ISSN: 2472-3703 DOI: 10.6911/WSRJ.202408_10(8).0010

Research on Application Methods of Network Fault Prediction Models

Yan Li[a], Jindai Qu[b], Kai Xin[c], Zirui Wang[d]

FAW Group Engineering and production logistics department-General Assembly Process Department, Changchun, China

[a]liyan41@faw.com.cn, [b]qujindai@faw.com.cn, [c]xinkai6@faw.com.cn, [d]wangzirui7@faw.com.cn

Abstract

With the rapid development of information technology, the internet has become an indispensable part of our daily life and work. However, the occurrence of network failures often has a significant impact on individuals, businesses, and even the country. Therefore, predicting network failures has become crucial. In recent years, machine learning and artificial intelligence technologies have achieved significant success in many fields, including network fault prediction. This article will explore the construction, application, and optimization of network fault prediction models.

Keywords

Network Failure; Prediction Model; Network Monitoring.

1. INTRODUCTION

The foundation of network fault prediction models is data. By collecting, analyzing, and processing data from network devices, we can identify patterns and trends of faults. These data may include network traffic, device temperature, CPU usage, memory usage, etc. By analyzing these data, we can establish a predictive model to predict future faults. Network fault prediction models have been applied in many scenarios, including the field of automotive monitoring[1][2]. Through predictive models, enterprises can detect potential faults in advance and take corresponding measures to prevent and solve them, thereby reducing losses caused by network failures. In addition, predictive models can also help enterprises optimize the configuration and maintenance of network equipment, improve the stability and efficiency of the network. Although existing network fault prediction models have achieved some success, there is still much room for optimization [3]. Firstly, we need to improve our data collection and processing methods to enhance the accuracy and completeness of the data. Secondly, we need to improve the algorithm and parameters of the model to enhance the accuracy and stability of predictions[4][5][6].

This article proposes a hybrid neural network model fault diagnosis method, which uses NetFlow technology to collect network traffic data and builds a hybrid neural network model composed of convolutional neural networks and artificial neural networks. Extract network traffic data using convolutional neural networks and detect network intrusion types using artificial neural networks [7]~[10]. The diagnostic method of this model has a lower false alarm rate than a single neural network algorithm model, making the detection more accurate.

2. SYSTEM TECHNICAL ARCHITECTURE

ISSN: 2472-3703 DOI: 10.6911/WSRJ.202408_10(8).0010

The platform adopts a multi-layer architecture and modular design pattern, with comprehensive system functions and independent module functions, which can be freely combined according to different needs[11]. At the same time, the system has good scalability, and can be seamlessly integrated with third-party products through third-party data interfaces, data buses, and portal portals. The main advantages are as follows:

2.1. The openness of the management system.

The system developed based on the ITIL architecture follows industry standards and provides management interfaces to achieve unified management of various resources and integration with other management software; Support application integration from certain third-party vendors, providing higher flexibility for system management selection[12][13][14]; The open Portal API supports interface integration of user application software, providing room for development in system management content expansion[15].

2.2. Security of management systems.

The security of the management system itself is a key factor in ensuring the normal operation of management work[16]~[19]. Therefore, when building the system, full consideration was given to the security of the management system, including:

Security login authentication and malicious login restriction mechanism.

Provide a complete strategy and framework that can adapt to organizational changes and flexibly set the roles and permissions of management personnel.

Save key information such as username and password in ciphertext.

2.3. Modular structure and scalability.

The scale of management will expand with the continuous expansion of applications, so the scalability of the system is of great significance for protecting investments. The scalability is mainly reflected in:

Expansion of management functions, which can expand resource management for network devices, hosts, applications, databases, power environments, business services, and more in the future[20].

Expansion of management scope, enabling the addition and expansion of multiple branches, and supporting distributed management. The system has excellent functional scalability and can add management modules, expand management nodes, and protect existing assets when needed.

The system architecture is divided into four levels: basic resource layer, data collection layer, analysis and processing layer, and display layer.

The menu function module of the system is divided into 9 items, including business management, topology management, resource management, and alarm management, and each item of data can be analyzed [21].

3. DESIGN AND IMPLEMENTATION OF NETWORK MONITORING SYSTEM

The design and implementation part of the network monitoring system is introduced from overall functions to sub modules, which are divided into two parts: the system business platform and the system algorithm platform.

