بسمه تعالی

به نام خدا

منشور اخلاقی پژوهش

بایاری از خداوند سبحان و اعتقاد به این که عالم محضر خداست و همواره ناظر بر اعمال انسان و به منظور پاس داشت مقام بلند دانش و پژوهش و نظر به اهمیت جایگاه دانشگاه در اعتلای فرهنگ و تمدن بشری، ما دانشجویان و اعضا هیأت علمی واحدهای دانشگاه آزاد اسلامی متعهد می گردیم اصول زیر را در انجام فعالیت های پژوهشی مد نظر قرار داده و از آن تخطی نکنیم:

۱- اصل حقیقت جویی: تلاش در راستای پی جویی حقیقت و وفاداری به آن و دوری از هرگونه پنهان سازی حقیقت.

۲- اصل رعایت حقوق: التزام به رعایت کامل حقوق پژوهشگران و پژوهیدگان (انسان، حیوان و نبات) و سایر صاحبان حق.

۳- اصل مالکیت مادی و معنوی: تعهد به رعایت کامل حقوق مادی و معنوی دانشگاه و کلیه همکاران پژوهش.

۴- اصل منافع ملی: تعهد به رعایت مصالح ملی و در نظر داشتن پیشبرد و توسعه کشور در کلیه مراحل پژوهش.

۵- اصل رعایت انصاف و امانت: تعهد به اجتناب از هرگونه جانب داری غیر علمی و حفاظت از اموال، تجهیزات و منابع در اختیار.

۶- اصل رازداری: تعهد به صیانت از اسرار و اطلاعات محرمانه افراد، سازمان ها و کشور و کلیه افراد و نهادهای مرتبط با تحقیق.

۷- اصل احترام: تعهد به رعایت حریم ها و حرمت ها در انجام تحقیقات و رعایت جانب نقد و خودداری از هرگونه حرمت شکنی.

۸- اصل ترویج: تعهد به رواج دانش و اشاعه نتایج تحقیقات و انتقال آن به همکاران علمی و دانشجویان به غیر از مواردی که منع قانونی دارد.

۹- اصل برائت: التزام به برائت جویی از هرگونه رفتار غیر حرفه ای و اعلام موضع نسبت به کسانی که حوزه علم و پژوهش را به شائبه های غیر علمی می آلایند.

دانشگاه آزاد اسلامی

واحد علوم و تحقیقات (تهران)

تعهدنامه اصالت رساله یا پایان‌نامه تحصیلی

اینجانب **مسعود دباغی** دانش‌آموخته مقطع کارشناسی‌ارشد در رشته مهندسی کامپیوتر گرایش نرم افزارکه در تاریخ ۱۴۰۵/۰۱/۰۱ از پایان‌نامه/رساله خود تحت عنوان **بهبود تشخیص ناهنجاری‌ها در لاگ‌های سیستمی مبتنی بر مدل‌های زبانی بزرگ با بهره‌گیری از مهندسی پرامپت** با کسب نمره ۰۰ و درجه ۰۰ دفاع نموده‌ام بدینوسیله متعهد می‌شوم:

۱) این پایان‌نامه / رساله حاصل تحقیق و پژوهش انجام شده توسط اینجانب بوده و در مواردی که از دستاوردهای علمی و پژوهشی دیگران (اعم از پایان‌نامه، کتاب، مقاله و ...) استفاده نموده‌ام، مطابق ضوابط و رویه موجود، نام منبع مورد استفاده و سایر مشخصات آنرا در فهرست مربوطه ذکر و درج کرده‌ام.

۲) این پایان‌نامه / رساله قبلاً برای دریافت هیچ مدرک تحصیلی (هم‌سطح، پایین‌تر یا بالاتر) در سایر دانشگاهها و مؤسسات آموزش عالی ارائه نشده است.

۳) چنانچه بعد از فراغت از تحصیل، قصد استفاده و هر گونه بهره‌برداری اعم از چاپ کتاب، ثبت اختراع و ... از این پایان‌نامه یا رساله داشته باشم، از حوزه معاونت پژوهشی واحد مجوزهای مربوطه را اخذ نمایم.

۴) چنانچه در هر مقطع زمانی خلاف موارد فوق ثابت شود، عواقب ناشی از آن را می‌پذیرم و دانشگاه آزاد اسلامی واحد علوم و تحقیقات مجاز است با اینجانب مطابق ضوابط و مقررات رفتار نموده و در صورت ابطال مدرک تحصیلی‌ام هیچگونه ادعایی نخواهم داشت.

نام و نام خانوادگی: **مسعود دباغی**

تاریخ و امضاء

دانشگاه آزاد اسلامی

واحد علوم و تحقیقات (تهران)

دانشکده برق و کامپیوتر، گروه نرم افزار

پایان نامه برای دریافت درجه ی کارشناسی ارشد در رشته کامپیوتر(M.Sc)

گرایش نرم افزار

عنوان:

بهبود تشخیص ناهنجاری‌ها در لاگ‌های سیستمی مبتنی بر مدل‌های زبانی بزرگ با بهره‌گیری از مهندسی پرامپت

استاد راهنما:

دکتر محمدرضا علی‌زاده

نگارش :

مسعود دباغی

تابستان ۱۴۰۵

بر خود لازم می‌دانم مراتب سپاس و قدردانی صمیمانه خود را از استاد ارجمند، **جناب آقای دکتر محمدرضا علی‌زاده،** استاد راهنمای محترم، ابراز نمایم.
راهنمایی‌های علمی و ارزشمند، دقت نظر، صبوری و حمایت‌های دلسوزانه ایشان در تمامی مراحل انجام این پژوهش، نقشی اساسی در شکل‌گیری و به سرانجام رسیدن این پایان‌نامه داشته است.
بی‌تردید، بهره‌مندی از دانش و تجربیات گران‌قدر ایشان، از ارزشمندترین بخش‌های این مسیر علمی برای من بوده است. بدین‌وسیله از تمامی زحمات، همراهی‌ها و راهنمایی‌های ایشان صمیمانه سپاسگزاری می‌نمایم.

تقدیم به

پدر و مادر عزیزم؛

آنان که با مهر بی‌پایان، صبوری و فداکاری‌های بی‌دریغشان، همواره استوارترین تکیه‌گاه و روشن‌ترین چراغ مسیر زندگی من بوده‌اند.

این پایان‌نامه را به پاس تمام محبت‌ها، رنج‌ها و حمایت‌های همیشگی‌شان، با نهایت عشق، احترام و قدردانی تقدیم می‌کنم.

فهرست مطالب

فصل اول: کلیات تحقیق ۲
مقدمه ۱-۱ ..... ۳
بیان مسئله ۲-۱ ..... ۳
اهمیت و ضرورت انجام پژوهش ۳-۱ ..... ۵
اهداف مشخص پژوهش ۴-۱ ..... ۶
سؤال‌ها و فرضیه‌های پژوهش ۵-۱ ..... ۶
سؤال‌های پژوهش ۶
فرضیه‌های پژوهش ۷
فصل دوم: مروری بر ادبیات و پیشینه تحقیق ۸
مقدمه ۱-۲ ..... ۹
مبانی نظری و مفاهیم موردنیاز ۲-۲ ..... ۹
لاگ سیستمی و تشخیص ناهنجاری ۱-۲-۲ ..... ۹
انواع ناهنجاری و واحد تحلیل ۲-۲-۲ ..... ۱۰
مجموعه‌داده BGL ۳-۲-۲ ..... ۱۰
تجزیه، نرمال‌سازی و قالب لاگ ۴-۲-۲ ..... ۱۰
روش‌های قاعده‌محور، یادگیری ماشین و مدل‌های ترتیبی ۵-۲-۲ ..... ۱۱
مدل‌های زبانی پیش‌آموزش‌دیده و مدل‌های زبانی بزرگ ۶-۲-۲ ..... ۱۱
مدل Qwen۲٫۵:۷B-Instruct ۷-۲-۲ ..... ۱۱
مهندسی پرامپت و خروجی ساخت‌یافته ۸-۲-۲ ..... ۱۲
استنتاج محلی با Ollama ۹-۲-۲ ..... ۱۲
کش سطح قالب و اعتبارسنجی تصمیم ۱۰-۲-۲ ..... ۱۲
دو معماری Prompt-only و Hybrid ۱۱-۲-۲ ..... ۱۳
معیارهای ارزیابی ۱۲-۲-۲ ..... ۱۳
پیشینه تحقیق ۳-۲ ..... ۱۴

۲-۳-۱ روش‌های ترتیبی و پیش‌آموزش‌دیده ۱۴
۲-۳-۲ روش‌های مبتنی بر آموزش یا تنظیم LLM ۱۴
۲-۳-۳ روش‌های مبتنی بر پرامپت ۱۴
۲-۳-۴ بازیابی، روش‌های ترکیبی و کاهش هزینه ۱۵
۲-۳-۵ تجزیه، انتقال، تبیین و کاربرد عملی ۱۵
۲-۴ دسته‌بندی، جدول مقایسه و تحلیل ۱۶
۲-۵ شکاف تحقیقاتی و جایگاه پژوهش حاضر ۱۸
۲-۶ جمع‌بندی ۱۹
۲-۷ منابع فصل دوم ۱۹

فهرست جدول ها

۱۶..... جدول ۲-۱: مقایسه مطالعات منتخب در تشخیص ناهنجاری و تحلیل لاگ
۱۷ .....جدول ۲-۲: مقایسه سازوکار مطالعات نزدیک با مؤلفه‌های پژوهش حاضر

فهرست نمودار ها

فهرست شکل ها

شکل ۲-۱: دسته‌بندی پژوهش‌های تشخیص ناهنجاری لاگ و جایگاه پژوهش حاضر.....۱۶

چکیده

رشد سامانه‌های توزیع‌شده موجب تولید حجم عظیمی از لاگ‌های ناهمگون شده است و تشخیص سریع ناهنجاری در آن‌ها برای افزایش پایداری و امنیت سامانه‌ها اهمیت دارد. روش‌های سنتی معمولاً به قواعد دست‌نویس، الگوهای ثابت یا داده‌های برچسب‌خورده وابسته‌اند و در مواجهه با تغییر ساختار لاگ‌ها و حجم بالای داده محدودیت دارند. هدف این پژوهش، ارائه چارچوبی مبتنی بر مدل زبانی بزرگ و مهندسی پرامپت برای طبقه‌بندی دودویی رویدادهای مجموعه‌داده Blue Gene/L ، بدون ریزتنظیم مدل است. در روش پیشنهادی، برچسب واقعی از ورودی مدل حذف شده و مقادیر پویایی مانند شناسه‌ها، مسیرها، نشانی‌های شبکه، مقادیر هگزادسیمال و اعداد به قالب‌های قابل‌بازاستفاده تبدیل می‌شوند؛ درعین‌حال، تفاوت‌های معنایی مهم مانند مقادیر صفر و غیرصفر حفظ می‌گردد. سپس مدل محلی Qwen2.5:7B-Instruct از طریق Ollama و با خروجی ساخت‌یافته، هر لاگ را عادی یا ناهنجار طبقه‌بندی می‌کند. برای کاهش هزینه استنتاج، نتایج معتبر در سطح قالب ذخیره شده و پیش از ورود به حافظه نهان اعتبارسنجی می‌شوند. سامانه در دو حالت ترکیبیِ قواعد قطعی و مدل زبانی، و حالت صرفاً مبتنی بر پرامپت پیاده‌سازی شده و نتایج خط‌به‌خط در MongoDB ذخیره می‌شوند. ارزیابی با معیارهای صحت، دقت، بازخوانی، امتیاز F1 ، نرخ خروجی نامعتبر، زمان پاسخ و آمار حافظه نهان انجام شد. نتایج نشان داد نرمال‌سازی قالب‌ها و استفاده از حافظه نهان، فراخوانی‌های تکراری مدل و زمان پردازش را کاهش می‌دهد، درحالی‌که ثبت خط‌به‌خط، قابلیت ارزیابی و ممیزی را حفظ می‌کند. چارچوب پیشنهادی راهکاری سبک و قابل‌استقرار برای تحلیل لاگ‌های سیستمی بدون آموزش مجدد مدل فراهم می‌سازد.

