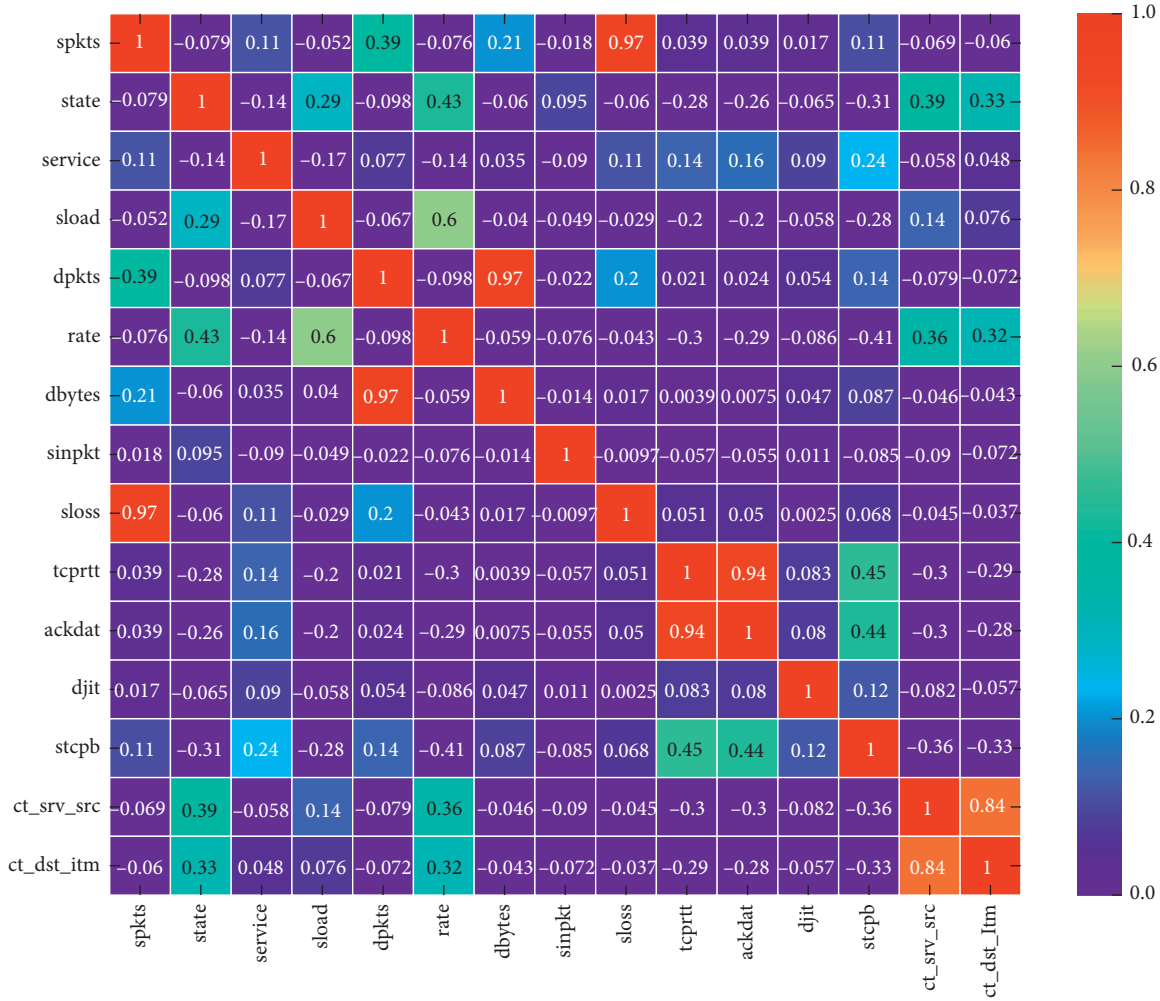

Figure 9: The Pearson coefficient value for UNSW_NB15.

UNSW_NB15 data test, five important features were selected according to the Pearson and Gini coefficients.

The number of output neurons is set to 10, and these 10 outputs correspond to 9 abnormal attack types and 1 normal type, respectively.

Generally speaking, under the same dataset, as the number of reserve pools increases, the time for model training will gradually increase, but the accuracy of model detection will not increase all the time, but will increase first and then decrease. Therefore, after comprehensive consideration, the number of reserve pools is initially set to 3.

The basic idea of ML-ESN is to generate a complex dynamic space that changes with the input from the reserve pool. When this state space is sufficiently complex, it can use these internal states to linearly combine the required output. In order to increase the complexity of the state space, this article sets the number of neurons in the reserve pool to 1000.

In Table 4, the reason why the tanh activation function is used in the reserve pool layer is that its value range is between −1 and 1, and the average value of the data is 0, which is more conducive to improving training efficiency. Second, when tanh has a significant difference in characteristics, the detection effect will be better. In addition, the neuron fitting training process in the ML-ESN reserve pool will continuously expand the feature effect.

The reason why the output layer uses the sigmoid activation function is that the output value of sigmoid is between 0 and 1, which just reflects the probability of a certain attack type.

In Table 4, the last three parameters are important parameters for tuning the ML-ESN model. The three values are set to 0.9, 50, and 1.0×10^{-6}, respectively, mainly based on relatively optimized parameter values obtained through multiple experiments.

6.3.1. Experimental Data Preparation and Experimental Environment. During the experiment, the entire dataset was divided into two parts: the training dataset and the test dataset.

The training dataset contains 175320 data packets, and the ratio of normal and attack abnormal packets is 0.46 : 1.

The test dataset contains 82311 data packets, and the ratio of normal and abnormal packets is 0.45 : 1.

