IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

SPECIAL SECTION ON CYBER-PHYSICAL SYSTEMS

Received December 29, 2018, accepted January 25, 2019, date of publication January 31, 2019, date of current version March 7, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2896129

Safety Risk Monitoring of Cyber-Physical Power Systems Based on Ensemble Learning Algorithm

QIANMU LI[1,2], SHUNMEI MENG[2,3], SAINAN ZHANG[2], MING WU[2], JING ZHANG[2], MILAD TALEBY AHVANOOEY[2], AND MUHAMMAD SHAMROOZ ASLAM[4]

[1]Intelligent Manufacturing Department, Wuyi University, Jiangmen 529020, China
[2]School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, China
[3]State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210023, China
[4]School of Automation, Nanjing University of Science and Technology, Nanjing 210094, China

Corresponding author: Shunmei Meng (mengshunmei@njust.edu.cn)

This work was supported in part by the Fundamental Research Funds for the Central Universities under Grant 30918012204, in part by the Jiangsu Province Key Research and Development Program through the Social Development Project under Grant BE2017739, in part by the Jiangsu Province Key Research and Development Program under Grant BE2017100, in part by the 2018 Jiangsu Province Major Technical Research Project—Information Security Simulation System, in part by the National Natural Science Foundation of China under Grant 61702264 and Grant 91846104, in part by the Open Research Project of the State Key Laboratory of Novel Software Technology (Nanjing University) under Grant KFKT2017B07.

ABSTRACT The traditional security risk monitoring technology cannot adapt to cyber-physical power systems (CPPS) concerning evaluation criteria, real-time monitoring, and technical reliability. The aim of this paper is to propose and implement a log analysis architecture for CPPS to detect the log anomalies, which introduces the distributed streaming processing mechanism. The processing mechanism can train the network protocol feature database precisely over the big data platform, which improves the efficiency of the network in terms of log anomaly detection. Moreover, we propose an ensemble prediction algorithm based on time series (EPABT) considering the characteristics of the statistical log analysis to predict abnormal features during the network traffic analysis. We then present a new asymmetric error cost (AEC) evaluation criterion to meet the characteristics of CPPS. The experimental results demonstrate that the EPABT provides an efficient tool for detecting the accuracy and reliability of abnormal situation prediction as compared with the several state-of-the-art algorithms. Meanwhile, the AEC can effectively evaluate the differences in the cost between the high and low prediction results. To the best of our knowledge, these two algorithms provide strong support for the practical application of power industrial network security risk monitoring.

INDEX TERMS Ensemble learning, security risk monitoring, big data, log analysis, evaluation criteria.

I. INTRODUCTION

In recent years, with the latest development of enterprise network technology and the popularity of Internet applications, a single computer system in the traditional sense has been unable to meet the current ubiquitous network application requirements. In this regard, new network technologies such as cloud computing and the Internet of Things have gradually penetrated various fields of industrial control. The Cyber-Physical Power Systems (CPPS) is a typical representative. The emergence of network anomalies will become inevitable because more and more devices are connected to the network. Therefore, to improve the reliability of systems has always been a focus for the researchers. In terms of processing methods of system exceptions, we divided this portion into two parts, and one is system recovery and other is anomaly monitoring. The analysis of anomaly monitoring is based upon the system history and current status. It helps to determine whether or not the system is abnormal, which in turn helps the system avoid anomalies and take steps to recover as soon as possible [1].

Anomaly monitoring is based upon the premise while abnormal behavior is different from normal user behavior, and this difference between them can be qualitatively or quantitatively described. The anomaly monitoring uses the normal samples of the network log to train a monitoring model, and to distinguish the unknown behavior by identifying the normal user behavior. This method does not require describing the

The associate editor coordinating the review of this manuscript and approving it for publication was Theodore E. Simos.

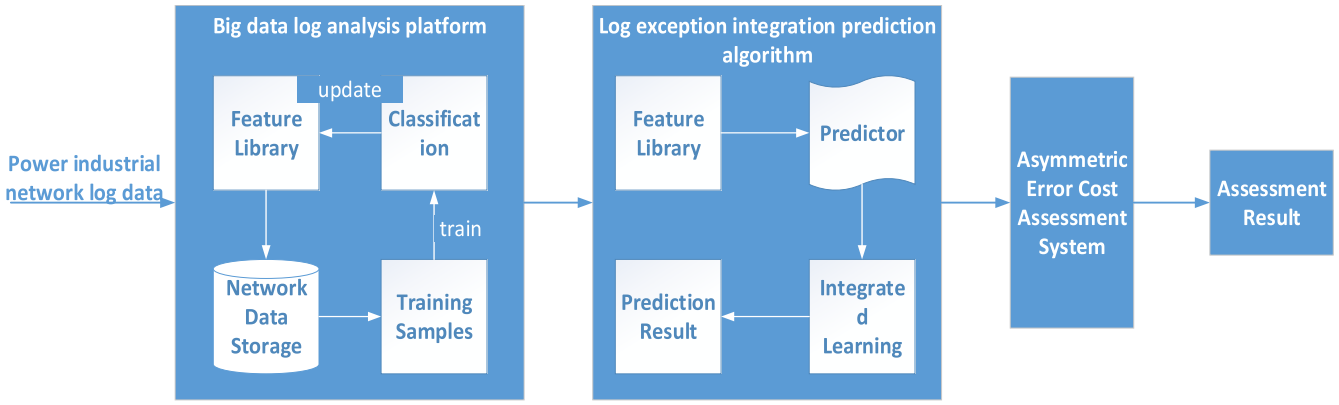

FIGURE 1. Safety risk monitoring framework of the CPPS based on ensemble learning algorithm.

feature of a web attack, and it can also be used to monitor unknown intrusion behavior. Most of the anomaly monitoring uses data mining and other technologies to achieve the best monitoring by learning the log data. At present, there is not too much research in this field of CPPS. We can only focus on some research results in the field of public safety.

Kruegel *et al.* [2] process and mines according to the request parameters input when the user requests the URL recorded into the log and performs sufficient learning to obtain the mode of normal access, and then conducts security monitoring and analysis according to the learned mode. However, it has only the ability to analyze the request parameters in the log but does not consider other attributes. Reference [3] uses a reverse proxy to perform anomaly analysis on HTTP requests. This method requires modification of the service source code and inserts an additional library to cause all SQL functions to point to the library. This method can analyze HTTP request and SQL query results at the same time, but this method requires modification of the source code, adding additional risk. In [4], they analyze the anomaly of the session by segmenting the log and using Bayesian parameter estimation. However, it only evaluates the session and lacks consideration of the request parameter properties and network topology in the access. Gao and Wu [5] mainly uses Web-based security detection by combining the unsupervised learning method with the valid rule detection model. These authors generated Web-based site architecture from the properties of the access structure and the process of learning from the Weblog processing analysis. The rule base performs security detection according to the mentioned rules, but this method does not consider the session attribute in the log access, and only considers the type of the parameter when considering the request parameter, rather than related characteristics such as the parameter length.

It is easy to find out that the research in the field of public security has indeed achieved a lot of results, and the methods are different. Although there are some shortcomings, these methods can achieve anomaly monitoring to a certain extent. However, to tackle the problem of anomaly monitoring for industrial control networks in hydropower generation, these methods generally have some shortcomings in dealing with data volume, real-time, accuracy, and reliability. At the same time, CPPS has its isolation measures to take timely measures. Also, the hydraulic control industrial control network has asymmetry for the cost of over-predicting and under-predicting. Therefore, the evaluation standard of traditional methods cannot meet the requirements for special occasions of CPPS.

To address the above problems, we propose a novel method for the CPPS log analysis. Figure 1 shows the main framework of this paper's research. The contributions of this paper are summarized as follows:

- Firstly, this paper proposes an architecture based on the big data platform for CPPS log anomaly detection architecture. The system adopts distributed streaming processing mechanism to achieve real-time monitoring, and utilizes the ability of distributed storage, data calculation, and analysis of big data platform to realize the distributed storage of network data and more accurately train the network data protocol feature database. The advantages of the big data platform improve the efficiency of network log anomaly detection, and have the ability to process and monitor the massive amounts of abnormal data. It is scalable, configurable, and has linear scalability in computing power.
- Secondly, this paper proposes an integrated forecasting algorithm based on time series for anomaly quantity of CPPS log in hydropower generation. Compared with other related algorithms, this algorithm maintains a high level of accuracy and reliability, and is more suitable for the security risk monitoring of hydropower generation industrial control networks.
- Finally, due to the asymmetry of the cost prediction of hydropower generation industrial control network logs, this paper proposes an asymmetric error cost evaluation standard, which is more suitable for the actual monitoring of CPPS security risk and application.

II. LOG ANALYSIS BIG DATA PLATFORM ARCHITECTURE AND ITS IMPLEMENTATION FOR CPPS

Log files are those files which record the system, application, and user behavior and system events according to

specific rules. This type of record can be used as an indispensable basis for assessing the effectiveness of cyber-security policy implementation and the reliability of the security defense system. Therefore, the log has become an essential tool for daily maintenance, and can even prevent the cyber-crime [6].

A. LOG FILE AND ITS CHARACTERISTICS

The logs in the network environment can be divided into three main types according to the source of the log generation:

1) OPERATING SYSTEM LOG

Such as UNIX operating system log data, Linux operating system log data, and Windows operating system log data.

2) NETWORK DEVICE LOG

Such as log data of routing and switching devices, firewalls, etc.

3) APPLICATION SERVICE LOG

Such as various web application applications.

According to the data storage form of the log, it can be further divided into two general types:

1) UNSTRUCTURED LOG DATA

The log data of Linux logs, Apache server logs, Hadoop logs, etc. are all recorded in a plain text log file. Each log or record is stored in the log file as a text message or short text.

2) STRUCTURED OR SEMI-STRUCTURED LOG EVENT DATA

Such as Windows Event Logs, database history query log, etc.

In general, each log is divided into three parts [7], [8]: Date, Java Class and Messages. The content of the message section is defined by Hadoop developers inside the Hadoop source code. The message section describes which task of the current program is performing or what error exceptions it encounters. In addition to Windows Event Logs, many systems monitoring the systems, such as IBM Tivoli Monitoring, log event data generated by HP Open View can also be stored in a relational database in a semi-structured form. The difference is that the professional monitoring system can customize various kinds of complex monitoring information through administrators to generate various log data.

Through the analysis of various log records themselves, the following main features are summarized as:

a) Diversity in format
b) Poor readability
c) Large data size
d) Difficult to obtain
e) Inevitable correlation between different logs
f) Easily tampering

Although the characteristics of the log are not good for log analysis, so the role of log analysis is not negligible. Logs are used as an auxiliary management tool for computer systems, and their main performances are as follows:

a) Auditing user behavior and monitoring malicious behavior
b) Detection of intrusion behavior
c) Effective monitoring of system resources
d) Auxiliary to the recovery of the system
e) Auxiliary to the assessment of system losses
f) Forensics of computer crimes
g) Auxiliary to the generation of the investigation report

B. LOG ANALYSIS PROCESSING

Because the log is not only massive in data, format, and storage are not uniform, and there are links between different logs, which makes more difficult for the analysis. If the network administrator can not only understand the meaning of this information, but also know how to analyze and use this information, then the value of the log for network security management will be incalculable.

However, with the complexity of the computer system architecture and the ever-expanding scale, the maintenance of such a large information system cannot be accomplished by simple manpower, especially for large distributed systems. Data mining can help to solve such kind of problems.