3.1. Overall System Capabilities.

The network monitoring system is a monitoring platform that integrates data collection, data analysis, data calculation, data display, data monitoring, and early warning. The overall system functions are divided into two major parts: business module and algorithm module [22].

ISSN: 2472-3703 DOI: 10.6911/WSRJ.202408_10(8).0010

(1) Business module

The business module is written and implemented in Java code, with the main functions of monitoring and displaying the overall network situation, including monitoring network device mib library indicators, visual topology management, traffic monitoring of all device ports, hardware information status monitoring of network devices, monitoring of wireless devices, automatic refreshing of device resource changes, and automatic recognition of changing interface board information; System management, including user management, role management, permission allocation, menu management, and log management.

(2) Algorithm module

The algorithm module is implemented by Java nested Python code, providing API calls to the business module. The overall technical architecture is shown in Fig. 1.

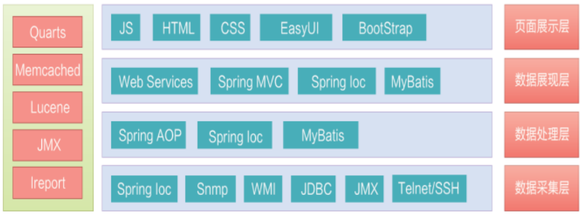

Fig. 1 System Technical Architecture.

3.2. System Business Platform.

The business platform is implemented using Java coding, and the mainstream development framework adopts SpringBoot+VUE architecture. The front-end and back-end are deployed separately. The relational database uses MySQL, the cache uses Redis, the message queue uses RabbitMQ, the middleware uses Kafka, and the big data storage uses Hadoop cluster technology stack, achieving monitoring and early warning of network data.

3.3. System algorithm platform.

The algorithm platform is implemented using Java integrated Python script coding. The purpose of this design is to fully utilize the flexibility and scalability of Python coding in big data analysis, and to fully utilize the shell of the Java framework to make it an independent service. It can also integrate with the provided API and business platform to achieve the effect of integrating models [23]. The technical approach adopted for the integration solution is to place Python code as a folder in the Resource folder of the Java framework, and call and execute Python scripts through the command line in the code.

4. NEURAL NETWORK ALGORITHM

The principle is to utilize the learning and prediction capabilities of neural networks to predict future trends and events. The neural network prediction model maps the input data to the specified output end by constructing multi-layer perceptrons, thereby achieving prediction [24]. Advantages: Neural networks can learn and simulate complex nonlinear relationships, which enables them to accurately predict complex trends and patterns. Neural networks can automatically extract useful features by learning historical data without the need for manual intervention. Neural networks can remember and learn historical trends and patterns, and consider these factors when making predictions in the future. Neural networks can process large-scale datasets and extract useful information from them. The specific technical scheme principle is shown in Fig. 2.

ISSN: 2472-3703 DOI: 10.6911/WSRJ.202408_10(8).0010

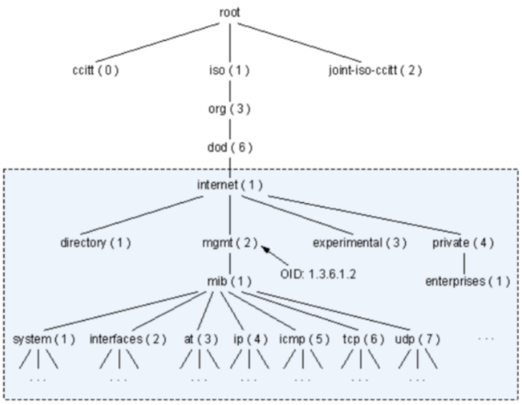

Fig. 2 Technical solution implementation.

5. APPLICATION METHODS

In the era of big data, algorithm models are widely used. The fusion model proposed in this article is widely used in the field of network monitoring, mainly focusing on monitoring device operation data. The following will elaborate on this application.

5.1. Network monitoring data

We can use information monitoring to support the monitoring and management of various mainstream middleware in the market. These middleware include, but are not limited to, Weblogic and TOMCAT in J2EE, RabbitMQ in message queues, reverse proxy server Nginx, Alibaba Nacos and Alibaba Gateway in configuration centers and service gateways, application servers JBOSS and Resin, distributed transaction processing system Tuxedo, web servers Apache and Resin, distributed object management system IceGrid, Java Enterprise Services SUN JES, collaboration platform Windows Sharepoint, and middleware tongweb [25].