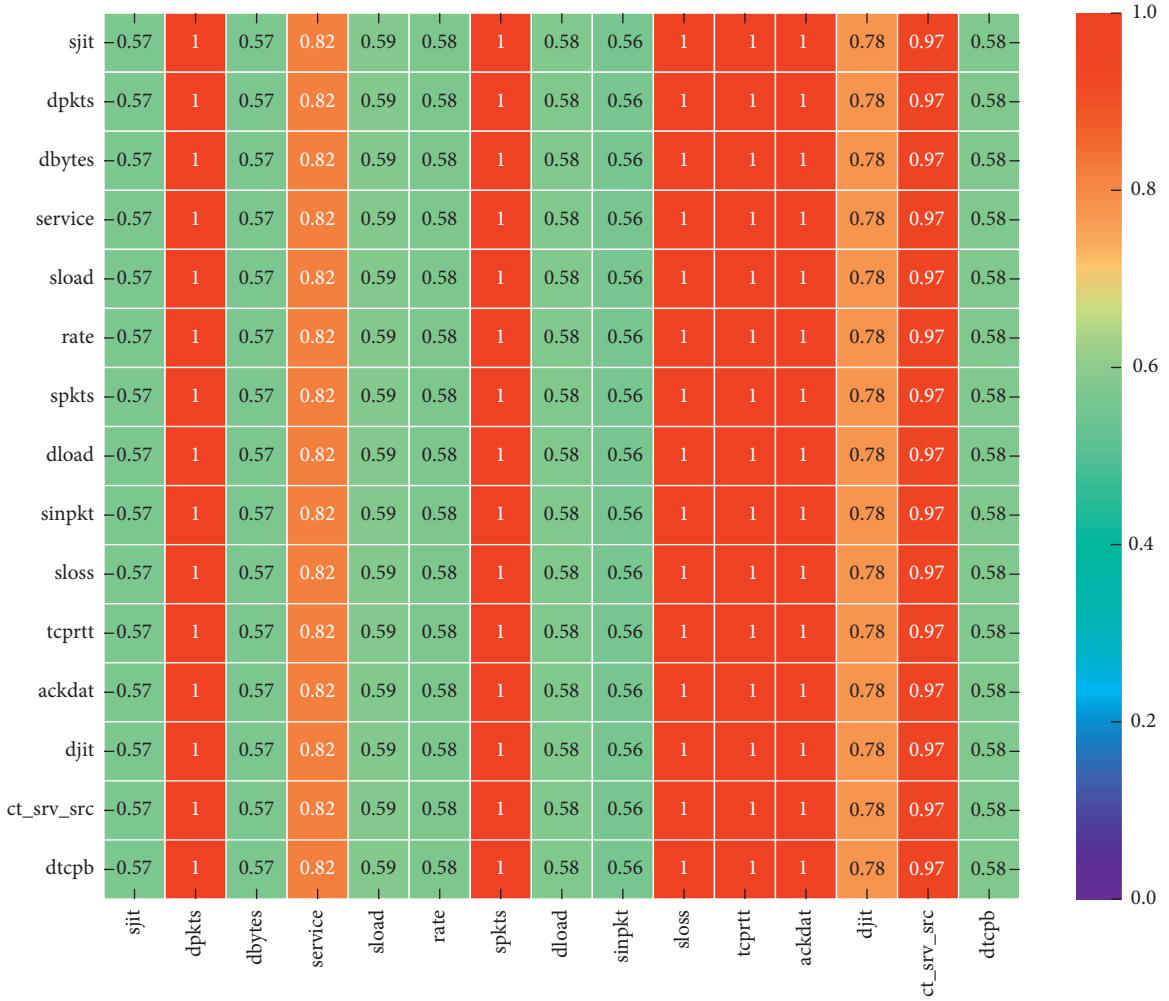

Figure 10: The Gini value for UNSW_NB15.

Table 4: The parameters of ML-ESN experiment.

Parameters	Values
Input dimension number	5
Output dimension number	10
Reservoir number	3
Reservoir neurons number	1000
Reservoir activation fn.	Tanh
Output layer activation fn.	Sigmoid
Update rate	0.9
Random seed	50
Regularization rate	1.0×10^{-6}

The experimental environment is tested in Windows 10 home version 64-bit operating system, Anaconda3 (64-bit), Python 3.7, 8.0 GB of memory, Intel (R) Core i3-4005U CPU @ 1.7 GHz.

6.3.2. The First Experiment in the Simulation Data. In order to fully verify the impact of Pearson and Gini coefficients on the classification algorithm, we have completed the method experiment in the training dataset that does not rely on these two filtering methods, a single filtering method and the combination of the two. The experimental results are shown in Figure 11.

From the experimental results in Figure 11, it is generally better to use the filtering technology than to not use the filtering technology. Whether it is a small data sample or a large data sample, the classification effect without the filtering technology is lower than that with the filtering technology.

In addition, using a single filtering method is not as good as using a combination of the two. For example, in the 160,000 training packets, when no filter method is used, the recognition accuracy of abnormal traffic is only 0.94; when only the Pearson index is used for filtering, the accuracy of the model is 0.95; when the Gini index is used for filtering, the accuracy of the model is 0.97; when the combination of Pearson index and Gini index is used for filtering, the accuracy of the model reaches 0.99.

6.3.3. The Second Experiment in the Simulation Data. Because the UNSW_NB15 dataset contains nine different types of abnormal attacks, the experiment first uses Pearson and Gini index to filter, then uses the ML-ESN training

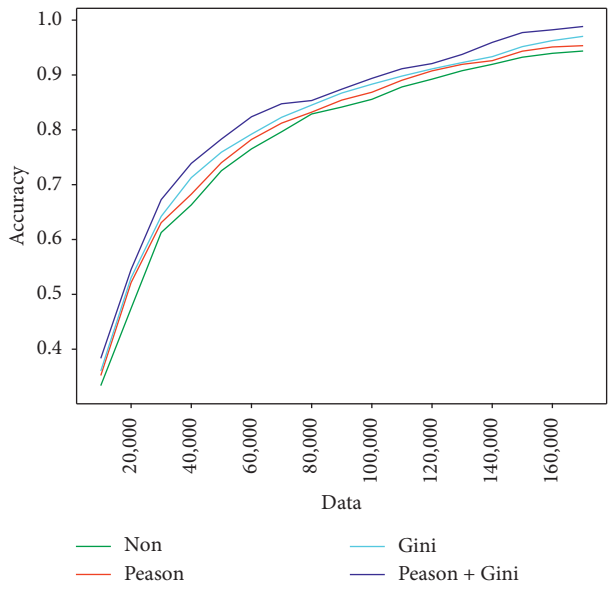

Figure 11: Classification effect of different filtering methods.

algorithm to learn, and then uses test data to verify the training model and obtains the test results of different types of attacks. The classification results of the nine types of abnormal attacks obtained are shown in Figure 12.