In general, the analysis of logs is normally divided into three steps:

1) LOG COLLECTION AND STORAGE

The collection of log data generally includes four aspects: collection of log data, preprocessing of log data, generation of log records, and storage of log data.

a: COLLECTION OF LOG DATA

In this scenario, we have to perform three functions: First, it is accepted by the standard protocol. The log analysis function should be able to receive log data sent from the log data source based on Syslog mode, SNMP Trap mode, Ftp mode or another standard protocol mode. Second is proxy mode collection. The log analysis function should have the ability to collect log data from the log data source through log proxy mode; the third one is to import log files. The log analysis function should be able to smoothly import log files in a common format.

b: PREPROCESSING OF LOG DATA:

It mainly includes two aspects of work: First, data screening. The log analysis function should be able to filter out the collected log data based on the established policy, selectively generate the optimized log records; the second is data conversion. The log analysis function should be able to convert raw log data of various formats into a unified data format, and when converting, it should ensure that key data items cannot be lost to provide the key features of the original log data.

c: GENERATION OF LOG RECORDS

The log analysis function should be able to generate the corresponding log records. Logging content should be understood by the administrator and generally includes the following information: the date and time of the event occurred including each description of an event like body, object, description, type, level, and the IP address with the name of the log data source.

d: STORAGE OF LOG DATA

There are two aspects need to be considered: On the one hand, security protection should be considered. Because the log data is easily falsified, and the falsified log data has a significant impact on the analysis results, the log analysis function should take necessary security mechanisms to protect the log records from unauthorized reading, deletion or modification. To ensure the security and authenticity of the log data; on the other hand, preventing the loss of log records should be considered into account. Sometimes, there is no external interference or damage from the system, due to various reasons; some negligence of the log storage itself can also easy to cause the loss of log data, thereby affecting the integrity and security of the log. Therefore, the log analysis function should also consider the following three aspects to ensure the security of the log data further. First, the log records should be stored in the non-volatile storage medium; secondly, when the storage capacity of the log records reaches the threshold, the system can issue Alarm information. Finally, before the log storage space is exhausted, the system can actively adopt the automatic dump technology to back up the log records to other storage spaces automatically.

2) LOG ANALYSIS AND PROCESSING

The analysis of the log records mainly completes the following seven aspects:

a) Discrimination analysis of events
b) Grading analysis of events
c) Statistical analysis of events
d) Potential hazard analysis of the event
e) Anomalous behavior analysis of events
f) Correlation attribute analysis of events
g) Mining analysis of log records

The processing of log records mainly includes two aspects:

a) Data integration capabilities
b) Data splitting ability

3) LOG DISPLAY AND ALARM

It mainly includes four aspects:

a) Query log
b) Output statistics report
c) Generate analysis report
d) Abnormal alarm

In practice, log analysis also faces many difficulties, which may result in the analysis results not genuinely reflecting the original intention of the log, or the desired results. The reasons for these situations are mainly the following:

a) The amount of log data is huge
b) There is not enough data
c) The type of logging is too complicated
d) The possibility of false alarms
e) Excessive duplicate data

C. LOGICAL STRUCTURE

Many data mining analysis algorithms are built on structured events. However, most computing systems only generate text logs containing many details, which are semi-structured or unstructured text. Discretized or structured events are more visually visible than text log messages and are more easily used by experts for in-depth research. Many visualization toolkits have been developed to macroscopically demonstrate system behavior on a large set of discrete events. Therefore, before we do this analysis, and further need to convert the text log into discrete or structured events.

Before introducing the log-structured approach, to illustrate that structured events are more conducive to data mining analysis, let's take an example for converting a text log into an event.

TABLE 1. An instance of a FileZilla log.

Serial number	Text log
s1	2016-12-06 00:22:35 Command: put "E:/Tomcat/apps/index.html" "/disk/…
s2	2016-12-06 00:22:36 Status: File transfer successful, transferred 246 bytes…
s3	2016-12-06 00:22:37 Command: cd "/disk/storage005/users/lt…
s4	2016-12-06 00:22:38 Command: cd "/disk/storage005/users/lt…
s5	2016-12-06 00:22:39 Command: cd "/disk/storage005/users/lt…
s6	2016-12-06 00:22:40 Command: put "E:/Tomcat/apps/record1.html" "/disk/…
s7	2016-12-06 00:22:40 Status: Listing directory /disk/storage005/users/lt...· ·
…	…

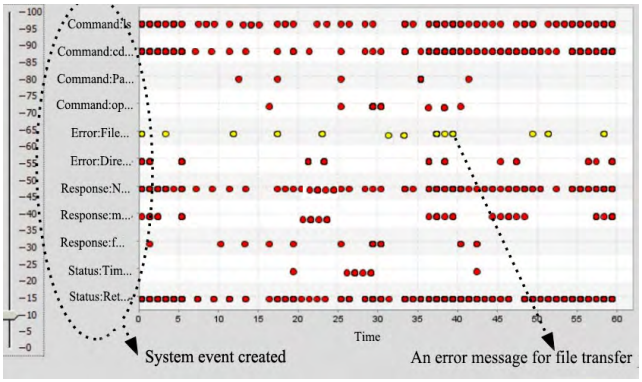

FIGURE 2. Event timeline for FileZilla log instance.

Table 1 shows a fragment of the SFTP (Simple File Transfer Protocol) log file collected from FileZilla. To analyze the behavior of FileZilla, we need to convert the original log message into some different types of events. Figure 2 shows the corresponding event timeline generated by analyzing log messages. The event timeline provides a convenient platform for us to understand log behavior and discover log patterns.

The transition from log messages to events provides the possibility to describe the semantics of the log data in a standardized manner and improves the ability to correlate from logs to multicomponent.

Due to the heterogeneous nature of modern systems, the log generation mechanism leads to the emergence of different formats and content for individual components. Each component generates data using its format and content. The diversity of journaling languages adds to the difficulty of parsing events and errors reported by multiple

TABLE 2. A summary of three types of solutions.

Type	Advantage	Disadvantage	Application scenario
log parser-based methods	Very accurate	Users need to know the system log. It is difficult to adapt to multiple system logs in different formats. Need manpower to develop log parser software	Applications that need to accurately generate system logs, such as alarm detection systems and monitoring systems
classification-based methods	Accurate and suitable for multiple system logs	The user is required to provide training log data. Marking log data with domain experts can be expensive and time consuming	Application tagged training log data readily available
cluster-based methods	Does not require a lot of manpower and is suitable for multiple system logs	Not very accurate	Application that can tolerate some errors or noise events

products and components. To support automated problem determination, it is necessary to compile semantics such as ''startup'' in a log-independent and system-independent manner. Converting log messages into events enable the consistency between similar fields and improve the ability to establish a correlation between multiple logs. By organizing the information in the log file into a common set of semantic events (also called ''scenarios'' or ''category''), that is, adding a semantic scenario type to a message, the converted representation can be standard describes the semantics of log data and the initial connection from syntax to semantics.

Three types of solutions can convert text log sets into system events, log parser-based methods, classification-based methods, and cluster-based methods. A summary of the three ways is given in Table 2.

The most straightforward solution is to use a log parser. Since the logged user may be very familiar with the contents of each log message, they can implement a simple text parser and accurately extract all required system event information from the log. However, for large complex systems, the log parser is not easy to implement. Implementing a log parser for each type of system log is not an efficient option. Although there are some common log formats, how to adapt a log parser to different types of logs is still a problem to be solved. In many event mining applications, since many mining algorithms are designed to discover unknown relationships between different event types, the user only needs to identify the event type, and it is not necessary to extract specific information of a system event, such as system metric information. Therefore, converting a text message into an event is equivalent to finding the type of log message; that is, the conversion problem becomes a text classification problem. Many classification algorithms, such as Support Vector Machine (SVM), can be used to solve text classification problems. The main disadvantage of these classification-based methods is that the classification algorithm requires the pre-marked training data to the user. In other words, the user must have a set of marked log messages ready. For a large complex system, only experts in this field can do this, so this is time-consuming and costly.

The cluster-based approach does not require the user to prepare the marked training data. It can infer the type of event from the log information itself. Although the type of event it concludes may not be as accurate as the classification-based or log-parser-based approach, the accuracy of the prediction is acceptable for both event mining algorithms and user manual analysis.

In the traditional field of natural language processing, this task is a typical information extraction. Information extraction is based on rules and also based on statistical models. The rule-based approach is simpler and more practical. In academia, more research is based on statistical models, such as Conditional Random Field [9] and so on. For Syslog data, the format and change are very small compared to human language. Whether it is based on rules or based on Conditional Random Field, it can achieve high precision and is better than general natural language processing. The result is more reliable.

In the absence of training data, unstructured log data conversion can also be done by analyzing the source code generated by its logs [10]. Much software now has specialized for log generation toolkits, such as ''java.util.logging'' and Apache ''Log4J'', to standardize the logs. In other words, the source code of these logs does not change too much. The function code that generates the log can be found by string matching in the source code file. Then, similar syntax processing is performed on the formal language, and the composition of the constant string and variables in the log can be extracted.

However, the source code for many commercial software systems is not available. Even if it is available, it may not be applicable, because a software system may use a myriad of third-party software libraries or toolkits. Third, the source code of large software systems is too large. In addition to

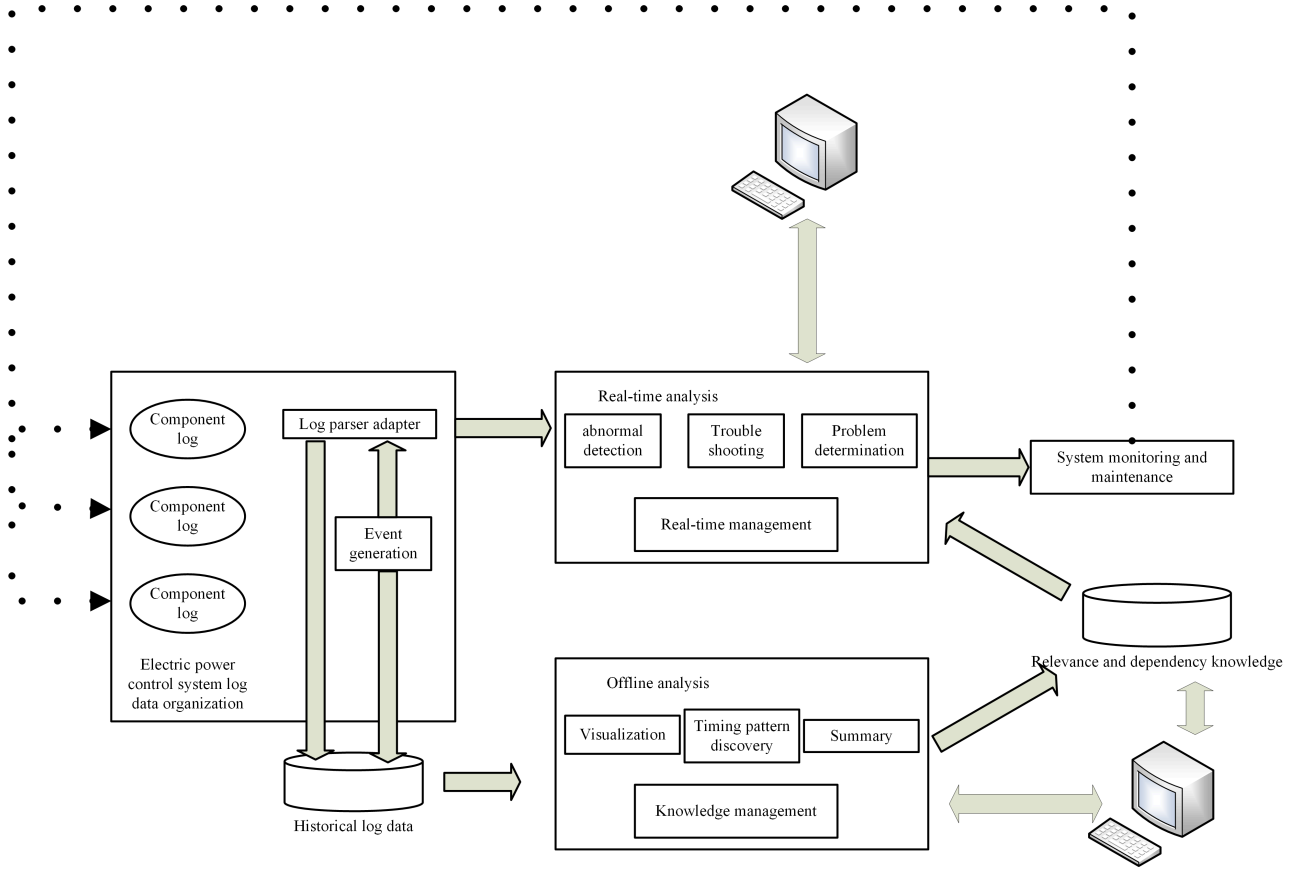

FIGURE 3. Water resources power generation industrial control system log analysis architecture.

