ABSTRACT

In Germany, there are demands to host large language models (LLMs) on internal servers by banks to satisfy the related data protection requirements on confidential information. For deploying an LLM on an internal server, monitoring the system and recognizing if a high rate of request failures occurs are important to ensure the system can answer every request before reaching its timeout, which can be done by forecasting with a machine learning model. Before bringing the system into production, it is important to test whether the time series features generated from the LLM server are relevant for training the model for forecasting and find some insights from it.

This study used a simulation that reflects a real-world situation to test whether the features extracted from an LLM deployment server are relevant for LLM server performance forecasting and afterward find the key features for the forecasting process. First, the simulation dataset was generated based on BurstGPT and then run on the server. After that, the request and server performance metrics data were extracted and used to forecast the server performance, i.e., the server's failure rate of processing requests at specific time intervals. An XGBoost model was compared for the forecasting process with ARMA, VAR, and univariate XGBoost model and then interpreted using Shapley additive explanations (SHAP) to find useful features for forecasting the server's failure rate.

Based on an example use case of a chatbot project, the XGBoost model had the best performance for forecasting the server's failure rate, beating other comparison models. The performance was determined based on MAE and RMSE metrics on the test data with rolling origin evaluation setups. The interpretation of the XGBoost model with SHAP revealed that request duration was the most important feature of the forecasting process. This result was consistent with the XGBoost feature importance method with weight, gain, and coverage.

The prompt sampling method from BurstGPT with Zipf distribution was also compared with a sampling method based on the empirical prompt length distribution of the prompt dataset. Both prompt sampling methods were compared based on the duration of the simulation run, the number of failed requests, and the feature distributions. It was determined that differences existed regarding those aspects. This outcome indicated that considering the prompt length distribution before choosing the prompt sampling method for the simulation is crucial, as it might offer a different result on the simulation, forecasting, interpretation, and resulting parameters to be used

in production.

Keywords: LLM, simulation, time series analysis, BurstGPT, XGBoost, rolling origin evaluation setups, SHAP, Zipf distribution.

ZUSAMMENFASSUNG

In Deutschland gibt es Nachfragen im Bankwesen, Large-Language-Models (LLMs) auf internen Servern zu hosten, um Datenschutzanforderungen für vertrauliche Informationen zu erfüllen. Für die Überwachung eines LLMs auf einem internen Server ist es wichtig zu erkennen, ob und wann eine hohe Rate von Anfrageausfällen auftritt. So kann sichergestellt werden, dass das System jede Anfrage beantworten kann, bevor es zu Zeitüberschreitungen kommt. Um das zu erreichen, erfolgt eine Vorhersage mit einem maschinellen Lernmodell. Dazu werden vom LLM-Server ermittelten Zeitreihenmerkmale für das Training des Analyse-Modells benutzt. Es wird dann getestet, inwieweit diese Merkmale relevant für die Vorhersage sind und welche die Schlüsselmerkmale sind.

In dieser Studie wurde eine Simulation verwendet, die eine reale Situation widerspiegelt, um zu testen, ob die von einem LLM-Server extrahierten Merkmale für die Leistungsprognose des LLM-Servers relevant sind, und um die Schlüsselmerkmale für den Prognoseprozess zu finden. Zunächst wurde der Simulationsdatensatz auf der Grundlage von BurstGPT erstellt und dann auf dem Server verarbeitet. Danach wurden die Anfragedaten und die Leistungsmetriken des Servers extrahiert und zur Vorhersage der Serverleistung verwendet, d. h. der Ausfallrate des Servers bei der Bearbeitung von Anfragen in bestimmten Zeitintervallen. Ein XGBoost-Modell wurde für den Vorhersageprozess mit ARMA-, VAR- und univariaten XGBoost-Modell verglichen und dann mit Hilfe von Shapley-Additive-Explanations (SHAP) interpretiert, um nützliche Merkmale für die Vorhersage der Ausfallrate des Servers zu finden.

Basierend auf einem Anwendungsfall eines Chatbot-Projekts zeigte das XGBoost-Modell die beste Leistung bei der Vorhersage der Ausfallrate des Servers und übertraf die anderen Modelle. Die Vorhersagegüte wurde auf der Grundlage der MAE- und RMSE-Metriken für die Testdaten mit Rolling-Origin-Evaluierungs-Setups ermittelt. Die Ergebnisse des XGBoost-Modells mit SHAP ergaben, dass die Anfragedauer das wichtigste Merkmal des Vorhersageprozesses war. Dieses Ergebnis stand im Einklang mit der XGBoost-Feature-Importance-Methode mit Weight, Gain, und Coverage.

Die Prompt-Sampling-Methode von BurstGPT mit Zipfscher Verteilung wurde auch mit einer Sampling-Methode verglichen, die auf der empirischen Prompt-Längenverteilung des Prompt-Datensatzes basiert. Beide Prompt-Sampling-Methoden wurden anhand der Dauer des Simulationslaufs, der Anzahl der fehlgeschlagenen Anfragen und der Merkmalsverteilungen verglichen. Es wurde festgestellt, dass hinsichtlich dieser Aspekte Unterschiede

bestehen. Dieses Ergebnis deutet darauf hin, dass die Berücksichtigung der Prompt-Längenverteilung vor der Wahl der Prompt-Sampling-Methode für die Simulation von entscheidender Bedeutung ist, da sie zu unterschiedlichen Ergebnissen bei der Simulation, Vorhersage, Interpretation und den daraus resultierenden Parametern, die in der Produktion verwendet werden sollen, führen kann.

Schlüsselwörter: LLM, Simulation, Zeitreihenanalyse, BurstGPT, XGBoost, Rolling-Origin-Evaluierungs-Setups, SHAP, Zipfsche Verteilung.