A Work Project, presented as part of the requirements for the Award of a Master's Degree in Business Analytics from the Nova School of Business and Economics.

Forecasting Cost-per-Click of Keywords in Google's Competitive Paid Search Advertising Market

A Semantic-based Clustering Approach

Marc Peters (58333)

Group Part co-written by:

Paul Reichert (59281)
Marc Peters (58333)
Frederic Liebald (61679)
Paul Schuhmann (59482)
Leon May (59765)

Work project carried out under the supervision of:

Qiwei Han, Maximilian Kaiser

31/12/2024

Abstract

Accurately forecasting Cost-per-Click of paid search advertising is essential for performance marketers to allocate budgets that optimize marketing campaign returns. In this study, we perform a comprehensive analysis using various time-series forecasting methods to predict daily average CPC of keywords in the car rental sector. Our results show the power of statistical models on noisy keyword-level CPC time-series on short to medium horizons, only being outperformed by more complex neural networks on longer horizons. Advanced forecasting approaches leveraging competition did not yield significant accuracy improvements. Additional experiments with fine-tuned foundational models for time-series showed promising results, optimizing practicality and accuracy.

Keywords

Time-Series Forecasting, Search Advertising, Deep Learning, Time-Series Clustering, Graph Neural Networks, Foundational Models, Digital Advertising, Cost-per-Click

This work used infrastructure and resources funded by Fundação para a Ciência e a Tecnologia (UID/ECO/00124/2013, UID/ECO/00124/2019 and Social Sciences DataLab, Project 22209), POR Lisboa (LISBOA-01-0145-FEDER-007722 and Social Sciences DataLab, Project 22209) and POR Norte (Social Sciences DataLab, Project 22

1. Introduction

In 2005, Bill Gates predicted that *"the future of advertising is the internet"* (Lithgow 2005). This has proven remarkably accurate: today, over 58% of the global advertising budget is spent on digital advertising (Dentsu 2024). Among the various online advertising methods available, search advertising stands out as a vital tool for businesses to connect with customers at the precise moment they express intent. Consequently, the paid search advertising market reached a size of USD 282 billion in 2023 and is projected to grow at a compound annual growth rate of 9.4% over the next five years (Statista 2024). Google dominates this space, holding an impressive market share of over 89% (StatCounter 2024).

Cost-Per-Click (CPC) is one of the most relevant metrics in Google Ads, Google's search advertising platform. It determines how much advertisers pay each time a user clicks on their ad on the search results page. CPC in paid search is influenced by various factors, including competition, keyword relevance, ad quality, and the user's search context. While CPC is not a direct measure of profitability, accurate forecasts are crucial for advertisers to manage cost risks effectively and to optimize bidding strategies to efficiently allocate budgets to attractive keywords (Najafi-Asadolahi and Fridgeirsdottir 2014).

Current research highlights significant limitations in the granularity of CPC forecasting. Google provides only broad, weekly estimates of CPC for specific keywords, which can diverge significantly from actual CPC values (Oldenburg, Han, and Kaiser 2024). This discrepancy leads to considerable uncertainty and provides only limited value for advertisers. While Oldenburg, Han, and Kaiser (2024) achieved promising results in predicting daily CPC at advertiser level, their approach does not account for the variability in CPC values across individual keywords within an advertiser's portfolio. This highlights a critical research gap: the lack of accurate keyword-level daily CPC forecasts. Addressing this gap would enable advertisers to allocate advertising resources more effectively and efficiently by targeting keywords, thus supporting more granular and data-

driven decision-making (Zia and Rao 2019). Hence, our research aims to solve the business problem: *Provide advertisers with precise, keyword-level daily CPC forecasts for Google's paid search advertising market independent of Google's proprietary broad, weekly estimates*.

To achieve this, the study seeks to answer the following research questions (RQs):

> *RQ_1: Which state-of-the-art modeling approach achieves the highest accuracy for keyword-level CPC prediction in Google's competitive paid search advertising?*
>
> *RQ_2: Does the integration of select exogenous variables enhance the forecasting accuracy of keyword-level CPC predictions?*

To address these research questions, we employ advanced statistical methods, ML algorithms, and neural network-based forecasting models to establish a robust baseline of forecasts across short-, mid- and long-term horizons. To further enhance the results of our baseline, we will:

i. **Incorporate competition dynamics** from Google's bidding mechanism to account for the influence of market behavior on CPC. This includes clustering approaches that range from distance- and characteristics-based methods (A) to semantic keyword clustering (B), as well as auction-based feature extraction to model competitive dynamics (C). Finally, Graph Neural Networks (D) are used to capture complex interactions within the bidding environment.

ii. **Apply and fine-tune pre-trained Foundational Models** (E), ranging from large language models to specialized models for time-series, to assess their capability in forecasting CPC.