Through data monitoring, we can obtain real-time information on the running status, performance indicators, and abnormal alarms of these middleware, helping us better grasp the health status of the entire middleware environment, timely discover and solve problems, and improve system availability and stability. At the same time, information monitoring can also provide statistics and analysis of historical data, providing effective reference basis for performance optimization and capacity planning.

5.2. Display of monitoring system

The network monitoring system based on the fusion model in this article is designed and implemented based on network monitoring data. As shown in Figure 3, the visual interface of the system after embedding the algorithm module into the business system is displayed, namely the model calculation interface and the model list interface.

Fig. 3 Indicator Monitoring Visualization Interface

ISSN: 2472-3703 DOI: 10.6911/WSRJ.202408_10(8).0010

In the system designed and implemented in this article, a fusion model solution is used to combine the business platform with the algorithm platform, and a visual interface for algorithm model operations is provided on the WEB end, providing users with an entry point to perform algorithm model operations, as shown in Figure 4.

Fig. 4 The entry point for executing algorithm model operations

Under the model list display menu, you can view all the operations that have been performed, view the operation parameters of saved or running models, pause the operation, or click Analyze to jump to the model creation page. The list clearly displays the model name, creator, creation time, and model running status, making it easy for users to view.

The model creation page is presented in the form of a flowchart, which allows users to clearly see the model calculation process, configure model parameters, select data sources, click on calculation to trigger a model operation, and after saving, add a new piece of data to the model list. The running progress can also be viewed through the calculation results, and the final result file can be exported.

6. CONCLUSION

This article integrates the business platform developed in Java and the algorithm platform developed in Python, and studies a fusion model optimized through technical solutions. Based on this, the design of a network monitoring system is implemented. This fusion model was proposed after conducting research on the current situation of the manufacturing industry, collecting user needs, and comparing and debugging Java and Python fusion solutions. Improvements and optimizations have been achieved in terms of functionality, technical architecture, technical operations, and business applications. The main achievements and innovative points are summarized as follows :

(1) Business requirement: Meeting the monitoring needs of users, it can monitor the real-time status of the network, network device port traffic, outbound and inbound traffic. By inputting time parameters, it can quickly view the curve graph of inbound and outbound traffic on a timeline, and intelligently identify the status of resources, including faults, configuration changes, additions and disappearance, achieving comprehensive visual monitoring of network resources.

(2) Technical implementation: Fully combining the advantages of two development languages. Compared to directly calling the Linux command line to trigger algorithm execution, embedding the command line and Python code into the Java framework to deploy independent services and provide external access APIs can achieve code decoupling, mutual independence, and no impact on each other. If there is a problem with the algorithm, it does not affect the interface requests of the WEB system. Both in terms of functionality and performance have been optimized.

ISSN: 2472-3703 DOI: 10.6911/WSRJ.202408_10(8).0010

(3) Deployment and operation: The basic business functions and algorithm parts are isolated, and the algorithm part is treated as a separate service, separate from the WEB visualization system, for easy code deployment and later system operation and maintenance.

(4) Widely applicable: The fusion model based monitoring system proposed in this article is suitable for monitoring both network data and device operation data, with a wide range of application data.

(5) Good user experience: The design and implementation of this fusion model provides users with a visual management interface for operating algorithm model execution, and provides operable buttons to control the model to start calculation, obtain calculation progress, export running results, and stop model operations. These all visualize the algorithm modules, bringing convenience to improving user experience.

(6) Strong scalability: The model platform interface has algorithm model configurable settings, allowing users to adjust the parameters of the computational model. This design concept also reflects the scalability of the fusion model, with parameters configurable, adjustable, modifiable, and viewable.

In summary, the network monitoring system based on the fusion model designed and implemented in this article is innovative and has achieved comprehensive monitoring of network devices. It has also been applied in vehicle enterprises and has practical reference value and exploratory significance. The network monitoring system that integrates models can not only be widely applied in the automotive industry, but also be extended to other fields such as finance and medicine [25].

