

融合语义金字塔与多层决策的跨平台配置稽核

徐玉华^{1, 2}, 徐杨^{1, 2}, 张梦^{1, 2}, 张正^{1, 3}, 孙知信^{1, 2}

¹ (南京邮电大学现代邮政学院 江苏省邮政大数据技术与应用工程研究中心, 南京 210003)

² (南京邮电大学现代邮政学院 国家邮政局邮政行业技术研发中心(物联网技术), 南京 210003)

³ (中国电信股份有限公司江苏分公司, 南京 210003)

E-mail: sunzx@njupt.edu.cn

摘要: 针对现有网络设备配置检测方法对于跨平台设备的配置错误识别精确度低、通用性差等问题, 本文提出了一种融合语义金字塔与多层决策的跨平台配置稽核方法。首先构建了由 RoBERTa 语义网络、特征金字塔、多头注意力机制及残差网络组成的动态语义金字塔模型, 实现配置文本多层跨尺度关键语义信息的特征提取与融合, 从而支持配置信息与模板信息之间的精准匹配; 其次提出了跨平台配置稽核的多层决策机制, 并设计了通用规则以实现各类网络设备配置文件的模板确定、段落定位和配置错误甄别。实验结果表明, 所提方法相比于现有方法在跨平台配置异常检测中具有更高的准确率, 并可以精准识别包含语句缺失、语句冗余、拼写异常和顺序错误的多种配置错误。

关键词: 语义金字塔模型; 多层决策; 配置稽核; 深度学习

中图分类号: TP391

文献标识码:

A

Integrating a Semantic Pyramid with Multi-level Decision-Making for Cross-platform Configuration Auditing

XU Yuhua^{1,2}, XU Yang^{1,2}, ZHANG Meng^{1,2}, ZHANG Zheng^{1,3}, SUN Zhixin^{1,2}

¹ (Post Big Data Technology and Application Engineering Research Center of Jiangsu Province, Modern Postal College of Nanjing University of Posts and Telecommunications, Nanjing 210003, China)

² (Post Industry Technology Research and Development Center of the State Posts Bureau (Internet of Things Technology), Modern Postal College of Nanjing University of Posts and Telecommunications, Nanjing 210003, China)

³ (China Telecom Corporation Limited Jiangsu Branch, Nanjing 210003, China)

Abstract: Given the low accuracy and poor versatility of the existing network device configuration detection methods for cross-platform configuration error identification, this paper proposes an approach to cross-platform configuration auditing by integrating a semantic pyramid with multi-level decision-making. Firstly, a dynamic semantic pyramid model consisting of RoBERTa semantic network, bidirectional feature pyramid, multi-head attention mechanism and residual network is constructed to realize the feature extraction and fusion of multi-layer cross-scale key semantic information of configuration text, thereby supporting the precise matching between configuration information and template in-formation. Secondly, a multi-layer decision-making mechanism for cross-platform configuration auditing is proposed, and general rules are designed to achieve template determination, paragraph positioning and configuration error identification of configuration files for various types of network devices. Experimental results show that the proposed method has a higher accuracy rate in cross-platform configuration anomaly detection than existing methods and can accurately identify different configuration errors, including missing sentences, redundant sentences, spelling anomalies, and sequence errors.

Keywords: dynamic semantic pyramid model; multi-layer decision-making; configuration audit; deep learning

收稿日期: 基金项目: 国家自然科学基金项目(62272239), 江苏省农业科技自主创新资金项目(CX(22)1007), 江苏省高等学校自然科学基金项目(22KJB520027), 南京邮电大学引进人才自然科学研究启动基金(NY222029)和贵州省科技支撑项目([2023]一般 272)。作者简介: 徐玉华, 女, 1989 年生, 博士, 讲师, CCF 会员, 研究方向为网络信息安全, 大数据智能分析等; 徐杨, 男, 2000 年生, 硕士研究生, 研究方向为深度学习、自然语言处理等; 张梦, 女, 2001 年, 硕士研究生, 研究方向为深度学习、网络信息安全等; 张正, 男, 1989 年生, 博士, 江苏电信 IT 架构师, 研究方向为网络运营智能化技术、网络大模型等; 孙知信, 男, 1964 年生, 博士, 教授, CCF 会员, 研究方向为物联网技术与应用、计算机网络及安全等。

0 引言

随着信息技术的迅猛发展以及5G、6G网络的广泛应用,通信网络的规模和复杂性显著增加^[1-3]。在应对增强型移动宽带、海量感知应用以及任务关键性应用等典型场景需求时,网络服务提供商通常与多个设备制造商合作,采用多种专有硬件和软件解决方案^[4-6]。当前的网络配置类型包括基础设施配置、安全配置、服务配置等,这些配置集中于配置文件中。由于配置类型多样,且不同厂商的设备协议和规范存在差异,甚至同一厂商的不同版本的配置文件也可能采用不同的书写规范,这使得配置过程中容易出现错误。传统的配置维护与异常检测方法严重依赖于运维人员的经验和技能,而随着网络的复杂性增加,人工操作导致的配置错误频发,消耗了大量的人力物力成本。因此,实现自动化和智能化运维已成为必然趋势^[7-9]。

在设备运维时,不同的设备厂家会提供相应的配置模板,即设备的配置标准文件。在配置稽核过程中,需要将人工完成的配置文件与配置模板进行比对以检测配置文件中是否存在错误信息。针对配置错误,相关厂商开发了基于自家设备的自动化网络巡检工具^[10],如思科的 Cisco Prime^[11]、瞻博网络的 Juniper Networks HealthBot^[12]和诺基亚的 Network Services Platform^[13]。这些企业解决方案主要基于语法规则对配置文件进行逐行检查,能够检测出拼写、命令格式和参数格式等简单异常。然而,这些工具在处理复杂的配置逻辑错误(如漏配、错配、多配和乱配)方面仍然存在局限性。此外,由于不同厂商和版本之间的配置文件存在显著差异,这些工具的跨设备平台的通用性受到限制,难以适应网络基础设施的快速升级。

与此同时,学界提出了多种配置检测的方法,主要包括基于静态分析、基于统计分析、基于语义关联分析与基于深度学习的配置检测方法。基于静态分析的方法依赖预定义的规则,通过解析网络配置文件并使用静态规则进行语法和逻辑检查。然而,由于依赖预定义规则,所以难以应对动态变化的网络环境和复杂的依赖关系^[14-16]。基于统计学的方法通过对配置数据的统计分布进行建模,检测异常配置,有效减少误报率,但不同设备的配置数据和配置语义差异性较大,基于统计学方法的模型泛化性较弱,处理非线性复杂配置和实时配置的能力有限^[17-18]。基于语义关联分析的方法通过挖掘配置命令之间的关联关系来检测异常,在捕捉配置间显性关联方面表现出色,但缺乏深度语义信息的提取以及隐性依赖的挖掘,导致该方法在新型或复杂配置场景中的覆盖面不足^[19-22]。基于深度学习的方法提升了复杂异常检测的自动化能力,但未考虑配置数据的内在逻辑关系,导致该方法无法识别具体的错误类型^[23-26]。虽然这些方法在各自的应用场景中具有一定优势,但在应对复杂且动态变化的跨平台网络

配置环境时仍存在检测精确度低、检测错误类型少、灵活性弱等局限。

针对以上不足,本文提出一种融合语义金字塔与多层决策的跨平台配置稽核方法,通过动态语义金字塔模型学习各类设备平台的配置模板的配置语法语义信息,实现待测配置文本信息与各类模板之间的有效匹配;通过多层决策方法实现具体配置错误的精准识别。

本文主要的研究工作如下。

1) 构建了动态语义金字塔模型,能够处理配置段落、语句等多种颗粒度的配置文本信息,生成包含丰富特征的文本向量,有效捕捉不同层次的语义信息,从而提升配置数据与模板数据之间的匹配精度。其中 RoBERTa 模块将配置语句转换为初始语义表示;自底向上特征提取层从局部到全局逐层提取特征,捕获不同粒度的配置文本结构与语法特性;自顶向下特征融合层将全局上下文信息反向传递,增强局部特征的语义表达;多头注意力机制在此基础上突出关键配置要素,确保模型关注重要的配置特征信息;残差连接则贯穿整个过程,保证信息流的完整性和稳定性。

2) 设计了配置稽核多层决策过程,从平台配置模板确定、段落定位、单个语句深层匹配到具体字符拼写识别,逐层处理配置数据,准确定位配置异常并精准识别缺失、冗余、拼写异常和顺序错误等多种配置错误。本文的多层决策过程不需要为每个设备编写特定的正则表达式,仅提供相应设备的配置标准文件,依靠动态语义金字塔模型进行语义匹配,通过模型训练和多层逻辑判断即可实现精准检测。

3) 本文设计了一系列实验验证所提出方法的有效性,通过消融实验评估了动态语义金字塔模型各组件的贡献,通过动态语义金字塔多层决策模型与主流预训练模型、现有配置稽核方法的对比实验,验证了本文方法能够准确检测多类设备的异常配置,并精准识别包括顺序错误、内容冗余、拼写错误、内容缺失等配置错误。

1 相关工作

近年来,随着网络规模和复杂性的不断增加,网络配置错误所带来的问题变得愈发突出。为了确保网络的稳定性和安全性,研究人员提出了多种网络配置检测方法,主要包括基于静态分析的方法、基于统计学的方法、基于语义关联分析的方法、基于深度学习的方法,如表1所示。

表1 网络配置检测方法功能对比

Table1 Functional Comparison of Network Configuration Detection Methods

功能点	基于静态分析 [14-16]	基于统计学 [17-18]	基于语义关联分析 [19-22]	基于深度学习 [23-26]	基于动态语义金字塔模型和多层次决策
配置异常预警	√	√	√	√	√
配置拼写错误检测	√	×	√	×	√
配置逻辑错误检测	√	×	√	×	√
语义特征深度提取	×	×	×	√	√
配置错误修改意见	√	×	√	×	√
跨平台通用性	×	×	√	√	√

1.1 基于静态分析的配置检测方法

基于静态分析的方法是一种通过检查配置文件中命令的语法和逻辑来检测配置错误的技术。这类方法主要根据平台配置文件的预定义规则来对配置文件进行检查。其步骤通常包括：将网络配置文件进行解析；使用一组静态规则对这些解析后的文本进行分析，检测出潜在的语法或逻辑错误；验证这些错误是否可能在实际网络环境中引发问题。

Fogel 等^[14]通过引入通用的配置分析方法，展示了如何将配置文件转化为数据平面模型以进行静态分析。文中提出的分析框架允许对复杂的网络配置进行语法和逻辑验证，并借助模型化的分析工具自动检测配置错误。Zhang, Khurat 等^[15-16]开发了自动化的网络安全配置验证工具，根据设备配置规则，分析网络配置文件的结构，验证其是否符合安全策略，并检测其中潜在的安全漏洞。

基于静态分析的方法能够有效发现语法错误和明显的逻辑错误，这类方法高度依赖设备配置文件预设的规则，难以适应复杂的网络环境，特别是含有多类平台设备时，静态规则的滞后性导致该类方法无法处理大规模数据，且缺乏跨平台通用性。

1.2 基于统计学的配置检测方法

基于统计学的异常检测方法通过对历史配置数据进行建模，识别潜在的异常配置。这种方法通常包括以下步骤：收集配置数据并进行统计建模；计算特定统计量，对数据分布进行分析；利用异常检测算法识别出异常配置点。该方法依赖于数据的统计特征，从而发现偏离正常模式的配置项。

Wang 等^[17]采用 Tukey 方法结合相对熵计算，分析配置数据的统计分布，识别超出正常范围的异常配置。Madhuri 等^[18]利用协方差矩阵对多维度配置进行分析，并通过假设检验发现与正常模式显著不同的异常配置，实现了数据中心环

境下大规模异常配置的准确检测。

基于统计学的异常检测方法在数据丰富且模式明确的场景中表现优异。然而，该方法依赖于大量的历史数据进行建模，当设备平台发生改变，数据特征也发生了变化，这就需要重新进行统计建模。此外，统计学方法在处理复杂的多维或非线性配置数据时往往难以捕捉异常行为的深层次特征，导致某些潜在的问题难以被发现。

1.3 基于语义关联分析的配置检测方法

基于语义关联分析的方法主要通过挖掘配置文件中的潜在关联关系，利用这些关系来检测配置中的异常。该方法一般包括以下几个步骤：对配置文件进行语义解析，提取出配置命令和参数之间的关联关系；运用关联规则挖掘算法对这些关系进行建模，生成频繁项集和强关联规则；利用这些规则检测异常配置，从而识别不符合正常关联模式的配置命令的正确性。

Yuan 等^[19]结合共现词频分析与 K-means 算法，提出通过构建配置语句树来分析网络设备的配置异常，该方法通过语义关联分析和机器学习相结合的方式，进行各类异常配置识别。He 等^[20]利用关联分析对大量配置数据进行建模，进而构建配置异常检测模型，通过关联规则模型实现大规模网络配置数据的异常检测。Xuanyuan 等^[21]提出了一种面向配置的语义表示方法，采用本体框架对配置中各部分的语义关联进行建模和分析，通过捕捉配置项之间的语义关系来实现配置异常检测。此外，Kobayashi 等^[22]提出了 dot2net 方法，该方法将网络配置表示为标记图，并通过与模板图进行比较来分析配置，利用图结构来捕捉和验证配置元素之间的关联关系并识别配置错误。