software supply developers, any other organization to organize and analyze is a vast project. In [11], a clustering algorithm is proposed to convert text logs into various types of events. Among them, the distance of each log text is obtained by simple word-by-word matching. If the two text logs match each other in more words, smaller the distance between the two text logs is considered. Reference [12] proposes a hierarchical clustering algorithm. Each layer extracts different log text format information for clustering. These two methods require too much format consistency for the log text. If the two log texts are not the same length, or some details change, the log text will be completely classified into different categories. Reference [13] proposed a method based on the phrase signature for the text log to a cluster. The phrase tag here can be seen as the most distinctive phrase structure for a class of log events. For example, ''database table space'', ''paging switch''. The log text is usually very short, and once such phrases appear, the log text can be classified accurately enough. The idea of phrase labeling comes from the Multiple Longest Common Subsequence [14]. It should be noted that when introducing dynamic programming in the general algorithm tutorial, two strings are used to find the longest common substring. The Multiple Longest Common subsequence here is an extension problem, assuming we have n strings and ask what the longest common substring is. The public substring does not necessarily have to be a continuous word composition, so its robustness is stronger than the previous method. However, the longest common substring problem of n strings is a well-known non-deterministic polynomial-time hard (NP-hard). The goal of the phrase tag-based clustering algorithm is to divide all log text sets into k clusters, find a phrase tag from each cluster, and then try to match the log text in the cluster to the phrase tag. Different from the longest common substring problem, the phrase tag is not required to be included in all the log texts in the cluster, but a quantitative index of the matching degree is calculated. But this problem can be proved further, which can be equivalent to the difficulty of the longest common substring problem of n strings. Due to its own characteristics, the CPPS needs a new platform and technology that can adapt to it.

D. LOG ANALYSIS OF THE CPPS BASED ON BIG DATA PLATFORM

According to the previous analysis, it is not difficult to find that the traditional log analysis system has been slowly unable to undertake the task of log analysis and in the field of CPPS, there is currently no suitable log analysis architecture. After carefully analyzing the various types of logs of the hydropower system, this paper proposes a new integrated framework for log processing of hydropower systems. At the same time, as shown in Figure 3, a distributed system based

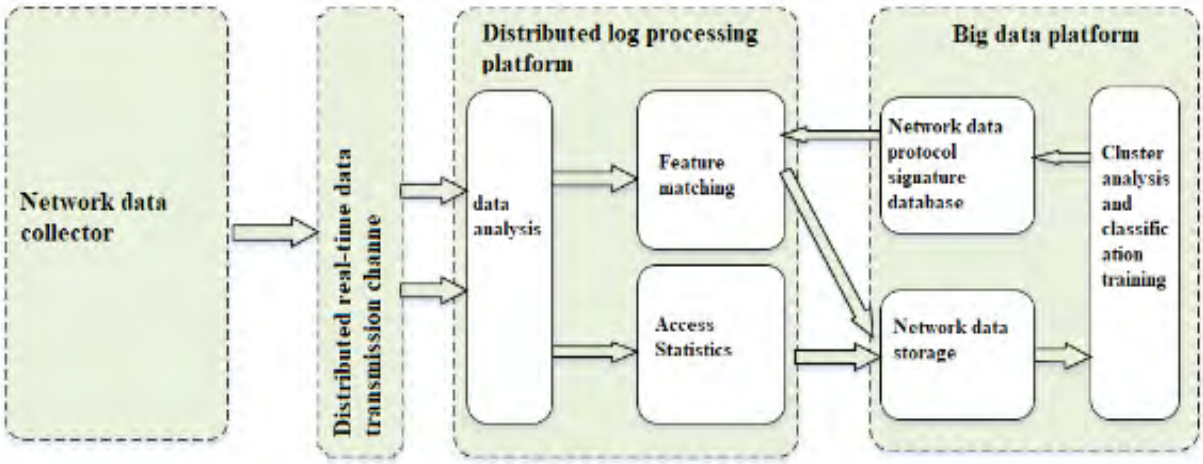

FIGURE 4. Architecture diagram of hydrological power generation industrial control network log analysis system based on big data.

on this framework is implemented, and the performance of the system is verified.

The framework's architecture consists primarily of three main components:

1) HYDROELECTRIC POWER GENERATION INDUSTRIAL CONTROL SYSTEM LOG DATA ORGANIZATION

The components of the computer system of the hydropower industrial control system, the host and the monitor embedded in the application provide the ability to collect log data from the computing system. The log parser/adapter, as well as the event generation process, enables data collection, data integration, and the ability to convert data from multiple heterogeneous data sources into historical data.

2) REAL-TIME ANALYSIS

The real-time analysis component is responsible for processing the newly generated log data in real time and completing online management operations based on the knowledge obtained by offline analysis. Typical real-time analysis techniques include anomaly detection, problem determination, and fault diagnosis.

3) OFFLINE ANALYSIS

The offline analysis component is responsible for obtaining knowledge (such as correlation and dependency knowledge) from historical log data and building a knowledge base. Typical offline analysis techniques include timing pattern discovery and abstracts.

Tasks involved in system monitoring and management include administrator announcements, active data collection, data collector and operational actuator deployment or passivation, monitoring configuration changes, and more. In addition to freeing system administrators from the closed-loop management structure, the CPPS log analysis framework can coordinate domain experts and autonomous intelligent technologies to establish advanced and used solutions for system management.

To verify the practicability and advancement of the architecture proposed based upon the log analysis system of the CPPS mentioned in this paper. We have established a log analysis system for the CPPS based on the big data platform. The system adopts distributed processing mechanism to achieve the purpose of real-time monitoring of system logs and utilizes the ability of distributed storage and data calculation and analysis of big data platform to realize distributed storage and computational analysis of network data.

The system consists of a network data collector, a distributed real-time data transmission channel, a distributed log processing platform, a network data protocol feature library, and a big data platform. The general architecture of the system is depicted in Figure 4.

The main functions of the system and its processing flow are as follows:

a) Data is transmitted via a distributed real-time data transmission channel.

b) The distributed log processing platform processes the obtained data packets in real time.

c) The big data platform performs cluster analysis and classification training on the stored network log data and dynamically updates the network data protocol feature database.

The network topology of the entire system is shown in Figure 5.

The system uses Cloudera CDH as the necessary supporting software of the big data platform and uses the distributed stream processing Spark Streaming platform to perform data analysis, feature matching and access statistics on network data packets. The cluster of the system consists of 5 nodes, each of which is configured with 24 core CPUs, 127 GB of memory, and 10 disks. We develop a program in Java language using the Eclipse platform, and the upper layer data query and presentation adopts the Restful API, which can conveniently issue instructions for querying data in the browser, obtain query results in real time and display them in the browser. We also use the TcpDump to collect data

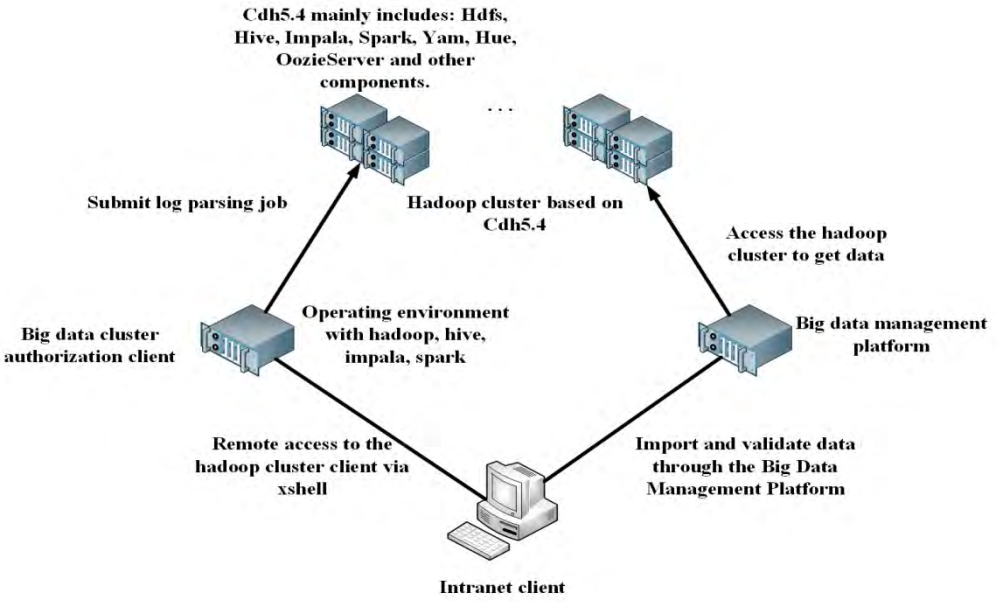

FIGURE 5. Topology diagram of hydropower power generation industrial control network log analysis system based on big data.

and configure the switch to forward data of one or more ports to a certain port to implement network monitoring without affecting the operation of normal programs. At the same time, we employ the Apache Kafka that uses Apache Kafka to transmit the collected data in real time and send the data to the Spark Streaming distributed stream platform for real-time processing. Thus, the analysis results are stored in the big data platform. Moreover, we use the support vector machine and Bayesian constructor for classification training to update the network data protocol feature library utilizing the MapReduce, Hive, Mahout, etc. components of the big data platform.

The steps are designed as follows:

1) LOG TYPE KEYWORD EXTRACTION

The extraction of feature keywords mainly relies on Spark interaction tools to complete the sorting and classification of feature data, and then extracts these data through manual tagging.

2) HOST DATA EXTRACTION

First, analyze and specify the de-duplication host data in the semi-structured log data, such as the hostname or IP address. The number of de-duplicated hosts in the specified semi-structured log data is then analyzed. In this way, you can find out the number of hosts from which the specified semi-structured log data comes from, and provide the basis for the subsequent log parsing classification.

3) PATH DATA STATISTIC

In this subsection, we analyze the de-duplication path data in the specified semi-structured log data, and then evaluate number of de-duplication paths in the specified semi-structured log data, as shown in Figure 6.

4) TYPE STATISTICS

Herein, we examine the de-duplication type data in the specific structured log data, and then analyze the number

```
/jkpt_app/jkpt_domain/jkpt.log
D:\bea\user_projects\domains\ykjxh_domain\servers\server2\logs\server2.log
/gwdaapp/gwda_domain/servers/baobiaoServer2/logs/baobiaoServer2.log
/robocop_app/robocop_domain/servers/roboServer_2/logs/roboServer_2.log
/app/gis/RRServiceProxy/bin/RRServiceProxy.log
/jkglapp/jkgl_domain/servers/Server2/logs/Server2.log
/uds_domain/uds_domain/udsServer2.log
/var/log/Xorg.1.log
/wip_app/wip_domain/servers/95598ManagedServer2/logs/95598ManagedServer2.log
/clgl_app/clgl_domain/servers/ClglServer1/logs/ClglServer1.log
/bea/user_projects/domains/sjzh_domain/servers/sjzhServer2/logs/sjzhServer2.log
/yyjc_app/yyjc_gzt_domain/gztServer1.log
/zsag/zsagpc_domain/servers/zsag1/logs/zsag1.log
```

FIGURE 6. Path data statistics.

of de-duplication types in the specific semi-structured log data.