کلمات کلیدی(فارسی)

تشخیص ناهنجاری، لاگ‌های سیستمی، مهندسی پرامپت، مدل‌های زبانی بزرگ

کلمات کلیدی(انگلیسی)

Anomaly Detection, System Logs, Prompt Engineering, Large Language Models (LLMs)

فصل اول: کلیات تحقیق

مقدمه ۱-۱

گسترش سامانه‌های نرم‌افزاری و معماری‌های توزیع‌شده، موجب تولید پیوسته حجم زیادی از لاگ‌های سیستمی شده است. این لاگ‌ها اطلاعات ارزشمندی درباره وضعیت عملکرد، خطاها و رخدادهای امنیتی سامانه‌ها در اختیار قرار می‌دهند. ازاین‌رو، تشخیص سریع و دقیق ناهنجاری‌ها در آن‌ها نقش مهمی در افزایش پایداری و قابلیت اطمینان سیستم‌ها دارد. بااین‌حال، حجم بالا، ساختار نامنظم و تغییرپذیری الگوهای لاگ، کارایی روش‌های سنتی را محدود می‌سازد. مدل‌های زبانی بزرگ، به دلیل توانایی درک متون غیرساختاریافته و روابط معنایی، ظرفیت مناسبی برای تحلیل این داده‌ها دارند. در این پژوهش، از مهندسی پرامپت برای بهره‌گیری از توانایی مدل‌های زبانی، بدون نیاز به آموزش مجدد و داده‌های برچسب‌خورده گسترده استفاده می‌شود. در ادامه این فصل، بیان مسأله، اهمیت و ضرورت، اهداف، پرسش‌ها و فرضیه‌های تحقیق تشریح می‌شوند.

بیان مسئله ۱-۲

در سامانه‌های مدرن با معماری توزیع‌شده، حجم عظیمی از لاگ‌های سیستمی به صورت پیوسته تولید می‌شود که حاوی اطلاعات مهمی برای عیب‌یابی، پایش سلامت و بررسی حوادث امنیتی هستند. کشف ناهنجاری‌ها در این لاگ‌ها نقش کلیدی در افزایش پایداری و قابلیت اطمینان سیستم دارد. با این‌حال، چند ویژگی مشکل‌ساز وجود دارد: ساختار نامنظم لاگ‌ها (متن آزاد همراه با پارامترها و شناسه‌ها)، حجم بسیار بالا و پویایی سریع الگوهای آن‌ها و تعداد کم رخدادهای ناهنجار نسبت به رویدادهای عادی. روش‌های سنتی مبتنی بر قوانین دست‌نویس یا الگوریتم‌های آماری، که اغلب به الگوهای ثابت یا داده‌های پارس شده نیاز دارند، در مواجهه با این چالش‌ها کارآمدی خود را از دست می‌دهند. برای مثال، LogLLM نشان می‌دهد که روش‌های معمول نیاز به استخراج الگو دستی دارند اما مدل‌های زبانی بزرگ می‌توانند با پیش‌پردازش ساده مانند عبارات با قاعده، فرایند را ساده کنند و به دقت بالا در تشخیص ناهنجاری برسند [۱].

مدل‌های زبانی بزرگ (مانند GPT، BERT، LLaMa و...) توانایی قوی در پردازش متون غیرساختاریافته و یادگیری روابط معنایی دارند. این مدل‌ها می‌توانند پیام‌های لاگ را مانند جمله‌های طبیعی در نظر گرفته و معنا و زمینه آن‌ها را استخراج کنند [۱].

همچنین استفاده از LLMها فراتر از خود تشخیص ناهنجاری رفته و در فرایندهای پیشین مانند تجزیه لاگ نیز تأثیرگذار بوده است. به عنوان نمونه، LogParser-LLM نشان می‌دهد که با ادغام دانش معنایی مدل‌های زبانی در فرایند پارس لاگ، می‌توان بدون نیاز به داده‌های برچسب‌دار یا تنظیمات دقیق، ساختار لاگ را استخراج کرد [۲].

این کار با بهره‌گیری هم‌زمان از اطلاعات معنایی و ویژگی‌های آماری، دقت بالایی در تشخیص الگوها ایجاد می‌کند. در کل، این مطالعات همگی حکایت از این دارند که مدل‌های زبانی بزرگ می‌توانند محدودیت روش‌های سنتی را با درک عمیق‌تر متن لاگ و تشخیص ناهنجاری حتی در شرایط تغییر الگو برطرف سازند [۳].

با وجود موفقیت اولیه، چندین چالش و سؤال پژوهشی باقی مانده است که باید بررسی شوند:

یکی از موانع اصلی در بکارگیری مدل‌های زبانی بزرگ، پارامترهای فراوان و محاسبات سنگینی است که منجر به افزایش تأخیر اجرا و هزینه‌های عملیاتی می‌شود. اکثر رویکردهای فعلی مانند LogLLM [۱] یا روش‌های مبتنی بر تنظیم دقیق[۱] مانند LogBERT [۴]، نیازمند زیرساخت‌های پردازشی قدرتمند (GPU) و زمان آموزش طولانی هستند. از سوی دیگر، در محیط‌های صنعتی، سرعت تشخیص ناهنجاری حیاتی است.

برای مقابله با این چالش، پژوهش‌هایی مانند AdaptiveLog [۵] استراتژی همکاری میان مدل‌های کوچک و بزرگ (SLM/LLM) را پیشنهاد داده‌اند تا تنها در موارد پیچیده از توان محاسباتی مدل بزرگ استفاده شود. با این حال، حتی در این مدل‌ها نیز نیاز به داده‌های برچسب‌دار و فرآیند آموزش همچنان پابرجاست.

علاوه بر چالش‌های محاسباتی و داده‌ای، یکی از موانع جدی در به‌کارگیری مدل‌های زبانی بزرگ در محیط‌های عملیاتی حساس، مسئله «توهم[۲]» و «شکاف دانش دامنه[۳]» است. مدل‌های زبانی عمومی اغلب بر روی داده‌های عمومی اینترنت آموزش دیده‌اند و ممکن است فاقد دانش تخصصی عمیق در مورد معماری‌های خاص سازمانی، کدهای خطای اختصاصی و وابستگی‌های پیچیده میان میکروسرویس‌ها باشند. این محدودیت می‌تواند منجر به تولید توضیحات نادرست اما با ظاهر موجه شود که در فرآیند عیب‌یابی سیستم‌های حیاتی، ریسک عملیاتی را افزایش می‌دهد. پژوهش‌های اخیر نشان می‌دهند که صرفاً تکیه بر دانش درونی مدل برای تشخیص ریشه خطا کافی نیست و مدل‌ها در مواجهه با لاگ‌های ناشناخته، ممکن است روابط علی و معلولی را به اشتباه استنتاج کنند[۶]. از این رو، مبحث تلفیق مدل‌های زبانی با پایگاه‌های دانش خارجی از طریق تکنیک‌هایی مانند «تولید افزوده به بازیابی» (RAG)[۴] مطرح شده است. سوال اساسی اینجاست که چگونه می‌توان مستندات فنی، سوابق تیکت‌های گذشته و گراف دانش سیستم را به صورت پویا در اختیار مدل قرار داد تا خروجی آن واقعیت‌محور باقی بماند. مطالعاتی نظیر OpsEval نشان داده‌اند که مدل‌های فعلی بدون دسترسی به دانش زمینه، در وظایف استدلال منطقیِ پیچیده دچار افت عملکرد می‌شوند[۷]. بنابراین، بررسی راهکارهایی که بتوانند دانش زمینه را بدون نیاز به بازآموزی سنگین به مدل تزریق کنند و نرخ توهم را در تشخیص ناهنجاری به حداقل برسانند، یک خلأ پژوهشی آشکار است که نیازمند تحقیقات متمرکز می‌باشد.

در این پژوهش، شکاف موجود با استفاده از «مهندسی پرامپت هوشمند[۵]» و تکنیک‌های یادگیری با مثال‌های محدود[۶] پر می‌شود. این رویکرد به جای بازآموزی مدل یا استفاده از معماری‌های دوگانه پیچیده، بر استفاده بهینه از توانایی‌های ذاتی مدل‌های پیش‌آموزش‌دیده تمرکز دارد. هدف این است که با طراحی استراتژی‌های پرامپت‌دهی کارآمد [۸]،

---

[۱] Fine-tuning
[۲] Hallucination
[۳] Domain Knowledge Gap
[۴] Retrieval-Augmented Generation
[۵] Prompt Engineering
[۶] Few-shot

دقت تشخیص در سطح مدل‌های ریزتنظیم‌شده حفظ شود، در حالی که هزینه محاسباتی و نیاز به داده‌های آموزشی برچسب‌دار به حداقل برسد.
هدف اصلی، دستیابی به سیستمی است که بتواند به‌صورت مؤثر ناهنجاری‌های معنایی و ترتیبی را در لاگ‌های سیستمی تشخیص دهد، به‌گونه‌ای که دقت تشخیص در سطح مدل‌های پیچیده حفظ شده، اما نیاز به داده‌های برچسب‌دار و هزینه‌های محاسباتی به حداقل برسد. این سیستم با استفاده از استراتژی‌های پرامپت‌دهی کارآمد (مانند زنجیره افکار[۱۰])، اختلالات واقعی را از هشدارهای کاذب تفکیک کرده و برخلاف روش‌های سنتی، در مواجهه با تغییرات ناگهانی در الگوهای لاگ، پویایی و تعمیم‌پذیری بالایی از خود نشان می‌دهد.
نتایج مورد انتظار این تحقیق شامل ارائه راهکاری سبک و قابل پیاده‌سازی در محیط‌های واقعی است که علاوه بر بهبود پایداری و امنیت سامانه‌ها، زمان و هزینه‌های اشکال‌یابی را از طریق ارائه توضیحات قابل‌فهم برای ناهنجاری‌ها کاهش دهد.
در نهایت، این تحقیق می‌تواند به توسعه سامانه‌های مانیتورینگ خودکار و پایدار در مقیاس بزرگ کمک کند، طوری که پایداری و امنیت سامانه‌ها بهبود یابد و زمان و هزینه‌های اشکال‌یابی کاهش پیدا کند.