It can be known from the detection results in Figure 12 that it is completely feasible to use the ML-ESN network learning model to quickly classify anomalous network traffic attacks based on the combination of Pearson and Gini coefficients for network traffic feature filtering optimization.

Because we found that the detection results of accuracy, $F1$-score, and FPR are very good in the detection of all nine attack types. For example, in the Generic attack contact detection, the accuracy value is 0.98, the $F1$-score value is also 0.98, and the FPR value is very low, only 0.02; in the Shellcode and Worms attack type detection, both the accuracy and $F1$-score values reached 0.99. The FPR value is only 0.02. In addition, the detection rate of all nine attack types exceeds 0.94, and the $F1$-score value exceeds 0.96.

6.3.4. The Third Experiment in the Simulation Data. In order to fully verify the detection time efficiency and accuracy of the ML-ESN network model, this paper completed three comparative experiments. (1) Detecting the time consumption at different reservoir depths (2, 3, 4, and 5) and different numbers of neurons (500, 1000, and 2000), the results are shown in Figure 13(a); (2) detection accuracy at different reservoir depths (2, 3, 4, and 5) and different number of neurons (500, 1000, and 2000), the results are shown in Figure 13(b); and (3) comparing the time consumption and accuracy of the other three algorithms (BP, DecisionTree, and single-layer MSN) in the same case, the results are shown in Figure 13(c).

As can be seen from Figure 13(a), when the same dataset and the same model neuron are used, as the depth of the model reservoir increases, the model training time will also increase accordingly; for example, when the neuron is 1000, the time consumption of the reservoir depth of 5 is 21.1 ms, while the time consumption of the reservoir depth of 3 is only 11.6. In addition, at the same reservoir depth, the more the neurons in the model, the more training time the model consumes.

As can be seen from Figure 13(b), with the same dataset and the same model neurons, as the depth of the model reservoir increases, the training accuracy of the model will gradually increase at first; for example, when the reservoir depth is 3 and the neuron is 1000, the detection accuracy is 0.96; while the depth is 2, the neuron is 1000, and the detection accuracy is only 0.93. But when the neuron is increased to 5, the training accuracy of the model is reduced to 0.95.

The main reason for this phenomenon is that at the beginning, with the increase of training level, the training parameters of the model are gradually optimized, so the training accuracy is also constantly improving. However, when the depth of the model increases to 5, there is a certain overfitting phenomenon in the model, which leads to the decrease of the accuracy.

From the results of Figure 13(c), the overall performance of the proposed method is better than the other three methods. In terms of time performance, the decision tree method takes the least time, only 0.0013 seconds, and the BP method takes the most time, 0.0024. In addition, in terms of detection accuracy, the method in this paper is the highest, reaching 0.96, and the decision tree method is only 0.77. These results reflect that the method proposed in this paper has good detection ability for different attack types after model self-learning.

Step 5. In order to fully verify the correctness of the proposed method, this paper further tests the detection

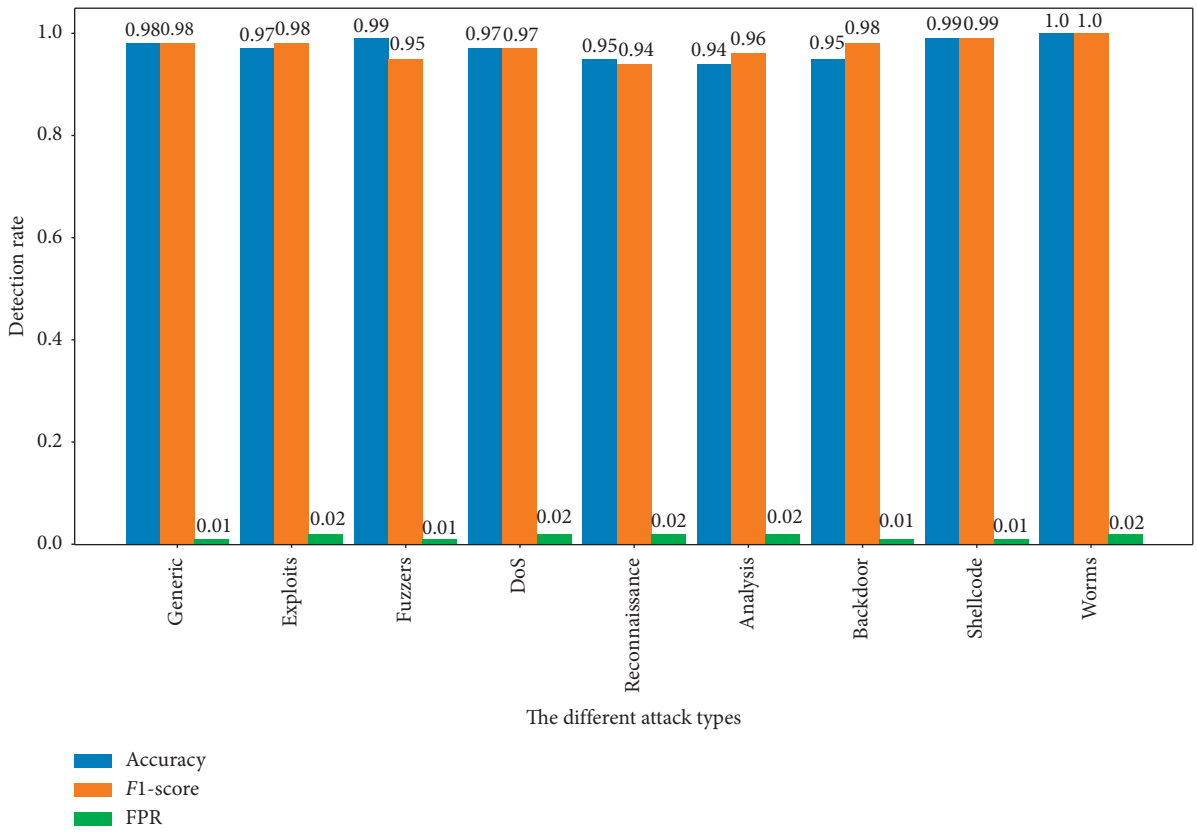

Figure 12: Classification results of the ML-ESN method.

performance of the UNSW_NB15 dataset by a variety of different classifiers.