基于语义关联分析的方法可以挖掘配置项之间的语义关联关系识别各类配置错误，但该类方法主要捕捉显性的语义关联关系，无法进行深度语义特征的挖掘，对于隐形语法依赖关系特征提取不全面，例如配置语法信息，配置语句上下文之间的结构依赖关系等，因此该类方法的错误识别精度还有待提高。

1.4 基于深度学习的方法

基于深度学习的方法利用深度学习模型学习大量配置文本数据的深度语义信息，从而实现自动化的配置检测。该方法通常包括以下步骤：收集和预处理网络配置数据；构建并训练深度学习模型；利用训练好的模型对待检测配置进行评估。

Mai 等^[23]利用深度神经网络学习历史配置及其性能数据，用于评估以太网网络配置的可行性。Lan^[24]提出了一种结合注意力机制、双向 LSTM 和卷积神经网络的深度学习架构，能够有效检测文本中的异常行为。将该架构引入配置检测领域并使用配置数据进行训练，实现精准的配置异常告警。

与此同时,随着 BERT 模型^[27]的问世及其在自然语言处理领域的广泛应用,配置异常检测领域也迎来了新的发展契机。BERT 及其变体模型^[28-29]通过深度双向编码器,能够深入理解配置命令的语法和语义结构,为异常检测提供了更为精确的特征表示。Hussain 等人^[25]提出了一种基于再训练模型的深度学习方法,利用 BERT 模型学习命令的结构特征,通过区分命令中的固定和随机元素,有效提升了对命令异常行为的检测能力。Li 等^[26]提出了 PreConfig 模型,该模型利用专门针对网络配置数据进行预训练的模型来实现配置任务的自动化。

尽管基于深度学习方法可以实现网络配置检测的智能化和自动化,但仍存在一些挑战。配置文本相比于自然语言,语义信息较为有限,语法规则更加严格,然而现有的深度学习方法通过学习配置语义特征来进行异常识别,缺乏对配置语法以及配置文本结构特征提取,因而检测精度有待提高。此外,基于深度学习的方法未考虑配置数据的内在逻辑关系,

导致该方法无法识别各种错误类型,无法提供具体修改意见,仅适用于初步异常检测。

综上所述,当前网络配置管理中配置数据复杂多样,而传统的配置稽核方法仍存在检测错误类型少、跨平台通用性差、配置错误识别精度有待提高等问题。因此,本文将深度学习和多决策分析相结合,提出了动态语义金字塔模型实现各类设备的配置文本语法和语义特征深度挖掘与融合;并构建多层决策机制,将待测配置文本逐级细粒度分解,从全局段落异常到局部拼写错误逐层定位,高精度检测跨平台的各类配置错误并生成修复建议,实现配置文本的自动化和智能化检测。该方法旨在大规模跨平台通信网络中,通过自动化手段高精度识别网络设备配置文件中的潜在错误,而无需根据不同设备的语义语法规则编写大量的不同的正则表达式来进行错误检测。

2 融合语义金字塔与多层决策的跨平台配置稽核框架

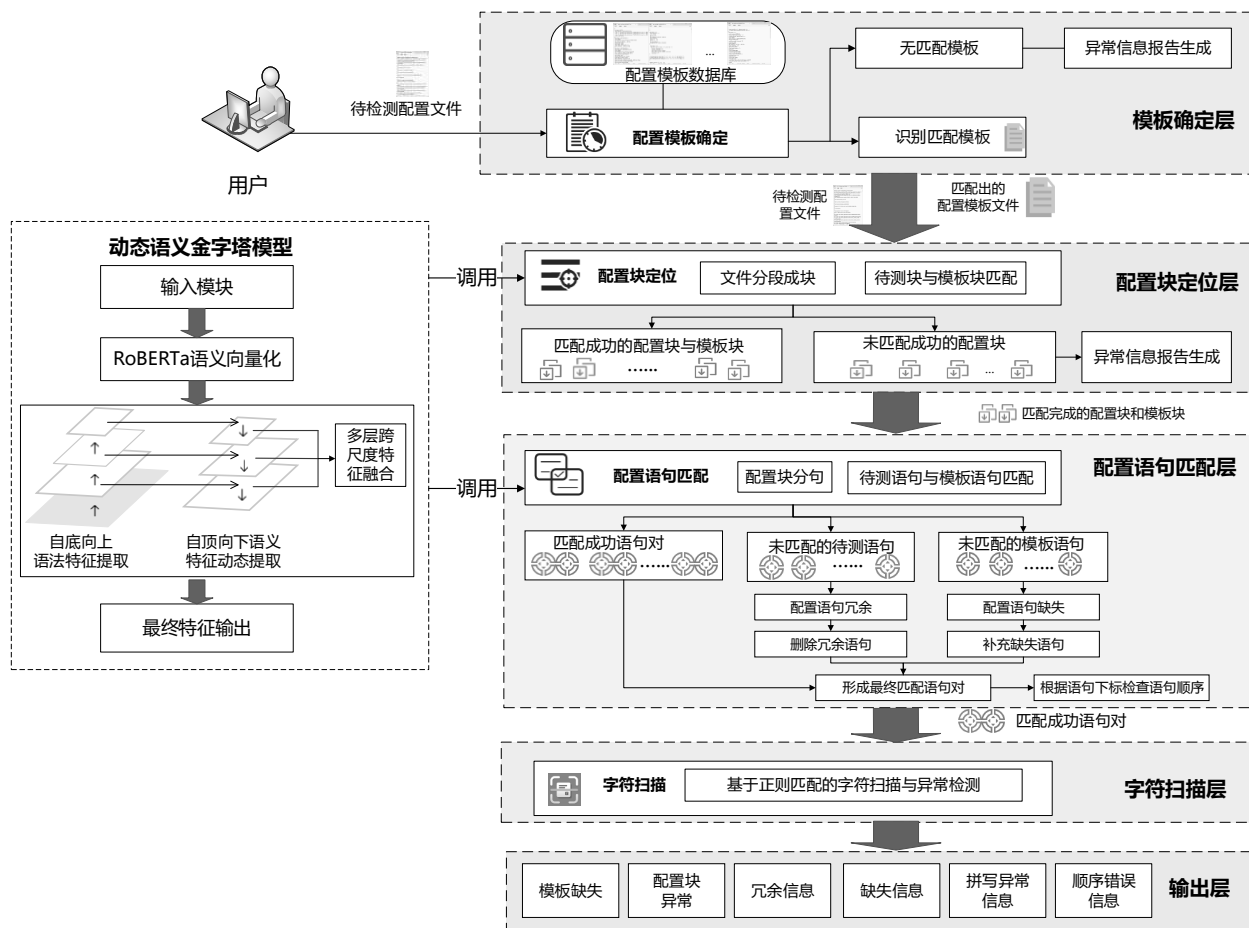


图 1 融合语义金字塔与多层决策的跨平台配置稽核模型架构图

Fig.1 Architecture of the Semantic Pyramid and Multi-level Decision Model for Cross-platform Configuration Auditing

图 1 展示了融合语义金字塔与多层决策的跨平台配置稽核的整体框架。待测配置文件一般只是配置模板中的部分内容,为了能够定位配置错误所在位置,该方案将配置文件进行文本切割,形成配置段落、配置语句等不同的颗粒度文本信息,并设计相应的处理模块,包括模板确定、配置段定

位、配置语句匹配,字符扫描匹配,通过逐层处理来逐步确定配置错误在待测文件和配置模板中的位置和范围。在确定错误位置后,设计了用于识别多种错误类型的通用的逻辑规则,从而实现面向不同网络设备的配置稽核。在配置段定位和配置语句匹配过程中,为了将待检配置信息与

配置模板间进行高效比对，设计了动态语义金字塔模型（Dynamic Semantic Pyramid Network, DSPN）以获取配置文本数据深度语义特征，然后通过相似度计算实现待测配置信息与模板信息的有效匹配。以下是该方法的整体流程：

（1）在模板确定层，通过段落标识符分析，从配置模板库中选择该设备对应的模板文件。若没有找到相应的模板，则进行异常报错。

（2）在配置段定位层，将待测的配置文件和确定好的模板根据段落形式切分为配置块和模板块，调用 DSPN 模型生成语义特征向量，通过特征向量相似度匹配，实现配置文件段落在模板上的定位，并筛选出乱码段落。

（3）在语句匹配层，将匹配成功的待测配置块和相关模板块进一步切分为各个语句，调用 DSPN 模型生成各语句语义特征向量，通过相似度计算进行语句对匹配，并检测语句的缺失、冗余；其次通过语句的位置差异检测语句顺序错误。根据这些语句错误进行自动修正，并更新语句对，使得待测语句与模板语句严格的两两匹配，以支持下一层字符的比对扫描。

（4）在字符扫描层，通过通用的正则表达式对语句中的字符进行扫描对比，识别语句中字符重复，缺失，拼写异常等错误。

（5）最后由输出层汇总各层识别出的异常，形成相应的错误报告和修改建议。

3 基于动态语义金字塔模型的配置文件信息多尺度融合特征提取

原始金字塔网络（Feature Pyramid Network, FPN）^[30]通过骨干网络将原始数据映射为特征图，并直接横向连接自下而上与自上而下的双向路径，实现图像的高层特征与浅层特征多尺度的融合，从而改善因图分辨率改变而带来的图像特征信息丢失的问题。由于在本方案多层决策过程中，需要对配置段落和配置语句进行特征提取，而这两类信息的颗粒度完全不同，类似于图像分辨率的改变，受此启发，为实现不同颗粒度的配置文本特征的有效提取，本文对 FPN 进行了改进，设计了适用于配置文本特征提取的动态语义金字塔网络 DSPN 模型。

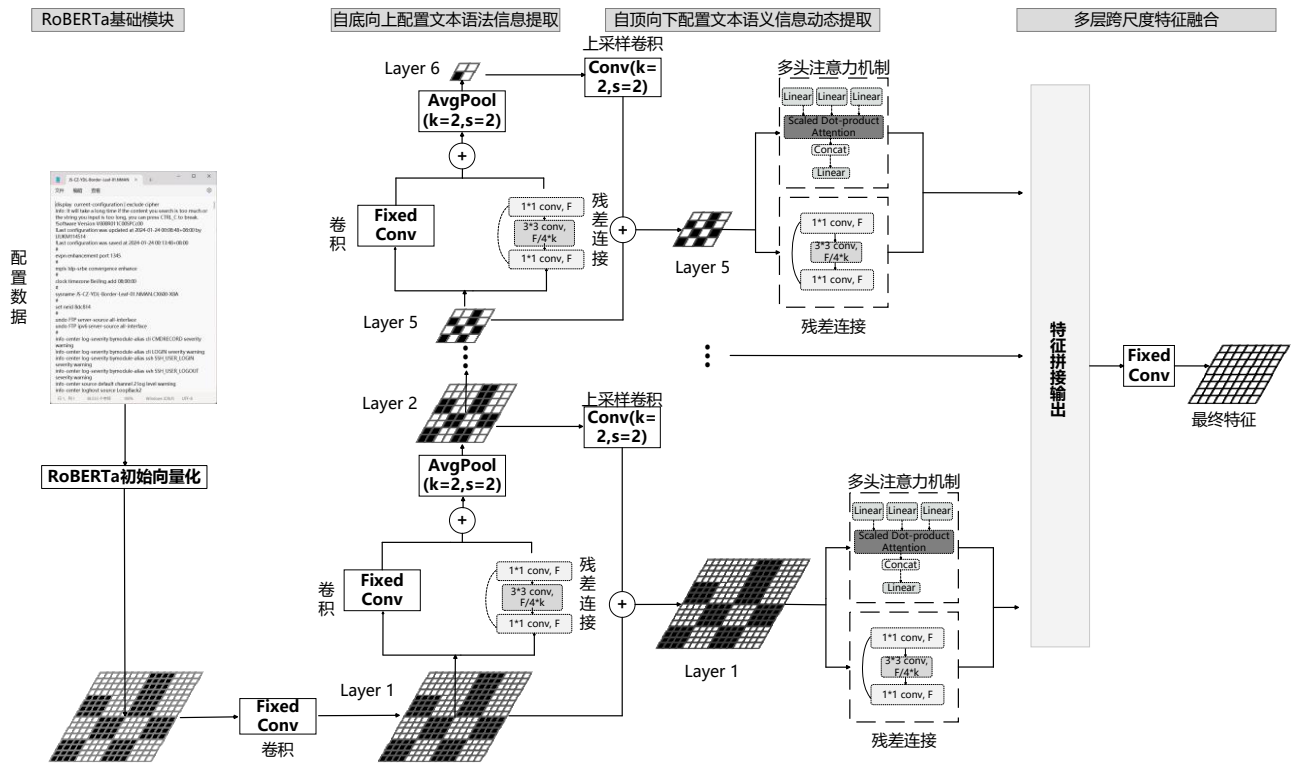


图2 动态语义金字塔模型架构图

Fig.2 Architecture of the Dynamic Semantic Pyramid Model

动态语义金字塔模型结构如图2所示，相比于 FPN，DSPN 模型在主干网络中增加了 RoBERTa 模型用于生成配置文本的基础语义向量；并保留了多层次的自底向上和自顶向下的双向路径结构，以提取包含语法规则、文本结构等信息的高层特征与包含配置文本内容语义信息的浅层特征；同时为保障网络模型的稳定性，提升多尺度特征表达能力，本