اهمیت و ضرورت انجام پژوهش ۳-۱

در دنیای امروز، سیستم‌های نرم‌افزاری و زیرساخت‌های توزیع‌شده به ستون فقرات صنایع گوناگون از جمله مالی، سلامت، مخابرات و سرویس‌های ابری تبدیل شده‌اند. پایداری و امنیت این سیستم‌ها وابستگی مستقیم به توانایی تشخیص سریع و دقیق ناهنجاری‌ها در لاگ‌های سیستمی دارد. با توجه به اینکه حجم و تنوع لاگ‌ها به‌طور مداوم در حال افزایش است، استفاده از روش‌های سنتی نظیر قواعد دست‌نویس یا الگوریتم‌های آماری دیگر پاسخگوی نیازهای فعلی نیست.
از نظر علمی، خلأهای تحقیقاتی قابل‌توجهی در این حوزه وجود دارد. نخست، بیشتر پژوهش‌های گذشته یا بر جنبه‌های آماری و ساختاری لاگ‌ها تمرکز داشته‌اند یا تنها به مدل‌سازی مبتنی بر توالی رویدادها پرداخته‌اند و کمتر به ترکیب ابعاد معنایی و زمانی توجه شده است. دوم، اگرچه مطالعات اخیر استفاده از مدل‌های زبانی بزرگ (LLMs) را پیشنهاد کرده‌اند، اما همچنان پرسش‌های مهمی درباره میزان کارایی، هزینه محاسباتی و قابلیت تعمیم‌پذیری LLMها در محیط‌های عملیاتی بدون پاسخ مانده است. اختلاف‌نظرهایی نیز درباره روش بهینه تنظیم این مدل‌ها (تنظیم پرامپت در مقابل فاین‌تیونینگ) و نیاز به داده‌های برچسب‌دار وجود دارد.
از جنبه کاربردی، اهمیت این تحقیق در افزایش پایداری و امنیت سیستم‌ها آشکار است. یک چارچوب هوشمند می‌تواند:

- زمان تشخیص و رفع خطا را به شکل چشمگیری کاهش دهد،
- هزینه‌های ناشی از توقف سیستم یا حملات سایبری را کم کند،
- و ابزار قدرتمندی برای تیم‌های عملیات و نگهداری فراهم آورد.

علاوه بر این، نتایج تحقیق می‌تواند به توسعه روش‌هایی منجر شود که حتی در محیط‌هایی با منابع محاسباتی محدود یا داده‌های برچسب‌دار اندک، عملکرد مؤثر و مقیاس‌پذیر داشته باشند. این موضوع به‌ویژه برای سازمان‌هایی که امکان استفاده از منابع ابری گسترده ندارند، اهمیت حیاتی خواهد داشت.
بنابراین، ضرورت این پژوهش هم در پاسخ به نیازهای عملی صنعت و هم در پر کردن شکاف‌های نظری موجود در ادبیات علمی نهفته است.

اهداف مشخص پژوهش ۴-۱

الف) هدف اصلی:
طراحی و ارائه یک چارچوب تشخیص ناهنجاری در لاگ‌های سیستمی مبتنی بر مهندسی پرامپت بهینه و مدل‌های زبانی بزرگ، با هدف دستیابی هم‌زمان به دقت و سرعت بالا و حذف نیاز به فرآیند پرهزینه تنظیم دقیق و داده‌های برچسب‌گذاری‌شده.

ب) اهداف فرعی:

۱. طراحی استراتژی‌های پرامپت‌دهی ساختاریافته برای استخراج هم‌زمان ویژگی‌های معنایی و ترتیبی از لاگ‌های سیستمی.
۲. ارزیابی و سنجش میزان افزایش سرعت پردازش و تشخیص خطا در روش پیشنهادی نسبت به روش‌های مبتنی بر تنظیم دقیق.
۳. تحلیل و ارزیابی دقت تشخیص ناهنجاری‌های پیچیده و شرطی در روش پیشنهادی در مقایسه با روش‌های سنتی و آماری.
٤. کاهش وابستگی به داده‌های آموزشی برچسب‌دار و بررسی کارایی مدل در شرایط محدودیت داده.
٥. کاهش نیاز به مداخله انسانی و دانش خبره در فرآیند تحلیل و تفسیر لاگ‌ها.

سؤال‌ها و فرضیه‌های پژوهش ۵-۱

سؤال‌های پژوهش

۱. آیا استفاده از روش‌های تنظیم پرامپت در مدل‌های زبانی پیش‌آموزش‌دیده مانند Qwen2.5:7B-Instruct می‌تواند دقت و کارایی سیستم‌های تشخیص ناهنجاری لاگ را بهبود بخشد؟
۲. آیا ترکیب اطلاعات معنایی و ترتیبی لاگ‌ها در مدل‌های زبانی پیش‌آموزش‌دیده موجب بهبود تشخیص ناهنجاری‌های نقطه‌ای و شرطی می‌شود؟
۳. آیا استفاده از داده‌های محدود در یک پرامپت می‌تواند مشکل عدم تعادل کلاس‌ها در داده‌های لاگ را کاهش داده و دقت تشخیص ناهنجاری‌های کمتر متداول را افزایش دهد؟

٤. آیا استفاده از روش‌های تنظیم پرامپت می‌تواند هزینه‌ها و زمان موردنیاز برای برچسب‌گذاری داده‌های لاگ را کاهش دهد، در حالی‌که دقت تشخیص ناهنجاری‌ها حفظ یا حتی بهبود یابد؟

فرضیه‌های پژوهش

۱. استفاده از روش‌های تنظیم پرامپت در مدل‌های زبانی پیش‌آموزش‌دیده مانند Qwen2.5:7B-Instruct می‌تواند دقت و کارایی سیستم‌های تشخیص ناهنجاری لاگ را بهبود بخشد.
۲. ترکیب اطلاعات معنایی و ترتیبی لاگ‌ها در مدل‌های زبانی پیش‌آموزش‌دیده موجب بهبود تشخیص ناهنجاری‌های نقطه‌ای و شرطی می‌شود.
۳. استفاده از داده‌های محدود در یک پرامپت می‌تواند مشکل عدم تعادل کلاس‌ها در داده‌های لاگ را کاهش داده و دقت تشخیص ناهنجاری‌های کمتر متداول را افزایش دهد.
٤. استفاده از روش‌های تنظیم پرامپت می‌تواند هزینه‌ها و زمان موردنیاز برای برچسب‌گذاری داده‌های لاگ را کاهش دهد، در حالی‌که دقت تشخیص ناهنجاری‌ها حفظ یا حتی بهبود می‌یابد.

فصل دوم: مروری بر ادبیات و پیشینه تحقیق

مقدمه ۱-۲

لاگ‌های سیستمی ردپای رخدادهای زمان اجرا، وضعیت اجزا و پیام‌های خطا را ثبت می‌کنند و ازاین‌رو یکی از ورودی‌های اصلی پایش، عیب‌یابی و تشخیص ناهنجاری در سامانه‌های نرم‌افزاری به‌شمار می‌روند. افزایش حجم و تنوع لاگ‌ها، بررسی دستی را دشوار کرده و پژوهش‌ها را از قواعد و شمارش رخدادها به‌سوی مدل‌های ترتیبی، مدل‌های زبانی پیش‌آموزش‌دیده و در سال‌های اخیر مدل‌های زبانی بزرگ سوق داده است [۱، ۲، ۲۲].

در روش‌های مبتنی بر یادگیری عمیق، توالی قالب‌های لاگ یا بازنمایی متنی پیام‌ها برای یادگیری الگوی رفتار عادی استفاده می‌شود. DeepLog با پیش‌بینی رخداد بعدی، LogBERT با تنظیم یک مدل BERT-Log (BERT ۱) برای طبقه‌بندی، سه مسیر شاخص این تحول را نشان می‌دهند [۳]. این روش‌ها توانایی مدل‌سازی الگو را افزایش داده‌اند، اما معمولاً به آموزش اختصاصی، ساخت توالی تجزیه لاگ یا داده برچسب‌خورده وابسته‌اند [۲، ۳، ۲۲].

مدل‌های زبانی بزرگ مسیر دیگری را فراهم کرده‌اند: انجام تشخیص از طریق دستور متنی، نمونه‌های درون‌متنی، بازیابی شواهد، تنظیم کامل یا تنظیم پارامترکارا. مطالعات LogPrompt، LogGPT، AdaptiveLog و FlexLog، LLM-LADE، LogLLM، RAGLog نشان می‌دهند که نقش مدل زبانی می‌تواند از طبقه‌بند مستقیم تا مفسر، تولیدکننده داده، جزء کمکی یا تصمیم‌گیر پشتیبان تغییر کند [۴–۸، ۱۰، ۱۲، ۱۵]. بنابراین مقایسه پژوهش‌ها تنها با نام مدل کافی نیست و باید نوع آموزش، واحد تحلیل، شیوه آماده‌سازی ورودی، محل استقرار و سازوکار کنترل هزینه نیز در نظر گرفته شود [۲۲، ۲۴].

فصل حاضر مبانی موردنیاز برای پژوهش تشخیص ناهنجاری در لاگ‌های BGL با Qwen۲٫۵:۷B-Instruct را ارائه می‌کند. سپس پیشینه در پنج دسته مرور و با جدول و شکل مقایسه می‌شود. در پایان، شکاف تحقیقاتی با اتکا به مجموعه منابع بررسی‌شده و جایگاه دو روش مورد مطالعه، یعنی Ollama و Hybrid در موارد LLM در این پژوهش به «قاعده قطعی با فراخوانی Hybrid تبیین می‌شود. واژه Prompt-only، در مطالعه پارک و همکاران، که ترکیبی از شواهد Hybrid Prompting باقیمانده» اشاره دارد و با اصطلاح سطح رخداد و توالی را در پرامپت قرار می‌دهد، یکسان نیست [۹].

مبانی نظری و مفاهیم موردنیاز ۲-۲

لاگ سیستمی و تشخیص ناهنجاری ۱-۲-۲

هر پیام لاگ معمولاً از سرآیندی مانند زمان، سطح شدت و مؤلفه تولیدکننده و یک بدنه متنی تشکیل می‌شود. در بدنه، بخش‌های ثابت بیانگر رخداد برنامه و بخش‌های متغیر شامل شناسه‌ها، نشانی‌ها، مسیرها، مقادیر عددی و سایر پارامترهای زمان اجرا هستند. پژوهش‌های LogParser-LLM و LogPPT فرایند تبدیل پیام خام به قالب ثابت و پارامترهای متغیر را «تجزیه لاگ» تعریف می‌کنند و نشان می‌دهند که کیفیت این مرحله بر تحلیل‌های پایین‌دستی، از جمله تشخیص ناهنجاری، اثر می‌گذارد [۱۸، ۱۹].