This thesis begins with the Literature Review, which establishes the foundational theory for CPC time-series forecasting. Next, the Exploratory Data Analysis (EDA) collects relevant insights from our dataset to inform the model setup. Once the initial models are configured and evaluated, the research questions are addressed. Subsequently, individual approaches (A-E) are introduced and discussed in dedicated chapters, aiming to enhance the baseline results. Ultimately, the results of all approaches are compared, leading to actionable recommendations for advertisers.

2. Literature Review

In this chapter, we lay the foundational theory required for CPC time-series forecasting. We cover the relevant background behind Google's paid search advertising business, special considerations in time-series forecasting as well as the current state of research within the domain of time-series forecasting.

2.1. Google's Paid Search Advertising Business

Our research focusses on Google's paid search business, which dominates the search engine advertising market with a market share of over 89% (StatCounter 2024). Our comprehensive literature review on Google's paid search advertising business encompasses key topics such as the role of search engines, the evolution of Google's paid advertising ecosystem and the underlying business model including the mechanics of its bidding system.

2.1.1. Search Engines

According to the Oxford English Dictionary (2024), *"a search engine is a computer program that searches the internet for information, especially by looking for documents containing a particular word or group of words"*. Search engines rely on sophisticated algorithms to evaluate factors such as keyword relevance, content quality, and user intent in real time. Subsequently, the most relevant information for each search query is ordered and presented on the search engine results pages (Laffey 2007).

This growing reliance on search engines offered Google and its competitors attractive monetization potentials. They generate revenue by selling advertisers the ability to display advertisements at the top or alongside regular unpaid search results. These paid advertisements look similar to organic results but are labeled as sponsored, giving advertisers a chance to appear more prominently when users search for relevant terms. Many companies are willing to invest considerable resources to achieve visibility on the first page of search results, as search engines organize the results of a user's query into pages, with the most relevant or popular links appearing

on the first page. In practice less than 1% of users reach the second page, meaning that appearing on the first page greatly increases a business's chances of being noticed by potential customers (Baluch 2023).

2.1.2. Google’s Search Engine Business Model

Google’s search engine advertising is essentially a two-sided market (Rochet and Tirole 2003) where Google acts as an intermediary connecting two distinct groups: users, who view and click on advertisements, and advertisers, which pay for those ads only when clicks are recorded. While users expect unrestricted and free access to search results, advertisers face a more competitive landscape. Visibility on search results is critical for advertisers to connect with consumers and drive revenue (Baluch 2023). Over the past decade, this dependency has grown further, as search engines have become central to the global digital economy (Bughin et al. 2011). Consequently, the global search advertising market grew with a compound annual growth rate (CAGR) of 16.7% in the past 5 years (Statista 2024).

Since information relevance depends on factors such as location, search history, and intent, Google utilizes the data it collects from a variety of sources within its ecosystem beyond its search engine, such as YouTube, Gmail, Google Maps, Android devices. This data is used to assess information relevance for each user individually to refine ad targeting, which ensures that advertisements are contextually relevant and personalized for individual users. Consequently, the search experience for users is optimized through personalization, which is why users are satisfied and return to Google to search for relevant information (Srinivasan 2020).

Simultaneously, these data network effects enhance the platform's attractiveness to information providers by expanding their market reach and enabling their ads to effectively target relevant audiences. This, in turn, increases Google’s market share in paid advertisements, leading to a virtuous cycle that explains Google’s unmatched position in the paid search advertisement market (Martens 2024).

2.1.3. Keyword Auctions

Advertisers set a maximum CPC in Google Ads for keywords they believe align with user search intent relevant to their business. The final CPC paid by advertisers is determined through an auction at the search query level, where Google matches user search queries to the most relevant keywords. For example, a user in San Francisco searches for "car rental" on Google. This search query is matched to keywords like "car rental," "car rental San Francisco," or "rental car San Francisco," which then compete in the auction for that specific query. The actual CPC an advertiser pays depends on their maximum CPC bid, the bids of competing keywords for the query, and the quality score (Evans 2009). The details of the auction mechanism are outlined below.