方案在双向结构中添加了残差连接^[31]，并在双向结构的每层横向连接处加入了多头注意力机制^[32]，实现上述特征信息的跨尺度关键特征的动态融合。通过这种设计方式增进不同网络层级的配置文本特征之间的关键信息传递，实现不同颗粒度配置文本特征有效提取，从而支持多层决策中待测配置文件的段落与语句在配置模板中的定位与匹配。

3.1 RoBERTa 基础模块

RoBERTa^[28]基础模块作为动态语义金字塔模型的初始模块,负责提取配置文件的基础语义特征。RoBERTa 采用字节对编码可以避免对罕见词的过度拆分,提升对配置文件特殊字符的处理能力,且该模型在每个训练周期动态生成不同的掩码,有利于单句内部的细粒度语义建模。因此本文选择 RoBERTa 模型提取配置文件的基础语义特征信息。

本文首先根据原始配置文件内容按段落分隔符分割得到的独立文本单元即配置块,再将配置块文本序列输入至此模块。

该模块首先对输入的配置块文本执行分词操作,将其转化为 RoBERTa 模型可处理的子词序列。随后,每个子词 w_i 通过嵌入层 E 映射为一个初始的向量表示 e_i ,如式(1)所示:

$$e_i = E(w_i) \quad (1)$$

为了使模型能够感知序列中子词的位置信息,初始的词嵌入 e_i 与其对应的位置编码 p_i 进行相加融合生成 $h_i^{(0)}$,如式(2)所示:

$$h_i^{(0)} = e_i + p_i \quad (2)$$

接下来,将包含位置信息的向量序列 $h_i^{(0)}$ 输入到 RoBERTa 模型内部堆叠的多层 Trans-former 编码器中。每一层 Transformer 通过自注意力机制等复杂运算^[26],深度建模配置文本序列内部的上下文依赖关系,并迭代优化,生成更具上下文感知能力的输出表示 $h_i^{(l)}$,如式(3)所示:

$$h_i^{(l)} = \text{TransformerLayer}(h_i^{(l-1)}) \quad (3)$$

此过程有 L 层,最终得到配置文本特征序列 $\{h_1^{(L)}, h_2^{(L)}, \dots, h_n^{(L)}\}$ 。

为获得代表整个输入序列的单一固定维度向量表示 V_s ,将配置文本特征序列 $\{h_1^{(L)}, h_2^{(L)}, \dots, h_n^{(L)}\}$ 进行平均池化操作,如式(4)所示:

$$V_s = \frac{1}{n} \sum_{i=1}^n h_i^{(L)} \quad (4)$$

通过以上步骤,利用 RoBERTa 基础模块将配置文本转化为包含基础语义信息的固定维度的向量。

3.2 自底向上配置文本语法信息提取

通过 RoBERTa 基础模块生成初始的语义向量后,动态语义金字塔模型建立自底向上结构将低层信息传递到高层的特征,低层高密度的语义特征中包含了丰富的配置语法规则、文本结构信息等,这些信息对于配置文本的精确匹配非常重要。通过自底向上结构,逐层提取低层特征关键信息,并传输至高层,使得高层特征提炼出具有长距离依赖的配置语法特征信息。

在自底向上的结构中,从最底层特征开始,逐层向上传递信息。对于每一层,由卷积操作提取局部细节特征,同时由残差连接提取残差特征,并将两类特征相加合并后,通过池化层进行降采样提取关键信息传输至上一层。其中,卷积

与残差连接的结合实现全局语法信息的提取与局部信息互补,同时缓解梯度消失。

初始特征提取表示为:

$$F_0 = \text{ReLU}(W_0 * V_s + b_0) + V_s \quad (5)$$

初始特征 F_0 由 RoBERTa 的输出 V_s 经过卷积 W_0 和偏置 b_0 处理后,通过 ReLU 激活函数生成。

初始层后的每层特征提取可以表示为:

$$F_i = \text{AvgPool}(\text{ReLU}((W_i * F_{i-1} + b_i) + \text{Residual}(F_{i-1}))) \quad (6)$$

每一层的输出 F_i 由前一层的输出 F_{i-1} 经过一个使用固定核大小的卷积核(权重为 W_i ,偏置为 b_i)进行卷积,并与残差网络 Residual 的提取特征相加。

其中,残差网络的特征提取形式表示为:

$$\text{Residual}(F_{\text{input}}) = F_{\text{input}} + W_{\text{res}} \cdot F_{\text{input}} + b_{\text{res}} \quad (7)$$

F_{input} 表示当前层的输入, W_{res} 和偏置 b_{res} 表示残差卷积的权重和偏置项。残差网络将其输入特征与经过变换后的输出特征相加,能够有效保留原始特征信息,有助于缓解深度网络训练过程中的梯度消失问题,增强模型的稳定性,确保信息在网络深层能够顺畅流动,因此在本模型的双向结构中都加入了残差连接。

卷积特征和残差特征相加之和通过 ReLU 激活函数后,进行平均池化。平均池化层使用大小为 2、步幅为 2 的池化核进行操作,将特征图的序列长度减半,在自底向上的过程中逐步缩小特征表示,形成特征金字塔。这一过程不断重复,直到达到预定的层数,有效捕捉不同尺度的特征。

3.3 自顶向下配置文本语义信息动态提取

自底向上配置文本语法信息特征提取完成后,模型自顶向下通过上采样拓宽高层特征维度,并与下层的高维配置语义特征相加融合,从而在语义特征中突出语法信息,以此强化特征表达。然后,利用多头注意力机制实现对融合信息的全局关键语义特征动态提取,并通过残差连接实现局部细节信息的补充,从而实现全局与局部特征相融合的配置语义信息动态提取。

自顶向下结构中,上采样操作通过核大小为 2、步幅为 2 的单层转置卷积,将上层特征 F_{i+1} 的序列维度加倍,使其与下层的特征 F_i 具有相同的序列维度,然后两者逐元素相加融合,如式(8)所示。

$$F'_i = \text{Upsample}(F_{i+1}) + F_i \quad (8)$$

将融合后的特征送入多头注意力机制,通过计算特征间的权重分配,从而动态获取全局关键特征,如式(9)所示。

$$\begin{aligned} Q_i &= \text{Conv}_Q(F'_i) \\ K_i &= \text{Conv}_K(F'_i) \\ V_i &= \text{Conv}_V(F'_i) \\ \text{Attention}(Q_i, K_i, V_i) &= \text{softmax}\left(\frac{Q_i K_i^T}{\sqrt{d_k}}\right) V_i \end{aligned} \quad (9)$$

其中, Q_i , K_i , V_i 分别代表查询、键和值, 这些都是通过不同的一维卷积层 (固定核大小为 1) 从 F'_i 中生成的, softmax 函数确保了权重的正规化。

将通过多头注意力机制得到的全局关键特征与残差网络特征相加, 并通过 ReLU 激活函数, 以增强特征表示的完整性和关联性, 提升模型对复杂结构文本的语义理解如式 (10) 所示。

$$F''_i = \text{ReLU}(\text{Attention}(Q_i, K_i, V_i) + \text{Residual}(F'_i)) \quad (10)$$

3.4 多层跨尺度特征融合

DSPN 模型通过自顶向下结构形成跨尺度的多层特征, 将这些特征进行对齐拼接, 旨在整合不同颗粒度的语义特征信息, 使最终特征同时包含小颗粒度的局部语义细节信息和大颗粒度的全局结构信息, 避免单一层次特征的局限性, 提升动态语义金字塔深层网络的特征表达能力。因而模型能够有效捕捉配置文本中从语句到段落的多层级结构特征, 满足不同大小配置文本的匹配计算需求。

首先, 将自顶向下路径生成的所有不同层级的精炼特征 $\{F''_1, F''_2, \dots, F''_{k-1}\}$ (其中 k 为金字塔的层数) 进行拼接。由于不同层级的特征具有不同的空间维度, 需要先通过填充 (Padding) 将它们对齐到这些特征中的最大空间尺寸。然

后, 将对齐后的特征沿着通道维度进行拼接, 形成一个包含了所有尺度信息的综合特征, 如式 (11) 所示。

$$F_{\text{concat}} = \text{Concat}(\{F''_1, F''_2, \dots, F''_{k-1}\}) \quad (11)$$

然后通过一个最终的单个卷积层 (使用固定的核大小) 对拼接后的特征 F_{concat} 进行卷积操作, 生成最终的特征表示 F_{final} 。

$$F_{\text{final}} = \text{Conv}(F_{\text{concat}}) \quad (12)$$

最终特征 F_{final} 将用于下游的配置匹配和分析任务。

3.5 损失函数设置

动态语义金字塔模型的损失函数设置采用自监督学习框架, 目标是使模型能够在没有外部标注的情况下, 通过内部生成的信号进行学习。损失函数的核心是通过一个自定义的损失函数来指导学习过程, 该损失函数采用滚动预测的策略, 即模型需要预测序列中被滚动过的下一个状态, 以此来捕捉配置数据中的长期依赖关系。

$$L = \frac{1}{N} \sum_{i=1}^N (y_i - \text{roll}(y_i))^2 \quad (13)$$

其中, y_i 是模型对第 i 个数据点的预测, $\text{roll}(y_i)$ 是该数据点序列中的下一个实际状态。

4 基于多层决策的配置稽核

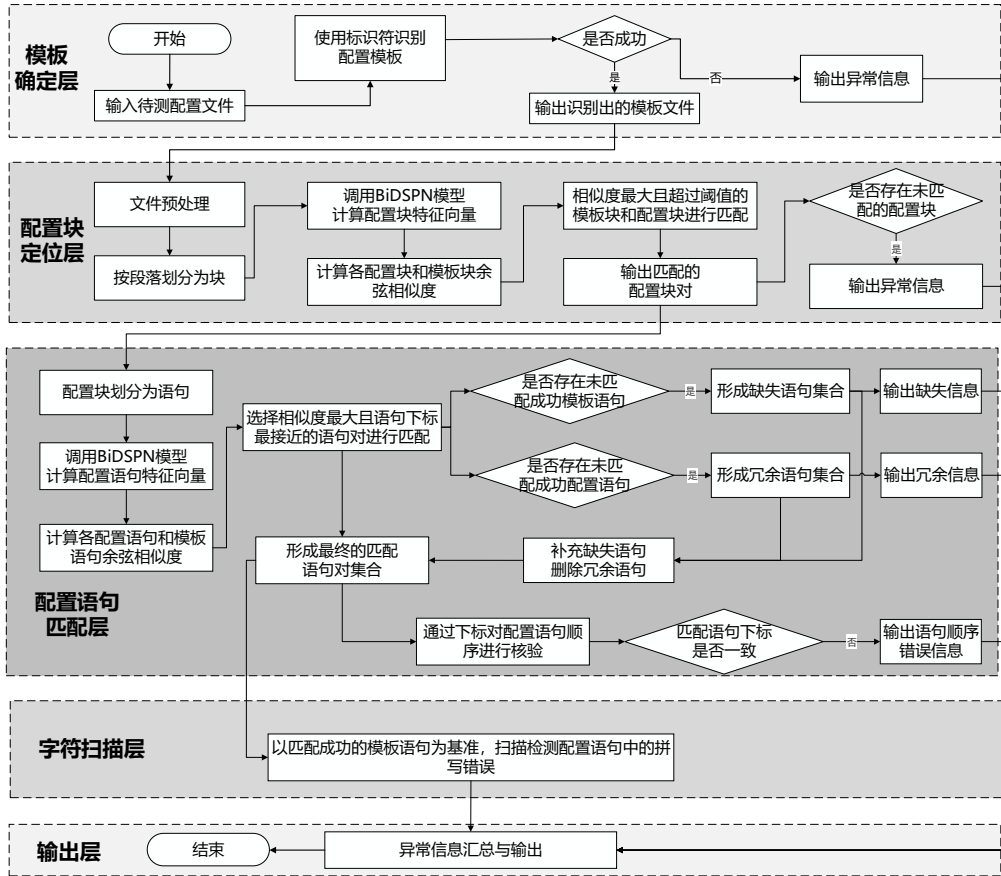


图3 基于多层决策的跨平台配置稽核

Fig.3 Cross-platform Configuration Auditing Based on Multi-level Decision

为了保障设备的可靠运行, 配置稽核过程不仅要能够预警配置文件的异常, 还要通过比对相关配置模板, 对配置

文件中的错误进行定位, 提供具体的配置错误信息, 以帮助技术人员进行快速的设备维护。这就需要首先根据待检

测的配置文件来确定相关的配置模板，即配置的标准文件，若无相关模板则说明配置文件异常；其次在确定配置模板后，需要对文件中的错误进行定位和识别。传统的配置错误识别方法，一般需要根据不同的设备配置模板逻辑结构、语法规则等编写多个的正则表达式来对配置文件所进行扫描识别，由于不同的网络设备的配置文件存在显著差异，这种方法费时费力，因此亟需一种跨平台的通用方法实现多类设备配置文件的错误定位与识别。