5) HOST + TYPE + PATH COMBINATION DEDUPLICATION

In this subsection, we investigate the de-duplication host + type + path data in the specified semi-structured log data, and then evaluate the de-duplication host + type + path number in the specified semi-structured log data, as shown in Figure 7.

6) WORD FREQUENCY STATISTICS

After counting the word frequency of the content and the word segmentation, we analyze the log type and application type. To do this, we need to select appropriate feature combination parameters as control variables and state variables. This paper implements the experimental results data to establish a test environment for all possible feature combination parameters and conduct experiments, according to the advantages and disadvantages of the experimental results. As illustrated in Figure 8, the operating system logs,

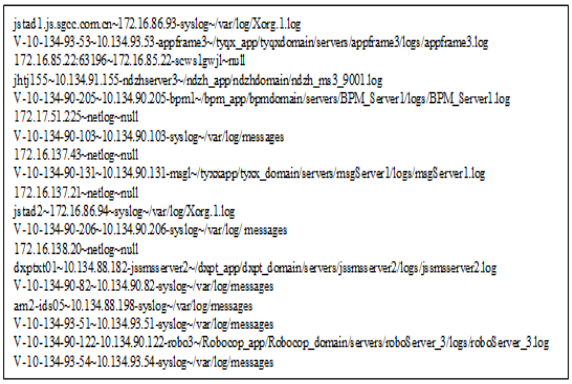

FIGURE 7. Host + type + path combination de-duplication.

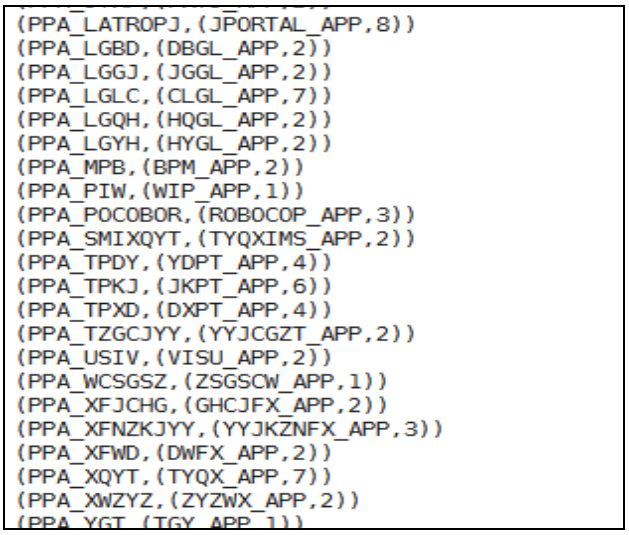

FIGURE 8. Word frequency statistics.

middleware logs, application running logs, and parsing results are used as standards for detection and evaluation. After implementing the frequency analysis of words, we summarized the obtained results in Table 3 and Table 4.

From Table 4 and the actual needs of the CPPS, we can observe that the Big data log analysis at the platform meets the requirements in terms of detection efficiency and other indicators. But the system has a significant problem: the trend of statistical characteristics of traffic attributes is miniaturized. To further improve the system platform and supplement the functions of the big data platform, we present a system log anomaly quantity analysis method which uses the EPABT.

III. SYSTEM LOG ANOMALY ANALYSIS ALGORITHM BASED ON EPABT

In this section, we use the network equipment alarm log of the CPPS as our framework basis. Then, we design an anomaly monitoring model using data mining. The details process of the anomaly monitoring model will be described as follows.

A. TARGET AND METHOD OF SYSTEM ABNORMAL MONITORING

The significance of network system anomaly monitoring can be predicted with the occurrence of abnormal conditions in advance, and then prevent some abnormalities by some preventive measures such as task scheduling or minimize the loss caused by the abnormality. This paper establishes the monitoring target based on the characteristics of the network environment and network equipment alarm log: the abnormal monitoring of the network as a whole. Anomaly monitoring for the whole network is to monitor the network log data to predict whether the entire network system will be abnormal in the next period or not. Once an abnormality is predicted, it can pass data backup, task scheduling, or restart the core device to reduce the loss caused by the abnormality and to improve the reliability of the network system to some extent. The significance of this type of monitoring is to improve the reliability of the network system and the efficiency of network management to save resources. The monitoring method needs to predict whether the network system will be abnormal or not, and the number of abnormalities in a certain period.

Based on the prediction targets established above, a corresponding anomaly prediction model is established for research. In the process of establishing anomaly prediction model, this paper selects ensemble learning algorithms based

TABLE 3. Data validation analysis results.

Large class	Small class	Total	Number of warning records	Number of abnormal records
Operating system log	syslog	809569	60145	39987
	netlog	4600467	58339	2395683
Middleware log	activemq	1714635	1001568	946
	……	……	……	……
	tomcat6	6834573	20	65926
Application run log	jkpt_app	610654		
	robocop_app	50489	20	2469
	……	……	……	……
	pm	106368	50361	706
	topoanalyse	12012456	6146	59978
Total number of articles		46946879	2001135	7988723

TABLE 4. Data validation analysis performance results.

Name	Time spent	Total	Number of bars / sec	MB / sec	Original file size	Cluster file size
Unknown semi-structured log data	30m20s	46946879	31124	21	28.5GB	86GB

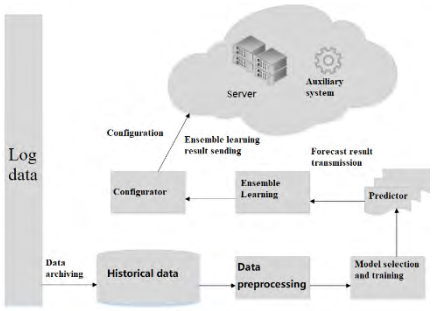

FIGURE 9. Ensemble learning system architecture.

on a multiple time series model method in data mining to develop the prediction model and uses two sets of evaluation systems to verify. One set is traditional time series analysis methods. Error cost functions such as Mean Square Error (MSE), Mean Absolute Error (MAE) and Mean Absolute Percentage Error (MAPE) are often used. Another set of evaluation criteria is the Asymmetric Error Cost (AEC), which is an evaluation criterion for the log anomaly of CPPS. The experiment scheme divides into two parts based on the data set: the training set and the test set. The data of the last month is utilized as the test set, and the remaining data is used as the training set to realize the function of abnormal monitoring through training test.

According to the previous industrial control-system log analysis system, we propose a solution based on multiple prediction algorithm integrations. Figure 9 shows the architecture of this new ensemble learning prediction system.

The main workflow of the platform includes the following steps:

a) Analysis, transformation, and data preprocessing of historical data of the network system

b) Model selection & training based on data characteristics.

c) Use a single trained model to predict the number of futures and reduce the variance of the predictions by Ensemble Learning.

B. COMMON PREDICTION ALGORITHM BASED ON TIME SERIES

For time series prediction, the typical solution is to use Move Average, Auto-Regressive, Neural Network, Support Vector Regressor Machine and Gene Expression Programming [6], [15]–[20]. The Ensemble algorithm proposed in this paper is based on the integration of several algorithms mentioned above. To illustrate the advantages of our algorithm, it is necessary to introduce the characteristics and deficiencies of several basic classification algorithms.

1) MOVE AVERAGE

Moving Average (MA) is the simplest way to predict time series. Essentially, it is a finite impulse response filter that analyzes by calculating the average of a subset over the entire data set. When using sliding window average prediction, it is often necessary to set a sliding window with a fixed length of n. Let the predicted time point be t, then the predicted value of the time point t is:

$$\hat{v}^{(t)} = MA_{t-1} = \frac{v^{(t-1)} + v^{(t-2)} + \cdots + v^{(t-n)}}{n} \tag{1}$$

MA_{t-1} represents the moving average at time point t. If incremental calculations are used, the moving average can be calculated according to the Eq. (1).

$$\hat{v}_{(t)} = MA_{t-1}\frac{v^{(t-n)}}{n} + \frac{v^{(t-1)}}{n} \tag{2}$$

The advantage of Moving Average prediction is that calculation is straightforward, but if the time series changes make the unstable, the prediction effect will be greatly reduced. Also, for the select of the length for the sliding window is a difficult problem. If n is too large, the predicted value change will be underestimated. If n is too small, the history is not fully utilized.

2) AUTO-REGRESSIVE

Auto-Regressive [21] is also known as the AR model. It is a statistical way of predicting time series. In essence, it is a Stochastic Process [22] analysis method. A basic assumption of autoregressive analysis is that the output variable is linearly dependent on its historical value. This model can be expressed as:

$$\hat{v}^{(t)} = c + \varepsilon_t + \sum_{i=1}^{n} \phi_t v^{(t-i)} \tag{3}$$

where c is a constant, ε_t is the random error value and ϕ_t is also the parameter of the autoregressive model.

Simply, it is assumed that there is an autocorrelation between the current value and the past value so that the previous values can predict the current values. It can be seen from Eq. (1) and Eq. (3) that the Moving Average prediction model is a particular case of the autoregressive prediction model, which is all parameters of the model are $1/n$.

The advantage of autoregressive prediction is that the model is easy to understand and simple to calculate. But its shortcomings are also quite obvious. First, time series data must have a linear relationship or can be approximated by a linear relationship. Secondly, autoregressive prediction can only be applied to the time series that are greatly influenced

by its historical factors, but not universally applicable to all types of time series.

3) ARTIFICIAL NEURAL NETWORK

As mentioned above, if there is a complex non-linear relationship data, Auto-Regressive fitting is difficult to predict the relationship between the data accurately. To predict data more effectively with non-linear relationships, Artificial Neural Network is a suitable model.

Artificial neural network (ANN) is a mathematical model that imitates biological neural networks. It is a data mining algorithm that can find complex relationships between data input/output. Structurally, artificial neural networks have some "artificial neurons" that are divided into several levels, and the adjacent two layers of neurons are related. A typical artificial neural network generally has three types of levels: An Input Layer, a Hidden Layer, and an Output Layer, where the hidden layer consists of 0 to multiple layers.

Figure 10 shows the topology of a three-layer artificial neural network, including a hidden layer, i.e., the number of neurons in the three layers is: n, m, and 1, respectively. Where the input layer and the hidden layer respectively contain one deviating neuron (the value of the deviating neuron is fixed at 1).

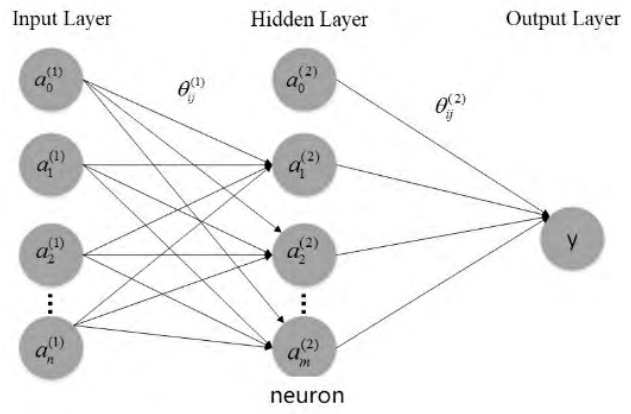

FIGURE 10. Artificial neural network topology.