To determine the order of advertisers' keywords in the auction system, an Ad Rank is calculated. The ad rank is calculated as the product of two factors - the advertiser's maximum CPC bid and their quality score, calculated as:

$$Ad\ Rank = Maximum\ CPC\ Bid\ x\ Quality\ Score$$

Equation 1: Ad Rank

The Quality Score, which lies at the core of Google's bidding mechanism due to its involvement in the Ad Rank calculation, measures the relevance and quality of an advertiser's ads (Geddes 2014). Google does not disclose the exact formula for calculating quality scores, but it is generally calculated by evaluating the following three factors on a scale from 1 to 10:

- **Expected click-through rate (CTR):** An estimate of the likelihood that users will click on the ad.
- **Ad relevance:** The degree to which the ad aligns with the user's search intent.
- **Landing page experience:** The quality and relevance of the landing page users see after clicking the ad.

The keyword with the highest Ad Rank among all advertisers bidding on the same query holds the top position on the search results page of the respective search query. For the above-mentioned example of the search query "car rental" of a searcher located in San Francisco, the highest Ad Rank could, for example, be achieved by the keyword "car rental San Francisco" by advertiser EasyCar. This would cause its ad to be shown at the very top of the search result page - above other paid ads, but also above all unpaid search results (Google 2024a).

The Actual CPC that is paid by the advertiser EasyCar when the searcher that initiated the search query "car rental" clicks on its ad is determined by the following formula:

$$Actual\ CPC = \frac{Ad\ Rank\ of\ the\ next\ competitor}{Own\ Quality\ Score} + \$0.01$$

Equation 2: Actual CPC

Google's ad ranking system therefore combines advertisers' CPC bids with quality scores to ensure that higher-quality, more relevant ads secure top placements on search engine results pages (SERPs) (Google 2024a). At first glance, this system benefits multiple stakeholders: advertisers gain better returns on their ad spend, users see more relevant ads, and Google maximizes its click-through revenue (Jansen, Liu, and Simon 2013). Moreover, this approach fosters a virtuous cycle. Satisfied users are more likely to return to Google for future searches, increasing ad clicks and reinforcing Google's network effects (Dan and Davison 2016).

On closer inspection however, the system has its drawbacks. The divergence between the CPC bid and the actual CPC paid, coupled with the undisclosed calculation of quality scores, turns Google's bidding mechanism into a "black box" for outsiders. This makes it challenging for advertisers to derive accurate forecasts, forcing them to rely on Google's proprietary predictions. By developing reliable methods independent of Google's tools, advertisers could allocate budgets more effectively, optimize performance, and benchmark their campaigns in a competitive auction landscape.

2.1.4. Complexity in CPC Forecasting on Keyword Level

CPC is heavily influenced by budget allocation, because of diminishing returns in clicks for increased budgets, as visible in Figure 1.

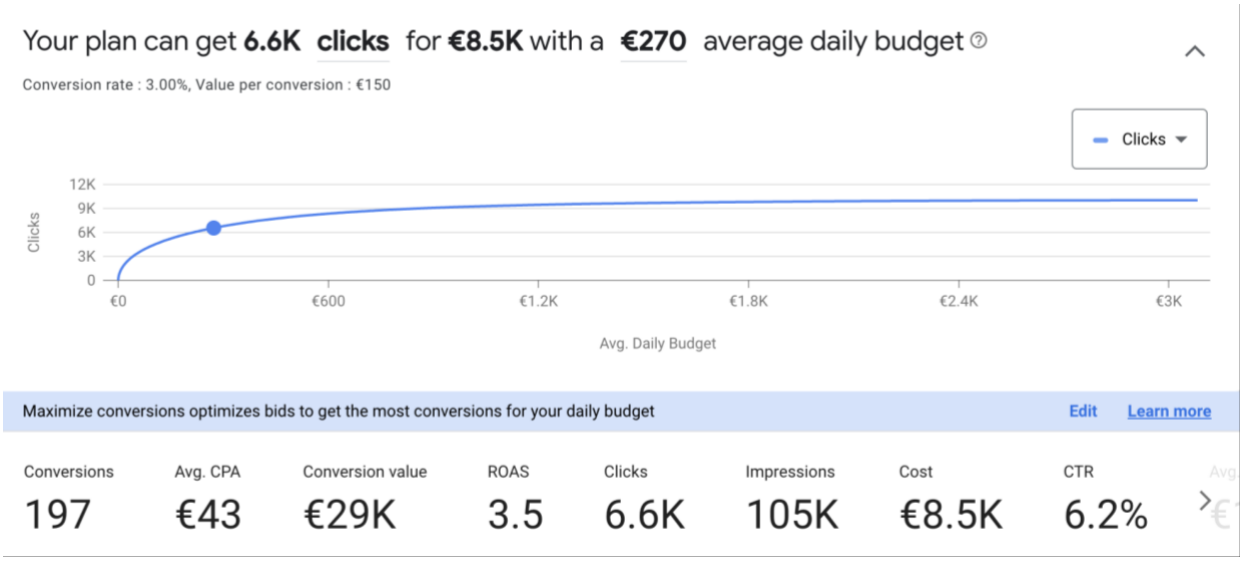

Figure 1: Clicks versus daily ad budget show diminishing returns for increased budget - Screenshot of Google Ads Keyword Planner