در ادبیات، ناهنجاری به انحراف مشاهده از الگوی عادی آموخته‌شده یا تعریف‌شده اطلاق می‌شود. این انحراف می‌تواند در یک پیام منفرد، فراوانی رخدادها، ترتیب قالب‌ها یا معنای یک توالی ظاهر شود. DeepLog هر پیام ساخت‌نیافته را ADALog بر انحراف توالی از رفتار عادی تمرکز دارند؛ در مقابل LogBERT مستقل تحلیل و احتمال بازسازی توکن‌ها را به امتیاز سطح پیام تبدیل می‌کند [۱، ۲، ۱۷]. این تفاوت، واحد تحلیل و در نتیجه نحوه ساخت نمونه آزمایشی را تعیین می‌کند.

انواع ناهنجاری و واحد تحلیل ۲-۲-۲

سه سطح رایج در پژوهش‌های لاگ قابل تفکیک است: ناهنجاری رخداد یا خط، ناهنجاری توالی و ناهنجاری کمّی. در سطح خط، هر پیام یک نمونه تصمیم است و برچسب به همان پیام تعلق دارد. در سطح توالی، پیام‌ها براساس شناسه نشست، گره یا پنجره زمانی گروه‌بندی می‌شوند و وجود یک رخداد غیرعادی می‌تواند برچسب کل توالی را تعیین کند. در ناهنجاری کمّی، تغییر غیرعادی فراوانی قالب‌ها در پنجره موردنظر بررسی می‌شود [۱–۳، ۲۲].

از پنجره لغزان BGL در BERT-Log ،واحد تحلیل بر قابلیت مقایسه نتایج اثر مستقیم دارد. برای نمونه توالی‌های یک‌دقیقه‌ای Hybrid Prompting مبتنی بر شناسه گره، اندازه پنجره و گام استفاده کرده است؛ تصمیم را برای یک پیام مستقل گزارش می‌کند [۳، ۹، ۱۷]. ازاین‌رو اعداد دقت یا ADALog می‌سازد؛ اما مطالعاتی که خط، نشست یا پنجره زمانی را نمونه می‌گیرند بدون همسان‌سازی پروتکل ارزیابی قابل F۱ مقایسه مستقیم نیستند.

مجموعه‌داده BGL ۳-۲-۲

در آزمایشگاه ملی لارنس Blue Gene/L از لاگ‌های ابررایانه BGL یا Blue Gene/L مجموعه‌داده لیورمور تهیه شده است. در این مجموعه ۴٬۷۴۷٬۹۶۳ پیام لاگ وجود دارد که ۳۴۸٬۴۶۰ پیام، معادل حدود ۷٫۳۴ درصد، با برچسب هشدار به‌عنوان ناهنجار مشخص شده‌اند [۲، ۶]. وجود برچسب در ابتدای هر خط امکان ارزیابی نظارت‌شده را فراهم می‌کند، اما آن برچسب باید پیش از ارسال متن به مدل حذف شود تا نشت پاسخ به ورودی رخ ندهد.

مطالعات مختلف BGL را با پروتکل‌های متفاوت به‌کار گرفته‌اند. LogBERT پنجره زمانی پنج‌دقیقه‌ای با Hybrid Prompting از پنجره مبتنی بر گره استفاده کرده و BERT-Log میانگین طول ۵۶۲ رخداد ساخته است؛ پنجره یک‌دقیقه‌ای و نمونه متوازن از هزار توالی عادی و هزار توالی ناهنجار را ارزیابی کرده است [۲، ۳، ۹]. در پژوهش حاضر، مسئله در سطح خط تعریف می‌شود؛ بنابراین برچسب مرجع هر خط فقط حذف می‌شود Qwen برای ارزیابی نگه‌داری و از ورودی.

تجزیه، نرمال‌سازی و قالب لاگ ۴-۲-۲

هدف تجزیه لاگ جداکردن کلمات ثابت قالب از پارامترهای متغیر است. روش‌های سنتی از فراوانی، شباهت با پرامپت‌تیونینگ و تعداد کمی نمونه برچسب‌خورده، و LogPPT یا قواعد نحوی استفاده می‌کنند؛ معنا را نیز وارد فرایند استخراج ،LLM با ترکیب درخت پیشوند و فراخوانی محدود LogParser-LLM با میانگین ۳٫۶ میلیون لاگ در هر LogPub در ارزیابی LogParser-LLM قالب کرده‌اند [۱۸، ۱۹]. گروه‌بندی ۹۰٫۶ درصد و دقت تجزیه ۸۱٫۱ درصد F۱ ،LLM مجموعه، به‌طور متوسط با ۲۷۲٫۵ فراخوانی گزارش کرده است [۱۸].

نرمال‌سازی در این پژوهش یک تجزیه کامل مبتنی بر خوشه‌بندی نیست؛ بلکه تبدیل قطعی مقادیر پویا به ،مقادیر هگزادسیمال ،IP جانشین‌های پایدار برای ساخت قالب نرمال‌شده است. شناسه گره، واحدها، نشانی تاریخ و زمان، مسیرها و اعداد پویا ماسک می‌شوند. بااین‌حال صفر و غیرصفر با دو جانشین متمایز نگه داشته می‌شوند، زیرا برخی پیام‌های عملیاتی تفاوت معنایی خود را از <NON_ZERO> و <ZERO> همین وضعیت عددی می‌گیرند. این طراحی با اصل جداسازی قالب و پارامتر در مطالعات تجزیه لاگ سازگار است، اما قاعده جانشینی آن به‌طور مشخص برای خط لوله این پژوهش تعریف شده است [۱۸، ۱۹].

یک نتیجه عملی نرمال‌سازی، کاهش تنوع ظاهری پیام‌هایی است که یک رخداد واحد را با شناسه یا عدد متفاوت بیان می‌کنند. در نتیجه، تصمیم سطح قالب می‌تواند میان خطوط هم‌معنا بازاستفاده شود. در این حالت دقت نرمال‌سازی اهمیت دارد: ماسک بیش‌ازحد ممکن است نشانه معنایی را حذف کند و ماسک کم نیز تعداد قالب‌ها و فراخوانی مدل را افزایش می‌دهد. مطالعات LogLAA نیز LogParser-LLM و LogPPT، حساسیت تحلیل پایین‌دستی به کیفیت آماده‌سازی و تجزیه را گزارش کرده‌اند [۱۸، ۱۹، ۲۱].

۲-۲-۵ روش‌های قاعده‌محور، یادگیری ماشین و مدل‌های ترتیبی

روش‌های قاعده‌محور الگوهای از پیش تعریف‌شده را با واژه کلیدی یا عبارت منظم تطبیق می‌دهند. مزیت آن‌ها تصمیم سریع و قابل ردیابی برای الگوهای شناخته‌شده است، ولی دامنه کشف به قواعد نوشته‌شده محدود می‌شود و نگه‌داری قواعد در برابر تغییر قالب‌ها هزینه‌بر است [۲، ۲۲]. روش‌های کلاسیک یادگیری ماشین، PCA، ماشین بردار پشتیبان پس از استخراج بردار فراوانی یا ویژگی‌های آماری، از الگوریتم‌هایی مانند تک‌کلاسه یا درخت تصمیم برای تشخیص انحراف استفاده می‌کنند [۱، ۱۰].

رخداد بعدی را از رخدادهای قبلی پیش‌بینی LSTM لاگ را همانند توالی زبان مدل‌سازی و با DeepLog می‌کند؛ رخدادی که در مجموعه پیش‌بینی‌های محتمل قرار نگیرد ناهنجار تلقی می‌شود. این روش آموزش افزایشی و یک جریان تشخیص و عیب‌یابی را نیز ارائه می‌کند [۱]. روش‌های بعدی مانند LogBERT، وابستگی دوطرفه و وظایف خودنظارتی را جایگزین اتکای صرف به پیش‌بینی رخداد بعدی کرده‌اند [۲]. این خانواده برای ناهنجاری‌های ترتیبی مناسب است، اما به تعریف توالی و آموزش روی داده سامانه وابستگی دارد.

۲-۲-۶ مدل‌های زبانی پیش‌آموزش‌دیده و مدل‌های زبانی بزرگ

بازنمایی زمینه‌مند واژه‌ها را از متن فرا می‌گیرند و سپس برای وظیفه BERT مدل‌های پیش‌آموزش‌دیده مانند دو وظیفه «پیش‌بینی کلید ماسک‌شده» و «کمینه‌سازی حجم ابرکره» را بر LogBERT خاص تنظیم می‌شوند. توالی‌های عادی به‌کار می‌گیرد BERT-Log مدل BERT و یک لایه تمام‌متصل را برای طبقه‌بندی توالی‌های عادی به‌کار می‌گیرد [۲]. LogFiT نیز BGL [۳] گزارش کرده است برابر ۹۹٫۴ درصد را برای F۱ تنظیم کرده و در پروتکل خود با مدل‌سازی توکن ماسک‌شده، فقط از لاگ‌های عادی برای یادگیری استفاده می‌کند [۱۳].

با ظرفیت زیاد و پیش‌آموزی گسترده است که می‌تواند Transformer مدل زبانی بزرگ معمولاً یک مدل دستور متنی را دنبال و خروجی تولید کند. در حوزه تشخیص ناهنجاری، این مدل ممکن است مستقیماً نقش آشکارساز داشته باشد یا برای تولید داده، تبیین و پشتیبانی از مدل کوچک‌تر استفاده شود؛ این دو نقش در «LLM برای تولید» [۲۴] نام‌گذاری شده‌اند. Ding و Xu رده‌بندی مرور «LLM برای تشخیص» و «LLM برای تولید» به‌ترتیب و همکاران نیز روش‌های لاگ را براساس آماده‌سازی، بازنمایی، مدل De la Cruz Cabello مرور نظام‌مند و شیوه ارزیابی دسته‌بندی می‌کند [۲۲].

۲-۲-۷ مدل Qwen۲٫۵:۷B-Instruct

خانواده‌ای از مدل‌های متن‌باز با نسخه‌های پایه و دستورمحور در اندازه‌های ۰٫۵ تا ۷۲ میلیارد Qwen۲٫۵ پارامتر است. گزارش فنی این خانواده، پیش‌آموزی بر ۱۸ تریلیون توکن و پس‌آموزی با بیش از یک میلیون نمونه تنظیم دستوری و مراحل تقویت یادگیری را شرح می‌دهد. نسخه ۷B در این گزارش حدود ۶٫۵ میلیارد پارامتر غیرتعبیه‌ای دارد و نسخه Instruct برای پیروی از دستور و تولید پاسخ ساخت‌یافته بهینه شده است [۲۵].