6.3.5. The Fourth Experiment in the Simulation Data. The experiment first calculated the data distribution after Pearson and Gini coefficient filtering. The distribution of the first two statistical features is shown in Figure 14.

It can be seen from Figure 14 that most of the values of feature A and feature B are mainly concentrated at 5.0; especially for feature A, their values hardly exceed 6.0. In addition, a small part of the value of feature B is concentrated at 5 to 10, and only a few exceeded 10.

Secondly, this paper focuses on comparing simulation experiments with traditional machine learning methods at the same scale of datasets. These methods include GaussianNB [44], KNeighborsClassifier (KNN) [45], DecisionTree [46], and MLPClassifier [47].

This simulation experiment focuses on five test datasets of different scales, which are 5000, 20,000, 60,000, 120,000, and 160,000, respectively, and each dataset contains 9 different types of attack data. After repeated experiments, the detection results of the proposed method are compared with those of other algorithms, as shown in Figure 15.

From the experimental results in Figure 15, it can be seen that, in the small sample test dataset, the detection accuracy of traditional machine learning methods is relatively high. For example, in the 20,000 data, the GaussianNB, KNeighborsClassifier, and DecisionTree algorithms all achieved 100% success rates. However, in large-volume test data, the classification accuracy of traditional machine learning algorithms has dropped significantly, especially the GaussianNB algorithm, which has accuracy rates below 50%, and other algorithms are very close to 80%.

On the contrary, ML-ESN algorithm has a lower accuracy rate in small sample data. The phenomenon is that the smaller the number of samples, the lower the accuracy rate. However, when the test sample is increased to a certain size, the algorithm learns the sample repeatedly to find the optimal classification parameters, and the accuracy of the algorithm is gradually improved rapidly. For example, in the 120,000 dataset, the accuracy of the algorithm reached 96.75%, and in the 160,000, the accuracy reached 97.26%.

In the experiment, the reason for the poor classification effect of small samples is that the ML-ESN algorithm generally requires large-capacity data for self-learning to find the optimal balance point of the algorithm. When the number of samples is small, the algorithm may overfit and the overall performance will not be the best.

In order to further verify the performance of ML-ESN in large-scale AMI network flow, this paper selected a single-layer ESN [34], BP [6], and DecisionTree [46] methods for comparative experiments. The ML-ESN experiment parameters are set as in Table 4. The experiment used ROC (receiver operating characteristic curve) graphs to evaluate the experimental performance. ROC is a graph composed of FPR (false-positive rate) as the horizontal axis and TPR

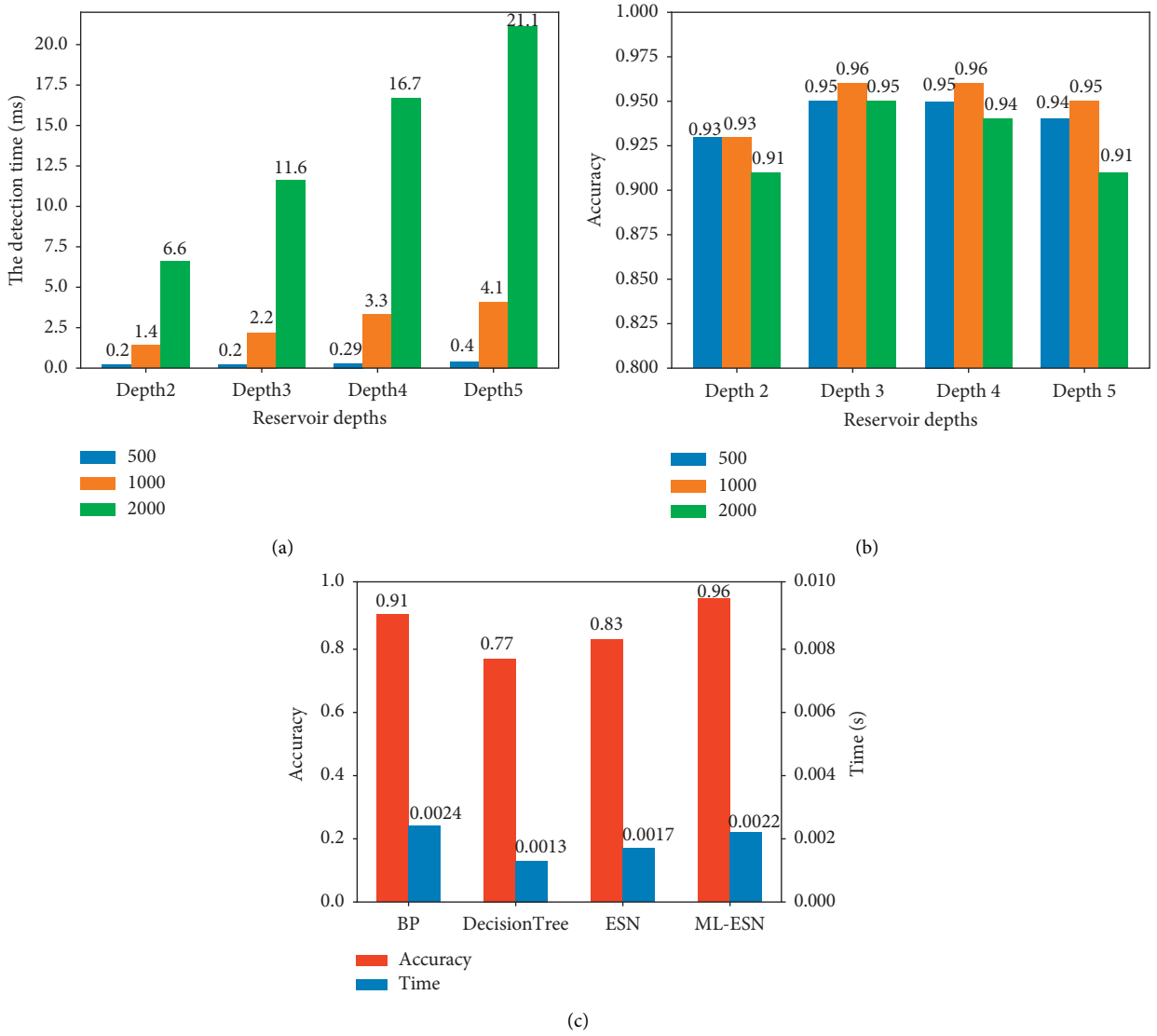

Figure 13: ML-ESN results at different reservoir depths.