Advertisers generally allocate one budget at campaign level and let Google set daily maximum CPC bids on individual keywords to optimize for their campaign's objective. Campaigns often encompass multiple, or even all keywords an advertiser is bidding on and typically span several weeks. Therefore, Google has a lot of influence on daily keyword-level CPC by possibly shifting budgets from day to day to arrive at desired campaign-level CPC. Hence, accurate predictions on keyword level are challenging, as fluctuations based on Googles allocation algorithms can induce a significant amount of noise. This may be a deliberate design by Google to make it harder for advertisers to predict daily CPCs at the keyword level, which would safeguard Google's revenue optimization as advertisers collectively rely on Google's proprietary budget distribution algorithms (Lanier 2023).

2.2. Special Considerations in Time-Series Forecasting

Since the business problem of this research focuses on improving the accuracy of time-series forecasts for CPC at the keyword level in Google's competitive advertising market, it is essential to clarify the unique characteristics of time-series data.

Time is the defining characteristic that distinguishes time-series data from other types of datasets. This temporal structure is both a limitation, which requires careful preservation of sequence, as well as a source of valuable insights, providing opportunities to analyze patterns such as trends, seasonality, and temporal dependencies. Unlike static datasets, time-series data exhibit sequential dependencies, where each observation is influenced by prior values. This dependency necessitates preserving temporal order during both training and evaluation to prevent data leakage and ensure realistic forecasting (Kontopoulou et al. 2023).

Time-series data is typically characterized by trends, seasonality, non-stationarity and noise. Trends capture the overall direction of the data over time, seasonality reflects fixed and repeating patterns, non-stationarity indicates that statistical properties such as mean and variance change over time, and noise represents random fluctuations that can obscure underlying patterns. To address these characteristics, techniques such as differencing, logarithmic transformations, or decomposition into trend, seasonal, and residual components are often required for effective modelling (Kontopoulou et al. 2023).

Beyond these patterns, more irregular behaviours like cyclic patterns without fixed frequencies and transient effects influenced by external conditions, such as the initial disruptions caused by Covid-19, can emerge (Kontopoulou et al. 2023). As in traditional machine learning, the presence of outliers – data points that deviate significantly - and white noise, which refers to random, patternless variations in data, add further complexity to the modeling process. This highlights the importance of robust anomaly detection and preprocessing techniques, which will be further explored in Chapter 3 as part of our Exploratory Data Analysis.

Exogenous variables are another unique aspect of time-series forecasting. These variables, external to the time-series and its values, can enhance model accuracy by providing additional context about factors influencing the target variable. However, their effective use requires precise alignment with the target variable to avoid data leakage. Allowing the model to access future information may result in overly optimistic performance during testing and poor real-world applicability. This contrasts with static datasets, where temporal alignment is not a concern. Time-series evaluation also differs substantially from traditional approaches, using methods such as rolling-origin splits and walk-forward validation to ensure that the temporal order of data is preserved, and future information does not influence past predictions. Metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), Mean Absolute Percentage Error (MAPE), and Symmetric Mean Absolute Percentage Error (SMAPE) are commonly used to evaluate forecasting and regression accuracy, including in time-series contexts, due to their ability to quantify prediction errors (Kontopoulou et al. 2023).

Lastly, the *No-Free-Lunch* theorem emphasizes that no single forecasting method can optimally handle all types of problems, including time-series data. As Wolpert and Macready (1997) highlight, the performance of any algorithm is inherently problem-dependent, meaning improvements for one dataset or context often come at the expense of others. In the realm of time-series forecasting, this underscores the importance of individually selecting models that align with the specific characteristics of the dataset rather than relying solely on a model's prior performance in different contexts.

2.3. State of Current Research in Time-Series Forecasting

This chapter provides an overview of the current state of research in time-series forecasting by highlighting key methodologies across statistical, machine learning (ML) and deep learning models. Each approach is examined for its strengths, limitations, and relevance to our forecasting challenge.