به‌صورت محلی و بدون تنظیم دقیق برای طبقه‌بندی دودویی هر Qwen۲٫۵:۷B-Instruct در این پژوهش قالب نرمال‌شده استفاده می‌شود. نبود تنظیم اختصاصی باعث می‌شود تفاوت دو روش آزمایش‌شده به معماری تصمیم و متن پرامپت مربوط باشد، نه تغییر وزن‌های مدل. انتخاب یک مدل و یک نسخه پرامپت ثابت برای Prompt-only یا Hybrid هر دو روش، شرط لازم برای نسبت‌دادن اختلاف نتایج به سازوکار است.

۲-۲-۸ مهندسی پرامپت و خروجی ساخت‌یافته

پرامپت مجموعه دستورها، زمینه دامنه، تعریف برچسب‌ها و قالب خروجی است که رفتار مدل را در زمان استنتاج هدایت می‌کند. LogGPT مبتنی بر ChatGPT انسانی اثر نوع پرامپت، اندازه پنجره و تزریق دانش از راهبردهای چندمرحله‌ای و درخواست LogPrompt بررسی کرده است [۶]. Spirit و BGL را روی تبیین برای تجزیه و تشخیص ناهنجاری استفاده می‌کند و بدون آموزش درون‌دامنه روی چند مجموعه لاگ نیز حساسیت وظایف عملیات فناوری اطلاعات به پرامپت، زنجیره فکر و OpsEval ارزیابی شده است [۷]. یادگیری درون‌متنی را در سطح معیار ارزیابی بررسی می‌کند [۲۳].

برای طبقه‌بندی خودکار، خروجی باید از متن آزاد به قرارداد ماشین‌خوان تبدیل شود. قرارداد این پژوهش فقط JSON یکی از دو شیء یعنی {"label":"۰"} و {"label":"۱"} برای عادی و برای ناهنجار را می‌پذیرد. JSON پاسخ دارای متن اضافی، برچسب نامعتبر یا ناقص با وضعیت INVALID ثبت می‌شود و وارد کش نامعتبر است، زیرا قرارداد Ollama نمی‌گردد. مستندات رسمی JSON امکان درخواست خروجی ساخت‌یافته با قالب یا طرح تصادفی‌بودن با دمای صفر را بیان می‌کند [۲۶]. اعتبارسنجی مستقل پاسخ همچنان لازم است، زیرا قرارداد خروجی بخشی از پروتکل آزمایش و محاسبه نرخ پاسخ نامعتبر است.

۲-۲-۹ استنتاج محلی با Ollama

پیش‌فرض API یک واسط محلی برای اجرای مدل و فراخوانی آن از طریق Ollama فراهم می‌کند. در دسترس است و پاسخ‌های آن علاوه بر محتوای مدل، شاخص‌های زمان در نشانی localhost:۱۱۴۳۴/api اجرا و شمار توکن را نیز گزارش می‌کنند [۲۶]. استنتاج محلی مانع ارسال لاگ به خدمت خارجی می‌شود و کنترل نسخه مدل و پارامترهای اجرا را ممکن می‌سازد؛ مطالعه کاربرد واقعی De la Cruz Cabello و همکاران نیز استقرار محلی مدل LogBERT را برای محیط دارای محدودیت محرمانگی به‌کار گرفته است [۲۷].

استنتاج محلی به معنی نبود هزینه محاسباتی نیست. زمان بارگذاری مدل، زمان ارزیابی پرامپت و زمان تولید فیلدهای eval_count، prompt_eval_duration، prompt_eval_count، load_duration، total_duration در پاسخ Ollama باید ثبت شوند و فراخوانی تکراری برای پیام‌های هم‌قالب کاهش یابد. مستندات را برای اندازه‌گیری این اجزا معرفی می‌کند [۲۶]. این داده‌ها در کنار زمان انتهابه‌انتهای eval_duration و برنامه، مبنای تحلیل کارایی قرار می‌گیرند.

۲-۲-۱۰ کش سطح قالب و اعتبارسنجی تصمیم

از کش و LLM برای کاهش فراخوانی FlexLog. کش نگاشت یک کلید پایدار به تصمیم معتبر قبلی است با واگذاری نمونه‌های مطمئن به مدل کوچک‌تر، تعداد فراخوانی AdaptiveLog بازیابی استفاده می‌کند و نیز با تجمیع الگوها در درخت LogParser-LLM. مدل بزرگ را تا ۷۳ درصد کاهش داده است [۱۰، ۱۲] برای پیام‌های مشابه جلوگیری می‌کند [۱۸]. این مطالعات نشان می‌دهند LLM پیشوند از تکرار فراخوانی کنترل هزینه استنتاج بخشی مستقل از کیفیت طبقه‌بندی است.

،در پژوهش حاضر کلید کش از نسخه پرامپت، نام و نسخه مدل، قالب نرمال‌شده و در صورت فعال بودن مؤلفه، سطح شدت و دسته پیام ساخته می‌شود. بنابراین با تغییر پرامپت یا مدل، تصمیم قبلی به‌اشتباه بازاستفاده خطای ارتباط یا پاسخ ناقص در کش قرار ،INVALID نمی‌شود. فقط پاسخ معتبر ۰ یا ۱ ذخیره می‌شود؛ نمی‌گیرد. نرخ برخورد کش، نرخ عدم برخورد و تعداد کلیدهای یکتا همراه با منبع تصمیم ثبت می‌شود تا اثر کش از اثر مدل و قواعد جدا شود.

۲-۲-۱۱ دو معماری Prompt-only و Hybrid

این پژوهش، خط پس از استخراج فیلدها و نرمال‌سازی ابتدا در کش جست‌وجو می‌شود Hybrid در معماری در نبود تصمیم کش‌شده، مجموعه‌ای محدود از قواعد قطعی و با اطمینان بالا بررسی می‌گردد. اگر هیچ ارسال و پاسخ پس از اعتبارسنجی ذخیره می‌شود. منبع هر تصمیم Qwen قاعده‌ای تصمیم ندهد، قالب به یک آبشار تصمیم است و هدف آن سنجش Hybrid ثبت می‌شود. در نتیجه llm یا guard ،cache به‌صورت است LLM اثر قاعده قطعی پیش از.
guard همان نرمال‌سازی، کلید کش، مدل و قرارداد خروجی حفظ می‌شود، اما Prompt-only در معماری برای هر قالبِ فاقد Qwen قطعی غیرفعال است. دانش دامنه و قواعد تشخیص در متن پرامپت بیان می‌شوند و کش تصمیم می‌گیرد. مقایسه منصفانه مستلزم ثابت ماندن مجموعه داده، ترتیب ورودی، نسخه مدل، گزینه‌های آن است که Hybrid Prompting نرمال‌سازی و سیاست اعتبارسنجی است. تفاوت این تعریف با ،Ollama روش پارک و همکاران شواهد سطح رخداد، شباهت توالی و نمونه عادی مشابه را در یک پرامپت ترکیب می‌کند و طبقه‌بندی آن در سطح توالی است [۹].

۲-۲-۱۲ معیارهای ارزیابی

خطوط FP ،تعداد خطوط عادیِ درست TN ،تعداد ناهنجاری‌های درست کشف‌شده TP ،برای مسئله دودویی ناهنجاری‌های ازدست‌رفته است. دقت کل نسبت همه تصمیم‌های درست به FN عادی با برچسب ناهنجار و سهم Recall سهم تشخیص‌های ناهنجار درست از همه پیش‌بینی‌های ناهنجار؛ Precision کل نمونه‌هاست؛ است Recall و Precision میانگین هارمونیک F۱ ناهنجاری‌های کشف‌شده از همه ناهنجاری‌های واقعی؛ و [۳، ۹].

$$\text{Accuracy} = (TP + TN) / (TP + TN + FP + FN)$$

$$\text{Precision} = TP / (TP + FP) \quad \text{Recall} = TP / (TP + FN) \quad F۱ = ۲ \times \text{Precision} \times \text{Recall} / (\text{Precision} + \text{Recall})$$

گزارش شوند. افزون Accuracy باید در کنار Recall و Precision ،F۱ ،BGL به دلیل نامتوازن بودن منبع تصمیم و آمار برخورد کش ،LLM زمان پاسخ، تعداد فراخوانی ،INVALID بر این چهار معیار، نرخ برای مقایسه دو معماری ضروری است. گزارش این شاخص‌ها کمک می‌کند افزایش سرعت ناشی از کش یا با کیفیت طبقه‌بندی اشتباه نشود guard.

پیشینه تحقیق ۳-۲

روش‌های ترتیبی و پیش‌آموزش‌دیده ۱-۳-۲

برای یادگیری توالی رخدادهای عادی، تشخیص LSTM را به‌عنوان یک مدل DeepLog و همکاران Du و دو هدف BERT از LogBERT و همکاران در Guo .انحراف و به‌روزرسانی برخط ارائه کردند [۱] و HDFS، BGL خودنظارتی برای یادگیری الگوی توالی‌های عادی بهره گرفتند و روش را روی را با طبقه‌بند BERT بازنمایی BERT-Log در Liao و Chen .ارزیابی کردند [۲] Thunderbird BGL و ۹۹٫۴ درصد برای HDFS برابر ۹۹٫۳ درصد برای F۱ تمام‌متصل ترکیب کردند و در پروتکل خود .گزارش دادند [۳]

top-k از تنظیم مدل زبانی پیش‌آموزش‌دیده با وظیفه توکن ماسک‌شده روی لاگ‌های عادی و قاعده LogFiT ADALog .آزمایش شده است [۱۳] Thunderbird و HDFS، BGL برای تشخیص استفاده می‌کند و روی روی لاگ‌های عادی، احتمال بازسازی توکن DistilBERT وابستگی به توالی و تجزیه را حذف و با تنظیم و آستانه صدکی تطبیقی، تصمیم سطح خط تولید می‌کند [۱۷]. این دو مطالعه نشان می‌دهند مدل زبانی پیش‌آموزش‌دیده می‌تواند بدون برچسب ناهنجاری به‌کار رود، اما همچنان به مرحله آموزش یا تنظیم روی داده .مقصد نیاز دارد

روش‌های مبتنی بر آموزش یا تنظیم LLM ۲-۳-۲

را برای پیش‌بینی رخداد بعدی آموزش و سپس با یادگیری GPT ابتدا LogGPT در Trabelsi و Yuan، Han انجام Thunderbird و HDFS، BGL تقویتی به هدف تشخیص ناهنجاری هم‌راستا کردند؛ ارزیابی روی برای بردارهای معنایی، یک نگاشت بین بازنمایی‌ها BERT از LogLLM و همکاران در Guan .شد [۴] به‌عنوان طبقه‌بند استفاده کردند و یک فرایند آموزشی چندمرحله‌ای را روی چهار مجموعه LLaMA و .گزارش کردند [۵] Thunderbird و HDFS، BGL، Liberty