在实际生产中，待测配置文件往往是配置模板中的一部分或几部分片段，且配置文件中可能存在不同颗粒度的多种错误，例如大颗粒度的段落乱码，中颗粒度的语句缺失、语句重复或者语句顺序错误，小颗粒度的拼写错误等。为实现不同颗粒度配置错误的定位与识别，本文采取从全局到局部逐层分解、分而治之的思想，设计了一种通用的多层决策配置稽核架构。该架构通过分层处理，将复杂的配置稽核任务分解为多个相对简单的子任务，包括模板确定、配置块定位、配置语句匹配、字符扫描等，每层子任务专注于解决特定问题从而提高整体的处理效率和准确性。各层任务之间根据配置文件处理的颗粒度的大小具有上下级关系，只有当大颗粒度的配置文件信息处理完毕后，才能将大颗粒信息分解为小颗粒度来进行处理，通过逐层细化，不断缩小异常信息的搜索范围，有效定位和识别相应的配置错误，并汇总各层的检测结果，生成全面的错误报告和修改建议，具体工作流程如图3所示。

4.1 模板确定层

# forward alarm cpu-usage multi-core threshold 95 forward alarm vcpu-usage multi-core threshold 95 # ptp enable ptp domain 24 ptp device-type bc ptp passive-measure enable ptp clock-source local priority2 200 # clock freq-deviation-detect enable clock ethernet-synchronization enable #	# address-family ipv4 route-distinguisher 65448:18030 apply-label per-instance # address-family ipv6 route-distinguisher 65448:18030 apply-label per-instance #
---	---

图4 华为某配置文件部分语句（左）和华三某配置文件部分语句（右）

Fig.4 Sample Configuration Statements: Huawei (Left) and H3C (Right)

各厂商的配置文件呈现明显的段落化结构，每一段配置信息都通过段落标识符进行分割，因此段落标识会有频繁的出现于配置文件中，且不同的厂商段落标识符具有明显的差异性。如图4所示，华为路由器配置文件通过“#”对配置信息进行分段；而华三路由器配置文不仅通过“#”进行分段，每条配置信息都通过空行进行分隔。根据这一特征，本文利用段落标识符来识别配置文件所属的厂商设备，并选择对应厂商设备的模板文件。

本文首先收集了各厂商设备的配置文件段落标识符集

合，如表2所示。

表2 部分设备配置段落标识符集合

序号	设备	段落标识符集合
1	中兴通讯路由器	{\$, !<*>}
3	华为路由器	{#}
4	华三通信路由器	{#, 空行}
5	思科路由器	{!, 空行}
.....

其次，通过统计待检测的配置文件的各符号在各集合内的出现次数总和，计算该文件与各厂商的匹配分数，计算公式如下：

$$Score_i = \sum_{j=1}^n w_{ij} \begin{cases} w_{ij} = 1, & \text{if } p_j \in L_i \\ w_{ij} = 0, & \text{if } p_j \notin L_i \end{cases} \quad (14)$$

其中， $Score_i$ 标识配置文件与第*i*个厂商的匹配分数。 p_j 表示配置文件中的第*j*个符号。 L_i 表示第*i*个厂商的段落标识符集合。 w_{ij} 表示配置文件中的第*j*个符号是否为第*i*个厂商的段落标识符的分数，如果是，则 $w_{ij} = 1$ ，否则为0。匹配分数计算完成之后，选择匹配分数超过设定阈值且最高的厂商作为配置文件的归属厂商，并选择该厂商对应的模板文件。

4.2 配置块定位层

由于配置文件一般是配置模板中的一段配置内容或几段配置内容的组合，通过配置块定位层可以在配置模板中找出其相应的段落。首先对接收到的配置文件及其模板进行预处理，包括删除空白行、特殊字符串等，减少这些噪声信息对于配置段落匹配的影响。

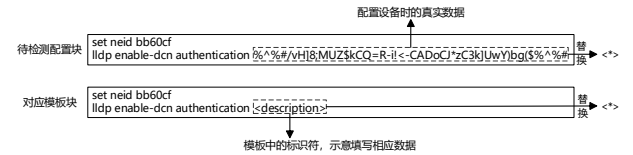


图5 通用标识符<*>替换示意图

Fig.5 Illustration of Substitution with the Generic Identifier

<*>

如图5所示，模板中的IP、端口等信息一般由标识符<XXXX>或<description>表示，在预处理过程中将这些符号直接替换为通用标识符<*>。而待检测的配置文件中IP、端口等信息是具体的字符串且字符串长度超过一般的词语，因此需要扫描配置文件中的字符串长度，超过设定的数值时，也被替换为通用标识符<*>。通过上述方法实现配置文件和模板中特殊字符串处理。

预处理完成后，根据段落标识符将配置文件和模板进行段落划分，分别形成配置块和模板块，然后采用动态语义金字塔模型计算双方的语义特征向量，计算公式如下：

$$V_{C_{b_i}} = F_{final}(C_{b_i}) \quad (15)$$

$$V_{T_{b_j}} = F_{final}(T_{b_j}) \quad (16)$$

其中, $V_{C_{b_i}}$ 表示当前的配置文件 C 的第 i 个配置块 b_i 的语义特征量; $V_{T_{b_j}}$ 表示配置模板 T 第 j 个模板块 b_j 向量; F_{final} 表示动态语义金字塔模型的输出函数。

然后计算单个配置块向量和模板块向量余弦相似度。余弦相似度的计算公式如下:

$$Sim_{vec}(V_{C_{b_i}}, V_{T_{b_j}}) = \frac{V_{C_{b_i}} \cdot V_{T_{b_j}}}{\|V_{C_{b_i}}\| \cdot \|V_{T_{b_j}}\|} \quad (17)$$

最后, 选择余弦相似度最高的模板块与配置块对, 并判断该相似度值是否达到设定阈值, 若达到, 说明配置块与该模板块之间具有较高的相似性, 选择此模板块作为配置块的比对基准, 即实现了配置文件段落在相应模板中的定位。若没有达到阈值, 说明该配置块与模板块的差异度过大, 则向输出层反馈异常信息。

4.3 配置语句匹配层

在一个配置段落中, 配置语句前后顺序和字符格式是固定的, 因而需要找到每个待检测语句的比对锚点, 还要确定该段落中语句顺序是否正确, 是否存在冗余或缺失语句。因此, 在配置语句匹配层, 将形成对应关系的配置块和模板块通过换行符进行语句切割, 并将待测语句与模板语句进行逐一匹配, 为后续的字符扫描检测提供参照。语句匹配过程有两个条件, 首先是两个语句的向量相似度最高且超过阈值; 其次是语句下标索引最接近, 如果有多模板语句与待测语句向量相似度最高且下标最接近, 则选下标较小的语句与待测语句进行匹配。

其中, 设待测配置块的语句集合为 $C_{b_i} = \{s_1, s_2, \dots, s_M\}$, 其中 s_g 表示待测配置块中第 g 条语句; 模板配置块的语句集合为 $T_{b_j} = \{t_1, t_2, \dots, t_N\}$, 其中 t_k 表示模板块中第 k 条语句。

各语句通过语义金字塔模型计算其特征向量值, 公式如下:

$$V_{s_g} = F_{final}(s_g) \quad (18)$$

$$V_{t_k} = F_{final}(t_k) \quad (19)$$

其中, V_{s_g} 表示配置块的第 g 条语句的语义特征向量, V_{t_k} 表示模板块中第 k 条语句的语义特征向量。 F_{final} 表示动态语义金字塔模型的输出函数。

然后, 计算各配置语句与模板语句向量余弦相似度, 公式如下:

$$Sim_{vec}(V_{s_g}, V_{t_k}) = \frac{V_{s_g} \cdot V_{t_k}}{\|V_{s_g}\| \cdot \|V_{t_k}\|} \quad (20)$$

对于每个待测语句 s_g 先找出与它向量相似度最高且超过设定阈值的那些模板语句形成集合, 如公式 (21) 所示:

$$T_{\max_sim}(s_g) = \{t_k | \max_{k=1, \dots, N} Sim_{vec}(V_{s_g}, V_{t_k})\} \quad (21)$$

并在向量相似度最高的模板集合 $T_{\max_sim}(s_g)$ 中, 通过式 (22) 计算各模板语句与待测语句 s_g 的下标接近程度。

$$Sim_{idx}(s_g, t_k) = \frac{1}{|k - g| + 1} \quad (22)$$

找出使得 $Sim_{idx}(s_g, t_k)$ 最大的模板语句 t'_k 作为待测配置语句 s_g 的匹配模板语句。若存在两个 t'_k 则选择下标较小的模板语句, 即 $i' = \min\{i | i \in \argmax_{t_i \in T_{\max_sim}(s_g)} Sim_{idx}(s_g, t_k)\}$ 。此时 t'_k 为待测语句 s_g 的最终匹配模板语句, 后续基于 t'_k 对 s_g 进行进一步修正。

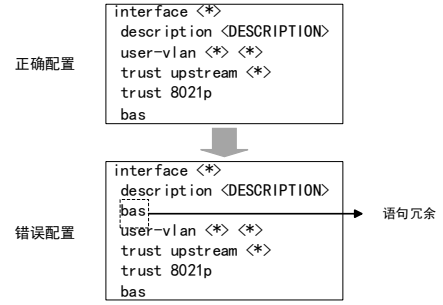


图 6 语句冗余示意图

Fig.6 Diagram of Statement Redundancy

匹配成功的语句对记录在集合 $P = \{(s_g, t_k) | g = 1, 2, \dots, M; k = 1, 2, \dots, N\}$ 中。待测配置块中存在未匹配成功的语句, 即 $R = \{s_g | s_g \notin \{s: (s, t) \in P\}\}$, 表示这些语句在模板块中没有出现, 根据规范, 配置块中的所有语句都应有模板语句对应, 因此这些空闲的配置语句为冗余语句, 应当删除, 如图 6 所示。

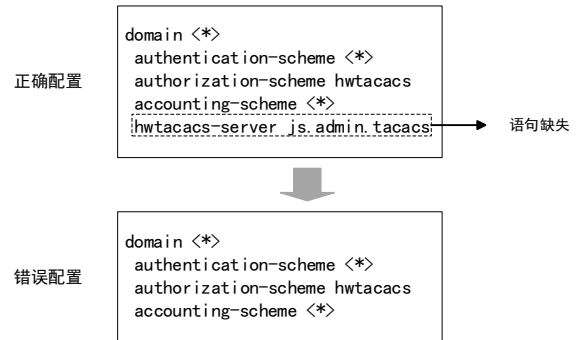


图 7 语句缺失示意图

Fig.7 Illustration of a Missing Statement

在模板块中存在未匹配成功的语句, 即 $M = \{t_k | t_k \notin \{t: (t, s) \in P\}\}$, 表示这些语句在待测的配置块中没有出现, 根据配置规范, 这些模板语句应当存在于配置文件中, 因此这些模板语句作为缺失语句, 如图 7 所示。

在配置块中删除冗余语句并添加缺失语句后, 形成最终语句匹配集合。将修复后的配置块的语句下标与模板块的语句下标进行核验, 检测语句的顺序是否存在错误如图 8 所示。

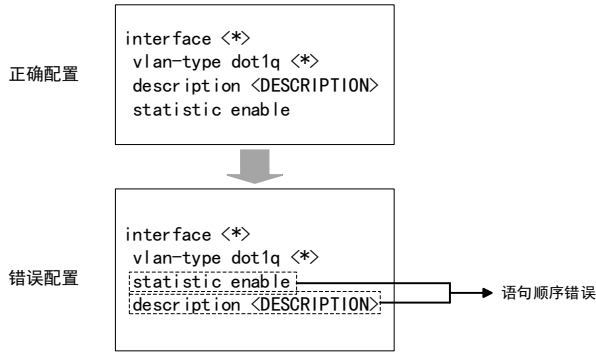


图 8 语句顺序错误示意图

Fig.8 Illustration of Incorrect Statement Order

4.4 字符扫描层

在字符扫描层，检测配置语句中的拼写异常，包括单词拼写错误、词缺失错误、词冗余错误等。针对语句匹配层中配置语句与模板语句的匹配集合 $P = \{(s_g, t_k) | g = 1, 2, \dots, M; k = 1, 2, \dots, N\}$ ，将配置语句与模板语句通过正则表达式进行字符扫描比对，若出现不一致则说明该配置语句存在拼写异常。

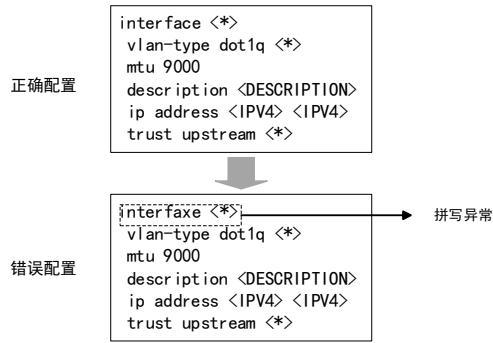


图 9 拼写异常示意图

Fig.9 Illustration of a Spelling Error

4.5 输出层

输出层的主要功能是整理和生成配置稽核信息，并通过用户界面反馈给用户。其中，若存在模板缺失的情况，则输出该配置文件无对应模板。若配置块未定位到对应的模板块，则输出该配置块无对应模板块，可能存在乱码，需人工检查。若能匹配到相应模板块或语句，则输出匹配到的内容和相似度分数、对应的具体错误和所处的文件位置，及其相应的修改建议。