2.3.1. Statistical Models

Statistical models have long served as the foundation of time-series forecasting, particularly in scenarios with limited data and well-defined temporal structures. Building upon the ARIMA framework, the SARIMA model introduces seasonal parameters to model periodic patterns alongside its core components—autoregressive terms (AR), moving averages (MA), and differencing for stationarity. Its suitability for univariate datasets with clear temporal patterns and ease of implementation (Khashei, Bijari, and Hejazi 2012) makes SARIMA a reliable baseline, particularly for shorter time-series or datasets where computational efficiency and interpretability are critical. Recent studies, such as those by Hamoudia et al. (2023) have reaffirmed SARIMA's strong performance as a baseline model, especially in well-structured, small-scale datasets. Furthermore, its success in outperforming ML models on shorter datasets has been demonstrated in comparative evaluations (Makridakis, Spiliotis, and Assimakopoulos 2018). However, its reliance on strict parametric assumptions, such as stationarity, poses challenges when dealing with multivariate, non-linear, or hierarchical data. Despite these limitations, SARIMA continues to be a robust and computationally efficient choice in applications that prioritize clarity and reliability (Cerqueira, Torgo, and Soares 2019). This makes it a reliable baseline model for our research problem.

2.3.2. Machine Learning Models

ML methods, such as Random Forests, Support Vector Machines, and Gradient Boosted Trees, have expanded the scope of time-series forecasting by capturing non-linear patterns and handling complex datasets without relying on strict parametric assumptions. Among the Gradient Boosted Trees algorithms, XGBoost has emerged as a leading model due to its scalability, flexibility, and computational efficiency (Chen and Guestrin 2016). As a boosting-based ensemble method, it combines the outputs of shallow decision trees to deliver accurate predictions, with interpretability derived from aggregated feature importance scores. By restructuring time-series data into a supervised learning format, XGBoost effectively models non-linear relationships and interactions

and fits separate models for each target time step. Its ability to integrate future and static exogenous variables, but not historical ones, makes it well-suited for forecasting scenarios where such inputs are available and relevant. However, when dealing with dynamic or real-time exogenous variables whose future values are not known, additional feature engineering and careful data preparation are required, limiting its direct applicability in such cases. Its efficiency in commercial applications and strong performance in forecasting competitions further underscore its value for our study (Makridakis, Spiliotis, and Assimakopoulos 2018).

2.3.3. Deep Learning Models

Deep learning methods have advanced time-series forecasting by addressing the limitations of traditional statistical and ML approaches, particularly for complex, multivariate datasets. Early models like Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU) were foundational in enabling sequence modelling and handling long-term dependencies in temporal data (Bandara, Bergmeir, and Smyl 2020). More recent architectures, such as Neural Basis Expansion Analysis for Time Series (N-BEATS), N-BEATS with exogenous variables (NBEATSx), Neural Hierarchical Time Series (NHITS), and Temporal Fusion Transformers (TFT), have set new benchmarks in forecasting accuracy (Oreshkin et al. 2020).

N-BEATS(x) employs a decomposition-based hierarchical architecture that separates time-series data into trend and seasonal components, excelling in univariate and multivariate forecasting tasks (Oreshkin et al. 2020). NHITS extends N-BEATS(x) by incorporating improvements for hierarchical time-series forecasting, achieving state-of-the-art results in datasets with multiple levels of granularity. The Temporal Fusion Transformer (TFT) introduces attention mechanisms, enabling the identification of the most critical time periods and features for forecasting. It integrates multiple components, such as variable selection networks, LSTM encoder-decoder layers, and temporal self-attention layers, making it particularly effective for handling complex and multivariate datasets (Lim et al. 2020). These advancements make deep learning models

valuable tools for time-series forecasting, particularly in scenarios with diverse temporal patterns and high-dimensional data.

To provide a comprehensive analysis, this study incorporates a variety of forecasting methodologies, each tailored to different aspects of the problem. This ensures a balanced evaluation of statistical, ML, and deep learning approaches. In the subsequent chapters, the selected models reflect this diversity: SARIMA is used as the statistical benchmark for its reliability with simpler datasets and clear temporal patterns. XGBoost is included for its ability to model non-linear relationships and handle complex datasets. Lastly, N-BEATS(x), NHITS, and TFT represent state-of-the-art deep learning models, designed to capture intricate patterns and dependencies in hierarchical and multivariate time-series data.

3. Exploratory Data Analysis

3.1. Purpose and Objectives of Exploratory Data Analysis

The Exploratory Data Analysis (EDA) phase serves as a foundational step in this study, aimed at uncovering essential patterns and insights within the CPC data aggregated at a daily level. Our analysis uses a proprietary dataset, which contains key digital marketing metrics for specific, anonymized advertisers and examines them in detail. The uniqueness of this dataset lies in its detailed coverage of the complete competitive landscape on keyword-level, offering a rare glimpse into the bidding behaviors that are typically not publicly disclosed by companies.