سه مرحله افزایش داده، تنظیم پارامترکارا و بهینه‌سازی برخط را LLM-LADE و همکاران در Zhang با Llama۳-۸B برای تولید نمونه و GPT-٤ برای تشخیص و تبیین هم‌زمان ترکیب کردند؛ در پیاده‌سازی از RoBERTa، را روی ReFT و LoRA نیز Pang و Zhu، Lim .برای مدل مقصد استفاده شد [۱۵] LoRA از نظر پایداری، کارایی نمونه و تعمیم بین مجموعه‌ها بررسی کردند [۱۴]. این دسته Llama-۳ و GPT-۲ کیفیت مدل را از طریق تغییر پارامترها یا نگاشت بازنمایی افزایش می‌دهد، اما هزینه آموزش و وابستگی به .داده تنظیم را وارد فرایند می‌کند

روش‌های مبتنی بر پرامپت ۳-۳-۲

Spirit و BGL برای تشخیص ناهنجاری ChatGPT امکان استفاده از LogGPT و همکاران در مطالعه Qi و Liu .را با پرامپت‌های مختلف بررسی کردند و علاوه بر برچسب، تبیین متنی را ارزیابی کردند [۶] مجموعه‌ای از راهبردهای پرامپت برای تجزیه و تشخیص ارائه و آن را بدون LogPrompt همکاران در آموزش درون‌دامنه روی ۹ مجموعه داده آزمایش کردند؛ نتایج مقاله بهبود تا ۵۵٫۹ درصد نسبت به روش‌های آموزش‌دیده و امتیاز تفسیرپذیری ۴٫۴۲ از ۵ در ارزیابی متخصصان را گزارش می‌کند [۷]. فایل سال ۲۰۲۳ موجود در مجموعه منابع، نسخه اولیه همین خط پژوهش است و در جدول به‌عنوان مطالعه مستقل تکرار نشده .است

سه نوع شواهد را در پرامپت ترکیب کردند: مشاهده Hybrid Prompting در Lee و Choi، Park و مدل تعبیه GPT-٤o-mini رخدادهای دیده‌نشده، شباهت یا توزیع توالی و نزدیک‌ترین نمونه عادی. روش با برابر F۱ آزمایش شد و میانگین بهبود Thunderbird و BGL، HDFS روی Qwen۳-Embedding-۸B در پروتکل توالی یک‌دقیقه‌ای ۷۷٫۸۱ BGL روش روی F۱ واحد درصد نسبت به خطپایه‌ها گزارش شد؛ ۹٫۹۳ درصد بود [۹]. این نتیجه به دلیل سطح توالی و نمونه متوازن با طبقه‌بندی خطبه‌خط این پژوهش قابل مقایسه عددی مستقیم نیست.

بازیابی، روش‌های ترکیبی و کاهش هزینه ۴-۳-۲

RAGLog را در تشخیص LLM ،پایگاه برداری از لاگ‌های عادی می‌سازد و با بازیابی نمونه‌های مرتبط Thunderbird [۸] و BGL بدون آموزش اختصاصی هدایت می‌کند؛ ارزیابی آن روی انجام شده است. AdaptiveLog همکاری مدل کوچک و بزرگ را به‌کار می‌گیرد: عدم‌قطعیت مدل کوچک زمان فراخوانی LLM را تعیین می‌کند و نمونه‌های دشوار مشابه به پرامپت افزوده می‌شوند؛ مقاله کاهش هزینه فراخوانی LLM تا ۷۳ درصد را گزارش می‌کند [۱۲].

Hadadi و همکاران در FlexLog خروجی درخت تصمیم ،KNN را با شبکه پیش‌خور و Mistral، کش و داده‌های مصنوعی LOGEVOL، ADFA بازیابی ترکیب کردند و روش را روی مجموعه‌های ناپایدار با ۶۲٫۸۷ واحد درصد داده برچسب‌خورده F۱ ارزیابی کردند. مقاله دست‌کم ۱٫۲ واحد درصد بهبود HDFS کمتر و در برخی تنظیم‌ها تا ۱۳ واحد درصد بهبود را گزارش می‌کند [۱۰]. مطالعه دیگر همین گروه مقایسه انجام داد و برتری تنظیم LOGEVOL را روی نسخه‌های ناپایدار GPT-٤ و مهندسی پرامپت GPT-۳ تنظیم اختصاصی را در پروتکل خود گزارش کرد [۱۱].

SemiRALD از ChatGPT، سپس BiLSTM و RoBERTa و یادگیری درون‌متنی برای تجزیه، برابر ۷٫۳ F۱ میانگین بهبود BGL و HDFS توجه‌محور برای تشخیص نیمه‌نظارتی استفاده می‌کند و روی و ۸٫۲ درصد را نسبت به دو گروه خطپایه گزارش کرده است [۱۶]. بنابراین واژه «ترکیبی» در این گروه یا ترکیب مرحله تجزیه و طبقه‌بندی اشاره کند و باید ،LLM می‌تواند به ترکیب چند مدل، ترکیب بازیابی و برای هر مطالعه به‌طور عملیاتی تعریف شود.

تجزیه، انتقال، تبیین و کاربرد عملی ۵-۳-۲

مستقیماً آشکارساز ناهنجاری نیستند، اما مرحله ورودی تحلیل لاگ را هدف LogPPT و LogParser-LLM Group با ۳۲ نمونه برچسب‌خورده روی ۱۶ مجموعه به میانگین بیش از ۰٫۹ برای LogPPT می‌گیرند. را LLM با درخت پیشوند، فراخوانی LogParser-LLM رسیده و Parsing Accuracy و Accuracy در مقیاس میلیون‌ها پیام محدود کرده است [۱۸، ۱۹]. این نتایج برای نرمال‌سازی و بازاستفاده تصمیم سطح قالب اهمیت دارند.

LogSynergy و استخراج ویژگی یکپارچه بین سامانه‌ها، انتقال آشکارساز LLM با تفسیر رخداد به‌وسیله بیش از ۸۹ درصد روی داده واقعی و F۱ به سامانه جدید را بررسی کرده و با ۵۰۰۰ توالی برچسب‌خورده تجزیه تطبیقی، ویژگی‌های LogLAA. بیش از ۸۳ درصد روی داده عمومی گزارش کرده است [۲۰]. برای تولید توضیح را در یک چارچوب LLM دوتوجهی و CNN-LSTM، شمارشی، ترتیبی و معنایی یکپارچه قرار داده است [۲۱].

را فقط با توالی‌های LogBERT و همکاران De la Cruz Cabello، AIOps در مطالعه کاربرد واقعی آموزش و به‌صورت محلی مستقر کردند. پنجره ۱۵ ثانیه‌ای با هم‌پوشانی ۱۰ ثانیه Linux syslog عادی بهترین موازنه گزارش‌شده بین اثربخشی و تأخیر بود و ارزیابی کیفی متخصصان کاربرد عملی روش را تأیید آموزش‌دیده LogBERT کرد [۲۷]. این مطالعه بر محرمانگی و استقرار محلی تأکید دارد، ولی طبقه‌بند آن دستورمحور بدون تنظیم LLM است، نه یک.

دسته‌بندی، جدول مقایسه و تحلیل ۲-۴

در تشخیص ناهنجاری، مطالعات بررسی‌شده را LLM براساس مرور نظام‌مند حوزه لاگ و رده‌بندی نقش LLM، می‌توان در چهار مسیر اصلی قرار داد: مدل‌سازی توالی، مدل زبانی پیش‌آموزش‌دیده با آموزش مقصد مبتنی بر پرامپت، بازیابی یا همکاری مدل‌ها [۲۲، ۲۴]. شکل ۲-۱ رابطه این LLM با تنظیم یا انتقال، و مسیرها و مؤلفه‌هایی را که در پژوهش حاضر کنار هم قرار می‌گیرند نشان می‌دهد.

ادبیات تشخیص ناهنجاری لاگ			
روش‌های ترتیبی	**مدل‌های پیش‌آموزش‌دیده**	**با آموزش/تنظیم LLM**	**پرامپت، بازیابی و ترکیب**
DeepLog؛ مدل‌سازی توالی رخداد	LogBERT، BERT-Log و LogFiT	LogGPT، LogLLM، LLM-LADE و PEFT	LogPrompt، RAGLog، FlexLog و AdaptiveLog
نرمال‌سازی قالب + کنترل تصمیم + کاهش فراخوانی مدل + LLM همگرایی مؤلفه‌های موردنیاز: معناشناسی			
دامنه و ورودی	**دو مسیر تصمیم**	**کارایی و قابلیت بازتولید**	**استقرار و ارزیابی**
BGL؛ تصمیم خطبه‌خط؛ فقط حذف برچسب از ورودی	Hybrid: قاعده قطعی سپس LLM Prompt-only: تصمیم صرفاً در پرامپت	کش سطح قالب نرمال‌شده؛ اعتبارسنجی JSON؛ نسخه‌بندی کلید کش	روی Qwen۲٫۵:۷B-Instruct محلی؛ Ollama، Accuracy، Precision، Recall، F۱، زمان و INVALID

شکل ۲-۱: دسته‌بندی پژوهش‌های تشخیص ناهنجاری لاگ و جایگاه پژوهش حاضر

در شکل ۲-۱، مسیر بالایی تحول روش‌ها را نمایش می‌دهد و بخش پایینی مشخصات عملیاتی پژوهش حاضر را از ادبیات متمایز می‌کند. قرارگرفتن یک مطالعه در یک دسته به معنای نبود اجزای دسته دیگر نیست؛ بازیابی و کش را به‌کار می‌گیرد [۱۰] و LLM، هم مدل‌های یادگیری ماشین و هم FlexLog برای نمونه را به مدل ترتیبی پیوند می‌دهد [۱۶] ChatGPT تجزیه با SemiRALD.