REFERENCES

[1] Yang Qingchuan Engine Fault Detection Based on Deep Learning [D]. Hangzhou University of Electronic Science and Technology, 2021. DOI: 10.27075/dc.cnki. ghzdc.2021.000341

[2] Single Triumph Research on Engine Fault Prediction Method Based on Deep Learning [D]. Civil Aviation University of China, 2019. DOI: 10.27627/d.cnki. gzmhy.2019.000271

[3] Mei Benxiang Research on Monitoring and Management System for Vehicle Service Platform Based on MVC Mode [D]. Guangdong University of Technology, 2018

[4] Yu Zhisheng Automotive Theory [M] Version 5 Beijing: Machinery Industry Press, 2009

[5] Hu Tianjie Research on Model Based ECU Hardware in the Loop Simulation [D] Kunming: Kunming University of Technology, 2020

[6] Wang Wangyu Automotive Design [M] Version 4 Beijing: Machinery Industry Press, 2004

[7] Zhang Wenjie, Li Changlong, Hong Yu, et al. The Application of Remote Diagnosis System Based on Big Data Technology in Automobiles [J]. Automotive Digest, 2022 (4): 30-33

[8] Zhang Jiaqi Design and Implementation of a Vehicle Test Management Information System Based on Java Web [D]. Beijing: Beijing Jiaotong University, 2016

[9] Lu Shaobing. Research on Hybrid Language Programming and Its Implementation Based on Python [J]. Science and Technology Information, 2022, 20 (14): 31-33

[10] Zhang Wei. Implementation of Vehicle Detection Algorithm Based on Python [J]. Mechanical Management Development, 2019,34 (12): 258-261

[11] Cheng Xueqi, Jin Xiaolong, Wang Yuanzhuo, et al. Overview of Big Data Systems and Analysis Technology [J]. Journal of Software, 2014,25 (9): 1889-1908

[12] Zhao Li Design and Implementation of Vehicle Monitoring System Based on Big Data Processing [D]. Beijing: Beijing Jiaotong University, 2019

ISSN: 2472-3703 DOI: 10.6911/WSRJ.202408_10(8).0010

[13] Zeng Xianyu Optimization and Implementation of Vehicle Monitoring System Based on Big Data Technology [D]. Changchun: Jilin University, 2016

[14] Li Guojie, Cheng Xueqi. Big data research: a major strategic area for future science and technology and economic and social development - research status and scientific thinking of big data [J]. Journal of the Chinese Academy of Sciences, 2012, 27 (6): 647-657

[15] Zeng Lei. Review of Big Data Research [J]. Software Guide, 2015,14 (8): 1-2

[16] Fu Yan, Li Hongru. Research on the Digital Transformation System Architecture of Vehicle Enterprises [J]. Automotive Digest, 2022 (7): 20-26

[17] Wang Ruwei The Application of Java Programming Language in Big Data Development [J]. Electronic Technology, 2022, 51 (1): 160-161

[18] Guo Yang, Chang Yingxian. On the Application of Java Language in Computer Software Development [J]. Digital Communication World, 2022 (1): 88-90+94

[19] Zhang Jiaqi Design and Implementation of a Vehicle Test Management Information System Based on Java Web [D]. Beijing: Beijing Jiaotong University, 2016

[20] Li Jian Diagnosis of Gas Path Faults in Variable Cycle Engines Based on Deep Learning [D]. Shanghai Jiao Tong University, 2019. DOI: 10.27307/d.cnki. gsjtu. 2019. 001850

[21] Hong Jiyu Reliability analysis of aircraft engines based on deep learning [D]. Nanjing University of Aeronautics and Astronautics, 2018

[22] Zhou Yiren, Qiu Xiaolin, Guo Zhiqiang. Fault identification of engine valve mechanism based on deep learning theory [J]. Manufacturing Automation, 2017, 39 (11): 89-93

[23] Yan Bing The Application of Deep Learning in Aircraft Engine Fault Diagnosis [D]. Shanghai Jiao Tong University, 2017

[24] Ostrowski K, Birman K, Dolev D. Extensible Architecture for High Performance, Scalable, Reliable Publish Subscription Events and Notification [J] International Journal of Web Services Research, 2007, 4 (4): 18-58

[25] YANG J. Web service componentization [J] Communications of the ACM, 2003, 46 (10): 35-40