The primary objectives of the EDA are twofold and interconnected. Firstly, the analysis seeks to understand the development of CPC by uncovering seasonal trends and recurring patterns, thereby enhancing the comprehension of CPC time-series forecasting. Secondly, it aims to evaluate the relationships between various features and the target variable, CPC, to assess the viability of incorporating exogenous variables into predictive models. By addressing these objectives collectively, the EDA provides a comprehensive foundation for building robust and accurate forecasting models.

Additionally, given the inherent challenges associated with keyword-level data—such as missing values due to tracking inconsistencies—the EDA will implement data filtering techniques to ensure completeness and reliability. By focusing on advertisers and keywords with high data integrity, the analysis ensures that the dataset is robust enough to support accurate forecasting. Ultimately, this chapter evaluates the dataset's readiness for forecasting purposes, laying the groundwork for subsequent modeling efforts.

3.2. Overview of the Dataset

3.2.1. Detailed Description of the Dataset

The initial dataset employed in this research comprises keyword-level advertising performance metrics for advertisers engaged in Google Ads campaigns from 2019 to 2023. It includes daily records of key metrics such as ad costs, clicks, and impressions. This dataset captures significant fluctuations in advertising dynamics, including the impacts of the COVID-19 pandemic on online marketing costs, thereby providing a robust foundation for forecasting models and offering rich insights into competitive bidding behaviors. The data is recorded at the keyword level, with entries for every keyword an advertiser allocates budget to, allowing for granular analysis of paid search activities.

3.2.2. Source of Data and Methods of Data Collection

This dataset was provided by Grips Intelligence, a European e-commerce research platform specializing in online advertising intelligence. Grips Intelligence collects advertising expenditure data globally through partnerships with various data providers. The data, aggregated and anonymized using first-party analytics tools such as Google Analytics under a "give to share" model, ensures a comprehensive and reliable dataset that enables in-depth analysis while safeguarding the confidentiality of individual advertisers.

The data captures advertisers' bidding behaviors, encompassing metrics such as ad spend, impressions, and clicks, which were pre-processed to calculate daily average CPC.

The full dataset collected this way encompasses information from over 249,000 advertisers, accounting for approximately 9.2% of Google's total advertising revenue. It demonstrates a strong correlation ($r > 0.9$) with the advertising revenue reported by Alphabet, Google's parent company (Oldenburg, Han, and Kaiser 2024).

3.2.3. Data Filtering and Preprocessing

3.2.3.1. Industry Selection: Focus on Car Rental Sector

Grips Intelligence gathers data across a wide range of industries. The objective of this study is to provide advertisers with an accurate prediction of CPC by incorporating, among other factors, data on keyword level competition. To maintain a manageable scope for this thesis, it is essential to focus on a single industry, allowing for a more in-depth analysis of the competitive dynamics within that specific domain. This study selected the car rental industry for three key reasons.

Firstly, the car rental market is mainly dominated by a limited number of major players, resulting in a highly competitive landscape. Secondly, renting a car is a relatively homogeneous service, as most companies essentially offer the same product—a vehicle. This contrasts with industries such as fashion, where offerings can vary widely, from specialized items like underwear to seasonal products such as winter jackets. Lastly, in the car rental industry, we assume search queries are more generic, as the product (a car) and its specifications, such as the brand or model, often take precedence over the identity of the rental company.

Before finalizing the dataset, a comprehensive preliminary dataset titled *"car_rental_original"* was utilized. This dataset contained 41,098,492 records and 16 features, encompassing detailed advertising metrics. These features included view ID, date, keyword, ad matched query, ad destination URL, device category, ad cost, ad clicks, impressions, transactions, transaction revenue, exchange rate in USD, search type, ad cost in USD, revenue in USD, and advertiser domain. This dataset served as the foundation for refining the focus of the study and optimizing the predictive modeling process.

3.2.3.2. Addressing Duplicate Data

The initial dataset we received from Grips Intelligence contained duplication, with around 34% of records repeated due to overlapping data entries across various advertising campaigns and dates. To address this issue, a systematic deduplication process was implemented. A combination of the

unique identifiers view ID (a unique ID assigned to a reporting view in Google Analytics), date, keyword and advertiser was utilized to identify redundant entries. Among duplicate entries, the view ID corresponding to the highest revenue value was retained. This purification step was crucial in preventing the inflation of metric calculations and ensuring the authenticity of the dataset. Consequently, the final dataset maintained high accuracy and reliability, essential for subsequent analytical procedures. By ensuring that only one view ID was used per keyword, the study guarantees that time-series tracking is not erroneously influenced by data collection methods.