Where θ_{ij}^{k} is the weight of the value from the j-th atom of the k-th layer to the i-th cell of the $(k+1)$-th layer. $a_i^{(k)}$ represents the value of the i-th neuron of the k-th layer. If it is an input layer, the value is directly derived from the data; otherwise the value is calculated from the previous layer of neurons (except for the deviating neurons). Let the (k+1)th computed neuron be $a^k = [a_1^{k+1}, \cdots, a_m^{k+1}]^T$, then the value can be obtained according to the Eq. (4), where $\theta^{(k)}$ is the matrix representation of the k-th to (k+1)-th layer of neuron weights, and a $(k) = [a_0^{(k)}, a_1^{(k)}, \cdots, a_2^{(k)}]^T$ is the vector representation of all neurons in the k-th layer. For example, the neurons of the hidden layer in the figure can be obtained from Eq. (5).

$$a^{(k+1)} = g(\theta^{(k)} a^{(k)}) \tag{4}$$

$$\begin{aligned} a_1^{(2)} &= g(\theta_{10}^{(1)} a_0^{(1)} + \theta_{11}^{(1)} a_1^{(1)} + \theta_{12}^{(1)} a_2^{(1)} + \cdots + \theta_{1n}^{(1)} a_n^{(1)}) \\ a_2^{(2)} &= g(\theta_{20}^{(1)} a_0^{(1)} + \theta_{21}^{(1)} a_1^{(1)} + \theta_{22}^{(1)} a_2^{(1)} + \cdots + \theta_{2n}^{(1)} a_n^{(1)}) \\ &\vdots \\ a_m^{(2)} &= g(\theta_{m0}^{(1)} a_0^{(1)} + \theta_{m1}^{(1)} a_1^{(1)} + \theta_{m2}^{(1)} a_2^{(1)} + \cdots + \theta_{mn}^{(1)} a_n^{(1)}) \end{aligned} \tag{5}$$

Due to the complexity of the neural network topology and the data size of the training set, the general gradient descent algorithm is inefficient and the training time is extremely long. The reason is that it needs to update the weights after learning the entire training sample set.

4) SUPPORT VECTOR REGRESSOR MACHINE

Support Vector Regressor Machine (SVM) [23]–[27] is a kind of support vector machine, which uses the basic principles of support vector machine: The characteristics of the training machine data are mapped from the original n-dimensional space to the higher-dimensional m-dimensional space Γ through the kernel function Φ, and then the linear regression is performed in the space. The basic idea can be expressed by the Eq. (6):

$$f(x) = \omega\Phi(x) + b, \quad \text{and } \Phi : R \to \Gamma, \quad \omega \in \Gamma \tag{6}$$

where $\omega = [\omega_1, \omega_2, \cdots, \omega_m]^T$ is the weight of each feature in the m (m > n) dimensional space after the original sample is mapped, and b is a threshold.

Given a trained support vector regression model, the predicted value can be calculated in O (m) time. For example, given the time series data $v = v^{(t-m)}, v^{(t-n+1)}, \cdots, v^{(t)}$ (assuming that it has been normalized to the interval $[-1, 1]$), the predicted value of the t+1 time point is $v^{(t+1)} = \omega \cdot \Phi(v) + b$.

Regarding the training of the model, the goal is to minimize the prediction error calculated by Eq.(7) given a kernel function, where x_i is the training sample, y_i is the actual value in the training set, and $C(g)$ is the cost function. The $\lambda||\omega||^2$ is used to control the complexity of the model and to penalize excessive weight values that increase the complexity of the model.

$$R_{reg} = \sum_{i=1}^{l} C(\mathrm{f}(x_i - y_i)) + \lambda||\omega||^2 \tag{7}$$

For most cost functions $C(g)$, Eq. (7) can be optimized by Quadratic Programming. With the regard of the cost function, common objects are the ε − int*ensitive* cost function and the Huber cost function.

However, the SVM algorithm is difficult to implement for large-scale training samples. Since SVM solves the support vector using the quadratic programming, solving the quadratic programming involves the calculation of the

m-order matrix (m is the number of samples), when the number of m is large. The storage and calculation of this matrix will consume a lot of machine memory and computation time.

5) GENE EXPRESSION PROGRAMMING

Gene Expression Programming (GEP) [28], [29] is a new type of bionics algorithm that originated in the field of biology and inherits the advantages of traditional Genetic Algorithm and Genetic Programming. Gene expression programming has a good performance for time series analysis.

Different from the time series analysis method mentioned above has only fixed hypothesis expressions (e.g., the expression of the autoregressive model is multiple linear equations). In other words, the hypothesis space of gene expression programming allows us for any expression. This gives gene expression programming to learn arbitrarily complex prediction models from theory correctly.

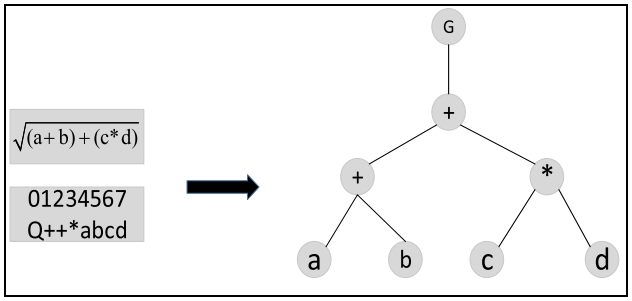

FIGURE 11. Array of gene expression programming and tree expression.

Gene expression programming is capable of expressing arbitrarily complex expressions because it uses ''chromosomes'' to encode expressions. Structurally, a ''chromosome'' is represented by an array; logically, a ''chromosome'' expresses an expression tree. Figure 11 shows a simple expression $\sqrt{(a+b)+(c*d)}$ with the corresponding array and expression tree. For time series analysis, gene expression programming measures prediction errors by evaluating the difference between the value of the ''chromosome'' and the actual value.

Many prediction algorithms have their shortcomings and advantages when based on time series prediction. However, due to the characteristics of the CPPS, the network equipment is complex, and the requirements for reliability and accuracy are high. At the same time, the prediction target of this research is based on many factors such as the number of abnormal time series, the traditional methods cannot adapt well to the needs of the CPPS (the latter experiment further validates this analysis), this paper still has to take other ways to achieve the goal.

C. ANOMALY QUANTITY ENSEMBLE LEARNING PREDICTION ALGORITHM BASED ON TIME SERIES AND ITS EVALUATION CRITERIA

Inspired by the work [30]–[36], this paper adopts the strategy of Ensemble Learning and proposes a time series based integrated prediction algorithm named as Ensemble Prediction Algorithm Based on Time series (EPABT) because it is difficult for a single forecasting model to predict accurately.

Ensemble learning strategies have widely used in the classification of data mining. In the classification problem, the data samples are considered to be independent and identically distributed. However, for time series analysis problems, there is a strong temporal dimension association between the samples. Also, unlike the classification problem, the class labels (e.g., the value that needs to be predicted) of the time series analysis problem is a continuous value, so it is impossible to use a voting mechanism similar to the classification to obtain the final result. For time series analysis ensemble learning, the ensemble learning strategy mainly updates the weight of each prediction algorithm by predictive evaluation criteria. In the ensemble prediction, we use five different time series prediction techniques, the algorithm name and related description are given in Table 5.

TABLE 5. Time series prediction algorithm for ensemble learning.

Method name	Method description
Moving Average	Naïve Prediction
Auto-Regressive	Linear Regression
Neural Network	Non-linear Regression
Support Vector Machine	Linear learning algorithm for non-linear kernels
Gene Expression Programming	Heuristic Algorithm

Moreover, for integrating their predictions, we propose a weighted linear combination strategy. Let us assume that the prediction results of the algorithm $p \in P$ at time point 't' is $\hat{v}_p^{(t)}$ and its weight corresponding to time point 't' is then the predicted value for a log at time 't' is:

$$\hat{v}^{(t)} = \sum_p w_p^{(t)} \hat{v}_p^{(t)}, \quad \text{and} \sum_p w_p^{(t)} = 1 \tag{8}$$

The initial state, t = 0, all prediction algorithms contribute the same amount to the prediction results, such as $w_p^{(t)} = \frac{1}{|P|}$.

The weight update strategy based on the ensemble algorithm is also different from the traditional classification basic strategy. In a classified forecast scenario, the results can only be expressed as ''correct'' or ''incorrect.'' However, the purpose of the ensemble algorithm weight update is to improve the weight of the classifiers with the correct classification results. In the predicted scenario, the prediction result is a continuous value, and the weight of the prediction algorithm directly affects the result of the ensemble algorithm.

Let us suppose that predicted value $\hat{v}_p^t$ and the true value $v^{(t)}$ at time 't', the relative error $e_i^{(t)}$ of the prediction algorithm i is as follows:

$$e_i^{(t)} = \frac{c_i^{(t)}}{\sum_p c_p^{(t)}} w_i^{(t)} \tag{9}$$

where $c_i^{(t)}$ (or $c_p^{(t)}$) represents the predicted cost of the prediction algorithm i or p calculated by the predictive evaluation cost function *MAE, LSE, MAPE.*

Note that the relative error cannot be used in the update weight of the prediction algorithm because it is not normalized. Since the final prediction result is a linear combination of multiple prediction algorithms the following equation can be used to normalize the weights.

$$w_i^{(t+1)} = \frac{e_i^{(t)}}{\sum_p e_p^{(t)}} \tag{10}$$

Using this weight update strategy, it can be ensured that the weight of the optimal prediction algorithm at each time point can be increased.

In practice, the traditional prediction algorithms cannot meet the demand in this particular scenario, and also, the conventional evaluation criteria cannot meet the needs of the abnormal quantity monitoring of the CPPS. A common predictive evaluation criterion is the error cost function.

The error cost function used by the traditional time series analysis method is usually Mean Square Error (MSE), Mean Absolute Error (MAE) and Mean Absolute Percentage Error (MAPE). In essence, these error cost functions are symmetric cost functions that mean the over-estimation and under-estimation of the method of equal treatment of the error cost function. The cost functions are shown in Eqs. (11-13):

$$MSE = \frac{1}{n}\sum_{t=1}^{n}(observed_t - predicted_t)^2 \tag{11}$$

where $observed_t$ represents the true value and $predicted_t$ represents the predicted value.

The MSE can evaluate the degree of change in the data. The smaller the value of the MSE, the better the accuracy of the predictive model describing the experimental data is.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|(observed_t - predicted_t)| \tag{12}$$

MAE is the average of the absolute errors. The MAE can better reflect the actual situation of the predicted value error.

$$MAPE = \sum_{t=1}^{n}\left|\frac{observed_t - predicted_t}{observed_t}\right| \times \frac{100}{n} \tag{13}$$

The larger the MAPE value means, the more significant the difference between the predicted value and the original value.

The purpose of these functions is to measure the geometric error of the time series analysis method and to guide the way to approach the correct value in the form of geometric minimal error during training.

These functions are broadly applicable to most time series prediction problems. However, they have certain limitations for scenarios that predict anomalous quantities for CPPS. There is a common between these evaluation criteria; that is, they only make erroneous estimates by the absolute errors of predicted and actual values, without distinguishing between over-predicted (predicted values are higher than actual values) and under-predicted (predicted values are lower than actual values).

There are overestimation and underestimation in the prediction of the abnormal number of hydropower generation industrial control systems, and these two cases have different semantics in the specific scenarios of this paper, which yield to different costs. Specifically, if the predicted number of anomalies is higher than the actual situation, the security of the system will not be affected, but as a whole system, it has to pay for these wasted resources. On the contrary, if the predicted abnormal number is lower than the actual situation, then, the system will not take enough exception handling measures, and the security of the system will be affected.

To deal with the application scenario of abnormal quantity prediction of CPPS, this paper proposes an asymmetric error cost function (AEC) to evaluate prediction errors. AEC is an asymmetric heterogeneous error cost function, which consists of two different costs: too high and too low. Correspondingly, they are represented by $R(v^{(s)}, \hat{v}^{(s)})$ and $P(v^{(s)}, \hat{v}^{(s)})$, where $v^{(s)}$ represents the number of anomalies at a future time point s, and $\hat{v}^{(s)}$ represents the number of anomaly predictions at a future time s.

The total cost function can be expressed as:

$$A = \beta P(v^{(s)}, \hat{v}^{(s)}) + (1-\beta)R(v^{(s)}, \hat{v}^{(s)}) \tag{14}$$

β is the parameter used to adjust the two cost weights. By this parameter, the cost of artificially improving the over-predicted and under-predicted can be artificially adjusted by the change the value of β. The specific representation of AEC can be changed according to the specific application. Basically, the specifically defined functions P and functions R only need to satisfy the two properties of non-negative and consistent.