جدول ۲-۱: مقایسه مطالعات منتخب در تشخیص ناهنجاری و تحلیل لاگ

مطالعه	سال	روش/مدل	داده‌ها	واحد تحلیل	نوع انطباق	یافته یا کارکرد گزارش‌شده
DeepLog [۱]	۲۰۱۷	LSTM و پیش‌بینی رخداد بعدی	HDFS، OpenStack	توالی	آموزش روی عادی	تشخیص برخط و به‌روزرسانی افزایشی
LogBERT [۲]	۲۰۲۱	BERT با دو هدف خودنظارتی	HDFS، BGL، Thunderbird	توالی	آموزش روی عادی	یادگیری زمینه دوطرفه و الگوی مشترک عادی
BERT-Log [۳]	۲۰۲۲	BERT و طبقه‌بند متصل	HDFS، BGL	توالی/پنجره	تنظیم نظارت‌شده	BGL گزارش‌شده ۹۹٫۴٪ روی F۱
LogGPT (Han) [٤]	۲۰۲۳	GPT، پیش‌بینی رخداد و RL	HDFS، BGL، Thunderbird	توالی	RL آموزش و	هم‌راستاسازی هدف تولید با تشخیص
LogLLM [٥]	۲۰۲۴	BERT + نگاشت + LLaMA	چهار مجموعه شامل BGL	توالی	آموزش چندمرحله‌ای	با طبقه‌بند BERT ترکیب بازنمایی LLaMA
LogGPT (Qi) [٦]	۲۰۲۳	ChatGPT و پرامپت	BGL، Spirit	خط/پنجره	بدون تنظیم	بررسی پرامپت، دانش دامنه و تبیین
LogPrompt [۷]	۲۰۲۴	راهبردهای پرامپت و تبیین	مجموعه لاگ ۹	قالب/توالی	بدون آموزش درون‌دامنه	تا ۵۵٫۹٪ بهبود و امتیاز تبیین ۴٫۴۲/۵

مطالعه	سال	روش/مدل	داده‌ها	واحد تحلیل	نوع انطباق	یافته یا کارکرد گزارش‌شده
RAGLog [۸]	۲۰۲۳	RAG و پایگاه برداری عادی	BGL، Thunderbird	توالی	بدون تنظیم	بازیابی نمونه عادی برای تصمیم معنایی
Hybrid Prompting [۹]	۲۰۲۵	توالی + پرامپت رخداد + نمونه	HDFS، BGL، Thunderbird	توالی یک‌دقیقه‌ای	بدون تنظیم طبقه‌بند	F۱ میانگین ۹٫۹۳ واحد درصد بهبود
FlexLog [۱۰]	۲۰۲۴/۲/۵	ML ensemble + Mistral + RAG + cache	چهار مجموعه ناپایدار	توالی	داده کم/درون‌متنی	کاهش داده برچسب‌خورده و استنتاج کمتر از یک ثانیه
Unstable Logs GPT [۱۱]	۲۰۲۴	GPT-۳ تنظیم‌شده و GPT-۴ پرامپت	LOGEVOL-Hadoop	توالی	تنظیم/پرامپت	برتری تنظیم در پروتکل لاگ ناپایدار
AdaptiveLog [۱۲]	۲۰۲۵	LLM + مدل کوچک بازیابی	چند وظیفه لاگ	وابسته به وظیفه	همکاری تطبیقی	LLM تا ۷۳٪ کاهش هزینه فراخوانی
LogFiT [۱۳]	۲۰۲۵	top-k مدل ماسک‌شده و	HDFS، BGL، Thunderbird	توالی	تنظیم روی عادی	تشخیص خودنظارتی بدون ناهنجاری برچسب‌خورده
PEFT-LAD [۱۴]	۲۰۲۵	LoRA/ReFT روی چند LM	HDFS، BGL، Spirit، Thunderbird	توالی	تنظیم پارامترکارا	تحلیل پایداری، کارایی نمونه و انتقال
LLM-LADE [۱۵]	۲۰۲۵	LoRA + افزایش داده بهینه‌سازی برخط	مجموعه‌های عمومی لاگ	پیام/توالی	PEFT	تشخیص و تبیین همزمان
SemiRALD [۱۶]	۲۰۲۵	ChatGPT parsing + RoBERTa + BiLSTM	HDFS، BGL	توالی	نیمه‌نظارتی	به‌ویژه با داده کم F۱ بهبود متوسط
ADALog [۱۷]	۲۰۲۵	DistilBERT MLM و آستانه تطبیقی	BGL، Thunderbird، Spirit	خط	تنظیم روی عادی	و مستقل از توالی parser بدون
LogParser-LLM [۱۸]	۲۰۲۴	LLM + درخت پیشوند	Loghub-۲k، LogPub	خط/قالب	بدون آموزش برچسبی	گروه‌بندی F۱ کاهش فراخوانی؛ ۹۰٫۶٪
LogPPT [۱۹]	۲۰۲۳	پرامپت‌تیونینگ RoBERTa	مجموعه ۱۶	خط/قالب	Few-shot۳۲ ؛ نمونه	بیش از ۰٫۹ میانگین دقت گروه/تجزیه
LogSynergy [۲۰]	۲۰۲۵	تفسیر LLM و انتقال	عمومی و واقعی	توالی	انتقال با ۵۰۰۰ برچسب	بیش از ۸۳٪ عمومی و ۸۹٪ واقعی F۱
LogLAA [۲۱]	۲۰۲۶	parser + CNN-LSTM + LLM explanation	چند مجموعه عمومی	توالی	آموزش چندجزئی	تشخیص یکپارچه و تولید توضیح
AIOps [۲۷] واقعی	۲۰۲۶	LogBERT محلی	Linux syslog واقعی	پنجره زمانی	خودنظارتی	استقرار محلی تحت محدودیت محرمانگی

در بسیاری از مطالعات حضور دارد، اما پروتکل‌های آن یکسان نیستند: BGL جدول ۲-۱ نشان می‌دهد که برخی پیام را مستقیماً طبقه‌بندی می‌کنند، برخی پنجره زمانی یا توالی مبتنی بر گره می‌سازند و برخی فقط به‌تنهایی BGL بخشی نمونه‌گیری‌شده را ارزیابی می‌کنند [۲، ۳، ۶، ۹، ۱۷]. بنابراین استفاده مشترک از نام هم‌ارزی آزمایش‌ها را تضمین نمی‌کند.

از نظر انطباق مدل، چهار حالت قابل مشاهده است: آموزش از ابتدا یا روی عادی، تنظیم نظارت‌شده یا پارامترکارا، پرامپت بدون تغییر وزن و همکاری چند مدل. پژوهش حاضر در حالت پرامپت بدون تغییر وزن قرار می‌گیرد، ولی با نرمال‌سازی و کش قطعی در سطح قالب و دو مسیر تصمیم کنترل‌شده تکمیل می‌شود. این ترکیب در جدول ۲-۲ با نزدیک‌ترین مطالعات مقایسه شده است.

جدول ۲-۲: مقایسه سازوکار مطالعات نزدیک با مؤلفه‌های پژوهش حاضر

مطالعه	سازوکار اصلی	سطح تصمیم	کنترل هزینه	نوع استقرار	تفاوت با پژوهش حاضر
LogGPT (Qi) [۶]	پرامپت ChatGPT	خط/پنجره	نمونه‌گیری به علت محدودیت API	خدمت تجاری	محلی، کش قالب و Qwen فاقد guard مقایسه

مطالعه	سازوکار اصلی	سطح تصمیم	کنترل هزینه	نوع استقرار	تفاوت با پژوهش حاضر
LogPrompt [۷]	راهبرد پرامپت و تبیین	قالب/توالی	بدون کش عملیاتیِ محور مقایسه	های مختلفLLM	تمرکز بر راهبرد پرامپت، نه cache و guard آبشار
RAGLog [۸]	بازیابی لاگ عادی	توالی	retrieval کاهش زمینه با	بیرونی LLM	پایگاه برداری نمونه است، نه کش تصمیم قالب
Hybrid Prompting [۹]	ترکیب شواهد در پرامپت	توالی	بازیابی نزدیک‌ترین نمونه	GPT-٤o-mini	به معنای ترکیب Hybrid پرامپت و در سطح توالی است
FlexLog [۱۰]	ensemble + LLM + RAG + cache	توالی	ML کش و واگذاری به	Mistral	داده‌های ناپایدار و ترکیب چند خطمحور BGL طبقه‌بند؛ نه
AdaptiveLog [۱۲]	LLM + مدل کوچک	چند وظیفه	فراخوانی براساس عدم‌قطعیت	همکاری دو مدل	به مدل کوچک آموزش‌دیده نیاز دارد
ADALog [۱۷]	و آستانه تطبیقی MLM	خط	مولد فراخوانی LLM نمی‌شود	مدل محلی	fine-tuning نیازمند ؛ فاقد پرامپتDistilBERT دودویی
LogParser-LLM [۱۸]	LLM + درخت پیشوند	قالب	تجمیع در درخت و فراخوانی محدود	قابل انتخاب LLM	هدف تجزیه است؛ تصمیم ناهنجاری تولید نمی‌کند
واقعی [۲۷] AIOps	محلی LogBERT	پنجره	تنظیم پنجره و مدل خودنظارتی	کاملاً محلی	syslog، مدل آموزش‌دیده روی دستورمحور Qwen نه
پژوهش حاضر	و Hybrid دو روش Prompt-only	BGL خط	نرمال‌سازی + کش تصمیم قالب	روی Qwen۲,۵:۷B Ollama	guard مقایسه کنترل‌شده بیرونی با قاعده درون پرامپت

:مطالعات نزدیک هرکدام بخشی از مسئله را پوشش داده‌اند، بر رفتار پرامپت LogGPT و LogPrompt بر واگذاری تطبیقی AdaptiveLog، بر ترکیب کش و مدل‌های کلاسیک FlexLog، بر بازیابی RAGLog بر کاهش فراخوانی در سطح قالب تمرکز دارند [۶–۱۰، ۱۲، ۱۸]. در مجموعه منابع LogParser-LLM و قطعی بیرون guard بررسی‌شده، مقایسه کنترل‌شده دو مسیر با مدل، داده، نرمال‌سازی و کش یکسان—یکی با مدل و دیگری با انتقال همان دانش به پرامپت—به‌صورت مستقیم گزارش نشده است.

۲-۵ شکاف تحقیقاتی و جایگاه پژوهش حاضر

نخستین شکاف به ناهمگونی واحد تحلیل و پروتکل BGL مربوط است. نتایج خطمحور، پنجره مبتنی بر گره، پنجره یک‌دقیقه‌ای و توالی پنج‌دقیقه‌ای در ادبیات با یک نام مجموعه داده گزارش شده‌اند [۲، ۳، ۹، ۱۷]. این به ورودی و گزارش ماتریس BGL وضعیت نیازمند تعریف صریح واحد تصمیم، جلوگیری از ورود برچسب اغتشاش در سطح همان واحد است. پژوهش حاضر واحد تصمیم را خط تعیین می‌کند و برچسب را فقط برای ارزیابی نگه می‌دارد.

دومین شکاف، تفکیک اثر کیفیت تشخیص از هزینه استنتاج است. برخی مطالعات کش، بازیابی یا مدل کوچک به‌کار برده‌اند [۱۰، ۱۲، ۱۸]، اما نرخ برخورد کش، منبع هر تصمیم، زمان LLM را برای کاهش تماس با گزارش نمی‌شود. در F۱ و Recall، Precision، Accuracy همیشه در کنار INVALID پاسخ و نرخ پاسخ یا بازاستفاده guard، پژوهش حاضر این متغیرها بخشی از خروجی آزمایش‌اند تا معلوم شود هر بهبود از مدل تصمیم ناشی شده است.

