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 ..... ۱۰
ماهیت و ساختار لاگ سیستمی ۱-۲-۲ ..... ۱۰
مفهوم ناهنجاری و سطوح تحلیل ۲-۲-۲ ..... ۱۰
مجموعه‌داده BGL ۳-۲-۲ ..... ۱۰
تجزیه، پاک‌سازی و نرمال‌سازی ۴-۲-۲ ..... ۱۱
پارادایم‌های یادگیری در تشخیص ناهنجاری لاگ ۵-۲-۲ ..... ۱۱
مدل‌های ترتیبی، Transformer و مدل‌های زبانی ۶-۲-۲ ..... ۱۱
نقش‌های LLM در خط لوله تحلیل لاگ ۷-۲-۲ ..... ۱۲
Qwen۲.۵:۷B-Instruct ۸-۲-۲ ..... ۱۲
مهندسی پرامپت، یادگیری درون‌متنی و خروجی ۹-۲-۲ ..... ۱۲
بازیابی، کش و کنترل هزینه استنتاج ۱۰-۲-۲ ..... ۱۳
استنتاج محلی با Ollama ۱۱-۲-۲ ..... ۱۳
در پژوهش حاضر Prompt-only و Hybrid دو معماری ۱۲-۲-۲ ..... ۱۳
معیارهای ارزیابی ۱۳-۲-۲ ..... ۱۴

۲-۲-۱۴ انواع الگوهای ناهنجاری در لاگ ۱۴
۲-۲-۱۵ چالش‌های روش‌شناختی و اجرایی ۱۵
۲-۳ پیشینه تحقیق ۱۵
۲-۳-۱ دامنه و شیوه مرور مطالعات ۱۵
۲-۳-۲ روش‌های ترتیبی و مدل‌های زبانی پیش‌آموزش‌دیده ۱۶
۲-۳-۳ آموزش LLM، تنظیم پارامترکارا و انتقال ۱۷
۲-۳-۴ مطالعات مبتنی بر پرامپت ۱۹
۲-۳-۵ بازیابی، همکاری مدل‌ها و کنترل هزینه ۲۰
۲-۳-۶ مطالعات تجزیه و تولید قالب لاگ ۲۰
۲-۳-۷ مرورها، معیار ارزیابی، مدل پایه و استقرار واقعی ۲۱
۲-۳-۸ جمع‌بندی تحلیلی مسیر پژوهش‌ها ۲۲
۲-۴ دسته‌بندی و تحلیل مقایسه‌ای ۲۳
۲-۴-۱ مقایسه براساس واحد تحلیل و پروتکل BGL ۲۶
۲-۴-۲ مقایسه براساس نیاز به آموزش ۲۶
۲-۴-۳ مقایسه کنترل هزینه و استقرار ۲۶
۲-۴-۴ مقایسه بازنمایی ورودی و آماده‌سازی ۲۷
۲-۴-۵ مقایسه سازوکار تصمیم و نوع خروجی ۲۷
۲-۴-۶ مقایسه تبیین‌پذیری و بازتولیدپذیری ۲۸
۲-۵ شکاف تحقیقاتی و جایگاه پژوهش حاضر ۲۹
۲-۵-۱ ناهمگونی پروتکل‌های BGL ۲۹
۲-۵-۲ جدانشدن کیفیت تصمیم از هزینه استنتاج ۲۹
۲-۵-۳ محل اعمال دانش دامنه ۲۹
۲-۵-۴ کش تصمیم سطح قالب ۲۹
۲-۵-۵ متن‌باز متوسط LLM استنتاج محلی یک ۳۰
۲-۵-۶ صورت‌بندی جایگاه پژوهش ۳۰
۲-۶ جمع‌بندی ۳۰

۳۰ .....منابع فصل دوم ۷-۲

فهرست جدول ها

جدول ۲-۱: مشخصات و مقایسه تمام ۲۷ مقاله موجود در پوشه پژوهش ۲۵
جدول ۲-۲: جمع‌بندی دسته‌های مطالعات و ارتباط آن‌ها با طراحی پژوهش ۲۹

فهرست نمودار ها

فهرست شکل ها

شکل ۲-۱: ارتباط ۲۷ مقاله بررسی‌شده و جایگاه پژوهش حاضر ۲۴

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 گزارش‌های متنی یا نیمه‌ساخت‌یافته‌ای هستند که رخدادهای زمان اجرا، وضعیت اجزای ،سامانه، پیام‌های هشدار و اطلاعات عیب‌یابی را ثبت می‌کنند. این داده‌ها در بیشتر زیرساخت‌های نرم‌افزاری شبکه‌ای و محاسباتی تولید می‌شوند و از منابع اصلی پایش سلامت سامانه، تشخیص خطا، تحلیل رخداد و یافتن علت ریشه‌ای به‌شمار می‌روند. با افزایش مقیاس سامانه‌ها، تعداد پیام‌ها و تنوع قالب‌های لاگ نیز افزایش یافته .و تحلیل دستی دیگر پاسخ‌گوی جریان پیوسته داده نیست [۱۴، ۲۰، ۲۱]

پژوهش تشخیص ناهنجاری لاگ از روش‌های قاعده‌محور و آماری به مدل‌های یادگیری توالی، مدل‌های زبانی LSTM توالی رخدادها را با DeepLog .پیش‌آموزش‌دیده و سپس مدل‌های زبانی بزرگ گسترش یافته است را وارد مسئله می‌کنند؛ و پژوهش‌هایی BERT بازنمایی دوطرفه BERT-Log و LogBERT مدل می‌کند؛ از توان پیروی از دستور و درک Hybrid Prompting و RAGLog، LogGPT، LogPrompt مانند .استفاده می‌کنند [۱–۶، ۱۲، ۱۸، ۱۹، ۲۱] LLM معنایی

مدل زبانی در مطالعات مرورشده نقش واحدی ندارد. در برخی پژوهش‌ها طبقه‌بند مستقیم است؛ در برخی دیگر برای پیش‌بینی رخداد بعدی آموزش داده می‌شود؛ در گروهی برای تولید داده یا توضیح به‌کار می‌رود؛ و در چارچوب‌های ترکیبی، مدل بزرگ فقط نمونه‌های دشوار را دریافت می‌کند. مرور نظام‌مند حوزه لاگ و مرور عمومی تشخیص ناهنجاری و داده خارج از توزیع نیز همین تنوع نقش را به‌عنوان یکی از محورهای .اصلی دسته‌بندی معرفی می‌کنند [۱۴، ۲۷]

به همین دلیل، مقایسه پژوهش‌ها باید افزون بر نام مدل، نوع ورودی، واحد تصمیم، نیاز به آموزش، سازوکار حاصل از طبقه‌بندی F۱ آماده‌سازی، نوع استقرار و روش کنترل هزینه را در نظر بگیرد. برای نمونه، عدد خطبه‌خط با نتیجه یک پنجره یک‌دقیقه‌ای یا توالی مبتنی بر نشست هم‌معنا نیست. همچنین کش تصمیم، بازیابی را LLM نمونه و واگذاری به مدل کوچک سه سازوکار متفاوت‌اند، هرچند هر سه تعداد یا هزینه فراخوانی .کاهش می‌دهند [۳، ۱۰، ۱۲، ۱۳]

محدود است. مدل از طریق Qwen۲٫۵:۷B-Instruct و مدل BGL دامنه پژوهش حاضر به مجموعه‌داده و Hybrid و به‌صورت محلی اجرا می‌شود و تنظیم دقیق انجام نمی‌گیرد. مقایسه میان دو روش Ollama اعمال می‌شوند و LLM صورت می‌گیرد: در روش نخست قواعد قطعی پیش از فراخوانی Prompt-only در روش دوم همان دانش دامنه در متن پرامپت قرار می‌گیرد. در هر دو روش، نرمال‌سازی، کش سطح قالب .و اعتبارسنجی خروجی ثابت نگه داشته می‌شوند

این فصل ابتدا مبانی نظری موردنیاز را تعریف می‌کند. سپس هر ۲۷ مقاله موجود در پوشه پژوهش به‌صورت ،هستند LogPrompt مستقل مرور می‌شود؛ حتی دو مقاله [۱] و [۶] که دو نسخه هم‌پوشان از خط پژوهشی جداگانه معرفی و رابطه آن‌ها تصریح می‌شود. در ادامه، جدول ۲۷ ردیفی، شکل ارتباط مطالعات، تحلیل مقایسه‌ای و شکاف تحقیقاتی ارائه می‌گردد. تمام تعریف‌ها و گزاره‌های مربوط به مطالعات به منابع ارجاع .داده شده‌اند و نتایج آزمایش‌های در حال تکمیل پروژه وارد پیشینه نشده‌اند

مبانی نظری و مفاهیم موردنیاز ۲-۲

ماهیت و ساختار لاگ سیستمی ۱-۲-۲

یک پیام لاگ معمولاً شامل سرآیند و بدنه است. سرآیند می‌تواند زمان، سطح شدت، نام مؤلفه، شناسه فرایند یا و LogPPT گره را در بر گیرد و بدنه، شرح رخداد برنامه و پارامترهای زمان اجرا را بیان کند. مطالعات بدنه را به بخش ثابت یا «قالب لاگ» و بخش متغیر یا «پارامتر» تفکیک می‌کنند. کلمات LogParser-LLM ثابت نوع رخداد را نشان می‌دهند و مقادیری مانند شناسه بلوک، نشانی شبکه، مسیر فایل و زمان اجرا میان نمونه‌ها تغییر می‌کنند [۲۳، ۲۴].

لاگ خام به دلیل وجود مقادیر پویا، اختصارات، کدهای خطا و قالب‌های متفاوت، متن طبیعی کامل نیست. بااین‌حال، روابط میان کلمات ثابت و زمینه عملیاتی حامل اطلاعات معنایی‌اند. مدل‌های زبانی پیش‌آموزش‌دیده LLM و LLM‌ها تلاش می‌کنند این اطلاعات را از متن یا توالی استخراج کنند، درحالی‌که روش‌های آماری معمولاً ابتدا پیام را به شناسه قالب یا بردار فراوانی تبدیل می‌کنند [۴، ۱۴، ۱۸، ۱۹].

مفهوم ناهنجاری و سطوح تحلیل ۲-۲-۲

در تشخیص ناهنجاری، مشاهده‌ای ناهنجار تلقی می‌شود که با الگوی رفتار عادی آموخته‌شده یا قواعد میان تغییر هم‌توزیعی که به ناهنجاری منجر می‌شود و تغییر Ding و Xu تعریف‌شده سازگار نباشد. مرور LLM» برای تشخیص» و LLM» را در دو گروه LLM معنایی داده خارج از توزیع تمایز می‌گذارد و نقش «برای تولید» طبقه‌بندی می‌کند [۲۷]. در حوزه لاگ، ناهنجاری می‌تواند در متن یک پیام، فراوانی قالب‌ها ترتیب رخدادها یا معنای یک پنجره زمانی ظاهر شود [۱۴].

واحد تحلیل در ادبیات شامل خط منفرد، قالب، پنجره زمانی، توالی مبتنی بر گره و نشست است. ADALog بر توالی رخدادها تکیه دارند؛ LogBERT و DeepLog پیام منفرد را مستقل تحلیل می‌کند؛ BERT-Log از پنجره یک‌دقیقه‌ای استفاده می‌کند Hybrid Prompting پنجره لغزان مبتنی بر گره می‌سازد؛ و BGL در واحد تحلیل، نحوه تعریف TP، FP، TN و FN و قابلیت مقایسه نتیجه را تعیین می‌کند [۲۲، ۲۱، ۱۹، ۱۸، ۳].

مجموعه‌داده BGL ۳-۲-۲

در آزمایشگاه ملی لارنس لیورمور Blue Gene/L از ابررایانه BGL یا Blue Gene/L مجموعه‌داده گردآوری شده است. این مجموعه ۴،۷۴۷،۹۶۳ پیام دارد و ۳۴۸،۴۶۰ پیام، حدود ۷،۳۴ درصد، با برچسب هشدار به‌عنوان ناهنجار مشخص شده‌اند [۲، ۱۸، ۱۹]. برچسب اصلی در ابتدای هر خط قرار دارد و برای ارزیابی قابل استفاده است، اما حضور آن در ورودی مدل باعث نشت پاسخ می‌شود؛ بنابراین در پژوهش حاضر پیش از تشکیل متن ورودی حذف می‌گردد.

پنجره زمانی پنج‌دقیقه‌ای با میانگین ۵۶۲ رخداد ایجاد LogBERT. در منابع یکسان نیستند BGL پروتکل‌های پنجره Hybrid Prompting شناسه گره، اندازه پنجره و گام را به‌کار می‌گیرد؛ BERT-Log می‌کند؛ پس از ADALog یک‌دقیقه‌ای و نمونه متوازن هزار توالی عادی و هزار توالی ناهنجار می‌سازد؛ و پاک‌سازی، مجموعه‌ای از پیام‌های یکتای عادی و ناهنجار را در سطح خط بررسی می‌کند [۳، ۱۸، ۱۹، ۲۲].