1) Non-negative: For any non-negative $v^{(s)}$ and $\hat{v}^{(s)}$, there are $P(v^{(s)}, \hat{v}^{(s)}) \geq 0$ and $R(v^{(s)}, \hat{v}^{(s)}) \geq 0$.

2) Consistency: If $v_1(s) - \hat{v}_1(s) \geq v_2(s) - \hat{v}_2(s)$, *then* $P(v_1(s), \hat{v}_1(s)) - P(v_2(s), \hat{v}_2(s))$ *should maintain positive and negative consistency; for the same reason, if* $v_1(s) - \hat{v}_1(s) \geq v_2(s) - \hat{v}_2$ *(s), then* $R(v_1(s), \hat{v}_1(s)) - R(v_2(s), \hat{v}_2(s))$ *should also maintain positive and negative consistency.*

In the process of anomaly forecasting for CPPS, we assume that the cost of accurate prediction is C_{normal}, the cost of too low prediction is C_{under}, and the cost of excessive prediction is C_{over}. Under normal circumstances, C_{under} is uncertain, but $C_{under} \gg C_{normal}$ is certain, the function P in the AEC cost function can be described as:

$$P(v^{(s)}, \hat{v}^{(s)}) = min(v^{(s)}, \hat{v}^{(s)})C_{normal} + max(0, v^{(s)} - \hat{v}^{(s)})C_{under} \tag{15}$$

The function in the AEC cost function R can be expressed as:

$$R(v^{(s)}, \hat{v}^{(s)}) = max(0, \hat{v}^{(s)} - v^{(s)})C_{over} \tag{16}$$

Combining the above two formulas, you can get the specific measure of the AEC cost function:

$$A = f(v^{(s)}, \hat{v}^{(s)}) = \beta P(v^{(s)}, \hat{v}^{(s)}) + (1-\beta)R(v^{(s)}, \hat{v}^{(s)}) = \begin{cases} \beta v^{(s)} C_{normal} + (1-\beta)(\hat{v}^{(s)} - v^{(s)})C_{over}, \\ \qquad\qquad \text{when } \hat{v}^{(s)} \geq v^{(s)} \\ beta(\hat{v}^{(s)} C_{normal} + (v^{(s)} - \hat{v}^{(s)})C_{under}), \\ \qquad\qquad when\ \hat{v}^{(s)} < v^{(s)} \end{cases} \quad (17)$$

IV. EXPERIMENT AND RESULTS

A. PROCESSING LOG FILES

The alarm log data mainly includes the following components:

1) LEVEL

Represent the urgency of abnormal information. The order of urgency from low to high is: L1, L2, L3, and L4. In this paper, the goal of abnormal prediction is to predict the occurrence of level L4 anomalies.

2) NAME

Describes the specific types of abnormal information, including link disconnection, OSPF interface state change, OSPF neighbor state change, and device offline.

3) ALARM SOURCE:

It indicates the specific device number that sent the abnormal information.

4) LOCATION

Describe the location of the failed device or interface.

5) OCCURRENCE TIME

Indicates the time when the abnormal information is recorded, which is accurate to the second.

Through the analysis of the original alarm log, we found that there are a large number of criminal records, redundant records, derivative alarms, flashing alarms and other noise data. Before the prediction study, this paper designed a corresponding filtering method to effectively filter the noise data in the original alarm log.

In the process of establishing a classification prediction model, the selection of feature items plays a crucial role. To select the feature items with a high degree of discrimination to the category as much as possible, the prediction target and the characteristics of the alarm log need to be analyzed first. The current best time window represents the current operating state of the network system, and it is closest to the predicted time window. Therefore, various levels and types of alarm events in the current time window are counted as feature items. The extracted feature items include: the number of four levels of alarm events within the current time window and the number of various types of alarm events in the current time window.

B. TIME GRANULARITY SELECTION

Since this experiment is based on the time series prediction, the time granularity selection of time series is a significant factor affecting the prediction effect. The time series is obtained by clustering the requests in the original data set according to a specific time granularity and sorting them according to time. The difference in time granularity selection can bring different difficulties in time series prediction.

Figure 12 shows that the time series obtained by clustering at different time granularities. The *X* axis represents different time granularities, and the *Y* axis represents an abnormal amount. As can be seen from the figure, the larger the time granularity selection is, the larger the abnormal amount at each time point.

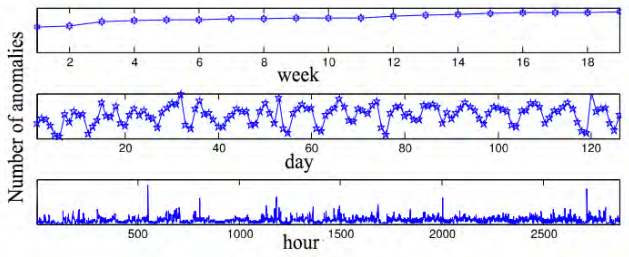

FIGURE 12. Time series of different time granularities.

TABLE 6. Metric statistics for different timegranularity clustering.

Metric	Week	Day	Hour
Coefficient of variation	0.1335	0.4123	0.6998
Skewness	-0.5463	-0.2498	2.7012
Kurtosis	2.8001	2.6002	21.4236

For large time granularity, even in small prediction bias may make the absolute error of the prediction large due to the large base number resulting in a large prediction error. However, short time of granularity would make the data at each point in time lack statistical significance. Also, the clustered time series shows irregularities on multiple indicators. Table 6 shows that the coefficient of variation (coefficient of variation = Mean Square Error / average × 100%), skewness, and kurtosis for each time series of different time granularity clusters. The coefficient of variation measures the variability of the time series. The higher the coefficient of variation, make easier to the series changes over time. Skewness measures the asymmetry of the –time series, and the larger the skewness value indicates the more asymmetrical happened at the time series. The kurtosis measures the variance of the time series, and the greater the kurtosis, the more irregular the distribution of the values of the time series. It can be seen from the table that when the hour is used as the time granularity, then there is an irregularity in the time series. Similarly, far exceeds the time series of weeks and days as the time granularity. In summary, based on the above observations and preliminary analysis, we can say that when case uses a time series with days as the time granularity.

C. RESULTS

To analyze the experimental results, we utilize a real dataset named ''event_warningdetail'' in the CPPS security project. Our project partner provided the data by collecting and

```
--- ----------------------------------
-- Records of event_warning
-- ----------------------------------
INSERT INTO 'event_warning' VALUES (nu11,'88893034','GJ00160509','1','2016-07-18 00:02:07','2','1','192.168.1.6','15','CJ991601150099','10.1.182.147','guman@sgcc.com.cn','u8D26u53F
INSERT INTO 'event_warning' VALUES (nu11,'88893057','GJ00160509','1','2016-07-18 00:02:07','1','1','192.168.1.6','7','CJ991601100002','10.1.182.147','guman@sgcc.com.cn','u5BC6u7801
INSERT INTO 'event_warning' VALUES (nu11,'88899521','GJ48161209','6','2016-12-09 17:43:14','3','2',null,'-1',null,'10.3.233.2','',null,'0',null,null,'',null,'6030500','-u670
INSERT INTO 'event_warning' VALUES (nu11,'88902182','GJ99161219','2','2016-12-19 12:01:40','2','2',null,'-1',null,'10.90.233.82','27.195.40.150',null,'0',null,null,'',null,'2
INSERT INTO 'event_warning' VALUES (nu11,'88902183','GJ99161219','2','2016-12-19 12:05:23','2','2',null,'-1',null,'10.99.2.11','10.99.2.8',null,'0',null,null,'9471BB61-DO50-4C4
INSERT INTO 'event_warning' VALUES (nu11,'88905904','GJ99161221','2','2016-12-21 13:54:29','1','2',null,'-1',null,'10.99.233.2','',null,'0',null,null,'C22FD820-D8FA-4C1A-A6D7-E
INSERT INTO 'event_warning' VALUES (nu11,'88905996','GJ99161223','2','2016-12-23 15:17:29','1','2',null,'-1',null,'10.3.22.14','104.223.133.27',null,'0',null,null,'',null,'20
INSERT INTO 'event_warning' VALUES (nu11,'88908013','GJ99161229','2','2016-12-29 15:40:54','1','2',null,'-1',null,'10.101.198.169','23.52.7.3',null,'0',null,null,'',null,'203
INSERT INTO 'event_warning' VALUES (nu11,'88909898','GJ74170110','6','2017-01-10 15:46:13','3','1','10.90.233.19','222.76.72.169','u8D26u53F7u5F02u5E38u767Bu
INSERT INTO 'event_warning' VALUES (nu11,'88910967','GJ99170123','2','2017-01-23 10:35:52','1','2',null,'-1',null,'10.99.2.11','',null,'0',null,null,'9471BB61-DO50-4C42-8290-45
INSERT INTO 'event_warning' VALUES (nu11,'88911401','GJ74170201','6','2017-02-01 00:01:45','3','1','10.90.233.24','1','3','10.99.233.24','120.40.96.180','u8D26u53F7u5F02u5E38u767Bu
INSERT INTO 'event_warning' VALUES (nu11,'88916416','GJ99170215','2','2017-02-15 13:41:24','2','2',null,'-1',null,'10.99.2.11','',null,'0',null,null,'9471BB61-DO50-4C42-8290-45
INSERT INTO 'event_warning' VALUES (nu11,'88923604','GJ74170301','6','2017-03-01 00:01:49','3','1','10.90.233.19','1','3','10.99.233.19','171.12.25.151','u8D26u53F7u5F02u5E38u767Bu
INSERT INTO 'event_warning' VALUES (nu11,'88925287','GJ48170328','2','2017-03-28 10:35:04','2','2',null,'-1',null,'10.99.2.1','',null,'0',null,null,'',null,'2020000','-u540E
INSERT INTO 'event_warning' VALUES (nu11,'88925710','GJ12170428','0','2017-04-28 10:43:35','1','2',null,'-1',null,'5.5.5.5','',null,'0',null,null,'','1','2020100','- Back door Trojan
INSERT INTO 'event_warning' VALUES (nu11,'88925711','GJ48170428','12','2017-05-28 15:17:43','2','1','192.168.1.3','0','0','192.168.0.57',null,'Host open high-risk port','0','0', '0' , '' ,
INSERT INTO 'event_warning' VALUES (nu11,'88925712','GJ48170428','12','2017-05-28 15:17:43','2','1','192.168.1.3','0','0','192.168.0.57',null,'Host open high-risk port','0','0', '0' , '' ,
INSERT INTO 'event_warning' VALUES (nu11,'88925713','GJ48170428','12','2017-05-28 15:17:43','2','1','192.168.1.3','0','0','192.168.0.57',null,'Host open high-risk port','0','0', '0' , '' ,
INSERT INTO 'event_warning' VALUES (nu11,'88925714','GJ48170428','12','2017-05-28 15:17:43','2','1','192.168.1.3','0','0','192.168.0.57',null,'Host open high-risk port','0','0', '0' , '' ,
INSERT INTO 'event_warning' VALUES (nu11,'88925715','GJ48170428','12','2017-05-28 15:17:43','2','1','192.168.1.3','0','0','192.168.0.57',null,'Host open high-risk port','0','0', '0' , '' ,
INSERT INTO 'event_warning' VALUES (nu11,'88925716','GJ48170428','12','2017-05-28 15:17:43','2','1','192.168.1.3','0','0','192.168.0.57',null,'Host open high-risk port','0','0', '0' , '' ,
INSERT INTO 'event_warning' VALUES (nu11,'88925717','GJ48170428','12','2017-05-28 15:17:43','2','1','192.168.1.3','0','0','192.168.0.57',null,'Host open high-risk port','0','0', '0' , '' ,
INSERT INTO 'event_warning' VALUES (nu11,'88925718','GJ48170428','12','2017-05-28 15:17:43','2','1','192.168.1.3','0','0','192.168.0.57',null,'Host open high-risk port','0','0', '0' , '' ,
INSERT INTO 'event_warning' VALUES (nu11,'88925719','GJ48170428','12','2017-05-28 15:17:43','2','1','192.168.1.3','0','0','192.168.0.57',null,'Host open high-risk port','0','0', '0' , '' ,
INSERT INTO 'event_warning' VALUES (nu11,'88925720','GJ48170428','12','2017-05-28 15:17:43','2','1','192.168.1.3','0','0','192.168.0.57',null,'Host open high-risk port','0','0', '0' , '' ,
INSERT INTO 'event_warning' VALUES (nu11,'88925721','GJ48170428','12','2017-05-28 15:17:43','2','1','192.168.1.3','0','0','192.168.0.57',null,'Host open high-risk port','0','0', '0' , '' ,
INSERT INTO 'event_warning' VALUES (nu11,'88925722','GJ48170428','12','2017-05-28 15:17:43','2','1','192.168.1.3','0','0','192.168.0.57',null,'Host open high-risk port','0','0', '0' , '' ,
INSERT INTO 'event_warning' VALUES (nu11,'88925723','GJ48170428','10','2017-05-12 05:00:11','2','1','192.168.1.3','0','0','192.168.0.57',null,'command：cat /etc/profile | grep –v \
```

FIGURE 13. Log data named event_waringdetail.