5 实验分析

本文的实验过程如图 10 所示，主要包括：（1）动态语义金字塔模型跨平台语义识别实验；（2）动态语义金字塔模型消融实验；（3）动态语义金字塔多层决策模型跨平台配置稽核实验。本文的实验环境为 32GB 内存，NVIDIA GeForce RTX 4080 GPU，Intel Core i9-14900KF CPU 的 Ubuntu 22.04.4 LTS 服务器，模型在 Python 3.10.14、PyTorch1.12.1 环境下运行。

5.1 数据准备

5.1.1 数据搜集

本文所使用的网络设备配置与模板文件来自华为、思科、中兴和华三通信四家厂商，涉及多类设备，其中华为设备配置模板文件 9 个，华三通信设备配置模板文件 5 个，中兴设备配置模板文件 9 个，思科设备模板文件 1 个。这些模板文件涵盖了路由器在实际部署中常见的多种功能特性，包括 MPLS、BGP、ISIS、ACL、SSH、NetConf、Telemetry、SRv6 等高级功能，并涉及了 IPv4/IPv6 路由、VPN 实例、AAA 认证与授权、接口管理、访问控制策略等多方面的网络配置内容。由这些模板文件形成 6979 个模板块和 99802 条模板语句，并收集了 301258 个配置块和 2299303 条配置语句。

上述四家厂商多个设备的配置文件横向上覆盖不同厂商的语法差异与功能侧重点，纵向上包含了从基础网络配置到高级路由策略、跨域互通、安全及运维管理等多层面的配置信息，为后续的跨平台语义提取与错误检测实验提供了充分的真实数据支撑。

5.1.2 数据集划分

表 3 测试数据集 1 明细

Table3 Test Dataset 1: Details

厂商	数据集类型	测试配置块数目	测试配置语句数目
华为	正常	440	1720
	异常	440	1720
思科	正常	160	640
	异常	160	640
中兴	正常	120	480
	异常	120	480
华三通信	正常	80	320
	异常	80	320

（1）训练数据集

训练数据集包含所有模板文件和所搜集的配置文件。该数据集用于模型预训练，使模型能够学习配置文本的深度语义信息。

（2）测试数据集 1

测试数据集 1 用于评估模型提取语义特征向量的性能。如表 3 所示，为了全面评估模型在不同粒度文本上的语义理解能力，该测试集包含配置块数据集和配置语句数据集两部分。模型将配置块和配置语句向量化，计算相似度寻找最优匹配模板配置块和配置语句的方式验证模型性能。在测试集包含正常数据集和异常数据集。其中，各厂商的正常数据集是从其各模板文件中随机抽取相同数量的模板块和模板语句组合生成，各厂商的异常数据集则包含相同比例的缺失、冗余、拼写错误和顺序四类错误。

表 4 测试数据集 2 明细

Table4 Test Dataset 2: Details

厂商	错误类型	配置块数目	配置语句数目
华为	缺失	350	13341
	冗余	350	14041
	拼写异常	350	4471
	顺序错误	350	5407
思科	缺失	150	2758
	冗余	150	3058
	拼写异常	150	874
	顺序错误	150	974
中兴	缺失	100	1942
	冗余	100	2142
	拼写异常	100	394
	顺序错误	100	536
华三通信	缺失	50	817
	冗余	50	917
	拼写异常	50	220
	顺序错误	50	437

配置块数据集中各厂商配置块数量的分配采用差异化设计, 主要基于模板文件中各厂商模板块的实际数量比例。华为设备占据 55% 的比重(440 个配置块); 思科设备占

20%(160 个配置块); 中兴设备占 15%(120 个配置块); 华三通信设备占 10%(80 个配置块)。各厂商配置语句数量的分配同样遵循了与配置块数据集相似的比例, 具体分配为: 华为设备配置语句 1720 条, 思科设备 640 条, 中兴设备 480 条, 华三通信设备 320 条。这种非均衡的数据分布设计更贴近实际网络环境中的设备分布情况, 能够更真实地验证模型性能。

(3) 测试数据集 2

测试数据集 2 用于评估模型对于各类配置错误的检测性能, 具体数据设置如表 4 所示。与测试数据集 1 类似, 为验证模型的有效性, 测试数据集 2 进行了差异性设置: 华为设备配置块占比为 54%(350 个), 思科为 23%(150 个), 中兴为 15%(100 个), 华三通信为 8%(50 个)。

(4) 用于匹配的模板数据集

实验过程中, 用于匹配的模板文件汇聚了所有厂商的模板文件, 并对所有的配置文件进行了文本切割, 去除了重复的模板块, 模板数据集中共包含 4619 个模板块和 55203 条模板语句。

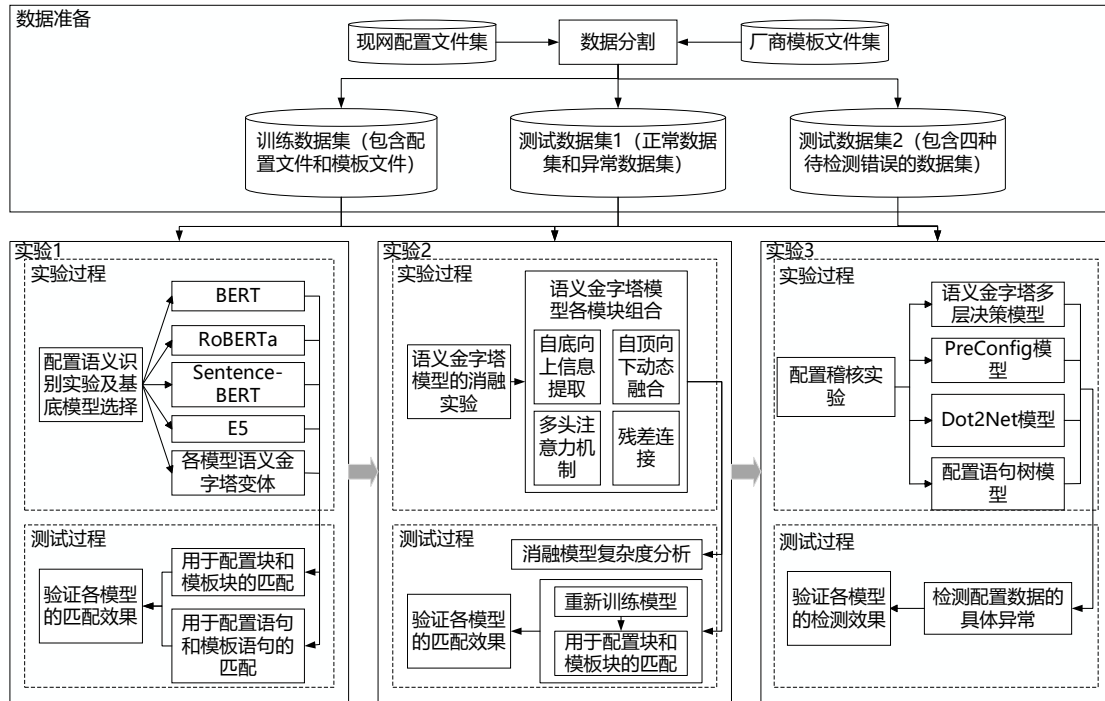


图 10 实验流程示意图

Fig.10 Experimental Procedure Flowchart

5.2 性能指标

(1) ACC

ACC 是指成功匹配模板的配置数据的数量占测试集数据总数目的比例, 用于衡量匹配效果的整体准确性。公式如下:

$$ACC = \frac{N_{matched}}{N_{total}} \quad (23)$$

其中, $N_{matched}$ 表示成功匹配模板的配置块或语句数量,

N_{total} 表示测试集中配置块或语句的总数量。

(2) 精确度

精确度是指配置稽核模型检测为正样本的结果中, 真正为正样本的比例。公式如下:

$$Precision = \frac{TP}{TP + FP} \quad (24)$$

其中, TP 表示真正例, 即被配置稽核模型正确检测为异常的异常数据; FP 表示假正例, 即被配置稽核模板错误检测

为异常的正确数据。精确度越高说明模型将正常配置识别为异常配置的误判率就越低。

(3) 召回率

召回率是指实际为正样本的结果中，被配置稽核模型正确检测为正样本的比例。公式如下：

$$Recall = \frac{TP}{TP + FN} \quad (25)$$

其中， FN 表示假负例，即被配置稽核模型错误检测为正常的异常数据。召回率越高说明异常配置的漏报率越低。

(4) F1 值 (F1-Score)

F1 值是精确度和召回率的调和平均数，用于综合评估模型的精确度和召回率。公式如下：

$$F_1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (26)$$

ACC 主要用于实验 1 和实验 2 中语义金字塔模型配置语义识别评估与分析，验证各模型提取配置语义特征的性能，在此过程中通过相似度匹配找到唯一最佳模板数据，也有可能找不到相应的模板，因而精确度、召回率和 F1 值不适用于实验 1 和实验 2，故选用 ACC 来评估成功匹配的整体比例。精确度、召回率和 F1 值主要用于实验 3 配置稽核过程中各类错误的性能检测与评估。

5.3 语义金字塔模型配置语义识别评估与分析

为准确评估本文提出的动态语义金字塔模型 (RoBERTa-DSPN) 在提取网络设备配置语义特征方面的性能，并确立其相较于现有方法的优势，本节设计了对比实验，与 BERT^[27]、RoBERTa^[28]、Sentence-BERT^[29]和 E5^[33]等主流文本深度学习模型进行对比。同时，为探究动态语义金字塔结构 DSPN 对不同基础模型语义提取能力的增益效果，构建了 BERT-DSPN、RoBERTa-DSPN (本文采用的模型)、Sentence-BERT-DSPN 和 E5-DSPN 四个增强模型进行比较。所有模型的评估均在包含配置块和配置语句两个子集的测试数据集 1 上进行，覆盖正常与异常两类数据，旨在全面检验模型在不同场景和粒度下的语义匹配能力。在训练含 DSPN 模块的模型时，关键超参数采用表 5 所示的固定设置，

以确保对比的公平性。

表 5 语义金字塔模块固定参数设置

Table5 Fixed Parameter Settings for the Semantic Pyramid

Module			
参数类	名称	简介	值
模型参数	deep_feature_layers	深度特征提取层的数量	6
	num_heads	多头注意力中的头数量	8
	kernel_size	卷积核的大小	3
训练参数	dropout_rate	dropout 层的 dropout 率	0.5
	learning_rate	训练学习率	5e-5
	epochs	训练的轮次	5
	batch_size	每批样本数量	4

实验结果如表 6 和表 7 所示，本文提出的 RoBERTa-DSPN 模型在各项评估中均展现出最优的综合性。在正常数据集上，所有模型均能较好地完成匹配任务。然而，性能的关键差异体现在异常数据集上，此类数据集包含各种细微语义语法偏差，更能考验模型的辨别力。

RoBERTa-DSPN 在此类数据上的表现优异，在配置块和配置语句两个粒度上的准确率均领先于其他模型，证明了其架构在识别异常配置方面的有效性。特别是在细粒度的配置语句数据集上，RoBERTa-DSPN 的优势更为显著，这得益于动态语义金字塔结构对指令级语义信息的精准捕捉能力，这对于后续的错误定位至关重要。

对比不同基础模型与 DSPN 模块的组合效果，实验数据明确支持了选择 RoBERTa 作为基础模块。虽然 RoBERTa 基线模型本身在某些指标上并非绝对最佳，但其与 DSPN 模块结合后所实现的性能提升幅度最大，且最终效果超越了所有其他组合（如 BERT-DSPN、Sentence-BERT-DSPN、E5-DSPN）。

表 6 不同模型在配置块数据集上的 ACC 性能验证

Table6 ACC Performance Comparison of Different Models on the Configuration Block Dataset

模型	正常数据集				异常数据集			
	华为	思科	中兴	华三通信	华为	思科	中兴	华三通信
BERT ^[27]	1.0000	1.0000	0.9750	0.9875	0.9159	0.8812	0.9417	0.9375
RoBERTa ^[28]	0.9977	0.9938	0.9917	0.9875	0.8523	0.8875	0.8917	0.9375
Sentence-BERT ^[29]	1.0000	0.9938	0.9917	0.9875	0.8227	0.8625	0.8750	0.8625
E5 ^[33]	1.0000	0.9938	0.9917	0.9875	0.9136	0.8812	0.9250	0.9750
BERT- DSPN	1.0000	1.0000	0.9750	0.9875	0.7455	0.6625	0.7917	0.7875
RoBERTa- DSPN (本文)	1.0000	1.0000	0.9917	0.9875	0.9200	0.9375	0.9167	0.9500
Sentence-BERT-DSPN	1.0000	1.0000	1.0000	1.0000	0.8932	0.9250	0.8333	0.9125
E5-DSPN	1.0000	0.9938	0.9917	0.9875	0.7341	0.8750	0.8333	0.8625

表 7 不同模型在配置语句数据集上的 ACC 性能验证

Table7 ACC Performance Comparison of Different Models on the Configuration Statement Dataset

模型	正常数据集				异常数据集			
	华为	思科	中兴	华三通信	华为	思科	中兴	华三通信
BERT ^[27]	1.0000	1.0000	0.9938	0.9875	0.9778	0.8969	0.9521	0.9594
RoBERTa ^[28]	1.0000	1.0000	0.9958	0.9969	0.9455	0.9219	0.9250	0.9156
Sentence-BERT ^[29]	1.0000	1.0000	0.9938	0.9875	0.8898	0.9000	0.8729	0.8812
E5 ^[33]	1.0000	1.0000	0.9958	0.9969	0.9790	0.9297	0.9571	0.9181
BERT- DSPN	1.0000	1.0000	0.9938	0.9875	0.8767	0.7500	0.8229	0.8031
RoBERTa- DSPN (本文)	1.0000	1.0000	0.9917	0.9875	0.9801	0.9791	0.9771	0.9787
Sentence-BERT-DSPN	1.0000	1.0000	1.0000	1.0000	0.8886	0.7609	0.8292	0.8094
E5- DSPN	1.0000	0.9938	0.9917	0.9875	0.9233	0.8734	0.9146	0.9187