دو هزار لاگ متوالی از بخش API، به دلیل محدودیت نرخ ChatGPT، مبتنی بر LogGPT در پژوهش K-means نیز برای ساخت خوشه‌ها و پایگاه برداری از نمونه‌گیری و RAGLog آزمون انتخاب شده است [۲]. در باید همراه با واحد «BGL» استفاده می‌شود [۱۲]. این تفاوت‌ها نشان می‌دهد عبارت «ارزیابی روی تحلیل، نمونه‌گیری، پنجره و نسبت کلاس‌ها گزارش شود.

۲-۲-۴ تجزیه، پاک‌سازی و نرمال‌سازی

،تجزیه لاگ فرایند تبدیل پیام خام به قالب ثابت و فهرست پارامترهای متغیر است. روش‌های سنتی از فراوانی را با پرامپت‌تیونینگ و تعداد اندکی نمونه RoBERTa مدل LogPPT .شباهت و قواعد نحوی استفاده می‌کنند را با درخت LLM نیز معناشناسی LogParser-LLM برای برچسب‌گذاری توکن‌های پارامتر تنظیم می‌کند؛ .پیشوند ترکیب می‌کند تا فقط برای الگوهای نیازمند تصمیم فراخوانی انجام شود [۲۳، ۲۴]

خوشه‌بندی‌شده کامل نیست، بلکه نگاشتی قطعی از مقادیر پویا به parser نرمال‌سازی پژوهش حاضر یک مقادیر هگزادسیمال، تاریخ و IP، جانشین‌های ثابت برای تشکیل قالب قابل کش است. شناسه‌های گره، نشانی <NON_ZERO> و <ZERO> زمان، مسیرها و اعداد پویا پوشانده می‌شوند. صفر و غیرصفر به‌ترتیب با متمایز می‌مانند، زیرا برخی رخدادها از وضعیت عددی خود معنا می‌گیرند. این تصمیم با اصل تفکیک قالب .و پارامتر در مطالعات تجزیه سازگار است [۲۳، ۲۴]

کیفیت نرمال‌سازی بر دو متغیر اثر دارد: اطلاعاتی که برای تشخیص باقی می‌ماند و تعداد قالب‌های یکتایی که به مدل فرستاده می‌شود. ماسک بیش‌ازحد ممکن است واژه یا مقدار معنادار را حذف کند؛ ماسک کمتر نیز نیز اثر کیفیت تجزیه بر LogParser-LLM و LogLAA .پیام‌های هم‌معنا را به کلیدهای متعدد تبدیل می‌کند .تحلیل پایین‌دستی و ضرورت کنترل دانه‌بندی قالب را گزارش می‌کنند [۱۷، ۲۳]

۲-۲-۵ پارادایم‌های یادگیری در تشخیص ناهنجاری لاگ

.روش نظارت‌شده از نمونه‌های عادی و ناهنجار برچسب‌خورده برای یادگیری مرز تصمیم استفاده می‌کند نمونه‌ای از تنظیم نظارت‌شده است [۱۸]. روش نیمه‌نظارتی از مقدار محدودی داده BERT-Log و RoBERTa این رویکرد را با SemiRALD برچسب‌خورده و داده بدون برچسب بهره می‌گیرد؛ ترکیب می‌کند [۱۶]. روش خودنظارتی از یک وظیفه کمکی مانند پیش‌بینی توکن یا رخداد BiLSTM در این دسته ADALog و LogFiT، LogBERT ماسک‌شده برای یادگیری الگوی عادی استفاده می‌کند؛ .قرار می‌گیرند [۸، ۱۹، ۲۲]

.تنظیم دقیق کامل، تنظیم پارامترکارا و یادگیری درون‌متنی سه شیوه متفاوت سازگارکردن مدل زبانی با دامنه‌اند Lim و پژوهش LLM-LADE مراحل آموزشی اختصاصی دارند؛ GPT مبتنی بر LogGPT و LogLLM LogGPT و LogPrompt استفاده می‌کنند؛ و PEFT یا سایر روش‌های ReFT، LoRA و همکاران از .بدون تغییر وزن، دستور و نمونه را در پرامپت قرار می‌دهند [۱، ۲، ۴–۷، ۹] ChatGPT مبتنی بر

۲-۲-۶ مدل‌های ترتیبی، Transformer و مدل‌های زبانی

رخداد بعدی LSTM با DeepLog .مدل ترتیبی احتمال رخداد بعدی یا سازگاری کل توالی را برآورد می‌کند با سازوکار توجه امکان Transformer .را از تاریخچه پیش‌بینی و انحراف را علامت‌گذاری می‌کند [۲۱]

با زمینه دوطرفه و دو هدف LogBERT .استفاده از زمینه دورتر و پردازش موازی‌تر را فراهم می‌کند و طبقه‌بند متصل، این معماری را برای لاگ به‌کار گرفته‌اند BERT با بازنمایی BERT-Log خودنظارتی و .[۱۸، ۱۹]

خودبازگشتی بنا می‌شوند و پاسخ را براساس دستور Transformer های دستورمحور عمدتاً بر معماریLLM ،فرایند توسعه یک مدل متن‌باز ۱۲ میلیارد پارامتری را در سه مرحله پیش‌آموزی YuLan .متنی تولید می‌کنند تنظیم دستوری و هم‌ترازی انسانی تشریح می‌کند و قابلیت یادگیری درون‌متنی را یکی از پیامدهای پس‌آموزی می‌داند [۲۶]. این مقاله آشکارساز لاگ نیست، اما برای توضیح منشأ توانایی مدل‌های دستورمحور در پیروی .از پرامپت استفاده می‌شود

۲-۲-۷ نقش‌های LLM در خط لوله تحلیل لاگ

در یک خط لوله لاگ، LLM می‌تواند parser، تبیین‌گر یا هماهنگ‌کننده باشد، تولیدکننده داده، طبقه‌بند. LogPrompt، LogGPT. در مرحله استخراج قالب قرار دارند [۲۳، ۲۴] LogPPT و LogParser-LLM از LLM-LADE. تصمیم تشخیص را مستقیماً از مدل دریافت می‌کنند [۱–۳، ۶] Hybrid Prompting و توضیح رخداد را پس از تشخیص تولید می‌کند LogLAA برای افزایش داده و توضیح استفاده می‌کند و LLM [۷، ۱۷].

در سامانه‌های چندمدلی، LLM لزوماً برای همه نمونه‌ها اجرا نمی‌شود. AdaptiveLog عدم‌قطعیت مدل نتیجه چند مدل یادگیری ماشین را با FlexLog. کوچک را معیار ارجاع به مدل بزرگ قرار می‌دهد [۱۳] را برای ساخت LLM نیز تفسیر رخداد با LogSynergy. بازیابی و کش ترکیب می‌کند [۱۰]، Mistral بازنمایی قابل انتقال میان سامانه‌ها به‌کار می‌برد [۱۵].

۲-۲-۸ Qwen۲٫۵:۷B-Instruct

خانواده‌ای از مدل‌های باز با نسخه‌های پایه و دستورمحور در اندازه‌های ۰٫۵ تا ۷۲ میلیارد پارامتر Qwen۲٫۵ پیش‌آموزی بر حدود ۱۸ تریلیون توکن و پس‌آموزی با بیش از یک میلیون Qwen۲٫۵ است. گزارش فنی حدود ۶٫۵ میلیارد پارامتر Bنمونه تنظیم دستوری و مراحل تقویت یادگیری را شرح می‌دهد. نسخه ۷ برای پیروی از دستور و تولید پاسخ ساخت‌یافته آماده شده است [۲۸] Instruct غیرتعبیه‌ای دارد و نسخه.

در پژوهش حاضر بدون تنظیم دقیق و با وزن‌های ثابت استفاده می‌شود. ثابت Qwen۲٫۵:۷B-Instruct نگه‌داشتن نسخه مدل، کوانتیزه‌سازی، دما و گزینه‌های تولید در دو روش ضروری است؛ زیرا در غیر این صورت اختلاف خروجی می‌تواند به تغییر مدل یا تنظیمات تولید نسبت داده شود، نه محل اعمال قاعده. مقاله برای تصمیم استفاده GPT-٤o-mini برای تعبیه و Qwen۳-Embedding-۸B از Hybrid Prompting کرده است و از نظر نقش مدل با این پژوهش تفاوت دارد [۳].

۲-۲-۹ مهندسی پرامپت، یادگیری درون‌متنی و خروجی

پرامپت شامل تعریف وظیفه، زمینه دامنه، معنای برچسب‌ها، نمونه‌های احتمالی و قرارداد خروجی است. پنج پرسش درباره نوع پرامپت، اندازه پنجره، دانش انسانی، مقایسه با ChatGPT مبتنی بر LogGPT اثر پرامپت ساده، پرامپت دارای دانش دامنه LogPrompt. روش‌های عمیق و تبیین‌پذیری مطرح می‌کند [۲] و زنجیره استدلال را بر تجزیه و تشخیص بررسی می‌کند [۱، ۶].

Few-shot نمونه‌های برچسب‌خورده را در خود پرامپت قرار می‌دهد و به تغییر وزن مدل نیاز ندارد. Hybrid Prompting نمونه‌های عادی مرتبط را از پایگاه برداری بازیابی می‌کند [۱۲] و RAGLog نیز حساسیت OpsEval. نزدیک‌ترین نمونه عادی را همراه با شواهد رخداد و توالی در متن قرار می‌دهد [۳] in-context learning و chain-of-thought، self-consistency وظایف عملیات فناوری اطلاعات به را در یک معیار مستقل بررسی می‌کند [۲۵].

قرارداد خروجی پژوهش حاضر فقط {"label":"۱"} یا {"label":"۰"}، را معتبر می‌داند. متن اضافی امکان تعیین Ollama ثبت و کش نمی‌شود. مستندات INVALID ناقص یا برچسب دیگر با وضعیت JSON و استفاده از دمای صفر برای کاهش نوسان را بیان می‌کنند [۲۹]. اعتبارسنجی در برنامه JSON قالب یا طرح مستقل از پذیرش ظاهری پاسخ انجام می‌شود.

بازیابی، کش و کنترل هزینه استنتاج ۲-۲-۱۰

بازیابی و کش دو سازوکار متفاوت‌اند. بازیابی، نمونه یا دانش مرتبط را برای ساخت زمینه پرامپت پیدا می‌کند؛ از این الگو استفاده می‌کنند [۱۲، ۱۳]. کش، پاسخ معتبر یک ورودی پایدار AdaptiveLog و RAGLog و مدل‌های RAG کش را در کنار FlexLog را نگه می‌دارد تا در مشاهده تکراری مدل دوباره اجرا نشود؛ یادگیری ماشین به‌کار می‌گیرد [۱۰].

در پژوهش حاضر کش در سطح قالب نرمال‌شده است. کلید شامل نسخه پرامپت، نام و نسخه مدل، قالب نرمال‌شده و در صورت فعال بودن، فیلدهای مؤلفه، شدت و دسته است. بنابراین تصمیم یک مدل یا پرامپت قدیمی به اجرای جدید منتقل نمی‌شود. نرخ برخورد، عدم برخورد، کلیدهای یکتا و منبع تصمیم برای تحلیل جداگانه اثر کش ثبت می‌شوند.

کاهش هزینه فراخوانی تا ۷۳ LLM با استفاده از مدل کوچک و ارجاع موارد نامطمئن به AdaptiveLog نیز با درخت پیشوند و تجمیع پیام‌های مشابه، در LogParser-LLM درصد را گزارش می‌کند [۱۳]. انجام داده LLM با میانگین ۳٫۶ میلیون پیام در هر مجموعه به‌طور متوسط فقط ۲۷۲٫۵ فراخوانی LogPub است [۲۳]. این اعداد نشان‌دهنده اهمیت گزارش فراخوانی و زمان در کنار معیارهای طبقه‌بندی هستند.

استنتاج محلی با Ollama ۲-۲-۱۱

Ollama پیش‌فرض API اجرای محلی مدل و دسترسی برنامه‌نویسی از طریق localhost:۱۱٤۳٤/api را می‌تواند زمان کل، زمان بارگذاری، شمار توکن پرامپت و خروجی و مدت ارزیابی API فراهم می‌کند. پاسخ هر بخش را گزارش کند [۲۹]. ثبت این داده‌ها امکان تفکیک زمان بارگذاری مدل از زمان پردازش پرامپت و تولید پاسخ را فراهم می‌سازد.

استقرار محلی از ارسال لاگ به سرویس خارجی جلوگیری و کنترل نسخه مدل را ممکن می‌کند. مطالعه De la Cruz Cabello و همکاران نیز LogBERT را برای محیط دارای محدودیت کاربرد واقعی محرمانگی به‌صورت درون‌سازمانی مستقر کرده است [۲۰]. تفاوت آن مطالعه با پژوهش حاضر در نوع مدل در این پژوهش بدون تنظیم Qwen عادی آموزش می‌بیند، درحالی‌که syslog و آموزش است: مدل آن‌ها روی اختصاصی استفاده می‌شود.