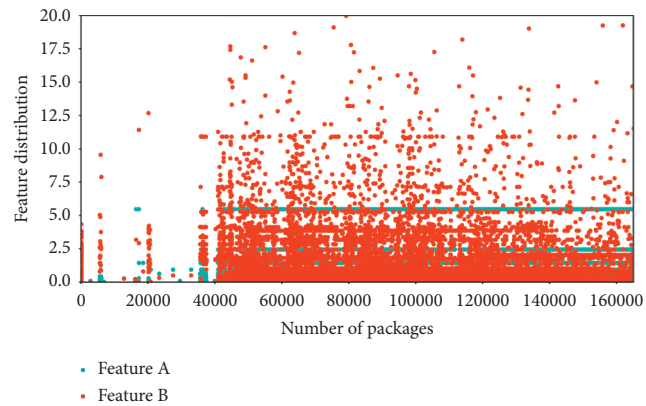

Figure 14: Distribution map of the first two statistical characteristics.

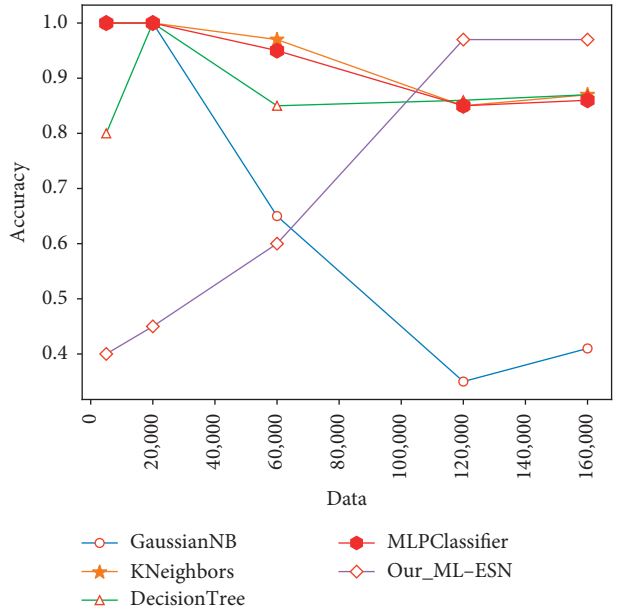

Figure 15: Detection results of different classification methods under different data sizes.

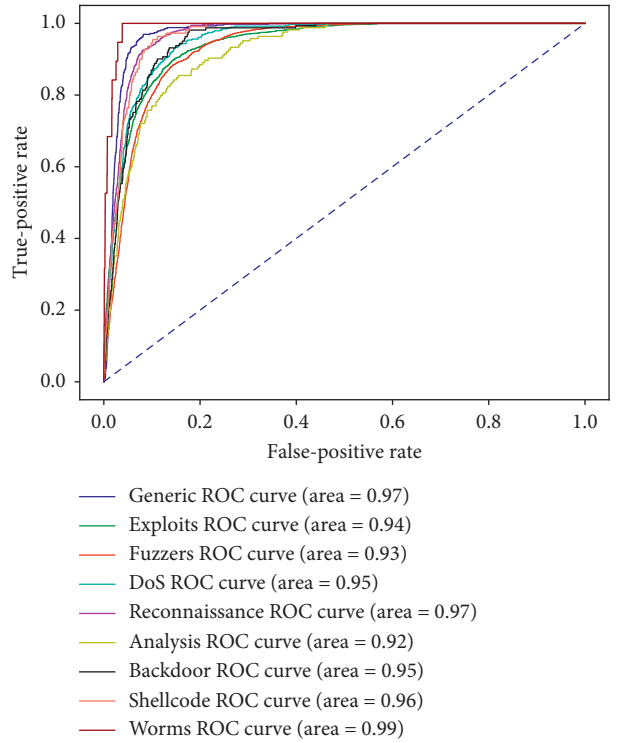

Figure 16: Classification ROC diagram of single-layer ESN algorithm.

(true-positive rate) as the vertical axis. Generally speaking, ROC chart uses AUC (area under ROC curve) to judge the model performance. The larger the AUC value, the better the model performance.

The ROC graphs of the four algorithms obtained in the experiment are shown in Figures 16–19, respectively.

From the experimental results in Figures 16–19, it can be seen that for the classification detection of 9 attack types, the optimized ML-ESN algorithm proposed in this paper is significantly better than the other three algorithms. For example, in the ML-ESN algorithm, the detection success rate of four attack types is 100%, and the detection rates for

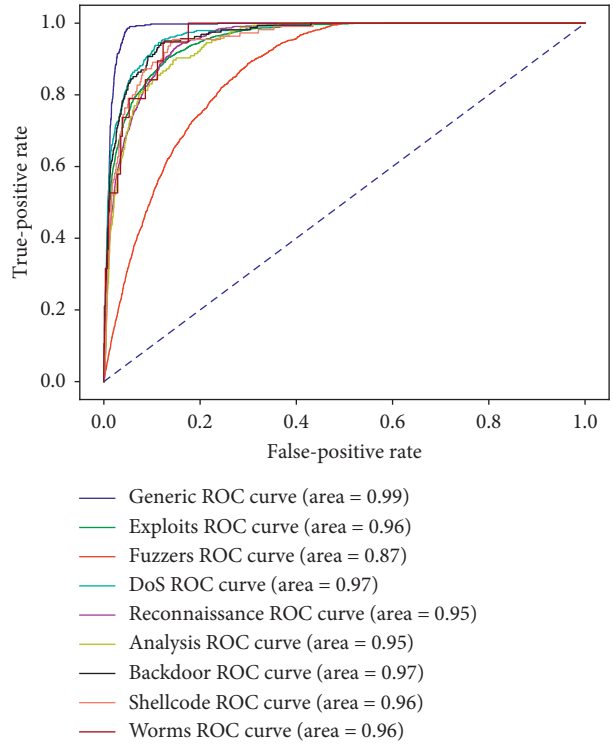

Figure 17: Classification ROC diagram of BP algorithm.