这表明 RoBERTa 提供的语义表示与本文设计的语义金字塔结构之间存在高度的适配性。RoBERTa 采用字节对编码、动态掩码等技术,舍弃了 BERT 的“下一句预测(NSP)”任务,仅通过掩码语言模型(MLM)训练,使模型更专注于句子内的深度语义建模。配置文本通常以段落或短句形式呈现(如配置文件中的独立配置项),句子间逻辑相对松散,而句子/短语内的参数结构(如键值对的语法正确性、参数取值的合法性)更为关键。RoBERTa 对句子内语义的学习为模型提供了强大的初始语义理解能力,而语义金字塔的多尺度特征提取、双向信息融合以及注意力机制,能够在此基础上高效地提炼和聚焦于网络配置文本特有的层次结构和关键语义信息。这种有效的协同作用,使得 RoBERTa-DSPN 能够更精确地处理复杂的跨平台配置语义识别任务。因此,实验结果充分验证了 RoBERTa 作为基础模型与双向动态金字塔结构结合的优越性。另外本文提出的动态语义金字塔模型在配置语句和配置块不同颗粒度的异常识别中准确性都优于 BERT、RoBERTa、Sentence-BERT 和 E5 等现在主流文本深度学习模型。

5.4 动态语义金字塔模型消融实验

为深入理解动态语义金字塔模型中各核心组件(自底向上特征提取、自顶向下特征融合、多头注意力机制、残差连接)对整体性能和计算复杂度的具体贡献,并验证模型超参数选择的有效性,本节设计并实施了参数优化对比实验和模块消融实验。

5.4.1 参数优化

为系统性地确定动态语义金字塔模型的最优超参数组合,并确保训练过程兼顾效率与精度,本文采用了一种基于 Optuna 框架的自动化超参数优化策略^[34]。该策略深度整合了分层搜索(Hierarchical Search)、贝叶斯优化(Bayesian Optimization)以及中途剪枝(Pruning)机制,在复杂的高维参数空间中定位最优解。

(1) 优化策略与核心算法

本文采用分层搜索策略(Hierarchical Search Strategy),将超参数依据其特性分为两类:决定模型拓扑结构的离散参数(如网络层数 `deep_feature_layers`、注意力头数 `num_heads`)和影响梯度下降动态的连续参数(如学习率 `learning_rate`)。该策略通过两个独立阶段进行:第一阶段集中优化离散参数,此时将连续参数固定为经验默认值;第二阶段则在已确定的最优结构上,对连续参数进行精细化搜索。这种分而治之的方法有效降低了搜索空间的维度,提升了优化效率。

为了提升搜索效率,本文采用树结构 Parzen 估计器(Tree-structured Parzen Estimator, TPE)^[35]进行贝叶斯优化。TPE 通过对历史试验结果建立概率模型来指导后续的参数选择。具体而言, TPE 将所有已完成的试验根据其损失值划分为“优良”试验集和“较差”试验集,并分别为这两组的参数 θ 建立条件概率密度模型 $l(\theta)$ 和 $g(\theta)$ 。其选择下一组候选参数 θ_{next} 的依据是最大化期望提升(Expected Improvement, EI),计算公式如下:

$$\theta_{next} = \underset{\theta}{argmax} \frac{l(\theta)}{g(\theta)} \quad (27)$$

通过这种方式, TPE 能够确定性能提升的参数区域。

为进一步节约计算资源,本文引入了中途剪枝机制(Pruning Mechanism)。该机制基于中位数性能进行判断,在每个训练轮次结束后,评估当前试验的中间性能(即该轮次的损失值)。若该性能劣于所有历史试验在同一轮次的中位数水平,则系统判定该参数组合没有潜力,并提前停止训练。设 $v_{i,s}$ 为当前第 i 次试验在第 s 步(epoch)的损失值, V_s 为所有历史已完成试验在第 s 步的损失值集合, $Median(V_s)$ 是对历史损失值集合取中间数,则剪枝的触发条件为:

$$Prune_i = \begin{cases} True, & \text{if } v_{i,s} > Median(V_s) \\ False, & \text{otherwise} \end{cases} \quad (28)$$

该机制能够剔除表现不佳的试验,将算力聚焦于更有希望的参数组合上。

表 8 动态语义金字塔模型参数设置

Table8 Parameter Settings for the Dynamic Semantic Pyramid

Mode			
参数类	名称	简介	值
模型参数	deep_feature_layers	深度特征提取层的数量	2~6
	num_heads	多头注意力中的头数量	{4,8,16}
	kernel_size	卷积核的大小	{3,5}
训练参数	dropout_rate	dropout 层的 dropout 率	0.1~0.5
	learning_rate	训练学习率	1e-5~5e-5
	weight_decay	优化器的权重衰减	1e-3~1e-2

(2) 参数搜索具体流程与决策依据

表 9 动态语义金字塔模型最优参数设置

Table9 Optimal Parameter Settings for the Dynamic Semantic

Pyramid Model		
参数类	名称	值
模型参数	deep_feature_layers	6
	num_heads	8
	kernel_size	5
训练参数	dropout_rate	0.1
	learning_rate	4.92e-5
	weight_decay	1.41e-3

参数搜索步骤如下。

步骤一：离散参数优化。此阶段的目标是确定最优的模型结构。决策依据是，首先固定对模型结构影响较小的连

表 10 不同参数模型在配置块数据集上的 ACC 性能验证

Table10 ACC Performance Comparison of Models with Different Parameters on the Configuration Block Dataset

模型	正常数据集				异常数据集			
	华为	思科	中兴	华三通信	华为	思科	中兴	华三通信
RoBERTa-DSPN(固定参数)	1.0000	1.0000	0.9917	0.9875	0.9200	0.9375	0.9167	0.9500
RoBERTa-DSPN(优化参数)	1.0000	1.0000	0.9917	0.9875	0.9250	0.9375	0.9667	0.9875

表 11 不同参数模型在配置语句数据集上的 ACC 性能验证

Table11 ACC Performance Comparison of Models with Different Parameters on the Configuration Statement Dataset

模型	正常数据集				异常数据集			
	华为	思科	中兴	华三通信	华为	思科	中兴	华三通信
RoBERTa-DSPN(固定参数)	1.0000	1.0000	0.9917	0.9875	0.9801	0.9791	0.9771	0.9787
RoBERTa-DSPN(优化参数)	1.0000	1.0000	0.9917	0.9875	0.9801	0.9812	0.9833	0.9844

5.4.2 不同模块消融

为了全面评估各个组件对动态语义金字塔模型性能的影响,分别去除自底向上的特征提取、自顶向下的特征融合、

续参数 (learning_rate=3e-5, weight_decay=1e-3), 然后对表 8 中定义的离散参数进行搜索。本文在此阶段进行了 60 次试验。为加速评估, 每次试验仅使用 30% 的训练数据子集, 并训练 2 个轮次。TPE 算法根据公式(27)推荐候选参数, 而中途剪枝机制则根据公式(28)在第一个轮次结束后开始剪枝。

步骤二：连续参数优化。在获得最优的离散参数组合后, 将其作为固定配置, 进入第二阶段。此阶段的目标是寻找与该结构最匹配的连续动态参数。本文对表 8 中定义的连续参数 (learning_rate, weight_decay) 进行 20 次试验搜索。与第一阶段类似, 此过程同样在 30% 的数据子集上进行, 训练 3 个轮次, 并启用 TPE 与中途剪枝机制。

步骤三：在完成上述两个阶段的搜索后, 将两阶段得到的最优参数进行组合, 形成了最终的超参数配置 (如表 9 所示)。最后, 使用这套最优参数, 在 100% 完整的训练数据集上进行完整的 5 个轮次的训练, 不进行任何剪枝, 从而得到性能最优、用于最终评估的动态语义金字塔模型。

(3) 对比实验

在获得优化参数后, 为验证优化参数的有效性, 本文使用优化参数对动态语义金字塔模型 (RoBERTa-DSPN) 进行训练, 并使用一组固定参数 (即 5.3 节表 5 中的参数设置) 对相同结构模型进行训练, 并将两个利用不同参数完成训练的模型在测试数据集 1 中的配置块数据集和配置语句数据集上进行性能对比, 结果如表 10 和表 11 所示。

对比实验结果显示, 使用优化参数进行训练的模型在正常数据集上保持了高水平性能, 而在更具挑战性的异常数据集上表现出明显的提升。这证明了参数优化的有效性, 所选定的最优参数组合能够提升模型对异常配置的辨识能力。后续的消融实验与配置稽核性能评估实验均基于此最优超参数组合进行。

多头注意力机制和残差连接, 构建相应的消融模型。每个消融模型在经过训练数据集的训练后, 使用测试数据集 1 对其进行评估, 通过配置块与模板块的匹配效果来分析各

个组件对模型语义特征提取能力的影响。评估指标为匹配准确率（ACC）。

表 12 消融实验不同模块 ACC 性能验证

Table12 Ablation Study on the ACC Performance of Different Modules

模块					正常数据集				异常数据集			
序号	Bottom-Up	Up-Down	MhAttention	Residual	华为	思科	中兴	华三通信	华为	思科	中兴	华三通信
1	×	×	×	×	0.9977	0.9938	0.9917	0.9875	0.8523	0.8875	0.8917	0.9375
2	√	×	×	×	1.0000	0.9938	0.9917	0.9875	0.7977	0.7188	0.8083	0.8375
3	×	√	×	×	1.0000	0.9938	0.9917	0.9875	0.8523	0.8625	0.8583	0.9125
4	×	×	√	×	1.0000	0.9938	0.9917	0.9875	0.7955	0.8375	0.8333	0.8500
5	×	×	×	√	1.0000	0.9938	0.9917	0.9875	0.6727	0.6697	0.7000	0.7750
6	√	√	×	×	1.0000	0.9938	0.9917	0.9875	0.8023	0.8438	0.8000	0.8375
7	√	×	√	×	1.0000	0.9938	0.9917	0.9875	0.7886	0.7625	0.8083	0.8500
8	√	×	×	√	1.0000	0.9938	0.9917	0.9875	0.8000	0.7250	0.7667	0.8625
9	×	√	√	×	1.0000	0.9938	0.9917	0.9875	0.8545	0.8688	0.9000	0.9125
10	×	√	×	√	1.0000	0.9938	0.9917	0.9875	0.6955	0.7125	0.7083	0.8000
11	×	×	√	√	1.0000	0.9938	0.9917	0.9875	0.6841	0.6937	0.7167	0.7125
12	√	√	√	×	0.9682	0.9563	0.9333	0.9875	0.7477	0.6500	0.7083	0.8250
13	√	√	×	√	1.0000	0.9938	0.9917	0.9785	0.7523	0.6937	0.7333	0.8750
14	√	×	√	√	1.0000	0.9938	0.9917	0.9875	0.8205	0.7937	0.8500	0.8625
15	×	√	√	√	1.0000	0.9938	0.9917	0.9875	0.7818	0.7875	0.8083	0.8000
16	√	√	√	√	1.0000	1.0000	0.9917	0.9875	0.9250	0.9375	0.9667	0.9875

实验结果如表 12 所示，完整模型在正常与异常数据集上的性能均优于所有消融变体，尤其在异常数据集上优势显著。移除任何一个核心组件均会导致模型在异常数据集上的准确率明显下降。去除自底向上特征提取层（Bottom-Up）后性能下降，说明该模块对于捕捉配置数据的基础多尺度语法与文本结构特征至关重要。去除自顶向下特征融合层（Up-Down）同样影响性能，反映出高层语法信息与底层语义细节特征的融合可以提升模型鲁棒性。去除多头注意力机制（MhAttention）导致性能大幅降低，表明该机制在动态聚焦关键语义差异、区分易混淆配置方面扮演了核心角色。去除残差连接（Residual）造成的性能损失最为严重，并同时影响了正常与异常数据集，证明了残差连接对于维持信息流畅通、保证模型训练稳定性和深层特征有效传递的决定性作用。

这五个模块在配置稽核过程中高效协同工作：RoBERTa 模块首先将配置语句转换为初始语义表示；自底向上特征提取层从局部到全局逐层提取特征，捕获不同粒度的配置语法与文本结构特性；自顶向下特征融合层将全局上下文信息反向传递，增强局部特征的语义表达；多头注意力

机制在此基础上突出关键配置要素，确保模型关注重要的配置项；残差连接则贯穿整个过程，保障信息流的完整性和模型的稳定性。

5.5 模型决策过程可解释性分析

本节通过一个具体案例，分析了第 4 章所述的基于多层决策的配置稽核流程，以展示模型识别与定位错误的具体决策过程。

5.5.1 案例设定与诊断结果

本节构建了一个包含拼写错误与语句缺失的复合型错误配置块作为分析对象，以全面地展示所提方法的诊断能力。待测配置块与系统在配置块定位层匹配到的正确模板块分别如图 11 与图 12 所示。