در پژوهش حاضر Prompt-only و Hybrid دو معماری ۲-۲-۱۲

پس از استخراج فیلدها، حذف برچسب و نرمال‌سازی، کش بررسی می‌شود. اگر کلید Hybrid، در روش برای guard موجود نباشد، مجموعه محدود قواعد قطعی با اطمینان بالا اعمال می‌شود. فقط نمونه‌هایی که ارسال می‌شوند. پاسخ معتبر در کش قرار می‌گیرد و منبع تصمیم به‌صورت Qwen آن‌ها تصمیم ندارد به llm یا guard، cache ثبت می‌شود.

قطعی غیرفعال است و دانش دامنه در پرامپت قرار می‌گیرد. نرمال‌سازی guard، Prompt-only در روش و ترتیب داده ثابت می‌مانند. در نتیجه متغیر اصلی مقایسه محل اعمال دانش JSON کش، نسخه مدل، قرارداد تفاوت دارد؛ مقاله [۳] شواهد سطح Hybrid Prompting در این پژوهش با Hybrid قاعده است. اصطلاح رخداد، توالی و نمونه بازیابی‌شده را در یک پرامپت ترکیب می‌کند و تصمیم آن در سطح توالی است.

معیارهای ارزیابی ۲-۲-۱۳

خط عادی با برچسب FP ،خط عادی درست TN ،ناهنجاری درست شناسایی‌شده TP ،در طبقه‌بندی دودویی سهم Precision ،نسبت کل تصمیم‌های درست Accuracy .ناهنجاری ازدست‌رفته است FN ناهنجار و و Precision میانگین هارمونیک F۱ سهم ناهنجاری‌های کشف‌شده و Recall ،پیش‌بینی‌های ناهنجار درست است [۲، ۳، ۱۸] Recall.

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

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

گزارش شود. نرخ F۱ و Recall ،Precision باید همراه Accuracy ،BGL به‌علت نامتوازن بودن منبع تصمیم و نرخ برخورد کش نیز برای این پژوهش ،LLM زمان پاسخ، تعداد فراخوانی ،INVALID با بهبود خود طبقه‌بند guard ضروری‌اند. این شاخص‌ها مانع آن می‌شوند که کاهش فراخوانی ناشی از کش یا اشتباه شود.

انواع الگوهای ناهنجاری در لاگ ۲-۲-۱۴

ناهنجاری رخدادی زمانی رخ می‌دهد که خود یک پیام یا قالب، مستقل از پیرامون آن، نشانه وضعیت غیرعادی باشد. وجود عبارت خطا، توقف مؤلفه، خرابی سخت‌افزار یا کد بازگشتی غیرمنتظره می‌تواند از این نوع باشد. بر همین سطح تمرکز می‌کند و احتمال بازسازی توکن‌های یک پیام را به امتیاز خط تبدیل می‌نماید ADALog طبقه‌بندی خطبه‌خط پژوهش حاضر نیز ابتدا برای این سطح از ناهنجاری تعریف شده است .[۲۲].

،ناهنجاری ترتیبی از کنار هم قرارگرفتن رخدادهایی ایجاد می‌شود که ممکن است هرکدام به‌تنهایی عادی باشند LogBERT با پیش‌بینی رخداد بعدی و DeepLog .اما ترتیب یا فاصله آن‌ها با الگوی عادی سازگار نباشد نیز اطلاعات Hybrid Prompting .با یادگیری زمینه دوطرفه، این نوع انحراف را هدف می‌گیرند [۱۹، ۲۱] توالی را همراه رخدادهای دیده‌نشده در پرامپت قرار می‌دهد [۳].

ناهنجاری کمّی به تغییر غیرعادی تعداد تکرار یک یا چند قالب در یک پنجره مربوط است. روش‌های مبتنی بر بردار شمارش و برخی مدل‌های ترکیبی، فراوانی رخداد را در کنار ترتیب یا معنا استفاده می‌کنند. سه نوع بازنمایی شمارشی، ترتیبی و معنایی را به‌صورت همزمان وارد آشکارساز می‌کند [۱۷] LogLAA. نیست و این محدودیت باید هنگام Qwen در تصمیم خطمحور این پایان‌نامه، شمارش پنجره ورودی مستقیم تحلیل خطا در نظر گرفته شود.

،ناهنجاری معنایی زمانی مطرح می‌شود که واژه‌ها یا مقادیر یک پیام از نظر نحوی مشابه نمونه عادی باشند استفاده LogLLM و RAGLog ،LogPrompt اما مفهوم آن‌ها وضعیت نامطلوبی را بیان کند. مطالعات از بازنمایی زبانی را برای بازیابی یا درک چنین تفاوتی بررسی کرده‌اند [۱، ۴، ۶، ۱۲]. حفظ صفر و غیرصفر در نرمال‌سازی پایان‌نامه برای جلوگیری از حذف بخشی از همین معنای عملیاتی انجام می‌شود.

لاگ ناپایدار به تغییر قالب، واژگان یا توزیع رخدادها پس از به‌روزرسانی نرم‌افزار یا تغییر پیکربندی اشاره این مسئله را با انتقال نسخه، مدل زبانی FlexLog و LogLLM ،LOGEVOL روی GPT دارد. مطالعات در پایان‌نامه به‌عنوان یک مجموعه ثابت ارزیابی BGL .یا ترکیب مدل‌ها بررسی کرده‌اند [۴، ۱۰، ۱۱] می‌شود و آزمون انتقال میان نسخه‌های نرم‌افزار خارج از دامنه است.

چالش‌های روش‌شناختی و اجرایی ۲-۲-۱۵

سهم پیام‌های ناهنجار حدود ۷٫۳۴ درصد BGL نامتوازن بودن کلاس‌ها یکی از چالش‌های اصلی است. در Accuracy است [۲، ۱۹]. در چنین شرایطی، مدلی که بیشتر نمونه‌ها را عادی پیش‌بینی کند ممکن است Hybrid و BERT-Log ناهنجاری پایین بماند. به همین دلیل، مطالعات Recall ظاهراً بالایی داشته باشد، اما یا به‌جای اتکای صرف به آن Accuracy را در کنار F۱ و Recall، Precision معیارهای Prompting گزارش می‌کنند [۳، ۱۸].

چالش دیگری است. اگر پارامتر به‌عنوان کلمه ثابت یا کلمه ثابت به‌عنوان پارامتر parser انتقال خطای برای سنجش LogParser-LLM و LogPPT. تشخیص داده شود، قالب نادرست به مدل پایین‌دستی می‌رسد و معیارهای دانه‌بندی را به‌کار می‌گیرند [۲۳، ۲۴] Parsing Accuracy، Group Accuracy. این مسئله در پایان‌نامه، قواعد نرمال‌سازی باید جدا از کیفیت طبقه‌بندی ممیزی شوند.

نشت داده می‌تواند از ورود برچسب به متن، هم‌پوشانی نمونه‌های آموزش و آزمون یا تکرار قالب یکسان در داده‌های یکتای عادی را میان آموزش، تعیین آستانه و آزمون جدا می‌کند ADALog. دو بخش ایجاد شود قبل از ساخت پرامپت حذف می‌شود و کش فقط در محدوده همان اجرای BGL در این پایان‌نامه برچسب [۲۲]. تعریف‌شده و با کلید نسخه‌بندی‌شده به‌کار می‌رود.

پاسخ مولد ممکن است از قرارداد خروجی پیروی نکند یا میان اجراها تغییر کند. مطالعات پرامپتی نشان می‌دهند متن دستور و درخواست تبیین بر نتیجه اثر دارد [۱، ۲، ۶، ۲۵]. استفاده از دمای صفر، طرح JSON، در پایان‌نامه برای قابل‌اندازه‌گیری کردن این رفتار انجام می‌شود INVALID اعتبارسنجی برچسب و ثبت [۲۹].

از مدل می‌خواهد دلیل تصمیم را تولید کند LogPrompt. تبیین‌پذیری در منابع به چند معنا به‌کار رفته است پس از آشکارسازی LogLAA تشخیص و توضیح را هم‌زمان آموزش می‌دهد [۷]؛ و LLM-LADE [۱۰، ۶]؛ برای توضیح استفاده می‌کند [۱۷]. خروجی پایان‌نامه عمداً دودویی است، بنابراین تبیین در سطح LLM از ثبت منبع تصمیم، قالب نرمال‌شده و قاعده فعال‌شده دنبال می‌شود، نه تولید شرح آزاد.

محرمانگی، بازتولیدپذیری و هزینه نیز به هم مرتبط‌اند. سرویس تجاری امکان تغییر نسخه مدل و محدودیت نرخ دارد، درحالی‌که استقرار محلی کنترل بیشتری بر مدل و داده فراهم می‌کند [۲، ۲۰]. بااین‌حال، استقرار و پارامترهای تولید نیاز دارد تا نتیجه قابل بازتولید Ollama محلی به ثبت سخت‌افزار، کوانتیزه‌سازی، نسخه باشد [۲۸، ۲۹].

پیشینه تحقیق ۲-۳

دامنه و شیوه مرور مطالعات ۲-۳-۱

پیشینه این فصل شامل تمام ۲۷ فایل مقاله موجود در پوشه پژوهش است. مقالات براساس نقش اصلی روش در شش دسته قرار گرفته‌اند: روش‌های ترتیبی و پیش‌آموزش‌دیده؛ آموزش و تنظیم LLM؛ بازیابی و پرامپت؛ کنترل هزینه؛ تجزیه لاگ؛ و مرور، معیار، مدل پایه یا استقرار واقعی. این دسته‌بندی با محورهای مطرح‌شده برای تشخیص ناهنجاری [۲۷] سازگار است LLM در مرور نظام‌مند لاگ [۱۴] و مرور.

هستند. مطابق درخواست پوشش LogPrompt دو فایل [۱] و [۶] نسخه‌های هم‌پوشان یک خط پژوهشی با نام همه مقالات، هر دو به‌صورت مستقل مرور و در جدول درج شده‌اند؛ بااین‌حال هنگام جمع‌بندی شواهد به‌عنوان آشکارساز مستقیم لاگ [۲۶] YuLan و [۲۵] OpsEval دو روش متفاوت شمرده نمی‌شوند. همچنین مقالات در عملیات و معماری مدل متن‌باز استفاده می‌شوند LLM نیستند، اما به‌ترتیب برای مبانی ارزیابی.

روش‌های ترتیبی و مدل‌های زبانی پیش‌آموزش‌دیده ۲-۳-۲

مقاله ۲۱ - DeepLog: Du و شبکه LSTM را مانند یک توالی زبان در نظر گرفتند و برای پیش‌بینی رخداد بعدی از روی تاریخچه رخدادهای عادی آموزش دادند. اگر رخداد واقعی در مجموعه رخدادهای محتمل مدل قرار نگیرد، توالی به‌عنوان مشکوک گزارش می‌شود. مقاله علاوه بر تشخیص، به‌روزرسانی افزایشی مدل و ساخت جریان‌های اجرایی برای کمک به عیب‌یابی را توضیح می‌دهد. [۲۱]

ارزیابی DeepLog روی داده HDFS و OpenStack نشان داد مدل ترتیبی می‌تواند وابستگی رخدادها را تشکیل کلید لاگ و توالی معتبر نیاز دارد، parser بهتر از روش‌های آماری مبنا ثبت کند [۲۱]. این روش به هدف آن تصمیم مستقل برای هر خط نیست. در پژوهش حاضر، ترتیب رخداد به‌عنوان ورودی مستقیم مدل و خط پایه مفهومی خانواده ترتیبی محسوب DeepLog است؛ بنابراین BGL استفاده نمی‌شود و واحد تصمیم خط پروتکل، نه روش هم‌پروتکل می‌شود.

مقاله ۱۹ - LogBERT: Guo، Yuan و Wu چارچوب خودنظارتی LogBERT را با دو وظیفه طراحی کردند: پیش‌بینی کلید لاگ ماسک‌شده و کمینه‌سازی حجم ابرکره بازنمایی توالی‌های عادی. زمینه دوطرفه باعث می‌شود بازنمایی هر رخداد از قبل و بعد توالی استفاده کند و وظیفه ابرکره نیز توالی‌های عادی BERT را در فضای بازنمایی به یکدیگر نزدیک می‌سازد. [۱۹]

LogBERT روی HDFS، BGL و Thunderbird ارزیابی شده است. برای BGL پنجره پنج‌دقیقه‌ای نیز BGL ساخته شده و مقاله از حدود پنج هزار توالی عادی برای آموزش استفاده می‌کند؛ میانگین طول توالی رخداد گزارش شده است [۱۹]. وابستگی به آموزش روی داده عادی و ساخت پنجره تفاوت اصلی آن با ۵۶۲ Qwen بدون تنظیم و تصمیم خطبه‌خط در پژوهش حاضر است.