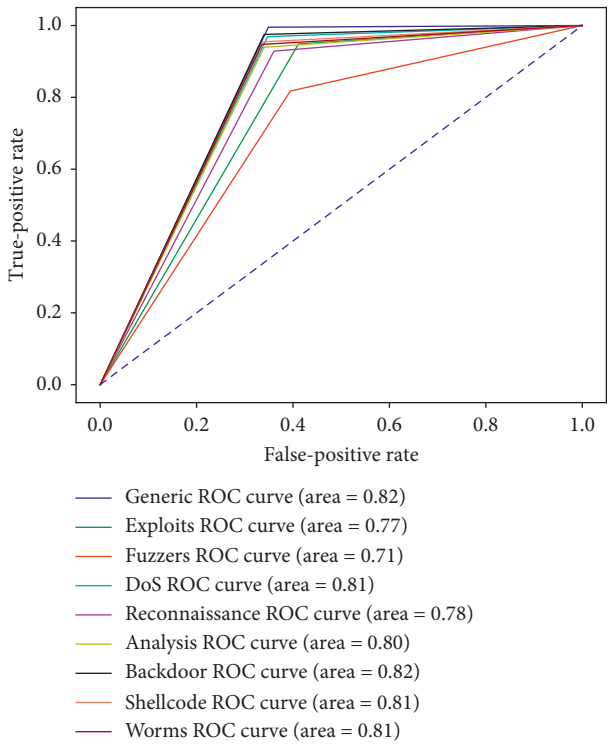

Figure 18: Classification ROC diagram of DecisionTree algorithm.

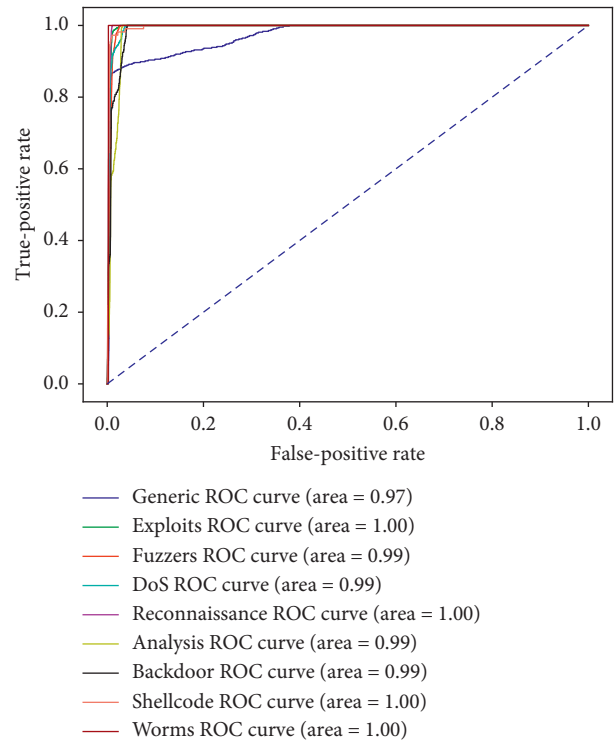

Figure 19: Classification ROC diagram of our ML-ESN algorithm.

other attack types are 99%. However, in the single-layer ESN algorithm, the best detection success rate is only 97%, and the general detection success rate is 94%. In the BP algorithm, the detection rate of the Fuzzy attack type is only 87%, and the false-positive rate exceeds 20%. In the traditional DecisionTree algorithm, its detection effect is the worst. Because the detection success rate is generally less than 80%, and the false-positive rate is close to 35%.

7. Conclusion

This article firstly analyzes the current situation of AMI network security research at home and abroad, elicits some problems in AMI network security, and introduces the contributions of existing researchers in AMI network security.

Secondly, in order to solve the problems of low accuracy and high false-positive rate of large-capacity network traffic data in the existing methods, an AMI traffic detection and classification algorithm based on ML-ESN deep learning was proposed.

The main contributions of this article are as follows: (1) establishing the AMI network streaming metadata standard; (2) the combination of Pearson and Gini coefficients is used to quickly solve the problem of extracting important features of network attacks from large-scale AMI network streams, which greatly saves model detection and training time; (3) using ML-ESN's powerful self-learning and storage and memory capabilities to accurately and quickly classify unknown and abnormal AMI network attacks; and (4) the proposed method was tested and verified in the simulation dataset. Test results show that this method has obvious advantages over single-layer ESN network, BP neural network, and other machine learning methods, with high detection accuracy and low time consumption.

Of course, there are still some issues that need attention and optimization in this paper. For example, how to establish AMI network streaming metadata standards that meet the requirements of different countries and different regions? At present, due to the complex structure of AMI and other electric power informatization networks, it is difficult to form a centralized and unified information collection source, so many enterprises have not really established a security monitoring platform for information fusion.

Therefore, the author of this article suggests that before analyzing the network flow, it is best to perform certain multicollection device fusion processing to improve the quality of the data itself, so as to better ensure the accuracy of model training and detection.