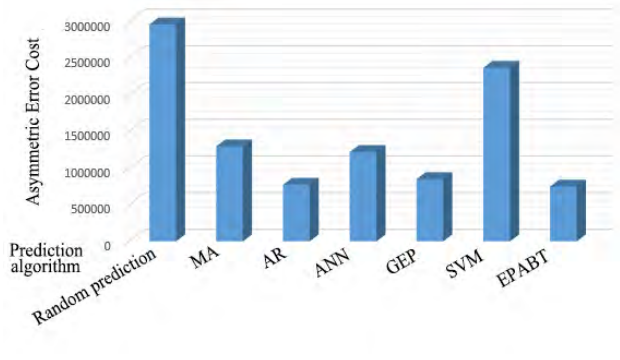

FIGURE 14. AEC for each algorithm.

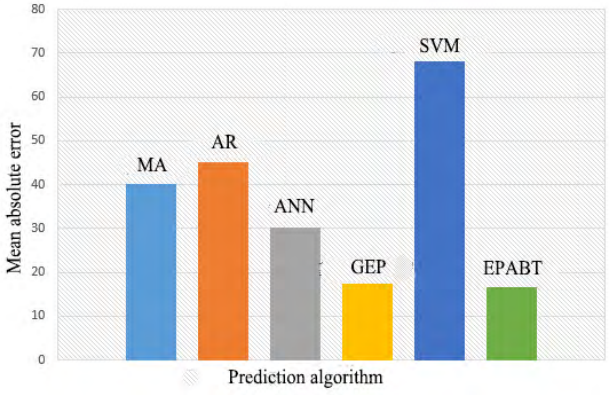

FIGURE 15. The results of each algorithm using mean absolute error.

collating the three-month CPPS alarm log data, as depicted in Figure 13. Since the prediction models used in ensemble learning are mostly supervised-based learning, the data set is split into two parts: the training set and the test set. The last month's data is employed as the test set, and the remaining data is used as the training set.

The comparative trial used a variety of prediction methods to compare, including Random Prediction, Moving Average, Auto-Regressive Prediction, Neural Network, Gene Expression Programming, Support Vector Regressor Machine, and Ensemble Learning method EPABT.

Figure 14 shows the total predictive cost of each forecasting method based on multiple metrics (total forecast errors on the three-month test set). The evaluation system of this experiment uses the Asymmetric Error Cost (AEC), which is proposed in this paper. From this figure, we can conclude that the method of ensemble learning is always optimal or close to optimal. The prediction results based on the three-month test data show that the optimal prediction models in different data sets are different, but the ensemble learning method is always close to the optimal model. It is because the predictive strategy of ensemble learning still tends to increase the proportion of the best performing predictive model by prior experience so that the performance of the whole ensemble learning model approaches the optimal model.

Figure 15-17 illustrate the results of six algorithms (Moving Average (MA), Auto-Regressive Prediction (AR), Artificial Neural Network (ANN), Gene Expression Programming (GEP), Support Vector Regressor Machine (SVM), and Ensemble Learning method(EPABT)) under three evaluation indicators (Mean Absolute Error, Mean Square Error and Mean Absolute Percentage Error).

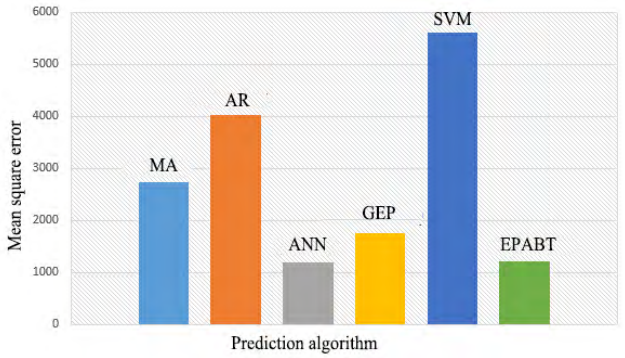

FIGURE 16. The results of each algorithm using mean square error.

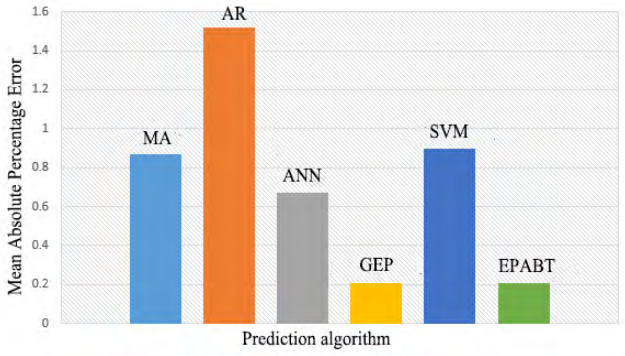

FIGURE 17. The results of each algorithm using mean absolute percentage error.

In Figure 15, we can see that when using the Mean Absolute Error as a measure, Gene Expression Programming works best, Support Vector Regressor Machine is the worst. The performance of Ensemble Learning algorithm and Gene Expression Programming is almost the same.

Figure 16 shows that when using Mean Square Error as a measurement, the performance of the Neural Network is the best. The performance of the Ensemble Learning algorithm is almost the same as that of the Neural Network. The performance of other algorithms is not efficient, especially the support vector regression machine.

From the analysis and comparison in Figure 17, it can be observed that, when using the Mean Absolute Percentage Error as a measurement, the Ensemble Learning algorithm and Gene Expression Programming get the best performance again, while other algorithms have poor performance, especially the Auto-Regressive model.

In summary, the results of various prediction algorithms in different evaluation models are not always optimal. In general, over the different evaluation systems, the best performance is the different prediction methods, which also validates the expression in the third section of this paper. However, the Ensemble Learning algorithm is always equivalent to the effect of a specific optimal method and can be used as the most accurate method.

To further verify the impact of the proposed algorithm and the evaluation system, we compare the actual number of anomalies in a certain period with the prediction results of each algorithm, as shown in Figure 18.

It is not difficult to see from the figure that Gene Expression Programming has achieved excellent results, followed by neural networks, but overall it is not able to meet the maximum number of abnormal predictions. In contrast, the Ensemble Learning algorithm is very obvious to take advantage of the various algorithms, as shown in Figure 19, which reveals that the EPABT and the asymmetric error cost evaluation system AEC can be used together to achieve optimal Predictive effect.

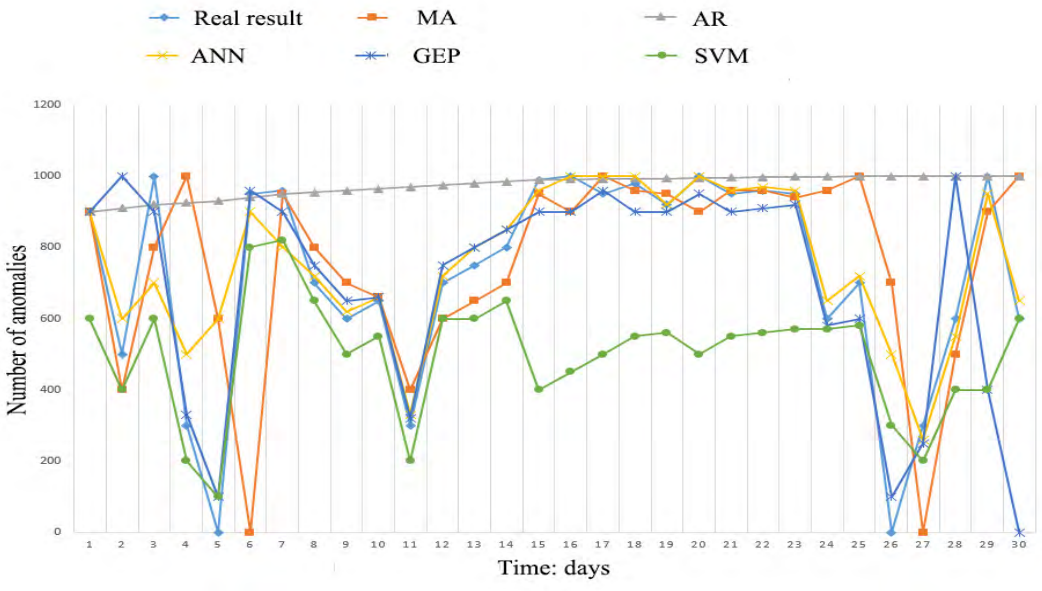

FIGURE 18. Comparison of actual results and prediction results of each algorithm.

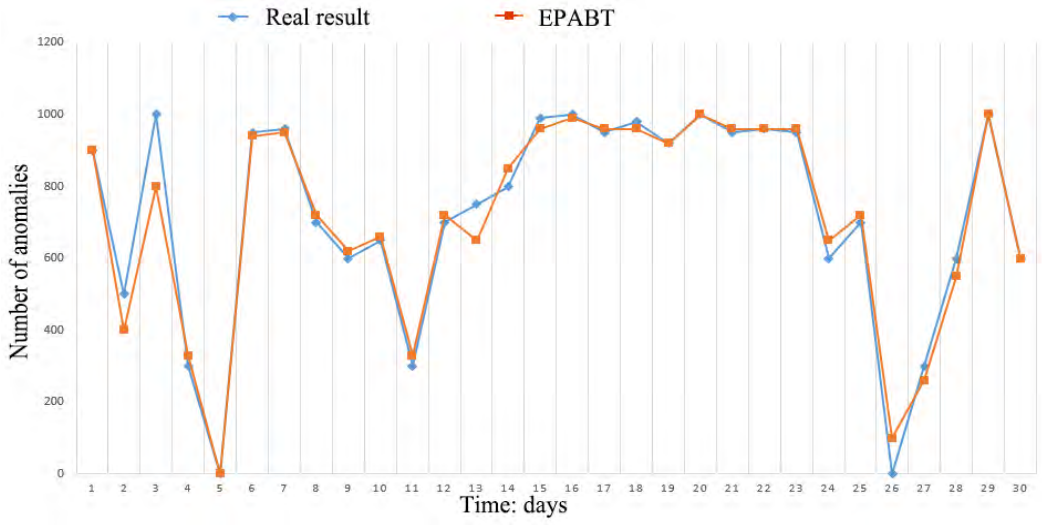

FIGURE 19. Comparison of ensemble learning method with real results.