مقاله ۱۸ - BERT-Log: Chen و Liao بازنمایی و مدل را همراه یک BERT پیام‌های تجزیه‌شده را با شناسه گره BGL توالی‌ها با شناسه بلوک و در HDFS لایه تمام‌متصل برای طبقه‌بندی تنظیم کردند. در اندازه پنجره و گام ساخته می‌شوند. مدل از اطلاعات متن و زمینه توالی برای تفکیک رفتار عادی و ناهنجار استفاده می‌کند. [۱۸]

مقاله F۱ را در پروتکل خود گزارش می‌کند BGL و ۹۹٫۴ درصد برای HDFS برابر ۹۹٫۳ درصد برای وابسته است و مستقیماً با تشخیص خطمحور BGL و پنجره parser، این نتیجه به آموزش نظارت‌شده [۱۸]. برای پژوهش حاضر نشان‌دادن ظرفیت بازنمایی BERT-Log قابل مقایسه نیست. اهمیت fine-tuning بدون ضرورت گزارش دقیق پروتکل پنجره است BGL زبانی در.

مقاله ۸ - LogFiT: Almodovar و همکاران LogFiT را بر پایه مدل زبانی تنظیم‌شده با وظیفه مدل‌سازی توکن ماسک‌شده ارائه کردند. مدل فقط روی لاگ‌های عادی آموزش می‌بیند و در آزمون، توکن‌های پیام یا پیش‌بینی باشد، شواهد ناهنجاری افزایش top-k توالی را پیش‌بینی می‌کند. اگر توکن واقعی خارج از مجموعه می‌یابد. [۸]

بررسی شده و هدف آن کاهش نیاز به نمونه ناهنجار Thunderbird و BGL، HDFS روی LogFiT برچسب‌خورده است [۸]. با وجود استفاده از مدل زبانی، روش مولد دستورمحور نیست و به تنظیم روی لاگ را ثابت نگه می‌دارد و پاسخ دودویی را از Qwen مقصد نیاز دارد. در مقابل، پژوهش حاضر وزن‌های پرامپت دریافت می‌کند.

مقاله ۲۲ - ADALog: Pospieszny و همکاران چارچوبی تطبیقی و بدون نظارت برای تحلیل پیام‌های خام روی پیام‌های عادی تنظیم می‌شود؛ احتمال MLM با وظیفه DistilBERT. و مستقل از توالی پیشنهاد کردند بازسازی توکن‌ها به امتیاز سطح پیام تبدیل و آستانه با صدک خطاهای داده عادی تعیین می‌گردد. روش به parser یا پیوستگی توالی وابسته نیست. [۲۲]

ارزیابی و اثر نرخ ماسک، پاک‌سازی و موقعیت توکن‌ها Spirit و Thunderbird، BGL روی ADALog را با مطالعات حذف مؤلفه بررسی کرده است [۲۲]. نزدیک‌بودن واحد تصمیم آن به سطح خط برای این و امتیاز fine-tuning DistilBERT به ADALog: پایان‌نامه مهم است، اما سازوکار تصمیم متفاوت است و کش قالب استفاده می‌کند JSON، دستورمحور Qwen بازسازی متکی است، درحالی‌که پژوهش حاضر از.

۲-۳-۳ آموزش، تنظیم پارامترکارا و انتقال LLM

مقاله ۵ - Log Anomaly Detection via GPT: Han، Yuan و Trabelsi مدلی با نام LogGPT را برای پیش‌بینی رخداد بعدی آموزش می‌دهد و سپس با یادگیری تقویتی، هدف تولید را GPT ارائه کردند که به تصمیم تشخیص ناهنجاری نزدیک می‌کند. طراحی روش از توان مدل خودبازگشتی برای یادگیری توالی رخداد و از بازخورد پاداش برای اصلاح مرز تصمیم استفاده می‌کند. [۵]

ارزیابی مقاله روی HDFS، BGL و Thunderbird انجام شده است [۵]. اشتراک نام آن با مقاله [۲] نباید به یکسان‌انگاری روش‌ها منجر شود؛ مقاله [۵] روش آموزش و یادگیری تقویتی است، ولی مقاله [۲] را با پرامپت بررسی می‌کند. پژوهش حاضر از نظر نبود آموزش به مقاله [۲] نزدیک‌تر است ChatGPT.

مقاله ۴ - LogLLM: Guan و همکاران LogLLM طراحی کردند با سه جزء اصلی: BERT برای به‌عنوان طبقه‌بند. آماده‌سازی LLaMA برای نگاشت بازنمایی و projector استخراج بردارهای معنایی، یک و فضای BERT با قواعد منظم انجام می‌شود و فرایند آموزشی چندمرحله‌ای برای هم‌ترازکردن بازنمایی به‌کار می‌رود. [۴] LLaMA

و در سناریوهای لاگ ناپایدار ارزیابی شده است Thunderbird و Liberty، BGL، HDFS روش روی می‌تواند اطلاعات معنایی و تصمیم را LLM کوچک‌تر با encoder نشان می‌دهد ترکیب LogLLM. [٤] است. پژوهش حاضر بدون آموزش، مستقیماً قالب LLaMA و projector جدا کند، اما نیازمند آموزش می‌دهد Qwen نرمال‌شده را به.

مقاله ۷ - LLM-LADE: Zhang و همکاران چارچوب سه‌مرحله‌ای شامل افزایش داده، تنظیم پارامترکارا LoRA با Llama۳-۸B برای تولید نمونه و few-shot با پرامپت GPT-٤. و بهینه‌سازی برخط ارائه کردند برای آشکارساز استفاده می‌شود. مدل علاوه بر برچسب ناهنجاری، توضیح متنی تولید می‌کند و دانش حاصل از نمونه‌های جدید در یک پایگاه دانش افزایشی نگه‌داری می‌شود. [۷]

تشخیص و توضیح را در یک فرایند مشترک قرار می‌دهد و سازوکار برخط را برای LLM-LADE سازگاری با تغییرات پیشنهاد می‌کند [۷]. در پژوهش حاضر خروجی برای جلوگیری از ابهام فقط برچسب وجود ندارد. این تفاوت دامنه LoRA است و توضیح تولید نمی‌شود؛ همچنین بهینه‌سازی برخط یا JSON سنجش را به کیفیت دودویی و هزینه استنتاج محدود می‌کند.

را روی ReFT و LoRA عملکرد Pang و Zhu ،Lim **:مقاله ۹ - تنظیم پارامترکارای مدل‌های زبانی**
برای تشخیص ناهنجاری لاگ بررسی کردند. پژوهش فقط یک مدل Llama-۳ و GPT-۲، RoBERTa پیشنهادی ارائه نمی‌کند، بلکه پایداری اجرا، کارایی نمونه، مقاومت در برابر لاگ ناپایدار و تعمیم میان مجموعه‌ها را در چهار مجموعه HDFS، BGL، Spirit و Thunderbird [۹] مقایسه می‌کند.

بر پایداری و انتقال اثر دارد [۹]. این مطالعه معیار PEFT نتایج مقاله نشان می‌دهد نوع مدل پایه و روش در پژوهش حاضر Qwen .و «استفاده بدون تنظیم» فراهم می‌کند «LLM مناسبی برای تفکیک «استفاده از fine-tuning تفسیر شود، نه prompt-only inference سازگار نمی‌شود و بنابراین نتیجه آن باید در طبقه PEFT با tuned detection.

و همکاران Hadadi **Anomaly Detection on Unstable Logs with GPT Models: مقاله ۱۱ -**
روی لاگ مقصد تنظیم و GPT-۳ .بررسی کردند GPT مسئله تغییر قالب‌ها میان نسخه‌های نرم‌افزار را با تشکیل می‌شود تا LOGEVOL-Hadoop با مهندسی پرامپت استفاده شد. داده آزمایش از دو نسخه GPT-٤ اثر ناپایداری و انتقال میان نسخه‌ها سنجیده شود. [۱۱]

.نتیجه بهتری گزارش کرده است [۱۱] prompt engineering نسبت به fine-tuning ،در پروتکل مقاله
این یافته محدود به مدل‌ها و داده همان مطالعه است، اما نشان می‌دهد نبود تنظیم می‌تواند در تغییر توزیع هزینه عملکرد داشته باشد. پژوهش حاضر مسئله انتقال نسخه‌ای را آزمایش نمی‌کند و تمرکز آن بر دو محل اعمال .ثابت است BGL قاعده در

و همکاران برای انتقال آشکارساز به سامانه نرم‌افزاری جدید، تفسیر رخداد Sui **LogSynergy: مقاله ۱۵ -**
تفاوت‌های نحوی رخدادها را LLM .و استخراج ویژگی یکپارچه میان سامانه‌ها را ترکیب کردند LLM با به توصیف‌های هم‌معنا نزدیک می‌کند و سپس ویژگی‌های استانداردشده به طبقه‌بند انتقالی داده می‌شوند. [۱۵]

بیش از ۸۹ درصد روی داده واقعی و بیش از ۸۳ درصد F۱ با پنج هزار توالی برچسب‌خورده LogSynergy روی داده‌های عمومی گزارش می‌کند [۱۵]. مسئله اصلی آن انتقال بین سامانه‌هاست، نه تصمیم بدون آموزش برای کاهش LLM در یک مجموعه. ارتباط آن با پژوهش حاضر به نقش نرمال‌سازی معنایی و استفاده از .ناهمگونی ورودی مربوط می‌شود

.را ارائه کردند SemiRALD و همکاران چارچوب نیمه‌نظارتی Sun **SemiRALD: مقاله ۱۶ -**
همراه با RoBERTa و یادگیری درون‌متنی برای تجزیه لاگ استفاده می‌شود و سپس بازنمایی ChatGPT توجه‌محور، الگوی توالی را برای طبقه‌بندی می‌آموزد. طراحی روش، مرحله تجزیه معنایی و BiLSTM طبقه‌بند آموزش‌پذیر را از یکدیگر جدا می‌کند. [۱۶]

برابر ۷٫۳ و ۸٫۲ درصد را در مقایسه با F۱ بررسی شده و مقاله میانگین بهبود BGL و HDFS روش روی .دو گروه خط پایه گزارش می‌کند؛ بهبود در نسبت ۰٫۱ درصد داده برچسب‌خورده برجسته‌تر است [۱۶]
پژوهش حاضر guard+LLM در نام این مقاله به ترکیب مدل زبانی و مدل ترتیبی اشاره دارد و با Hybrid .متفاوت است

را برای تجزیه، تشخیص LogLAA و همکاران چارچوب یکپارچه و تطبیقی Gao **LogLAA: مقاله ۱۷ -**
از طول، فراوانی کلمات، جایگزینی متغیر و درخت پیشوند استفاده می‌کند؛ parser .و توضیح طراحی کردند توضیح نتیجه LLM دوتوجهی داده می‌شوند؛ و CNN-LSTM بازنمایی‌های شمارشی، ترتیبی و معنایی به را تولید می‌کند. [۱۷]

تشخیص نسبت به میانگین Precision مقاله افزایش ۰٫۸ درصدی کیفیت تجزیه، بهبود حدود ۶ درصدی سامانه‌ای چندمرحله‌ای و LogLAA .روش‌های مقایسه و امتیاز ۰٫۷ برای توضیح را گزارش می‌کند [۱۷] آموزش‌پذیر است؛ پژوهش حاضر بخش توضیح و مدل ترتیبی را حذف و بر کنترل تصمیم، کش و استنتاج ثابت تمرکز می‌کند LLM محلی یک.

مطالعات مبتنی بر پرامپت ۴-۳-۲

مقاله ۱ - نسخه اولیه LogPrompt: Liu و همکاران در مقاله Interpretable Online Log Analysis Using Large Language Models with Prompt Strategies، LLM استفاده از را برای تجزیه و تشخیص آنلاین لاگ بدون آموزش درون‌دامنه بررسی کردند. هسته روش، ارائه دستور مرحله‌بندی‌شده، دانش وظیفه و درخواست توجیه تصمیم است تا مدل از معنا و زمینه رخداد استفاده کند. [۱]

LogPrompt نسخه سال ۲۰۲۳ پایه خط پژوهشی را معرفی و چند وظیفه تحلیل لاگ را در یک چارچوب پرامپتی جمع می‌کند [۱]. مقاله [۶] نسخه توسعه‌یافته همین کار است؛ بنابراین نتایج آن‌ها مستقل از یکدیگر جمع نمی‌شوند. حضور جداگانه هر دو در پیشینه برای نشان‌دادن تحول نسخه و پوشش کامل فایل‌های موجود است.

مقاله ۲ - Exploring ChatGPT for Log-Based Anomaly Detection: Qi و همکاران بررسی کردند. پژوهش پنج محور را مطالعه Spirit و BGL را برای تشخیص ناهنجاری روی ChatGPT BGL می‌کند: تغییر پرامپت، اندازه پنجره، تزریق دانش انسانی، مقایسه با مدل‌های عمیق و تبیین‌پذیری. برای اطلاعات اصلی شامل ۴٬۷۴۷٬۹۶۳ پیام و ۳۴۸٬۴۶۰ ناهنجاری گزارش شده است. [۲]

مقاله دو هزار پیام متوالی از بخش آزمون را انتخاب و نسبت وجود ناهنجاری، API به دلیل محدودیت نرخ BGL را به‌صورت دستی کنترل کرده است [۲]. این محدودیت برای تفسیر نتیجه مهم است، زیرا آزمایش کل نرخ F۱، نیست. پژوهش حاضر با مدل محلی و کش قالب برای اجرای گسترده‌تر طراحی می‌شود و علاوه بر زمان را ثبت می‌کند INVALID.