将图 11 所示的待测配置块输入本文所提出的稽核系统后，经过多层决策处理，在输出层最终生成的诊断报告如下。该报告是对后续各层诊断结果的汇总。

（1）拼写错误（Spelling Error）：在字符扫描层检测到，待测配置块第 4 行的内容"shutdonw"存在拼写错误，应为"shutdown"。

（2）缺失语句（Missing Statement）：在配置语句匹配层

检测到，待测配置块缺失了模板块（图 12）中第 2 行的配置语句"description <description>"。

```
interface GigabitEthernet0/0/1
undo portswitch
ip address 192.168.1.1 255.255.255.0
shutdonw
```

图 11 包含复合错误的待测配置块

Fig.11 Test Configuration Block with Compound Errors

```
Interface <*>
description <description>
undo portswitch
ip address <IPV4> <IPV4>
shutdown
```

图 12 与待测块匹配的正确模板块

Fig.12 Correct Template Block Matched with the Test Block

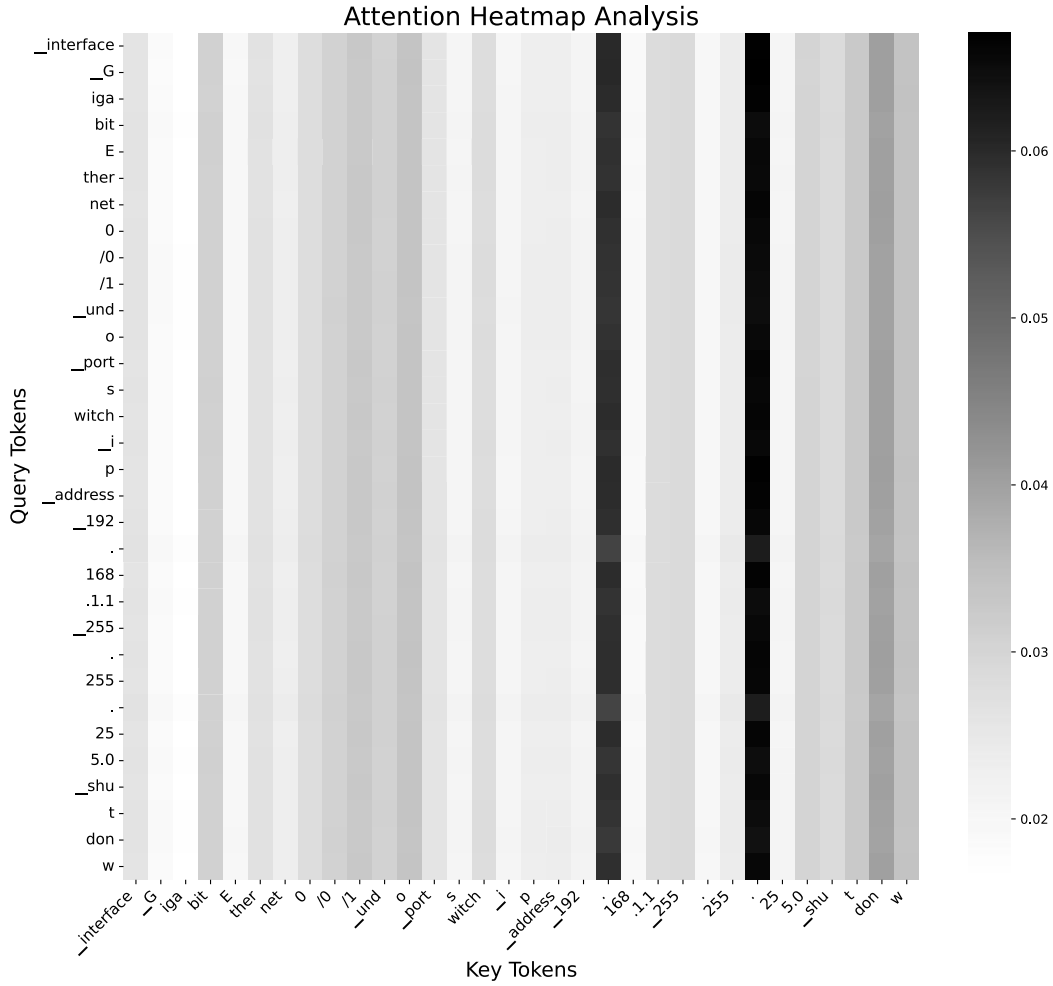


图 13 待测配置块的注意力热力图

Fig.13 Attention Heatmap of the Test Configuration Block

5.5.2 决策过程分析

(1) 配置块定位层该层的核心任务是，即便待测配置块（图 11）存在错误信息，也要从全局语义层面准确地将其与模板库中对应的正确模板块（图 12）进行匹配。此过程由动态语义金字塔模型（DSPN）完成，具体如下：

首先，待测配置块文本被送入 DSPN 的 RoBERTa 基础模块。根据 3.1 节的描述，该模块利用字节对编码将输入文本转化为子词序列。值得注意的是，拼写错误的单词 shutdonw 被分解为 _shu, t, don, w 等词元。

随后，该序列经过 DSPN 的自底向上与自顶向下双向路径处理。为了探究模型在此过程中的内部决策依据，本文

提取并可视化了其自顶向下路径中顶层多头注意力机制（如 3.3 节式(9)所述）的权重分布，如图 13 所示。

图 13 的热力图（颜色越亮表示注意力权重越高）直观地反映了 DSPN 模型的关注焦点。图中显示，模型对由错误单词 shutdonw 分解出的词元给予了异常的关注。这证明了 DSPN 模型在进行跨尺度特征融合时，已在语义层面感知到该区域不符合其通过大规模预训练所学到的语言范式。这种感知到的异常信息，连同全局的上下文语义，被一同编码在 DSPN 输出的最终特征表示 F_{final} 中。

最后，系统利用该也正向量 $V_{c_{b_i}}$ ，通过式 (17) 计算与所有模板块的余弦相似度。尽管存在局部异常，但 F_{final} 所

包含的整体语义信息使得系统能够成功定位到正确的模板块。

(2) 配置语句匹配层

在成功定位到匹配的模板块后，系统进入配置语句匹配层。该层接收上一层输出的匹配对（图 11 和图 12），并根据 4.3 节描述的流程进行处理：

首先，将配置块和模板块按换行符切割为语句集合。系统再次调用 DSPN 模型，计算每个待测语句 s_g 和模板语句 t_k 的特征向量 V_{s_g} 和 V_{t_k} 。

然后，对每个待测语句 s_g ，根据式(21)和式(22)计算得到的向量相似性与下标接近度，在模板块中寻找其最佳匹配语句。在此过程中，模板语句 `<description>` 在待测配置块中没有找到任何满足匹配条件的对应语句。因此，根据 4.3 节中对未匹配模板语句的定义（如图 7 所示），系统将该语句判定为缺失语句。

(3) 字符扫描层

该层处理上一层输出的已成功匹配的语句对。当处理到待测语句 `shutdonw` 及其匹配的模板语句 `shutdown` 时，系统根据第 4.4 节的定义，通过正则表达式进行字符扫描比对。由于两者字符串不完全一致，系统判定待测语句存在拼写异常（如图 9 所示）。

通过对上述案例的分析，本文所提出的多层决策配置稽核方法具有清晰、严谨且可解释的逻辑层次。动态语义金字塔模型（DSPN）在宏观的配置块定位层展现了强大的语义理解与抗干扰能力，其内部的注意力机制为定位异常特征提供了关键的视觉证据。该阶段的准确输出，为后续的配置语句匹配层和字符扫描层进行更细粒度的差异比对和错误诊断，提供了决策基础。这种从宏观到微观、层层递进的决策流程，确保了方法在处理复杂错误场景时的准确性与逻辑清晰性。

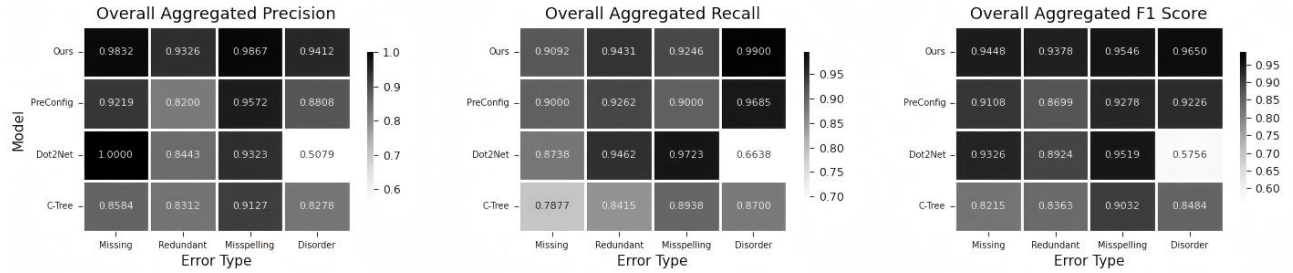


图 14 各方法在整个测试数据集上的配置异常检测性能对比

Fig.14 Comparison of Configuration Anomaly Detection Performance among Different Methods on the Entire Test Dataset

5.6 配置稽核性能评估与结果分析

为了验证动态语义金字塔多层决策配置稽核的检测性能，本文选择深度学习方法 PreConfig^[26]、语义关联方法 dot2net^[22]和配置语句树模型（C-Tree）^[19]作为对比模型。由于 PreConfig 方法本身主要进行配置块级别的异常判断，无法直接提供细粒度的错误类型和定位，因此在本实验中 PreConfig 模型的实验结果是指在其输出的特征基础上，应用了本文构想的多层决策模型进行后续错误分类与定位后的综合表现。

首先，实验在包含华为、思科、中兴和华三通信四家厂商的配置数据的测试数据集 2 上进行，评估本文方法对不同厂商不同错误类型的检测性能。其次，为了进一步验证模型的泛化性，本文额外构建了包含烽火通信（FIBER）和锐捷网络（RUIJIE）两家厂商的异常数据集。其中，新增的烽火通信数据集包含 700 个配置块、7220 条配置语句；锐捷网络数据集包含 500 个配置块、5106 条配置语句。本节将分别展示模型在测试数据集 2 上的性能，以及在全部六个数据集上的综合性能。实验对语句缺失、语句冗余、拼写异常和语句顺序错误四种错误类型进行识别，详细结果如图 14-20 所示。

动态语义金字塔多层决策模型在所有六家厂商的配置错误类型上都取得了优异的性能。如图 14 各方法在全部六个数据集的配置异常检测综合性能对比中，本文方法在精确率、召回率和 F1 值上均优于其他方法。缺失、冗余、拼写错误和顺序错误的配置检测 F1 值分别为 0.9448、0.9378、0.9546 和 0.9650，均在 0.93 以上，相比于其他三种方法在 F1 值上分别平均高出了 0.0428、0.1124 和 0.0982。特别是在新增的烽火通信和锐捷网络数据集上，本文方法同样表现出色，例如在烽火通信数据集上，语句缺失、冗余和顺序错误的 F1 值分别达到了 0.9903、0.9720 和 0.9804；在锐捷网络数据集上，冗余语句检测的 F1 值更是高达 0.9904。在思科数据集上，本文模型在排序错误类型上的召回率达到 1.0000，F1 分数为 0.9677。这充分说明动态语义金字塔多层决策模型在跨平台配置异常检测过程中具有良好的准确率和泛化性，这主要得益于动态语义金字塔多层决策模型采用了多层次的特征提取和决策机制，能够全面捕捉配置块之间的语义关联及层次依赖关系。同时，模型的多层决策结构能够逐层细化对不同错误类型的分析，有效降低了误报和漏报的概率。

如图 15-20 所示，PreConfig 模型的整体表现优于

dot2net 和 C-Tree 两种方法但是却次于动态语义金字塔多层决策模型。这是因为 PreConfig 模型的核心在于利用预训练语言模型的通用语义理解能力,能够借助大规模语料库学习到的知识,对词汇层面的异常有较好的感知。然而,PreConfig 模型缺乏对网络配置所特有的严格结构化规则、命令间精确依赖以及语句顺序的建模能力。这导致其对结构规则违反和顺序错误的敏感度不足,限制了检测精度,本文方法通过动态语义金字塔模型深度获取配置文件的语义与语法结构特征,弥补了这一缺陷。例如在思科数据集上冗余和顺序错误的检测精度为 0.7083 和 0.8773,本文方法检测精度提升至 0.7901 和 0.9375;在中兴数据集上冗余和顺序错误的检测精度为 0.5893 和 0.8857,而本文方法检测精度分别为 1.000 和 0.9429。此外,在新增的烽火通信数据集上,对于拼写异常的检测,PreConfig 的精确率为 0.8333,而本文方法达到了 0.9884,展示了更强的细粒度语义辨别能力。



图 15 测试数据集 2 的华为数据集配置异常检测性能对比
Fig.15 Comparison of Configuration Anomaly Detection Performance among Different Methods on Test Dataset 2 (HW)