V. CONCLUSION

To achieve the characteristics of the field equipment layer of the hydropower generation industrial control network, it is essential to meet the blank of the CPPS during the log analysis system. So for this, we proposed a log analysis architecture for the CPPS and also design an abnormal detection framework based on big data mining. To predict the abnormal traffic of the network, we introduced an Ensemble Prediction Algorithm Based on Time series (EPABT) which evaluates the characteristics of the log analysis architecture to detect the abnormal features during the network traffic analysis.

To analyze the efficiency of the EPABT, we performed the proposed prediction algorithm on a real dataset named ''event_warningdetail,'' and compared the experimental results with the existing state of the art methods. The analysis of the EPABT confirms that it provides better prediction accuracy as compared to the other approaches.

For future work, we can introduce more novel technologies into the field of power industrial control systems to achieve better improvement, such as event mining, tickets mining.

ACKNOWLEDGMENT

(Qianmu Li and Shunmei Meng contributed equally to this work.)

REFERENCES

[1] Z. Wang, ''Research on network equipment fault prediction based on log analysis,'' Chongqing Univ., Chongqing, China, Tech. Rep., 2015.
[2] C. Kruegel, G. Vigna, and W. Robertson, ''A multi-model approach to the detection of Web-based attacks,'' *Comput. Netw.*, vol. 48, pp. 717–738, Aug. 2005.
[3] C. Criscione and S. Zanero, ''Masibty: An anomaly based intrusion prevention system for Web applications,'' Univ. Politecnico di Milano, Milan, Italy, Tech. Rep., 2009.
[4] S. Cho and S. Cha, ''SAD: Web session anomaly detection based on parameter estimation,'' *Comput. Secur.*, vol. 23, no. 4, pp. 312–319, 2004.
[5] G. Gao and H. Wu, ''Design and application of Web application security monitoring system,'' *Comput. Eng. Des.*, vol. 31, no. 17, pp. 3760–3762, 2010.
[6] T. Li, ''Application and practice of data mining,'' Xiamen Univ. Press, Xiamen, China, Tech. Rep., 2013, pp. 8–61.
[7] T. Li, J. Xu, and L. Zhang, ''Theoretical algorithm and application of event mining,'' Xiamen Univ. Press, Xiamen, China, Tech. Rep., 2016, pp. 10–36.
[8] T. Li, *Event Mining: Algorithms and Applications*. Boca Raton, FL, USA: CRC Press, 2015.
[9] J. Lafferty, A. McCallum, and F. C. N. Pereira, ''Conditional random fields: Probabilistic models for segmenting and labeling sequence data,'' in *Proc. 18th Int. Conf. Mach. Learn. (ICML)*, 2001, pp. 282–289.
[10] W. Xu, L. Huang, A. Fox, D. Patterson, and M. I. Jordan, ''Detecting large-scale system problems by mining console logs,'' in *Proc. 22nd ACM SIGOPS, Symp. Oper. Syst. Princ.*, 2009, pp. 117–132.
[11] M. Aharon, G. Barash, I. Cohen, and E. Mordechai, ''One graph is worth a thousand logs: Uncovering hidden structures in massive system event logs,'' in *Machine Learning and Knowledge Discovery in Databases*. Springer, 2009, pp. 227–243.
[12] A. A. O. Makanju, A. N. Zincir-Heywood, and E. E. Milios, ''Clustering event logs using iterative partitioning,'' in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2009, pp. 1255–1264.
[13] L. Tang, T. Li, and C. S. Perng, ''LogSig: Generating system events from raw textual logs,'' in *Proc. ACM Conf. Inf. Knowl. Manage. (CIKM)*, 2011, pp. 785–794.
[14] D. Maier, ''The complexity of some problems on subsequences and supersequences,'' *J. ACM*, vol. 25, no. 2, pp. 322–336, Apr. 1978.
[15] R. Agrawal and R. Srikant, *Fast Algorithms for Mining Association Rules*. San Francisco, CA, USA: Morgan Kaufmann, 1998, pp. 580–592.
[16] P. Ning, Y. Cui, and D. S. Reeves, ''Analyzing intensive intrusion alerts via correlation,'' in *Recent Advances in Intrusion Detection* (Lecture Notes in Computer Science), vol. 25. no. 16, 2003, pp. 74–94.
[17] P.-N. Tan, M. Steinbach, and V. Kumar, *Introduction to Data Mining*. Beijing, China: People's Posts and Telecommunications Press, 2011, pp. 36–75.
[18] A. J. Oliner, A. Aiken, and J. Stearley, ''Alert detection in system logs,'' in *Proc. IEEE Int. Conf. Data Mining (DBLP)*, Dec. 2008, pp. 959–964.
[19] Y. Liang, Y. Zhang, A. Sivasubramaniam, M. Jette, and R. Sahoo, ''BlueGene/L failure analysis and prediction models,'' in *Proc. Int. Conf. Dependable Syst. Netw.*, Jun. 2006, pp. 425–434.
[20] Y. Liang, Y. Zhang, H. Xiong, and R. Sahoo, ''Failure prediction in IBM BlueGene/L event logs,'' in *Proc. 7th IEEE Int. Conf. Data Mining*, Oct. 2007, pp. 583–588.
[21] G. U. Yule, ''On a method of investigating periodicities in disturbed series, with special reference to Wolfer's sunspot numbers,'' *Philos. Trans. Roy. Soc. London A, Containing Papers Math. Phys. Character*, vol. 226, pp. 267–298, Apr. 1927.
[22] H. M. Taylor and S. Karlin, *An Introduction to Stochastic Modeling*, vol. 36. no. 4, 2011, pp. 387–388.

[23] H. Drucker, C. J. C. Burges, L. Kaufman, A. J. Smola, and V. Vapnik, "Support vector regression machines," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 28, no. 7, 1997, pp. 779–784.
[24] S. Ding, S. Qu, Y. Xi, and S. Wan, "A long video caption generation algorithm for big video data retrieval," *Future Gener. Comput. Syst.*, vol. 93, pp. 583–595, Apr. 2019.
[25] H. Qin, Q. Zhang, and S. Wan, "The continuous Galerkin finite element methods for linear neutral delay differential equations," *Appl. Math. Comput.*, vol. 346, pp. 76–85, Apr. 2019.
[26] S. Wan, Y. Zhao, T. Wang, Z. Gu, Q. H. Abbasi, and K.-K. R. Choo, "Multi-dimensional data indexing and range query processing via Voronoi diagram for Internet of Things," *Future Gener. Comput. Syst.*, vol. 91, pp. 382–391, Feb. 2019.
[27] Z. Gao, D. Y. Wang, S. H. Wan, H. Zhang, and Y. L. Wang, "Cognitive-inspired class-statistic matching with triple-constrain for camera free 3D object retrieval," *Future Gener. Comput. Syst.*, vol. 94, pp. 641–653, May 2019.
[28] C. Ferreira, "Gene expression programming: Mathematical modeling by an artificial intelligence," *Eng. Appl. Artif. Intell.*, vol. 1, no. 3, pp. 223–225, 2002.
[29] Y. Cheng, X. Zhou, S. Wan, and K.-K. R. Choo, "Deep belief network for meteorological time series prediction in the Internet of Things," *IEEE Internet Things J.*, to be published.
[30] Y. Jiang, C.-S. Perng, T. Li, and R. N. Chang, "Cloud analytics for capacity planning and instant VM provisioning," *IEEE Trans. Netw. Service Manage.*, vol. 10, no. 3, pp. 312–325, Sep. 2013.
[31] Y. Jiang, C.-S. Perng, T. Li, and R. Chang, "Self-adaptive cloud capacity planning," in *Proc. IEEE 9th Int. Conf. Services Comput.*, Jun. 2012, pp. 73–80.
[32] Y. Jiang, C.-S. Perng, T. Li, and R. Chang, "ASAP: A self-adaptive prediction system for instant cloud resource demand provisioning," in *Proc. IEEE Int. Conf. Data Mining*, Dec. 2011, pp. 1104–1109.
[33] S. Wan, Y. Zhang, and J. Chen, "On the construction of data aggregation tree with maximizing lifetime in large-scale wireless sensor networks," *IEEE Sensors J.*, vol. 16, no. 20, pp. 7433–7440, Oct. 2016.
[34] R. Wiśniewski, G. Bazyło, and P. Szcześniak, "Low-cost FPGA hardware implementation of matrix converter switch control," *IEEE Trans. Circuits Syst., II, Exp. Briefs*, to be published.
[35] R. Wiśniewski, A. Karatkevich, M. Adamski, A. Costa, and L. Gomes, "Prototyping of concurrent control systems with application of Petri nets and comparability graphs," *IEEE Trans. Control Syst. Technol.*, vol. 26, no. 2, pp. 575–586, Mar. 2018.
[36] R. Wiśniewski, G. Bazyło, L. Gomes, and A. Costa, "Dynamic partial reconfiguration of concurrent control systems implemented in FPGA devices," *IEEE Trans. Ind. Informat.*, vol. 13, no. 4, pp. 1734–1741, Aug. 2017.

QIANMU LI received the B.Sc. and Ph.D. degrees from the Nanjing University of Science and Technology, China, in 2001 and 2005, respectively, where he is currently a Full Professor with the School of Computer Science and Engineering. His research interests include information security, computing system management, and data mining. He received the China Network and Information Security Outstanding Talent Award, in 2016, and multiple Education Ministry Science and Technology Awards, in 2012.

SHUNMEI MENG received the Ph.D. degree from the Department of Computer Science and Technology, Nanjing University, China, in 2016. She is currently an Assistant Professor with the School of Computer Science and Technology, Nanjing University of Science and Technology, Nanjing, China. She has published papers in international journals and international conferences, such as TPDS, ICWS, and ICSOC. Her research interests include recommender systems, service computing, and cloud computing.

SAINAN ZHANG is currently pursuing the degree with the Nanjing University of Science and Technology, China. Her main research direction is industrial control network security.

MING WU is currently pursuing the Ph.D. degree with the Nanjing University of Science and Technology, China. From 2017 to 2018, he was an Exchange Scholar with the School of Computing and Information Sciences, Florida International University. His research interests include data mining and crowdsourcing.

JING ZHANG received the B.Eng. degree from Anhui University, Hefei, China, in 2003, the M.S. degree in computer science from the Graduate University of Chinese Academy of Sciences, Beijing, China, in 2006, and the Ph.D. degree in computer science from the Hefei University of Technology, Hefei, in 2015. He is currently an Associate Professor with the School of Computer Science and Engineering, Nanjing University of Science and Technology. He has published dozens of articles in some prestigious journals, such as TNNLS, TKDE, TCYB, JMLR, and TMM, and top-tier international conferences, such as AAAI, SIGKDD, SIGIR, and CIKM. His research interests include data mining, machine learning, and their applications in business and industry.

MILAD TALEBY AHVANOOEY received the B.Sc. degree in software engineering from UAST, Semnan, Iran, in 2012, and the M.Sc. degree in computer engineering from IAU Science and Research, Tehran, Iran, in 2014. He is currently pursuing the Ph.D. degree in computer science with the Nanjing University of Science and Technology, Nanjing, China. From 2014 to 2016, he was a Lecturer with the School of Mathematics and Computer Science, Damghan University, Iran. His research interests include modern coding theory, text mining, text hiding, and genetic programming. He is also an External Reviewer for various international journals, including the IEEE Access, the *Computers in Human Behavior*, and the *KSII Transactions on Internet and Information Systems*.

MUHAMMAD SHAMROOZ ASLAM received the B.Sc. and M.S. degrees in electronics and electrical engineering from COMSATS University, Abbottabad and Attock campus, Pakistan, in 2009 and 2013, respectively. He is currently pursuing the Ph.D. degree in control science and engineering with the School of Automation, Nanjing University of Science and Technology, China. He was a Lecturer with the Department of Electrical Engineering, COMSATS University, Attock campus, from 2010 to 2015. His research interests include fuzzy systems, time-delay systems, non-linear systems, and network control systems.