مقاله ۶ - نسخه توسعه‌یافته LogPrompt: مقاله Prompt Engineering Towards Zero-Shot and Interpretable Log Analysis نسخه کامل‌تر LogPrompt است و راهبردهای پرامپت را روی ۹ مجموعه داده و چند وظیفه تجزیه و تشخیص ارزیابی می‌کند. اجزای پرامپت شامل تعریف وظیفه، دانش ضمنی یا صریح و زنجیره استدلال برای هدایت مدل به تصمیم و توضیح است. [۶]

مقاله بهبود تا ۵۵٫۹ درصد نسبت به روش‌های آموزش‌دیده در برخی تنظیم‌ها و امتیاز ۴٫۴۲ از ۵ برای نشان می‌دهد افزودن BGL تفسیرپذیری در ارزیابی متخصصان را گزارش می‌کند [۶]. تحلیل حذف مؤلفه در سطح نشست و قالب را تغییر دهد. پژوهش حاضر برای کاهش F۱ استدلال یا درخواست توجیه می‌تواند خروجی آزاد، استدلال را به JSON دودویی محدود می‌کند.

مقاله ۳ - Hybrid Prompting: Park، Choi و Lee سه نوع شواهد را در یک پرامپت ترکیب کردند: وجود رخداد دیده‌نشده، آمار و شباهت توالی و نزدیک‌ترین نمونه عادی. برای تعبیه از Qwen۳-Embedding-۸B و برای طبقه‌بندی از GPT-٤o-mini استفاده شده است. BGL، HDFS و Thunderbird [۳] با پروتکل‌های توالی ارزیابی می‌شوند.

F۱ برابر ۹۹٫۶۰ و Recall ،برابر ۶۳٫۶۵ Precision ،با پنجره یک‌دقیقه‌ای و نمونه متوازن BGL روش در روی سه مجموعه ۹٫۹۳ واحد درصد است [۳] F۱ برابر ۷۷٫۸۱ درصد گزارش می‌کند و میانگین بهبود است LLM و guard پایان‌نامه یک آبشار Hybrid ترکیب اطلاعات در متن است؛ Hybrid Prompting و سطح تصمیم آن خط است.

بازیابی، همکاری مدل‌ها و کنترل هزینه ۵-۳-۲

- **مقاله ۱۲ RAGLog:** Pan، Wong و Yuan پایگاه برداری را از نمونه‌های عادی تشکیل دادند و برای K-means ارائه کردند. نمونه‌های پایگاه با خوشه‌بندی LLM هر ورودی، نمونه مرتبط را بازیابی و همراه پیام به انتخاب می‌شوند و مدل بدون دریافت نمونه ناهنجار، براساس شباهت و Thunderbird و BGL از تحلیل معنایی تصمیم می‌گیرد. [۱۲]

RAGLog در تنظیم گزارش‌شده Precision ۰٫۹۱ ،برابر Recall و ۰٫۸۸ برابر F۱ را در ۰٫۸۹ برابر ارائه می‌کند [۱۲]. مقاله مصرف منابع و تأخیر تحلیل خطبه‌خط را محدودیت می‌داند LogPrompt مقایسه با نمونه را برای زمینه انتخاب می‌کند، درحالی‌که کش پایان‌نامه تصمیم معتبر قالب RAGLog پایگاه برداری را بازاستفاده می‌کند.

- **مقاله ۱۳ AdaptiveLog:** Ma و همکاران همکاری مدل زبانی کوچک و بزرگ را برای چند وظیفه تحلیل لاگ پیشنهاد کردند. مدل کوچک ابتدا نمونه را پردازش می‌کند؛ معیار عدم‌قطعیت تعیین می‌کند کدام ارجاع شود؛ و موارد دشوار مشابه از مخزن خطا بازیابی و در پرامپت قرار می‌گیرند LLM نمونه به نیز برای بهبود تدریجی مدل کوچک استفاده می‌شود. [۱۳] LLM بازخورد

AdaptiveLog تا ۷۳ درصد را گزارش می‌کند [۱۳]. این روش برای کنترل LLM کاهش هزینه فراخوانی پایان‌نامه به‌جای مدل Hybrid .هزینه به یک مدل کوچک آموزش‌پذیر و معیار عدم‌قطعیت وابسته است همان دروازه را حذف می‌کند؛ در Prompt-only کوچک، قواعد قطعی محدود را دروازه قرار می‌دهد و نتیجه مقایسه بدون آموزش مدل جانبی انجام می‌شود.

- **مقاله ۱۰ FlexLog:** Hadadi و همکاران FlexLog را برای تشخیص داده‌کارآمد روی لاگ‌های ناپایدار بازیابی نمونه و کش ترکیب ،Mistral و شبکه پیش‌خور با KNN ،طراحی کردند. خروجی درخت تصمیم برای نمونه‌های دشوار یا نیازمند معنا به‌کار LLM می‌شود. مدل‌های کلاسیک موارد آشنا را پوشش می‌دهند و می‌رود. [۱۰]

آزمایش روی ADFA-U، LOGEVOL-U، SynHDFS-U و SYNEVOL-U انجام شده است. مقاله با ۶۲٫۸۷ واحد درصد داده برچسب‌خورده کمتر، در برخی تنظیم‌ها تا ۱۳ F۱ حداقل ۱٫۲ واحد درصد بهبود از نظر FlexLog .واحد درصد بهبود و زمان استنتاج کمتر از یک ثانیه برای هر توالی گزارش می‌کند [۱۰] کش و آبشار تصمیم نزدیک است، اما داده، واحد تحلیل و ترکیب چند طبقه‌بند آن با پایان‌نامه تفاوت دارد.

مطالعات تجزیه و تولید قالب لاگ ۶-۳-۲

- **مقاله ۲۴ LogPPT:** Le و Zhang پرامپت‌تیونینگ برای تجزیه لاگ، few-shot را روی RoBERTa به‌کار گرفتند. الگوریتم نمونه‌گیری تصادفی تطبیقی مجموعه‌ای کوچک و متنوع از پیام‌ها را انتخاب می‌کند و مدل یاد می‌گیرد توکن‌های پارامتر را با یک برچسب مجازی تشخیص دهد. روش به عبارت منظم دستی وابسته نیست و تنظیمات یکسان را در دامنه‌های مختلف حفظ می‌کند. [۲۴]

Group روی ۱۶ مجموعه عمومی و با ۳۲ نمونه برچسب‌خورده ارزیابی شده و میانگین LogPPT Accuracy و Parsing Accuracy نیست ناهنجاری تشخیص هدف آن [۲۴]. گزارش می‌کند ۰٫۹ بیش از، ولی تعریف قالب و پارامتر مبنای نرمال‌سازی و کش قالب در پژوهش حاضر است.

مقاله ۲۳ - LogParser-LLM: Zhong و همکاران معناشناسی LLM را با یک درخت پیشوند برای تجزیه برخط ترکیب کردند. پیام‌ها ابتدا با ساختار موجود تطبیق داده می‌شوند و فقط الگوهای جدید یا مبهم به LLM برای کنترل دانه‌بندی parser می‌روند. مقاله یک معیار و امکان بازخورد انسانی نیز پیشنهاد می‌کند. [۲۳]

به‌طور متوسط ۲۷۲٫۵ LogParser-LLM، با میانگین ۳٫۶ میلیون لاگ در ۱۴ مجموعه LogPub در برابر ۸۱٫۱ درصد گزارش می‌کند Parsing Accuracy گروه‌بندی ۹۰٫۶ درصد و F۱، LLM فراخوانی این روش نشان می‌دهد تجمیع سطح قالب می‌تواند تعداد فراخوانی را کاهش دهد، اما خروجی آن قالب [۲۳]. است و برچسب ناهنجاری تولید نمی‌کند.

۲-۳-۷ مرورها، معیار ارزیابی، مدل پایه و استقرار واقعی

مقاله ۱۴ - مرور نظام‌مند تشخیص ناهنجاری لاگ در عصر LLM: De la Cruz Cabello، Prince Sales و تشخیص ناهنجاری لاگ را تحلیل کردند AIOps با روش مرور نظام‌مند، ۳۳ مقاله حوزه Machado و مطالعه مراحل آماده‌سازی داده، تجزیه، استخراج ویژگی، مدل تشخیص، مجموعه‌های معیار و شیوه ارزیابی را جمع‌بندی می‌کند. [۱۴] LLM را طبقه‌بندی و روند حرکت از مدل‌های عمیق به

ارائه نمی‌کند و نقش آن ایجاد چارچوب دسته‌بندی و استخراج BGL این مقاله یک روش آزمایشی جدید روی مسائل باز مانند هزینه، تفسیرپذیری، تعمیم و محرمانگی است [۱۴]. ساختار دسته‌بندی فصل حاضر از محورهای این مرور استفاده می‌کند، اما جدول ۲-۱ به مجموعه ۲۷ فایل محلی محدود است.

مقاله ۲۰ - پیاده‌سازی واقعی AIOps: De la Cruz Cabello و همکاران چارچوب خودنظارتی مبتنی بر عادی آموزش و در محیط دارای محدودیت محرمانگی به‌صورت Linux syslog را روی LogBERT محلی مستقر کردند. اثر حداقل تعداد توکن، اندازه و هم‌پوشانی پنجره و نرخ ماسک بر تشخیص و تأخیر بررسی شده است. [۲۰]

پنجره ۱۵ ثانیه‌ای با هم‌پوشانی ۱۰ ثانیه بهترین موازنه گزارش‌شده میان اثربخشی و تأخیر را ایجاد کرد و ارزیابی متخصصان کاربرد عملی روش را تأیید نمود [۲۰]. مقاله اهمیت استقرار محلی و اندازه‌گیری تأخیر بدون Qwen واقعی است، نه syslog آموزش‌دیده و داده آن LogBERT را پشتیبانی می‌کند، اما مدل آن BGL. تنظیم روی

مقاله ۲۵ - OpsEval: Liu و همکاران یک مجموعه معیار برای ارزیابی توان LLM در عملیات فناوری شامل ۷٬۳۳۴ سؤال چندگزینه‌ای و ۱٬۷۳۶ سؤال پاسخ‌آزاد در چند زیرحوزه OpsEval. اطلاعات ارائه کردند. مقایسه می‌کند in-context learning و chain-of-thought، self-consistency است و ۲۴ مدل را تحت [۲۵]

پاسخ را از نظر روانی، صحت و شواهد می‌سنجد و هم‌بستگی حدود ۰٫۹۱۸۵ با ارزیابی FAE-Score معیار آشکارساز لاگ نیست، اما حساسیت مدل‌های عملیات به OpsEval .متخصصان گزارش شده است [۲۵] پرامپت و ضرورت تعریف معیار متناسب با خروجی را نشان می‌دهد. در پایان‌نامه، چون خروجی دودویی .جای معیار متنی را می‌گیرند INVALID است، معیارهای ماتریس اغتشاش و

مقاله ۲۶ - YuLan: Zhu و همکاران گزارش فنی توسعه YuLan، یک مدل متن‌باز ۱۲ میلیارد پارامتری، را ارائه کردند. مدل با حدود ۱٫۷ تریلیون توکن چندزبانه پیش‌آموزی شده و پس از آن تنظیم دستوری و و داده tokenizer ،هم‌ترازی انسانی انجام گرفته است. مقاله برنامه‌درسی آسان‌به‌دشوار و جزئیات معماری را تشریح می‌کند. [۲۶]

روی ۲۲ معیار عمومی ارزیابی شده و مقاله‌ای درباره تشخیص ناهنجاری لاگ نیست [۲۶]. حضور YuLan آن در فصل دوم برای تعریف مدل متن‌باز، مراحل پیش‌آموزی تا دستورپذیری و تفاوت مدل پایه با مدل در آزمایش مقایسه‌ای YuLan است و Qwen۲٫۵:۷B-Instruct است. انتخاب عملی پایان‌نامه Instruct .استفاده نمی‌شود

مقاله ۲۷ - مرور آثار تشخیص ناهنجاری و داده خارج از Ding و Xu: **OOD و Anomaly برای LLM** LLM برای تشخیص و LLM را در حوزه‌های مختلف مرور و براساس نقش مدل به دو گروه LLM توزیع با برای تولید تقسیم کردند. مطالعه تفاوت تغییر هم‌توزیعی و تغییر معنایی، مجموعه‌های داده و مسیرهای آینده را توضیح می‌دهد. [۲۷]

دامنه مقاله فراتر از لاگ و شامل متن، سری زمانی و سایر انواع داده است [۲۷]. در این فصل از آن برای تعریف نقش مدل، تفکیک تشخیص از تولید و قرار دادن مطالعات تبیین یا افزایش داده در طبقه مناسب استفاده .در آن وجود ندارد BGL یا Qwen می‌شود؛ ارزیابی مستقیمی از