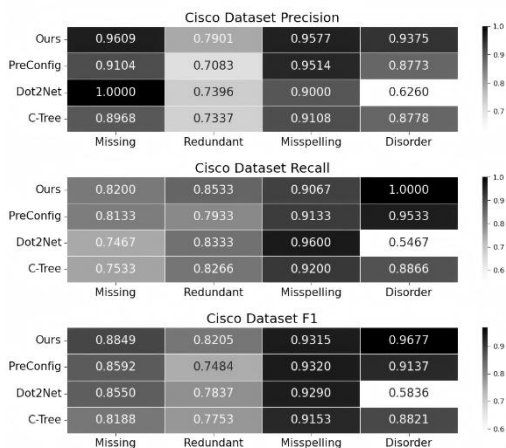


图 16 测试数据集 2 的思科数据集配置异常检测性能对比
Fig.16 Comparison of Configuration Anomaly Detection Performance among Different Methods on Test Dataset 2 (Cisco)

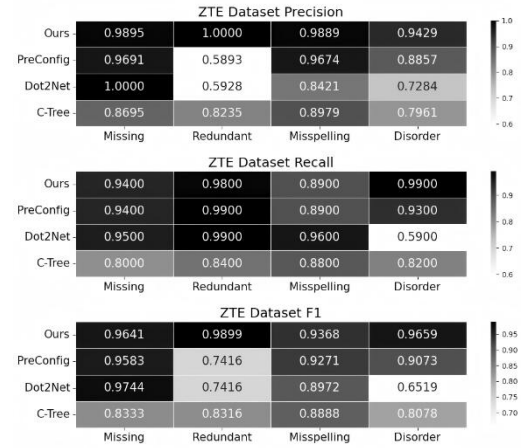


图 17 测试数据集 2 的中兴数据集配置异常检测性能对比
Fig.17 Comparison of Configuration Anomaly Detection Performance among Different Methods on Test Dataset 2 (ZTE)



图 18 测试数据集 2 的华三通信数据集配置异常检测性能对比
Fig.18 Comparison of Configuration Anomaly Detection Performance among Different Methods on Test Dataset 2 (H3C)

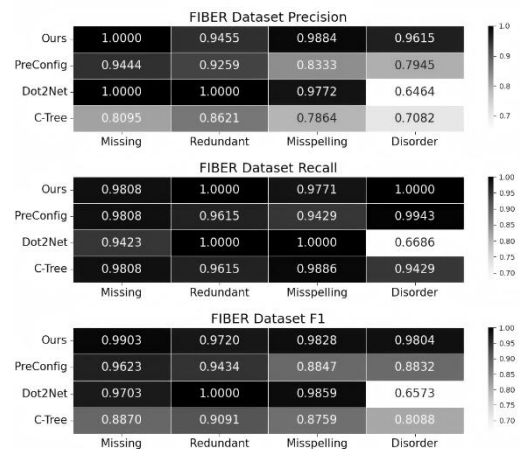


图 19 烽火通信数据集配置异常检测性能对比
Fig.19 Comparison of Configuration Anomaly Detection Performance on FIBER Dataset

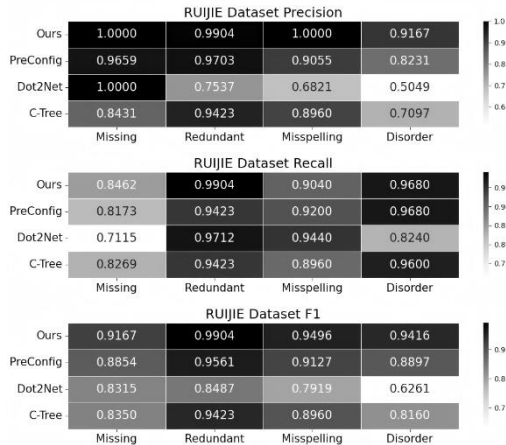


图 20 锐捷网络数据集配置异常检测性能对比

Fig.20 Comparison of Configuration Anomaly Detection Performance on RUIJIE Dataset

Dot2Net 模型采用将配置文本转为图形结构分析的策略,构建节点和边代表配置行及关系,利用图匹配算法识别错误。但此过程缺乏深度语义理解,削弱了原始文本的细粒度语法信息,直接影响顺序错误检测性能,导致多个数据集上相关指标偏低。在华为、思科、中兴、华三通信、烽火通信和锐捷网络六个数据集上的顺序错误检测的 F1 值仅为 0.4620、0.5836、0.6519、0.5962、0.6573 和 0.6261,远低于本文方法。

配置语句树模型将网络设备的配置文件构建成结构化的语句树,通过无监督的 K-Means 聚类算法找出异常的配置文件,但在应对配置块中内容语句缺失和语义顺序变化时,缺乏灵活感知能力。如图 15-20 所示,动态语义金字塔多层决策模型相比于配置语句树模型,在六个数据集上的语句缺失检测 F1 值分别提升了 0.1412、0.0661、0.1308、0.1369、0.1033 和 0.0817,语句顺序错误检测 F1 值分别提升了 0.128、0.0856、0.1581、0.1049、0.1716 和 0.1256。此外,当有新设备加入时,配置语句树模型则需要重新编写相关正则表达式并统计关键词,而动态语义金字塔多层决策模型只需提供待检测配置数据对应的模板文件而无需重新编写匹配规则,即可实现新设备的配置错误的精准识别。

通过以上实验说明本文提出的配置稽核方法能够有效识别复杂通信网络中更广泛的跨平台设备的多种配置错误,其中利用动态语义金字塔模型能够的提取跨平台设备的配置语义特征,从而能够有效检测异常配置信息,并通过多层决策来精准定位与识别多种具体的配置错误,为跨平台的通信设备运维提供有效的数据支持。

6 结语

随着信息通信技术的迅速发展,各类网络设备日益剧增,而设备的配置检测是网络运维的重要内容,也是保障网络可靠运行的重要环节。为实现多类通信网络设备的配置异常检测,本文提出了一种融合语义金字塔与多层决策的跨平

台配置稽核方法。其中动态语义金字塔模型实现了多类设备配置语义信息全局与局部关键特征动态融合提取;而在多层决策过程中则构建通用的匹配规则实现跨平台设备多类配置错误的精准识别。本文设计了详尽的实验过程,包括消融实验、语义识别实验、错误配置稽核实验,有效验证了基于动态语义金字塔模型和多层决策的跨平台配置稽核的有效性。

在未来的研究工作中,将致力于配置稽核模型与大语言模型相结合,探索基于 AI Agent 的交互式人工智能检测方法,以进一步提升检测的智能化和个性化,增强网络设备运维的适应性和灵活性。

References:

- [1] Wang X, Gao L. When 5G Meets Industry 4.0[M]. Berlin/Heidelberg, Germany: Springer, 2020.
- [2] Na M, Lee J, Choi G, et al. Operator's perspective on 6G: 6G services, vision, and spectrum[J]. IEEE Communications Magazine, 2024, 62(8): 178-184.
- [3] ZHANG H J, CHEN A Q, LI Y B, et al. Key technologies of 6G mobile network[J]. Journal on Communications, 2022, 43(7): 189-202.
- [4] WANG H T, SONG L H, WANG X M, et al. Service-oriented 5G Network Architecture—Inheritance and Innovation[J]. Computer Technology and Development, 2020, 30(11): 89-93.
- [5] Han B, Zhang J, Cai H, et al. 5G wireless technology evolution: identifying evolution pathways of core technologies based on patent networks[J]. Wireless Networks, 2024, 30(8): 6875-6886.
- [6] Mendonça S, Damásio B, de Freitas L C, et al. The rise of 5G technologies and systems: A quantitative analysis of knowledge production[J]. Telecommunications Policy, 2022, 46(4): 102327.
- [7] BAO H Y, YIN K L, CAO L, et al. AIOps in Practice: Status Quo and Standardization[J]. Journal of Software, 2023, 34(09): 4069-4095.
- [8] Tyagi A K, Fernandez T F, Mishra S, et al. Intelligent automation systems at the core of industry 4.0[C]//International conference on intelligent systems design and applications. Cham: Springer International Publishing, 2020: 1-18.
- [9] Yang Y, Yang S, Zhao C, et al. TelOps: AI-driven operations and maintenance for telecommunication networks[J]. IEEE Communications Magazine, 2023, 62(4): 104-110.
- [10] Yong Z, Jian L Y, Cui X W. Design and research on PTN

- technology in local transport network[C]//2011 IEEE 3rd International Conference on Communication Software and Networks. IEEE, 2011: 479-482.
- [11] Cisco Systems, Inc. Cisco Prime Collaboration Deployment Administration Guide, Release 11.5(3)[M]. San Jose, CA: Cisco Systems, Inc., 2016.
- [12] Du W, Côté D, Barber C, et al. Forecasting loss of signal in optical networks with machine learning[J]. *Journal of Optical Communications and Networking*, 2021, 13(10): E109-E121.
- [13] Chen W T, Hu C N. Entering the mobile service market via mobile platforms: Qualcomm's BREW platform and Nokia's Preminet platform[J]. *Telecommunications Policy*, 2008, 32(6): 399-411.
- [14] Fogel A, Fung S, Pedrosa L, et al. A general approach to network configuration analysis[C]//12th USENIX Symposium on Networked Systems Design and Implementation. 2015: 469-483.
- [15] Bringhenti D, Marchetto G, Sisto R, et al. Automation for network security configuration: state of the art and research trends[J]. *ACM Computing Surveys*, 2023, 56(3): 1-37.
- [16] Khurat A, Sangkhachantharanan P. An automatic networking device auditing tool based on cis benchmark[C]// 18th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology. 2018: 409-412.
- [17] Wang C, Viswanathan K, Choudur L, et al. Statistical techniques for online anomaly detection in data centers[C]//12th IFIP/IEEE international symposium on integrated network management and workshops. IEEE, 2011: 385-392.
- [18] Madhuri G S, Usha Rani M. Statistical approaches to detect anomalies[C]//Emerging Research in Data Engineering Systems and Computer Communications: Proceedings of CCODE 2019. Springer Singapore, 2020: 499-509.
- [19] SHEN Y C, BAN R, CHEN X, et al. Anomaly Detection Algorithm for Network Device Configuration Based on Configuration Statement Tree[J]. *Computer Science*, 2023, 50(S2): 626-635.
- [20] He X, Li S, He Z, et al. Research on Network Configuration Verification Based on Association Analysis[C]//Proceedings of the 6th International Conference on Computer Science and Application Engineering. 2022: 1-6.
- [21] Xuanyuan S, Li Y, Patil L, et al. Configuration DSPNs representation: A rule-based ontology for product configuration[C]//2016 SAI Computing Conference. IEEE, 2016: 734-741.
- [22] Kobayashi S, Shiiba R, Miura R, et al. dot2net: A labeled graph approach for template-based configuration of emulation networks[C]//2023 19th International Conference on Network and Service Management. IEEE, 2023: 1-9.
- [23] Mai T L, Navet N. Deep learning to predict the feasibility of priority-based Ethernet network configurations[J]. *ACM Transactions on Cyber-Physical Systems*, 2021, 5(4): 1-26.
- [24] Lan Y. Chat-oriented social engineering attack detection using attention-based Bi-LSTM and CNN[C]//2021 2nd International Conference on Computing and Data Science. IEEE, 2021: 483-487.
- [25] Hussain Z, Myllyaho L, Nurminen J K. Learning the Structure of Commands by Retraining a Language Model[C]//Proceedings of the 2024 9th International Conference on Machine Learning Technologies. 2024: 239-244.
- [26] Li F, Lang H, Zhang J, et al. PreConfig: A Pretrained Model for Automating Network Configuration[J]. *arXiv preprint arXiv:2403.09369*, 2024.
- [27] Devlin J, Chang M W, Lee K, et al. Bert: Pre-training of deep bidirectional transformers for language understanding[C]//Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers). 2019: 4171-4186.
- [28] Liu Y, Ott M, Goyal N, et al. Roberta: A robustly optimized bert pretraining approach[J]. *arXiv preprint arXiv:1907.11692*, 2019.
- [29] Reimers N, Gurevych I. Sentence-bert: Sentence embeddings using siamese bert-networks[J]. *arXiv preprint arXiv:1908.10084*, 2019.
- [30] Lin T Y, Dollár P, Girshick R, et al. Feature pyramid networks for object detection[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2017: 2117-2125.
- [31] He K, Zhang X, Ren S, et al. Deep residual learning for image recognition[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 770-778.

- [32] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[J]. Advances in neural information processing systems, 2017, 30(1): 5998-6008.
- [33] Wang L, Yang N, Huang X, et al. Text embeddings by weakly-supervised contrastive pre-training[J]. arXiv preprint arXiv:2212.03533, 2022.
- [34] Claret R, O'Neill M, Cotofrei P, et al. Investigating Hyperparameter Optimization and Transferability for ES-HyperNEAT: A TPE Approach[C]//Proceedings of the Genetic and Evolutionary Computation Conference Companion. 2024: 1879-1887.
- [35] Bergstra J, Bardenet R, Bengio Y, et al. Algorithms for hyper-parameter optimization[C]//Proceedings of the 25th International Conference on Neural Information Processing Systems. 2011: 2546-2554.

附中文参考文献:

- [3] 张海君,陈安琪,李亚博,等.6G 移动网络关键技术[J].通信学报,2022,43(07):189-202.
- [4] 王海涛,宋丽华,王雪梅,等.面向服务的 5G 网络架构—传承与创新[J].计算机技术与发展,2020,30(11):89-93.
- [7] 包航宇,殷康璘,曹立,等.智能运维的实践:现状与标准化[J].软件学报,2023,34(09):4069-4095.
- [19] 沈袁程,班瑞,陈昕,等.基于配置语句树的网络设备配置异常检测算法[J].计算机科学,2023,50(S2):626-635.