سومین شکاف به جایگاه قواعد دامنه مربوط است. در پژوهش‌های پرامپتی، دانش در متن دستور یا نمونه‌ها قرار می‌گیرد [۶، ۷، ۹]؛ در روش‌های قاعده‌محور تصمیم خارج مدل است؛ و در سامانه‌های ترکیبی، مدل با همان منطق LLM کلاسیک یا بازیابی نقش دروازه را دارد [۱۰، ۱۲]. مقایسه مستقیم قاعده قطعی خارج از به‌صورت دستور درون پرامپت، تحت شرایط ثابت، در مجموعه مطالعات مرورشده مشاهده نشد. دو روش برای سنجش همین تمایز تعریف شده‌اند Prompt-only و Hybrid.

متن‌باز متوسط در استقرار کاملاً محلی و بدون تنظیم LLM چهارمین شکاف، محدود بودن شواهد درباره یک GPT- یا ChatGPT است. بخشی از مطالعات از خدمات تجاری مانند BGL دقیق برای طبقه‌بندی خطبه‌خط را آموزش یا تنظیم کرده‌اند [۵، ۱۰، ۱۵] و Mistral یا LLaMA استفاده کرده‌اند [۶، ۹]، بخشی ۴o-mini خودنظارتی بهره گرفته است [۲۷]. پژوهش حاضر LogBERT مطالعه محلی سال ۲۰۲۶ از محلی و وزن‌های ثابت ارزیابی می‌کند [۲۵، ۲۶] Ollama را با Qwen۲٫۵:۷B-Instruct.

شامل BGL برآیند این شکاف‌ها جایگاه پژوهش را چنین مشخص می‌کند: یک خط لوله بازتولیدپذیر برای JSON استخراج و حذف برچسب، نرمال‌سازی حساس به معنا، کش نسخه‌بندی‌شده در سطح قالب، خروجی اعتبارسنجی‌شده و ثبت منبع تصمیم؛ سپس مقایسه دو معماری که فقط در محل اعمال دانش قطعی تفاوت دارند. این صورت‌بندی ادعای برتری پیشینی هیچ مسیر را مطرح نمی‌کند و نتیجه باید از آزمایش کامل و یکسان دو روش به‌دست آید.

۲-۶ جمع‌بندی

در این فصل مفاهیم لاگ، ناهنجاری، واحد تحلیل، BGL، مدل‌های ترتیبی، تجزیه و نرمال‌سازی، مهندسی پرامپت، استنتاج محلی، کش و معیارهای ارزیابی، Qwen۲٫۵، LLM، زبانی پیش‌آموزش‌دیده، LLM به تنظیم BERT تعریف شد. مرور پیشینه نشان داد مسیر پژوهش از پیش‌بینی توالی و مدل‌های پرامپت، بازیابی و سامانه‌های چندجزئی گسترش یافته است [۱–۲۴].

مقایسه مطالعات نشان داد که نوع واحد تحلیل، نیاز یا عدم نیاز به آموزش، سازوکار کنترل هزینه و محل استقرار برای تفسیر نتایج ضروری‌اند. شکاف استخراج‌شده از منابع، نبود مقایسه مستقیم و کنترل‌شده بین محلی در طبقه‌بندی Qwen و انتقال دانش آن به پرامپت، همراه با کش قالب و LLM قطعی بیرون guard است. فصل بعد باید معماری نرم‌افزار، قواعد نرمال‌سازی، ساختار پرامپت‌ها، کلید کش BGL خطبه‌خط، پروتکل اجرای دو روش و شیوه محاسبه معیارها را به‌صورت عملیاتی تشریح کند، Ollama تنظیمات.

۲-۷ منابع فصل دوم

[۱] Du, M.; Li, F.; Zheng, G.; Srikumar, V. DeepLog: Anomaly Detection and Diagnosis from System Logs through Deep Learning. ACM CCS, ۲۰۱۷. DOI: ۱۰,۱۱۴۵/۳۱۳۳۹۵۶,۳۱۳۴۰۱۵.

[۲] Guo, H.; Yuan, S.; Wu, X. LogBERT: Log Anomaly Detection via BERT. arXiv:۲۱۰۳,۰۴۴۷۵, ۲۰۲۱.

[۳] Chen, S.; Liao, H. BERT-Log: Anomaly Detection for System Logs Based on Pre-trained Language Model. Applied Artificial Intelligence, ۳۶(۱), ۲۰۲۲. DOI: ۱۰,۱۰۸۰/۰۸۸۳۹۵۱۴,۲۰۲۲,۲۱۴۵۶۴۲.

[۴] Han, X.; Yuan, S.; Trabelsi, M. LogGPT: Log Anomaly Detection via GPT. ۲۰۲۳.

[۵] Guan, W.; Cao, J.; Qian, S.; Gao, J.; Ouyang, C. LogLLM: Log-based Anomaly Detection Using Large Language Models. ۲۰۲۴.

[۶] Qi, J.; Huang, S.; Luan, Z.; Fung, C.; Yang, H.; Qian, D. LogGPT: Exploring ChatGPT for Log-Based Anomaly Detection. ۲۰۲۳.

[۷] Liu, Y.; Tao, S.; Meng, W.; Wang, J.; Ma, W.; Chen, Y.; Zhao, Y.; Yang, H.; Jiang, Y. Prompt Engineering Towards Zero-Shot and Interpretable Log Analysis. ۲۰۲۴. (نسخه اولیه: Interpretable Online Log Analysis Using Large Language Models with Prompt Strategies, ۲۰۲۳).

[۸] Pan, J.; Wong, S. L.; Yuan, Y. RAGLog: Log Anomaly Detection using Retrieval Augmented Generation. ۲۰۲۳.

[۹] Park, J.; Choi, E.; Lee, J. Hybrid Prompting for LLM-based Log Anomaly Detection. Journal of KIISE, ۵۲(۱۲), ۲۰۲۵. DOI: ۱۰.۵۶۲۶/JOK.۲۰۲۵.۵۲.۱۲.۱۰۶۷.

[۱۰] Hadadi, F.; Xu, Q.; Bianculli, D.; Briand, L. LLM meets ML: Data-efficient Anomaly Detection on Unstable Logs. ۲۰۲۴–۲۰۲۵.

[۱۱] Hadadi, F.; Xu, Q.; Bianculli, D.; Briand, L. Anomaly Detection on Unstable Logs with GPT Models. ۲۰۲۴.

[۱۲] Ma, L.; Yang, W.; Li, Y.; Fei, B.; Zhou, M.; Li, S.; Jiang, S.; Xu, B.; Xiao, Y. AdaptiveLog: An Adaptive Log Analysis Framework with the Collaboration of Large and Small Language Model. arXiv:۲۵۰۱,۱۱۰۳۱, ۲۰۲۵.

[۱۳] Almodovar, C.; Sabrina, F.; Karimi, S.; Azad, S. LogFiT: Log Anomaly Detection using Fine-Tuned Language Models. ۲۰۲۵.

[۱٤] Lim, Y. F.; Zhu, J.; Pang, G. Adapting Large Language Models for Parameter-Efficient Log Anomaly Detection. ۲۰۲۵.

[۱۵] Zhang, Z.; Li, S.; Zhang, L.; Ye, J.; Hu, C.; Yan, L. LLM-LADE: Large language model-based log anomaly detection with explanation. Knowledge-Based Systems, ۳۲٦, ۱۱٤۰٦٤, ۲۰۲۵. DOI: ۱۰.۱۰۱۶/□.□□□□□□.۲۰۲۵.۱۱۴۰۶۴.

[۱٦] Sun, Y.; Keung, J. W.; Yang, Z.; Liu, S.; Yu, H. K. SemiRALD: A semi-supervised hybrid language model for robust Anomalous Log Detection. Information and Software Technology, ۱۸۳, ۱۰۷۷٤۳, ۲۰۲۵. DOI: ۱۰,۱۰۱٦/j.infsof.۲۰۲۵,۱۰۷۷٤۳.

[۱۷] Pospieszny, P.; Mormul, W.; Szyndler, K.; Kumar, S. ADALog: Adaptive Unsupervised Anomaly Detection in Logs with Self-attention Masked Language Model. arXiv:۲۵۰۵,۱۳٤۹٦, ۲۰۲۵.

[۱۸] Zhong, A.; Mo, D.; Liu, G.; Liu, J.; Lu, Q.; Zhou, Q.; Wu, J.; Li, Q.; Wen, Q. LogParser-LLM: Advancing Efficient Log Parsing with Large Language Models. arXiv:۲٤۰۸,۱۳۷۲۷, ۲۰۲٤.

[۱۹] Le, V.-H.; Zhang, H. Log Parsing with Prompt-based Few-shot Learning. arXiv:۲۳۰۲,۰۷٤۳۵, ۲۰۲۳.

[۲۰] Sui, Y.; Wang, X.; Cui, T.; Xiao, T.; He, C.; Zhang, S.; Zhang, Y.; Yang, X.; Sun, Y.; Pei, D. Bridging the Gap: LLM-Powered Transfer Learning for Log Anomaly Detection in New Software Systems. ۲۰۲۵.

[۲۱] Gao, Y.; Luo, T.; Huang, K.; Tang, J.; Li, X. LogLAA: an adaptive integrated log anomaly analysis framework. Cybersecurity, ۹:۱٤۱, ۲۰۲٦. DOI: ۱۰,۱۱۸٦/s٤۲٤۰۰-۰۲٦-۰۰۵۷۳-۸.

[۲۲] De la Cruz Cabello, M.; Prince Sales, T.; Machado, M. R. AIOps for log anomaly detection in the era of LLMs: A systematic literature review. Intelligent Systems with Applications, ۲۸, ۲۰۰٦۰۸, ۲۰۲۵. DOI: ۱۰.۱۰۱۶/□.□□□□.۲۰۲۵.۲۰۰۶۰۸.

[۲۳] Liu, Y.; Pei, C.; Xu, L.; et al. OpsEval: A Comprehensive Benchmark Suite for Evaluating Large Language Models' Capability in IT Operations Domain. FSE Companion, ۲۰۲۵. DOI: ۱۰.۱۱۴۵/۳۶۹۶۶۳۰.۳۷۲۸۵۷۲.

[۲٤] Xu, R.; Ding, K. Large Language Models for Anomaly and Out-of-Distribution Detection: A Survey. arXiv:۲٤۰۹,۰۱۹۸۰, ۲۰۲۵.

[۲۵] Qwen Team. Qwen۲,۵ Technical Report. arXiv:۲٤۱۲,۱۵۱۱۵, ۲۰۲٤.

[۲٦] Ollama. API Introduction; Structured Outputs; Usage Metrics. مستندات رسمی Ollama، مشاهده‌شده در ۱۴ اوت ۲۰۲۶.

[۲۷] De la Cruz Cabello, M.; Prince Sales, T.; Machado, M. R. Log anomaly detection in AIOps: A real-world implementation using Large Language Models. Systems and Soft Computing, ۸, ۲۰۰٤۷۵, ۲۰۲٦. DOI: ۱۰.۱۰۱۶/□.□□□□.۲۰۲۶.۲۰۰۴۷۵.