جمع‌بندی تحلیلی مسیر پژوهش‌ها ۸-۳-۲

رخداد بعدی را پیش‌بینی می‌کند DeepLog .مرحله نخست تحول ادبیات با مدل‌سازی توالی شکل گرفته است مسیر BERT-Log .زمینه دوطرفه و وظیفه خودنظارتی را به توالی می‌افزاید [۱۹، ۲۱] LogBERT و را با طبقه‌بند نظارت‌شده ترکیب می‌نماید [۱۸]. این سه مطالعه BERT دیگری را دنبال می‌کند و بازنمایی .مبنای تفاوت میان یادگیری الگوی عادی و طبقه‌بندی برچسب‌محور را مشخص می‌کنند

MLM از وظیفه ADALog و LogFiT .مرحله دوم بر کاهش وابستگی به برچسب ناهنجار تمرکز دارد الگوی توکن LogFiT :است parser روی داده عادی استفاده می‌کنند [۸، ۲۲]. تفاوت آن‌ها در واحد تحلیل و پیام خام و مستقل را با آستانه تطبیقی ارزیابی می‌کند. این تفاوت ADALog در توالی را می‌سنجد، درحالی‌که .نشان می‌دهد حذف برچسب به‌تنهایی واحد تصمیم را تعیین نمی‌کند

LLM-، LogLLM، GPT مبتنی بر LogGPT .به‌عنوان مدل آموزش‌پذیر است LLM مرحله سوم ورود از پیش‌آموزی عمومی آغاز می‌کنند، اما برای دامنه لاگ وزن‌ها یا نگاشت‌های PEFT و روش‌های LADE تفاوت بنیادی دارد، زیرا هزینه prompt-only مدل را تغییر می‌دهند [۴، ۵، ۷، ۹]. این خانواده با روش‌های .آموزش و داده انطباق بخشی از روش است

و مطالعه LogPrompt مرحله چهارم استفاده از توان دستورپذیری بدون تغییر وزن است. دو نسخه Hybrid .اثر دستور، دانش دامنه و استدلال را بررسی می‌کنند [۱، ۲، ۶] ChatGPT-LogGPT ،همین مسیر را با شواهد سطح توالی و نمونه بازیابی‌شده توسعه می‌دهد [۳]. در این خانواده Prompting .متن پرامپت و روش انتخاب زمینه بخشی از مدل عملیاتی محسوب می‌شود

AdaptiveLog ،از بازیابی نمونه عادی RAGLog .مرحله پنجم به کنترل هزینه و مقیاس اختصاص دارد ،و کش استفاده می‌کند [۱۰ RAG ،از ترکیب مدل‌های کلاسیک FlexLog از واگذاری مبتنی بر عدم‌قطعیت و نیز در وظیفه تجزیه نشان می‌دهد ساختار پیشوند می‌تواند فراخوانی را در □□□-□□□□□□□□□□ .[۱۳، ۱۲ .مقیاس میلیون‌ها پیام محدود کند [۲۳]

توضیح را در پاسخ مستقیم مدل قرار LogPrompt .مسیر دیگری از ادبیات بر تفسیرپذیری متمرکز است آشکارساز LogLAA توضیح را همراه تشخیص آموزش می‌دهد [۷]؛ و LLM-LADE می‌دهد [۱، ۶]؛ عمیق و تولید توضیح را در دو جزء جدا ترکیب می‌کند [۱۷]. این سه الگو نشان می‌دهند تفسیرپذیری می‌تواند .خروجی زبانی، هدف آموزشی یا جزء پس‌پردازش باشد

روی GPT مطالعات انتقال و ناپایداری، محدودیت ارزیابی روی یک توزیع ثابت را مطرح می‌کنند. پژوهش بازنمایی یکپارچه میان سامانه‌ها LogSynergy تغییر نسخه را مستقیماً آزمایش می‌کند [۱۱]؛ LOGEVOL مجموعه‌های ناپایدار واقعی و مصنوعی را بررسی می‌کند [۱۰]. پایان‌نامه حاضر FlexLog می‌سازد [۱۵]؛ و .ثابت است BGL این مسئله را به‌صورت تجربی پوشش نمی‌دهد و دامنه آن

و few-shot با LogPPT .پژوهش‌های تجزیه نشان می‌دهند آماده‌سازی ورودی یک مسئله مستقل است .و درخت پیشوند، کیفیت قالب را به‌عنوان خروجی اصلی می‌سنجند [۲۳، ۲۴] LLM با LogParser-LLM تفکیک شود. پایان‌نامه parser رخ دهد باید از اثر خود parser بنابراین هر بهبود آشکارساز که پس از تغییر .نرمال‌سازی را در هر دو روش ثابت نگه می‌دارد

فاصله میان معیار عمومی و استقرار را نشان می‌دهد؛ پارامتر پنجره و تأخیر AIOps مطالعه کاربرد واقعی در عملیات به LLM نیز نشان می‌دهد ارزیابی OpsEval .به اندازه دقت برای کار عملی اهمیت دارند [۲۰] معیار و پرامپت تخصصی نیاز دارد [۲۵]. این دو منبع مبنای گزارش زمان، نسخه مدل و قرارداد خروجی .هستند F۱ در کنار

را AIOps دو مقاله مروری نقش کنترل دامنه را دارند. مرور نظام‌مند لاگ، مراحل خط لوله و شکاف‌های .را به تشخیص و تولید تقسیم می‌نماید [۲۷] LLM نقش Anomaly/OOD استخراج می‌کند [۱۴] و مرور جدول فصل از اولی برای محورهای لاگ و از دومی برای نقش مدل استفاده می‌کند؛ هیچ‌یک به‌عنوان نتیجه .شمرده نمی‌شود BGL آزمایش

نیز در گروه نتیجه تشخیص قرار نمی‌گیرد، اما مراحل ساخت مدل پایه، تنظیم دستوری و هم‌ترازی YuLan محصول چنین Qwen۲٫۵:۷B-Instruct را مستند می‌کند [۲۶]. این تفکیک برای پایان‌نامه لازم است، زیرا .مراحل عمومی است، ولی وزن آن در پروژه تغییر نمی‌کند و فقط در زمان استنتاج به‌کار می‌رود [۲۸]

برآیند ۲۷ مقاله نشان می‌دهد پنج تصمیم طراحی باید صریح باشند: واحد تحلیل، نوع انطباق مدل، روش در پایان‌نامه فقط Prompt-only و Hybrid آماده‌سازی، سازوکار کنترل هزینه و محل استقرار. دو روش .یکی از این تصمیم‌ها—محل اعمال دانش قاعده—را تغییر می‌دهند و سایر عوامل برای مقایسه ثابت می‌مانند

دسته‌بندی و تحلیل مقایسه‌ای ۴-۲

شکل ۲-۱ تمام ۲۷ مقاله را در شش مسیر روش‌شناختی قرار می‌دهد و سپس نشان می‌دهد کدام مؤلفه‌ها در .طراحی پژوهش حاضر به هم می‌رسند. شماره‌های شکل با شماره فایل‌های پوشه و ارجاع‌های فصل یکسان‌اند هم مدل FlexLog قرارگرفتن یک مقاله در یک دسته به معنای نبود مؤلفه‌های دسته دیگر نیست؛ برای نمونه .و کش دارد [۱۰] RAG، LLM، کلاسیک

نقشه مطالعات بررسی‌شده در فصل دوم (۲۷ مقاله)			
ترتیبی و پیش‌آموزش‌دیده	**LLM آموزش، تنظیم و انتقال**		**پرامپت و استدلال درون‌متنی**
DeepLog [۲۱]؛ LogBERT [۱۹]؛ BERT-Log [۱۸]؛ LogFiT [۸]؛ ADALog [۲۲]	LogGPT [۵]؛ LogLLM [٤]؛ LLM-LADE [۷]؛ PEFT [۹]؛ GPT ناپایدار [۱۱]؛ LogSynergy [۱۵]؛ SemiRALD [۱٦]؛ LogLAA [۱۷]		LogPrompt [۱۰، ٦]؛ ChatGPT-LogGPT [۲]؛ Hybrid Prompting [۳]
بازیابی، همکاری و کنترل هزینه	**تجزیه و ساخت قالب**		**مرور، معیار و استقرار**
RAGLog [۱۲]؛ AdaptiveLog [۱۳]؛ FlexLog [۱۰]	LogPPT [۲٤]؛ LogParser-LLM [۲۳]		SLR [۱٤]؛ AIOps واقعی [۲۰]؛ OpsEval Anomaly/OOD ؛ مرور[۲٦] YuLan [۲٥]؛ [۲۷]
ترکیب شواهد ادبیات برای طراحی پژوهش حاضر: معناشناسی مدل زبانی + قالب پایدار + کنترل تصمیم + استنتاج محلی			
دامنه و واحد تحلیل	**متغیر مستقل مقایسه**	**کارایی**	**استقرار و سنجش**
؛ طبقه‌بندی خطبه‌خط؛ حذف برچسب BGL فقط از ورودی	LLM قطعی پیش از Hybrid: guard انتقال دانش قاعده به پرامپت: Prompt-only	نرمال‌سازی و کش سطح قالب؛ رد INVALID	Qwen۲,٥:۷B-Instruct روی Ollama؛ کیفیت، زمان و منبع تصمیم

شکل ۲-۱: ارتباط ۲۷ مقاله بررسی‌شده و جایگاه پژوهش حاضر

پرامپت، بازیابی و استقرار، LLM تا BERT در بخش بالایی شکل، مسیر تاریخی و فنی از مدل‌های ترتیبی و واقعی دیده می‌شود. بخش پایینی نوآوری عملیاتی مورد سنجش را نشان می‌دهد: ثابت نگه‌داشتن مدل و بیرونی و پرامپت. این شکل ادعای برتری هیچ guard پیش‌پردازش و تغییر محل اعمال دانش قاعده میان مسیر را بیان نمی‌کند و فقط رابطه اجزای برگرفته از منابع را نمایش می‌دهد.