The main points of the next work in this paper are as follows: (1) long-term, large-scale test verification of the proposed method in the real AMI network flow, so as to find out the limitations of the method in the real environment; (2) carry out unsupervised ML-ESN AMI network traffic classification research to solve the problem of abnormal network attack feature extraction, analysis, and accurate detection; (3) further improve the model learning ability, such as learning improvement through parallel training, greatly reducing the learning time and classification time; (4) study the AMI network special protocol, and establish an optimized ML-ESN network traffic deep learning model that is more in line with the actual application of AMI, so as to apply it to actual industrial production.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the Key Scientific and Technological Project of "Research and Application of Key Technologies for Network Security Situational Awareness of Electric Power Monitoring System (no. ZDKJXM20170002)" of China Southern Power Grid Corporation, the project of "Practical Innovation and Enhancement of Entrepreneurial Ability (no. SJCX201970)" for Professional Degree Postgraduates of Changsha University of Technology, and Open Fund Project of Hunan Provincial Key Laboratory of Processing of Big Data on Transportation (no. A1605).

References

[1] A. Maamar and K. Benahmed, "A hybrid model for anomalies detection in AMI system combining k-means clustering and deep neural network," *Computers, Materials & Continua*, vol. 60, no. 1, pp. 15–39, 2019.

[2] Y. Liu, *Safety Protection Technology of Electric Energy Measurement, Collection and Billing*, China Electric Power Press, Beijing, China, 2014.

[3] B. M. Nasim, M. Jelena, B. M. Vojislav, and K. Hamzeh, "A framework for intrusion detection system in advanced metering infrastructure," *Security and Communication Networks*, vol. 7, no. 1, pp. 195–205, 2014.

[4] H. Ren, Z. Ye, and Z. Li, "Anomaly detection based on a dynamic Markov model," *Information Sciences*, vol. 411, pp. 52–65, 2017.

[5] F. Fathnia and D. B. M. H. Javidi, "Detection of anomalies in smart meter data: a density-based approach," in *Proceedings of the 2017 Smart Grid Conference (SGC)*, pp. 1–6, Tehran, Iran, 2017.

[6] Z. Y. Wang, G. J. Gong, and Y. F. Wen, "Anomaly diagnosis analysis for running meter based on BP neural network," in *Proceedings of the 2016 International Conference on Communications, Information Management and Network Security*, Gold Coast, Australia, 2016.

[7] M. Stephen, H. Brett, Z. Saman, and B. Robin, "AMIDS: a multi-sensor energy theft detection framework for advanced metering infrastructures," *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 7, pp. 1319–1330, 2013.

[8] Y. Chen, J. Tao, Q. Zhang et al., "Saliency detection via improved hierarchical principle component analysis method," *Wireless Communications and Mobile Computing*, vol. 2020, Article ID 8822777, 12 pages, 2020.

[9] Y. Mo, H. J. Kim, K. Brancik et al., "Cyber–physical security of a smart grid infrastructure," *Proceedings of the IEEE*, vol. 100, no. 1, pp. 195–209, 2012.

[10] The AMI network engineering task Force (AMI-SEC) ," 2020, http://osgug.ucaiug.org/utilisec/amisec/default.aspx.

[11] Y. Park, D. M. Nicol, H. Zhu et al., "Prevention of malware propagation in AMI," in *Proceedings of the IEEE International Conference on Smart Grid Communications*, pp. 474–479, Vancouver, Canada, 2013.

[12] P. Jokar, N. Arianpoo, and V. C. M. Leung, "Electricity theft detection in AMI using customers' consumption patterns," *IEEE Transactions on Smart Grid*, vol. 7, no. 1, pp. 216–226, 2016.

[13] Q. R. Zhang, M. Zhang, T. H. Chen et al., "Electricity theft detection using generative models," in *Proceedings of the 2018 IEEE 30th International Conference on Tools with Artificial Intelligence (ICTAI)*, Volos, Greece, 2018.

[14] N. Y. Jiang, "*Anomaly intrusion detection method based on AMI*," M.S. thesis, Southeast University, Dhaka, Bangladesh, 2018, in Chinese.

[15] S. Neetesh, J. C. Bong, and G. Santiago, "Secure and privacy-preserving concentration of metering data in AMI networks," in *Proceedings of the 2017 IEEE International Conference on Communications (ICC)*, Paris, France, 2017.

[16] C. Euijin, P. Younghee, and S. Huzefa, "Identifying malicious metering data in advanced metering infrastructure," in *Proceedings of the 2014 IEEE 8th International Symposium on Service Oriented System Engineering*, pp. 490–495, Oxford, UK, 2014.

[17] P. Yi, T. Zhu, Q. Q. Zhang, Y. Wu, and J. H. Li, "Puppet attack: a denial of service attack in advanced metering infrastructure network," *Journal of Network & Computer Applications*, vol. 59, pp. 1029–1034, 2014.

[18] A. Satin and P. Bernardi, "Impact of distributed denial-of-service attack on advanced metering infrastructure," *Wireless Personal Communications*, vol. 83, no. 3, pp. 1–15, 2015.

[19] C. Y. Li, X. P. Wang, M. Tian, and X. D. Feng, "AMI research on abnormal power consumption detection in the environment," *Computer Simulation*, vol. 35, no. 8, pp. 66–70, 2018.

[20] A. A. A. Fadwa and A. Zeyar, "Real-time anomaly-based distributed intrusion detection systems for advanced metering infrastructure utilizing stream data mining," in *Proceedings of the 2015 International Conference on Smart Grid and Clean Energy Technologies*, pp. 148–153, Chengdu, China, 2015.

[21] M. A. Faisal and E. T. Aigng, "Securing advanced metering infrastructure using intrusion detection system with data stream mining," in *Proceedings of the Pacific Asia Conference on Intelligence and Security Informatics*, IEEE, Jeju Island, Korea, pp. 96–111, 2016.

[22] K. Song, P. Kim, S. Rajasekaran, and V. Tyagi, "Artificial immune system (AIS) based intrusion detection system (IDS) for smart grid advanced metering infrastructure (AMI) networks," 2018, https://vtechworks.lib.vt.edu/handle/10919/83203.

[23] A. Saad and N. Sisworahardjo, "Data analytics-based anomaly detection in smart distribution network," in *Proceedings of the 2017 International Conference on High Voltage Engineering and Power Systems (ICHVEPS)*, IEEE, Bali, IndonesiaIEEE, Bali, Indonesia, 2017.