جدول ۲-۱: مشخصات و مقایسه تمام ۲۷ مقاله موجود در پوشه پژوهش

شماره و مطالعه	سال	دسته	روش/مدل	داده‌ها	واحد تحلیل	نتیجه یا نقش گزارش‌شده
اولیه LogPrompt [۱]	۲۰۲۳	پرامپت	راهبرد دستور و تبیین + LLM	چند مجموعه لاگ	قالب/توالی	نسخه اولیه تحلیل بدون آموزش درون‌دامنه
ChatGPT-LogGPT [۲]	۲۰۲۳	پرامپت	و دانش دامنه ChatGPT	BGL، Spirit	خط/پنجره	بررسی پرامپت، پنجره و تبیین روی نمونه تایی۲۰۰۰
Hybrid Prompting [۳]	۲۰۲۵	پرامپت ترکیبی	رخداد + توالی + بازیابی	HDFS، BGL، Thunderbird	توالی یک‌دقیقه‌ای	BGL: P=۶۳,۶۵، R=۹۹,۶۰، F۱=۷۷,۸۱
LogLLM [٤]	۲۰۲۴	LLM آموزش	BERT + projector + LLaMA	HDFS، BGL، Liberty، Thunderbird	توالی	LLM و طبقه‌بند encoder ترکیب بازنمایی
LogGPT via GPT [٥]	۲۰۲۳	RL/آموزش	RL + پیش‌بینی رخداد GPT	HDFS، BGL، Thunderbird	توالی	هم‌راستاسازی پیش‌بینی رخداد با هدف تشخیص
LogPrompt توسعه‌یافته [٦]	۲۰۲۴	پرامپت	و تبیین zero-shot، CoT	مجموعه ۹	قالب/توالی	تا ۵۵,۹٪ بهبود؛ تفسیرپذیری ۴,۴۲/۵
LLM-LADE [۷]	۲۰۲۵	PEFT/تبیین	LoRA + افزایش داده + بهینه‌سازی برخط	مجموعه‌های عمومی	پیام/توالی	تشخیص و توضیح همزمان با پایگاه دانش
LogFiT [۸]	۲۰۲۵	خودنظارتی	top-k مدل زبانی ماسک‌شده و	HDFS، BGL، Thunderbird	توالی	آموزش فقط روی لاگ عادی
PEFT-LAD [۹]	۲۰۲۵	PEFT	LoRA/ReFT روی RoBERTa، GPT-۲، Llama-۳	HDFS، BGL، Spirit، Thunderbird	توالی	تحلیل پایداری، کارایی نمونه و انتقال
FlexLog [۱۰]	۲۰۲۴/۲۵	ترکیبی	ML ensemble + Mistral + RAG + cache	چهار مجموعه ناپایدار	توالی	داده کمتر؛ تا ۱۳ واحد درصد بهبود؛ کمتر از ۱ ثانیه
Unstable Logs GPT [۱۱]	۲۰۲۴	تنظیم/پرامپت	GPT-٤ تنظیم‌شده و GPT-۳ پرامپت	LOGEVOL-Hadoop	توالی	در پروتکل مطالعه fine-tuning برتری
RAGLog [۱۲]	۲۰۲۳/۲۴	بازیابی	LLM + پایگاه برداری عادی	BGL، Thunderbird	خط/توالی	Precision=۰,۹۱، Recall=۰,۸۸، F۱=۰,۸۹
AdaptiveLog [۱۳]	۲۰۲۵	همکاری مدل	بازیابی + LLM + مدل کوچک	چند وظیفه لاگ	وابسته‌به‌وظیفه	LLM تا ۷۳٪ کاهش هزینه فراخوانی
مرور نظام‌مند AIOps [۱٤]	۲۰۲۵	مرور	و طبقه‌بندی ۳۳ مقاله SLR	ادبیات حوزه	مطالعه	چارچوب مراحل، داده‌ها، مدل‌ها و شکاف‌ها
LogSynergy [۱٥]	۲۰۲۵	انتقال	ویژگی یکپارچه + LLM تفسیر	عمومی و واقعی	توالی	بیش از ۸۳٪ عمومی و ۸۹٪ واقعی F۱
SemiRALD [۱٦]	۲۰۲۵	نیمه‌نظارتی	ChatGPT parsing + RoBERTa + BiLSTM	HDFS، BGL	توالی	٪برابر ۷,۳ و ۸,۲ F۱ بهبود متوسط
LogLAA [۱۷]	۲۰۲۶	یکپارچه	parser + CNN-LSTM + LLM explanation	چند مجموعه	توالی	و تولید توضیح Precision، parser بهبود
BERT-Log [۱۸]	۲۰۲۲	نظارت‌شده	طبقه‌بند + BERT	HDFS، BGL	پنجره/توالی	BGL گزارش‌شده ۹۹,۴٪ روی F۱
LogBERT [۱۹]	۲۰۲۱	خودنظارتی	و ابرکره MLM با BERT	HDFS، BGL، Thunderbird	توالی	یادگیری الگوی عادی با دو هدف خودنظارتی
واقعی AIOps [۲۰]	۲۰۲۶	استقرار	محلی LogBERT	واقعی Linux syslog	پنجره زمانی	پنجره ۱۵ ثانیه با همپوشانی ۱۰ ثانیه
DeepLog [۲۱]	۲۰۱۷	ترتیبی	و پیش‌بینی رخداد بعدی LSTM	HDFS، OpenStack	توالی	تشخیص برخط، به‌روزرسانی و جریان عیب‌یابی
ADALog [۲۲]	۲۰۲۵	بدون نظارت	آستانه + DistilBERT MLM تطبیقی	BGL، Thunderbird، Spirit	خط	و مستقل از توالی parser بدون
LogParser-LLM [۲۳]	۲۰۲۴	تجزیه	درخت پیشوند + LLM	Loghub-۲k، LogPub	خط/قالب	٪گروه‌بندی ۹۰,۶ F۱ فراخوانی؛ ۲۷۲,۵
LogPPT [۲٤]	۲۰۲۳	تجزیه	few-shot prompt tuning روی RoBERTa	مجموعه ۱۶	خط/قالب	بیش از ۰,۹ با ۳۲ نمونه برچسب‌خورده
OpsEval [۲٥]	۲۰۲۵	معیار	MCQ/QA + CoT/ICL + FAE-Score	Ops پرسش ۹۰۷۰	پرسش	حدود ۰,۹۱۸۵ FAE ؛ همبستگی۲۴ LLM

شماره و مطالعه	سال	دسته	روش/مدل	داده‌ها	واحد تحلیل	نتیجه یا نقش گزارش‌شده
YuLan [۲۶]	۲۰۲۴	مدل پایه	۱۲ LLM و سه Bمتن‌باز مرحله آموزش	توکن؛ ۲۲ معیار ۱٫۷T	متن	مرجع ساخت مدل دستورمحور؛ نه آشکارساز لاگ
مرور Anomaly/OOD [۲۷]	۲۰۲۵	مرور	برای LLM رده‌بندی تشخیص/تولید	چند حوزه داده	مطالعه	LLM و نقش‌های OOD و anomaly تفکیک

جدول ۲-۱ حضور هر ۲۷ فایل را قابل ردیابی می‌کند. سطرهای [۱۴]، [۲۵]، [۲۶] و [۲۷] نتیجه تشخیص ندارند، زیرا به‌ترتیب مرور نظام‌مند، معیار عملیات، گزارش مدل پایه و مرور عمومی‌اند. درج آن‌ها BGL در جدول برای تفکیک نقش منبع نظری از مطالعه آزمایشی انجام شده است.

۲-۴-۱ مقایسه براساس واحد تحلیل و پروتکل BGL

نزدیک‌ترین ADALog. واحد تحلیل از خط تا توالی و پنجره تغییر می‌کند، BGL در میان مطالعات دارای پنجره یا توالی Hybrid Prompting و LogBERT، BERT-Log مطالعه به تصمیم خطمحور است [۲۲]؛ بخشی نمونه‌گیری‌شده از آزمون را بررسی می‌کند ChatGPT مبتنی بر LogGPT می‌سازند [۳، ۱۸، ۱۹]؛ و بنابراین مقایسه عددی فقط زمانی معتبر است که واحد و نمونه یکسان باشد. [۲].

Hybrid اصلی حدود ۷٫۳۴ درصد پیام ناهنجار دارد [۲، ۱۹]، اما BGL. نسبت کلاس‌ها نیز متفاوت است در داده متوازن و نامتوازن رفتار Accuracy. هزار توالی از هر کلاس انتخاب می‌کند [۳] Prompting و ماتریس اغتشاش در سطح خط برای F۱ و Recall، Precision یکسانی ندارد؛ به همین دلیل گزارش پایان‌نامه ضروری است.

۲-۴-۲ مقایسه براساس نیاز به آموزش

روی داده مقصد آموزش یا تنظیم می‌شوند ADALog و LogFiT، LogBERT، BERT-Log، DeepLog آموزش چندمرحله‌ای یا یادگیری تقویتی GPT مبتنی بر LogGPT و LogLLM. [۲۲، ۲۱، ۱۹، ۱۸، ۸] ReFT، LoRA نیز از SemiRALD و LogSynergy، PEFT-LAD، LLM-LADE. دارند [۴، ۵] انتقال یا آموزش نیمه‌نظارتی بهره می‌گیرند [۷، ۹، ۱۵، ۱۶].

تغییر وزن طبقه‌بند RAGLog و Hybrid Prompting، ChatGPT-LogGPT، LogPrompt، در مقابل را شرط اصلی قرار نمی‌دهند و تصمیم را از پرامپت و زمینه می‌گیرند [۱–۳، ۶، ۱۲]. پژوهش حاضر در بیرونی با دانش درون پرامپت را به پروتکل اضافه guard این گروه قرار دارد، اما کش سطح قالب و مقایسه می‌کند.

۲-۴-۳ مقایسه کنترل هزینه و استقرار

با بازیابی، زمینه مرتبط را انتخاب می‌کند [۱۲] RAGLog. سه الگوی کنترل هزینه در منابع دیده می‌شود و FlexLog. واگذار می‌کند [۱۳] LLM با عدم‌قطعیت مدل کوچک، نمونه را به AdaptiveLog، از کش یا ساختار تجمیعی برای جلوگیری از پردازش تکراری استفاده می‌کنند [۱۰ LogParser-LLM. این سازوکارها از نظر هدف مشترک اما از نظر داده ذخیره‌شده و منطق تصمیم متفاوت‌اند. [۲۳]

مطرح شده است [۲۰]. بیشتر مطالعات پرامپتی AIOps استقرار محلی به‌صورت صریح در مطالعه واقعی Mistral یا LLaMA استفاده می‌کنند [۲، ۳، ۱۱] و برخی از GPT-٤o-mini یا GPT-٤، ChatGPT از fine-tuning بدون Ollama، روی Qwen۲,۵:۷B-Instruct تنظیم‌شده بهره می‌گیرند [۴، ۷، ۱۰]. استفاده از ترکیب عملیاتی مورد سنجش پایان‌نامه است [۲۸، ۲۹]، INVALID و با ثبت زمان و tuning.

۲-۴-۴ مقایسه بازنمایی ورودی و آماده‌سازی

ابتدا پیام را به کلید رخداد تبدیل می‌کند DeepLog. خانواده ترتیبی معمولاً پیام خام را مستقیماً دریافت نمی‌کند با BERT-Log. قرار می‌دهد [۱۹، ۲۱] BERT نیز توالی کلیدهای تجزیه‌شده را ورودی LogBERT و می‌تواند به مرحله parser و پنجره، پیام‌ها را برای مدل آماده می‌کند [۱۸]. در این روش‌ها خطای parser تشخیص منتقل شود.

کلاسیک، پاک‌سازی یا قواعد منظم را parser برای کاهش وابستگی به LogLLM و ADALog روش‌های مسیر، عدد و نشانی را با جانشین عمومی جایگزین می‌کند و ADALog. روی متن به‌کار می‌گیرند [۴، ۲۲] به «parser» را از متن پردازش‌شده می‌سازد. بنابراین اصطلاح «بدون BERT بردارهای معنایی LogLLM معنی نبود پیش‌پردازش نیست.

نمونه عادی RAGLog. در روش‌های پرامپتی، آماده‌سازی علاوه بر پاک‌سازی شامل انتخاب زمینه نیز هست ویژگی‌های رخداد و توالی را محاسبه می‌کند [۳] و Hybrid Prompting، بازیابی‌شده را می‌افزاید [۱۲] LLM دانش وظیفه را در دستور می‌نویسد [۱، ۶]. در نتیجه بخشی از توان روش خارج از خود LogPrompt و در ساخت ورودی قرار دارد.

پایان‌نامه ورودی را به قالب نرمال‌شده تبدیل می‌کند، اما نمونه بازیابی‌شده یا توالی زمانی را به پرامپت نمی‌افزاید. ثابت ماندن این بازنمایی در دو روش باعث می‌شود اختلاف نتیجه به تغییر آماده‌سازی نسبت داده نشود. تعداد قالب‌های یکتا و نرخ برخورد کش نیز اندازه عملی اثر نرمال‌سازی را نشان می‌دهند.

۲-۴-۵ مقایسه سازوکار تصمیم و نوع خروجی

LogFiT رخداد بعدی و DeepLog. سازوکار تصمیم در مدل‌های پیش‌بینی‌محور، احتمال رخداد یا توکن است top-k توکن ماسک‌شده را پیش‌بینی می‌کنند [۸، ۲۱، ۲۲]. آستانه یا قرارگرفتن رخداد واقعی در ADALog و خروجی پیوسته مدل را به برچسب دودویی تبدیل می‌کند. انتخاب آستانه بخشی از پروتکل این روش‌هاست.

BERT-Log. تنظیم‌شده مستقیماً احتمال کلاس را تولید می‌کند LLM در مدل‌های طبقه‌بندی‌شده، یک سر طبقه‌بند یا نمونه‌های این گروه‌اند [۴، ۷، ۱۶، ۱۸]. کیفیت خروجی SemiRALD و LLM-LADE، LogLLM، Log به داده آموزش، تابع زیان و روش تنظیم وابسته است.

می‌توانند ChatGPT-LogGPT و LogPrompt. در روش پرامپتی، پاسخ متنی باید به برچسب تبدیل شود برای پاسخ آزاد معیار جداگانه می‌سازد [۲۵]. در OpsEval توضیح همراه برچسب تولید کنند [۱، ۲، ۶] و یک سامانه خودکار، متن اضافی یا برچسب مبهم می‌تواند پردازش پایین‌دستی را مختل کند؛ به همین دلیل دودویی محدود می‌کند JSON پایان‌نامه خروجی را به.

پاسخ مدل را از خطای طبقه‌بندی جدا می‌کند. اگر پاسخ نامعتبر به‌صورت عادی یا ناهنجار INVALID ثبت رفتار مدل را به‌درستی نشان نمی‌دهند. همچنین ذخیره نکردن F۱ و Accuracy، پیش‌فرض‌گذاری شود در کش مانع تکثیر یک خطای قالب در خطوط تکراری می‌شود INVALID.

مقایسه تبیین‌پذیری و بازتولیدپذیری ۶-۴-۲

برای کاربر انسانی قابل خواندن است، اما ارزیابی آن به معیار یا قضاوت متخصص نیاز LLM تبیین آزاد از طراحی همزمان تشخیص و توضیح و LLM-LADE ،از امتیاز متخصصان LogPrompt .دارد طبقه‌بندی نیازمند F۱ از امتیاز کیفیت توضیح استفاده می‌کند [۱، ۶، ۷، ۱۷]. مقایسه این امتیازها با LogLAA تفکیک دو هدف است.

تبیین ساختاری به‌جای متن آزاد می‌تواند شامل شناسه قالب، منبع تصمیم، قاعده فعال و زمان باشد. این اطلاعات برای ممیزی خطبه‌خط قابل ذخیره و مقایسه‌اند. طراحی پایان‌نامه این نوع تبیین را انتخاب می‌کند تا خروجی دو روش یکسان بماند و تفاوت طول پاسخ، زمان را مخدوش نکند.

ChatGPT- دشوارتر است؛ مطالعه API بازتولیدپذیری مدل‌های تجاری با تغییر نسخه خدمت و محدودیت نیز به مدل تجاری مشخص وابسته است Hybrid Prompting ناچار به نمونه‌گیری محدود شده و LogGPT ،را فراهم می‌کند runtime استقرار محلی مدل باز امکان ثبت فایل مدل، کوانتیزه‌سازی و نسخه .[۳، ۲] هرچند سخت‌افزار و تنظیمات تولید همچنان باید گزارش شوند [۲۰، ۲۸، ۲۹].

تعداد تلاش ،timeout ،بذر یا ترتیب داده، دما، مدل، نسخه پرامپت ،Prompt-only و Hybrid برای مقایسه و خروجی خطبه‌خط امکان run identifier مجدد و سیاست کش باید ثابت باشند. ثبت این متغیرها همراه بازسازی آزمایش و تحلیل منشأ اختلاف را فراهم می‌کند. این الزامات از شکاف بازتولید و ارزیابی عملی پشتیبانی می‌شوند [۱۴، ۲۵] OpsEval مطرح‌شده در مرور نظام‌مند و.

جدول ۲-۲: جمع‌بندی دسته‌های مطالعات و ارتباط آن‌ها با طراحی پژوهش

دسته	مقالات	نقش مدل زبانی	نیاز غالب به آموزش	ارتباط با پژوهش حاضر
ترتیبی/پیش‌آموزش‌دیده	[۸، ۱۸، ۱۹، ۲۱، ۲۲]	یا مدل توالی encoder	بله؛ عادی یا برچسب‌خورده	واحد تحلیل و خط پایه BGL، تعریف زبانی
آموزش/تنظیم/انتقال LLM	[٤، ٥، ٧، ٩، ١١، ١٥، ١٦، ١٧]	طبقه‌بند، مولد داده یا تبیین	یا انتقال PEFT، آموزش	بدون تنظیم Qwen مرز تمایز با
پرامپت	[١، ٢، ٣، ٦]	تصمیم‌گیر مستقیم	بدون تغییر وزن	Prompt-only نزدیک‌ترین خانواده به
بازیابی/همکاری/هزینه	[١٠، ١٢، ١٣]	RAG جزء آبشار یا	متغیر	مبنای مقایسه کش، دروازه و فراخوانی
تجزیه	[٢٣، ٢٤]	استخراج قالب	بدون آموزش گسترده یا few-shot	مبنای نرمال‌سازی و کش قالب
مرور/معیار/مدل/است قرار	[١٤، ٢٠، ٢٥، ٢٦، ٢٧]	چارچوب نظری یا زیرساخت	وابسته به منبع	رده‌بندی، محرمانگی، ارزیابی و مدل دستورمحور

۲-۵ شکاف تحقیقاتی و جایگاه پژوهش حاضر

۲-۵-۱ ناهمگونی پروتکل‌های BGL

مطالعات BGL ۱۲ ،۹ ،۸ ،۵–۲] استفاده کرده‌اند از پیام، توالی، پنجره گره، پنجره زمانی و نمونه متوازن، [۱۶، ۱۸، ۱۹، ۲۲]. در نتیجه، یک نتیجه عددی بدون مشخصات نمونه و واحد تصمیم قابل انتقال مستقیم نیست. پژوهش حاضر تصمیم را خطبه‌خط تعریف، برچسب را از ورودی حذف و ماتریس اغتشاش را در همان سطح محاسبه می‌کند.

۲-۵-۲ جدانشدن کیفیت تصمیم از هزینه استنتاج

هزینه را با کش، واگذاری یا تجمیع کاهش می‌دهند [۱۰ LogParser-LLM و AdaptiveLog، FlexLog بازیابی را وارد زمینه می‌کند [۱۲]. بااین‌حال، همه مطالعات نرخ برخورد کش، منبع RAGLog و [۲۳، ۱۳ و زمان انتهابه‌انتها را همراه معیارهای کیفیت گزارش نمی‌کنند. پایان‌نامه این INVALID، تک‌تک تصمیم‌ها شاخص‌ها را همزمان ثبت می‌کند.

۲-۵-۳ محل اعمال دانش دامنه

شواهد Hybrid Prompting دانش در پرامپت است [۱، ۲، ۶]؛ ChatGPT-LogGPT و LogPrompt در یک جزء بیرونی را FlexLog و AdaptiveLog رخداد، توالی و نمونه را در پرامپت ترکیب می‌کند [۳]؛ قطعی guard به‌کار می‌گیرند [۱۰، ۱۳]. در مجموعه بررسی‌شده، مقایسه کنترل‌شده LLM برای واگذاری به بیرون مدل با انتقال همان دانش به پرامپت، تحت مدل، داده، نرمال‌سازی و کش یکسان، گزارش نشده است.

۲-۵-۴ کش تصمیم سطح قالب

ساختار قالب را برای کاهش فراخوانی استفاده LogParser-LLM پایگاه نمونه می‌سازد [۱۲] و RAGLog نیز کش را در سامانه‌ای چندمدلی به‌کار می‌گیرد [۱۰]. کش نسخه‌بندی‌شده تصمیم FlexLog می‌کند [۲۳]؛ BGL دودویی در سطح قالب نرمال‌شده، همراه با رد پاسخ نامعتبر و مقایسه دو مسیر تصمیم، در مطالعات مرورشده محور آزمایش مستقل نبوده است.

متن‌باز متوسط LLM استنتاج محلی یک ۲-۵-۵

یا LLaMA استفاده کرده‌اند [۲، ۳، ۱۱، ۱۲] و روش‌های مبتنی بر GPT مطالعات پرامپتی عمدتاً از خدمات LogBERT از AIOps معمولاً آموزش یا ترکیب چندمدلی دارند [۴، ۷، ۱۰]. مطالعه محلی Mistral و در Ollama با وزن ثابت روی Qwen۲٫۵:۷B-Instruct آموزش‌دیده بهره می‌گیرد [۲۰]. ارزیابی شکاف استقرار موردنظر این پژوهش است [۲۸، ۲۹] BGL طبقه‌بندی خطبه‌خط.

صورت‌بندی جایگاه پژوهش ۲-۵-۶

و تصمیم خطمحور؛ نرمال‌سازی حساس به معنا؛ BGL :جایگاه پژوهش از ترکیب پنج جزء تشکیل می‌شود Prompt-only با Hybrid اعتبارسنجی‌شده؛ و مقایسه JSON کش نسخه‌بندی‌شده در سطح قالب؛ خروجی در شرایط ثابت. این صورت‌بندی ادعای پیشینی درباره برتری روش‌ها ندارد و نتیجه باید از اجرای کامل هر دو روش روی پروتکل یکسان حاصل شود.

جمع‌بندی ۲-۶

،تجزیه و نرمال‌سازی، پارادایم‌های یادگیری ،BGL ،در این فصل مفاهیم ساختار لاگ، سطوح ناهنجاری و معیارهای Ollama ،پرامپت، بازیابی، کش ،Qwen۲٫۵ ،LLM نقش ،Transformer مدل‌های ترتیبی و ارزیابی تعریف شد. سپس هر ۲۷ مقاله موجود در پوشه پژوهش به‌صورت مستقل مرور و در جدول مقایسه درج گردید.

،پرامپت ،LLM مرور مطالعات نشان داد ادبیات از مدل‌سازی توالی به مدل‌های زبانی پیش‌آموزش‌دیده، تنظیم ،و چارچوب‌های چندمدلی گسترش یافته است. در عین حال، تفاوت واحد تحلیل، آموزش، نمونه‌گیری RAG محل استقرار و کنترل هزینه سبب می‌شود نتایج عددی تنها در پروتکل‌های همسان مقایسه شوند.

،با دانش قاعده درون پرامپت LLM قطعی خارج از guard شکاف استخراج‌شده، مقایسه مستقیم و کنترل‌شده خطمحور است. فصل بعد باید BGL محلی، کش سطح قالب و خروجی اعتبارسنجی‌شده در Qwen همراه با اجرای دو روش و ،Ollama معماری نرم‌افزار، قواعد نرمال‌سازی، متن پرامپت‌ها، کلید کش، تنظیمات محاسبه معیارها را به‌صورت عملیاتی تشریح کند.

منابع فصل دوم ۲-۷

[۱] Liu, Y.; Tao, S.; Meng, W.; Wang, J.; Ma, W.; Chen, Y.; Zhao, Y.; Yang, H.; Jiang, Y. Interpretable Online Log Analysis Using Large Language Models with Prompt Strategies. ۲۰۲۳.

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

[۳] 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.

[٤] Guan, W.; Cao, J.; Qian, S.; Gao, J.; Ouyang, C. LogLLM: Log-based Anomaly Detection Using Large Language Models. ۲۰۲٤.

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

[٦] 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. ۲۰۲٤.

[۷] 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, Article ۱۱٤۰٦٤, ۲۰۲٥.

[۸] Almodovar, C.; Sabrina, F.; Karimi, S.; Azad, S. LogFiT: Log Anomaly Detection using Fine-Tuned Language Models. ۲۰۲٥.

[٩] Lim, Y. F.; Zhu, J.; Pang, G. Adapting Large Language Models for Parameter-Efficient Log Anomaly Detection. ٢٠٢٥.

[١٠] Hadadi, F.; Xu, Q.; Bianculli, D.; Briand, L. LLM meets ML: Data-efficient Anomaly Detection on Unstable Logs. ٢٠٢٤–٢٠٢٥.

[١١] Hadadi, F.; Xu, Q.; Bianculli, D.; Briand, L. Anomaly Detection on Unstable Logs with GPT Models. ٢٠٢٤.

[١٢] Pan, J.; Wong, S. L.; Yuan, Y. RAGLog: Log Anomaly Detection using Retrieval Augmented Generation. ٢٠٢٣–٢٠٢٤.

[١٣] 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:٢٥٠١,١١٠٣١, ٢٠٢٥.

[١٤] 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, ٢٨, Article ٢٠٠٦٠٨, ٢٠٢٥.

[١٥] 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. ٢٠٢٥.

[١٦] 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, ١٨٣, ١٠٧٧٤٣, ٢٠٢٥. DOI: ١٠,١٠١٦/j.infsof.٢٠٢٥,١٠٧٧٤٣.

[١٧] Gao, Y.; Luo, T.; Huang, K.; Tang, J.; Li, X. LogLAA: an adaptive integrated log anomaly analysis framework. Cybersecurity, ٩:١٤١, ٢٠٢٦. DOI: ١٠,١١٨٦/s٤٢٤٠٠-٠٢٦-٠٠٥٧٣-٨.

[١٨] Chen, S.; Liao, H. BERT-Log: Anomaly Detection for System Logs Based on Pre-trained Language Model. Applied Artificial Intelligence, ٣٦(١), ٢٠٢٢. DOI: ١٠,١٠٨٠/٠٨٨٣٩٥١٤,٢٠٢٢,٢١٤٥٦٤٢.

[١٩] Guo, H.; Yuan, S.; Wu, X. LogBERT: Log Anomaly Detection via BERT. arXiv:٢١٠٣,٠٤٤٧٥, ٢٠٢١.

[٢٠] 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, ٨, Article ٢٠٠٤٧٥, ٢٠٢٦.

[٢١] Du, M.; Li, F.; Zheng, G.; Srikumar, V. DeepLog: Anomaly Detection and Diagnosis from System Logs through Deep Learning. ACM CCS, ٢٠١٧. DOI: ١٠,١١٤٥/٣١٣٣٩٥٦,٣١٣٤٠١٥.

[٢٢] Pospieszny, P.; Mormul, W.; Szyndler, K.; Kumar, S. ADALog: Adaptive Unsupervised Anomaly Detection in Logs with Self-attention Masked Language Model. arXiv:٢٥٠٥,١٣٤٩٦, ٢٠٢٥.

[٢٣] 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:٢٤٠٨,١٣٧٢٧, ٢٠٢٤.

[٢٤] Le, V.-H.; Zhang, H. Log Parsing with Prompt-based Few-shot Learning. arXiv:٢٣٠٢,٠٧٤٣٥, ٢٠٢٣.

[٢٥] 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, ٢٠٢٥. DOI: ١٠.١١٤٥/٣٦٩٦٦٣٠.٣٧٢٨٥٧٢.

[٢٦] Zhu, Y.; Zhou, K.; Mao, K.; et al. YuLan: An Open-source Large Language Model. arXiv:٢٤٠٦,١٩٨٥٣, ٢٠٢٤.

[٢٧] Xu, R.; Ding, K. Large Language Models for Anomaly and Out-of-Distribution Detection: A Survey. arXiv:٢٤٠٩,٠١٩٨٠, ٢٠٢٥.

[٢٨] Qwen Team. Qwen٢,٥ Technical Report. arXiv:٢٤١٢,١٥١١٥, ٢٠٢٤.

[٢٩] Ollama. API Introduction; Structured Outputs; Usage Metrics. Official documentation. Accessed ١٤ August ٢٠٢٦.