[24] R. Berthier, W. H. Sanders, and H. Khurana, "Intrusion detection for advanced metering infrastructures: requirements and architectural directions," in *Proceedings of the IEEE International Conference on Smart Grid Communications*, IEEE, Dresden, Germany, pp. 350–355, 2017.

[25] V. B. Krishna, G. A. Weaver, and W. H. Sanders, "PCA-based method for detecting integrity attacks on advanced metering infrastructure," in *Proceedings of the 2015 International Conference on Quantitative Evaluation of Systems*, pp. 70–85, Madrid, Spain, 2015.

[26] G. Fernandes, J. J. P. C. Rodrigues, L. F. Carvalho, J. F. Al-Muhtadi, and M. L. Proença, "A comprehensive survey on network anomaly detection," *Telecommunication Systems*, vol. 70, no. 3, pp. 447–489, 2019.

[27] W. Wang, Y. Sheng, J. Wang et al., "HAST-IDS: learning hierarchical spatial-temporal features using deep neural networks to improve intrusion detection," *IEEE Access*, vol. 6, pp. 1792–1806, 2018.

[28] N. Gao, L. Gao, Y. He et al., "A lightweight intrusion detection model based on autoencoder network with feature reduction," *Acta Electronica Sinica*, vol. 45, no. 3, pp. 730–739, 2017, in Chinese.

[29] M. Yousefi-Azar, V. Varadharajan, L. Hamey, and U. Tupalula, "Autoencoder-based feature learning for cyber security applications," in *Proceedings of the 2017 International Joint Conference on Neural Networks (IJCNN), IEEE, Neural Networks*, pp. 3854–3861, Anchorage, AK, USA, 2017.

[30] Y. Wang, H. Zhou, H. Feng et al., "Network traffic classification method basing on CNN," *Journal on Communications*, vol. 39, no. 1, pp. 14–23, 2018, in Chinese.

[31] S. Kaur and M. Singh, "Hybrid intrusion detection and signature generation using deep recurrent neural networks," *Neural Computing and Applications*, vol. 32, no. 12, pp. 7859–7877, 2019.

[32] H. Jaeger, M. Lukoševičius, D. Popovici, and U. Siewert, "Optimization and applications of echo state networks with leaky- integrator neurons," *Neural Networks*, vol. 20, no. 3, pp. 335–352, 2007.

[33] S. Saravanakumar and R. Dharani, "Implementation of echo state network for intrusion detection," *International Journal of Advanced Research in Computer Science Engineering and Information Technology*, vol. 4, no. 2, pp. 375–385, 2015.

[34] Y. Kalpana, S. Purushothaman, and R. Rajeswari, "Implementation of echo state neural network and radial basis function network for intrusion detection," *Data Mining and Knowledge Engineering*, vol. 5, no. 9, pp. 366–373, 2013.

[35] X. X. Liu, "*Research on the network security mechanism of smart grid AMI*," M.S. thesis, National University of Defense Science and Technology, Changsha, China, 2014, in Chinese.

[36] Y. Wang, "*Research on network behavior analysis and identification technology of malicious code*," M.S. thesis, Xi'an University of Electronic Science and Technology, Xi'an, China, 2017, in Chinese.

[37] A. Moore, D. Zuev, and M. Crogan, "*Discriminators for use in flow-based classification*," M.S. thesis, Department of Computer Science, Queen Mary and Westfield College, London, UK, 2005.

[38] Data standardization, Baidu Encyclopedia," 2020, https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E6%A0%87%E5%87%86%E5%8C%96/4132085?fr=aladdin.

[39] H. Li, *Statistical Learning Methods*, Tsinghua University Press, Beijing, China, 2018.

[40] Z. K. Malik, A. Hussain, and Q. J. Wu, "Multilayered echo state machine: a novel architecture and algorithm," *IEEE Transactions on Cybernetics*, vol. 47, no. 4, pp. 946–959, 2017.

[41] C. Naima, A. Boudour, and M. A. Adel, "Hierarchical bi-level multi-objective evolution of single- and multi-layer echo state network autoencoders for data representation,"

2020, https://arxiv.org/ftp/arxiv/papers/1806/1806.01016.pdf.

[42] M. Nour and S. Jill, "UNSW-NB15: a comprehensive data set for network intrusion detection systems," in *Proceedings of the 2015 Military Communications and Information Systems Conference (MilCIS)*, pp. 1–6, Canberra, Australia, 2015.

[43] UNSW-NB15 dataset," 2020, https://www.unsw.adfa.edu.au/unsw-canberra-cyber/cybersecurity/ADFA-NB15-Datasets/.

[44] N. B. Azzouna and F. Guillemin, "Analysis of ADSL traffic on an IP backbone link," in *Proceedings of the GLOBECOM'03. IEEE Global Telecommunications Conference (IEEE Cat. No. 03CH37489)*, IEEE, San Francisco, CA, USAIEEE, San Francisco, CA, USA, 2004.

[45] P. Cunningham and S. J. Delany, "K-nearest neighbour classifiers," *Multiple Classifier System*, vol. 34, pp. 1–17, 2007.

[46] K. J. Manas, R. S. Subhransu, and T. Lokanath, "Decision tree-induced fuzzy rule-based differential relaying for transmission line including unified power flow controller and wind-farms," *IET Generation Transmission & Distribution*, vol. 8, no. 12, pp. 2144–2152, 2014.

[47] K. J. Manas, R. S. Subhransu, and T. Lokanath, "Decision tree-induced fuzzy rule-based differential relaying for transmission line including unified power flow controller and wind-farms," *IET Generation Transmission & Distribution*, vol. 8, no. 12, pp. 2144–2152.

[48] L. V. Efferen and A. M. T. Ali-Eldin, "A multi-layer perceptron approach for flow-based anomaly detection," in *Proceedings of the 2017 International Symposium on Networks, Computers and Communications (ISNCC)*, IEEE, Marrakech, MoroccoIEEE, Marrakech, Morocco, 2017.