3.2.3.3. Ensuring Data Completeness: Thresholds and Imputation

In addition to addressing the view ID issue, the *"car_rental_original"* dataset revealed the presence of missing values, with over 99% of the records exhibiting gaps in the target variable, CPC. Given the critical importance of complete time-series data for accurate forecasting, a stringent completeness threshold of 98% was applied during the data preprocessing phase. This threshold ensured that each time-series included at least one CPC record per date for each advertiser and keyword on 98% of days, minimizing biases stemming from incomplete data entries. Setting a 98% completeness threshold strikes a balance between minimizing missing data and ensuring enough time-series are kept to capture a relevant revenue share.

To handle the remaining 2% of incomplete data, linear interpolation was employed as the imputation method. This approach balanced simplicity and effectiveness, preserving temporal consistency without introducing undue complexity.

Although the number of keywords was reduced from approximately 140,000 to 78 due to this stringent preprocessing, the downsized dataset still represents 54% of the overall revenue, ensuring its relevance and representativeness for advertisers focusing on high-value keywords (see Figure 2). While our analysis primarily targets CPC as an independent metric, the inclusion of revenue as

a proxy for keyword importance helps retain focus on keywords with significant commercial impact, which aligns with the strategic priorities of performance marketing.

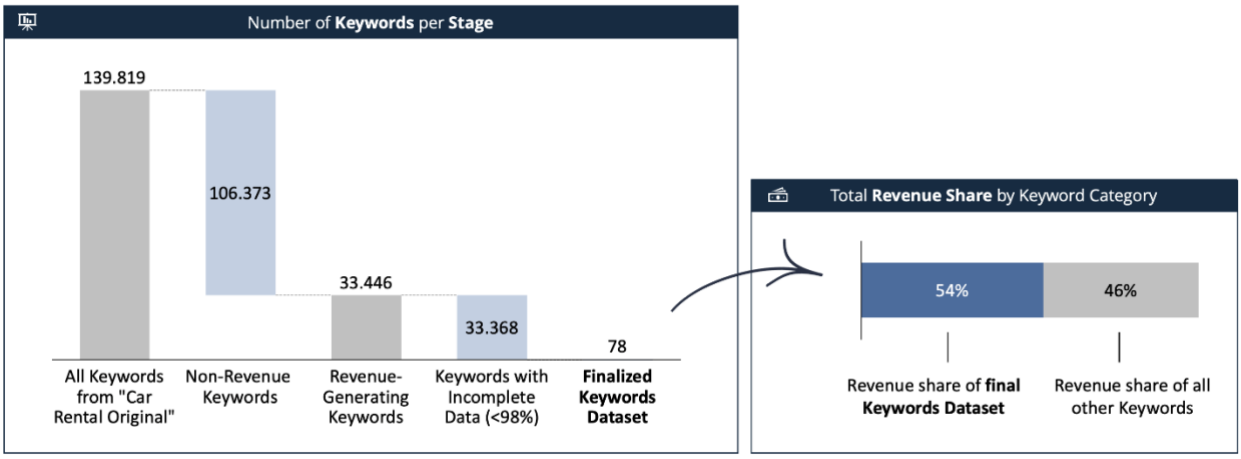

Figure 2: Keyword Filtering and its Impact on Remaining Revenue Share in the Dataset

3.2.3.4. Device Category Filtering and Outlier Management

To further maintain consistency across the dataset, it was filtered to include only keywords searched on desktop devices. This choice was motivated by the fact that desktop searches have been around since the early days of search engines. They represent a more established and consistent category compared to mobile searches, which only became significant with the emergence of modern smartphones and still tend to vary more. By concentrating on desktop data, we use the most impactful data, thereby enhancing the relevance of the forecasting models.

Furthermore, we also removed rows with revenue but no ad costs, as every click generating revenue must incur ad costs. This way, the study safeguards the dataset against the influence of anomalies, ensuring that the analysis remains grounded in reliable and consistent information.

3.2.4. Structure and Feature Overview of the Final Dataset

The final dataset consists of 42.432 records with 13 features, providing daily performance metrics for 78 keywords. Each record represents a unique combination of keyword-level identifiers (unique_id), date (ds), and advertising performance metrics, such as revenue, adclicks, transactions, and impressions. The temporal scope of January 2021 to June 2022 was chosen due to limitations in data quality. Extending beyond this range was not feasible as less complete

keyword data was available for longer periods, which could compromise the accuracy and reliability of the analysis. The table below provides a summary of the features, their data types, and brief descriptions:

Feature	Data Type	KPI type	Description
unique_id	Categorical	-	A unique identifier representing keyword-level per advertiser
ds	Temporal	-	The date corresponding to each record
revenue	Continuous	Absolute	Total revenue generated in USD
adclicks	Discrete	Absolute	Total number of ad clicks recorded
transactions	Discrete	Absolute	Count of completed transactions
impressions	Discrete	Absolute	Number of ad impressions served
adcostusd	Continuous	Absolute	Total expenditure on advertisements in USD
ROAS	*Continuous*	*Relative*	*Return on ad spend, calculated as revenue divided by ad cost*
CPC	*Continuous*	*Relative*	*Cost-per-click, computed as ad cost divided by ad clicks*
CR	*Continuous*	*Relative*	*Conversion rate, representing transactions per ad click*
CTR	*Continuous*	*Relative*	*Click-through rate, calculated as ad clicks per impression*
TotalMarket_CPC	*Continuous*	-	*Market-level average CPC used for benchmarking*
DomainLevel_CPC	*Continuous*	-	*Advertiser-specific average CPC used for competitive analysis*

Table 1: Feature overview of final dataset (italic = engineered feature)

3.3. Feature Engineering for Predictive Modeling

3.3.1. Development of Derived Features Descriptive Statistics

To enhance analytical insights and capture the multifaceted dynamics of advertising performance, this study developed several derived metrics, as outlined in Chapter 3.2.5. Key metrics include Return on Ad Spend (ROAS), Conversion Rate (CR), Click-Through Rate (CTR), TotalMarket_CPC and DomainLevel_CPC.

In addition to these primary metrics, time-related features, such as the day of the week and month of the year, were extracted from the existing temporal feature "ds". These temporal variables provided additional context for understanding performance patterns over time.

While not all the newly created features were directly incorporated into the forecasting models, their development enriched the exploratory data analysis.

3.3.2. Transformation and Scaling of Features

During the EDA phase, no scaling or normalization techniques were applied to the dataset. Most features were retained in their original scales to preserve interpretability and maintain the intrinsic relationships within the data.

However, in the time-series forecasting models implemented later, tools such as Nixtla's Robust Scaler were applied to the input data. This step was necessary to standardize the range of features, enhancing model performance and convergence. Detailed justifications for this approach are provided in the methodology chapter of this study.

3.4. Descriptive Statistics

3.4.1. Target Variable

Our forecasting objective focuses on predicting CPC. To identify the most effective modeling approaches and optimize outcomes, it is essential to conduct a thorough analysis of the target variable's behavior within our dataset.

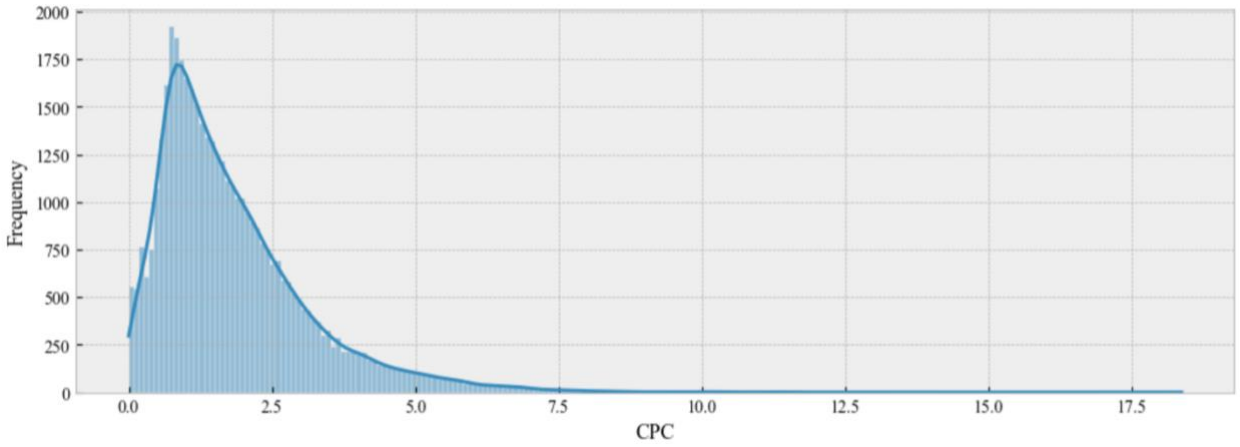

Figure 3: Histogram of CPC values across the dataset

The total distribution of values of CPC, shows a right-skewed distribution, visible in Figure 3. Most values lie between 0.95 (25%-percentile) and 2.47 (75%-percentile) with an average value

of 1.94 and a median value of 1.59. CPC across all time-series has a variance of 2.02 and a standard deviation of 1.42.

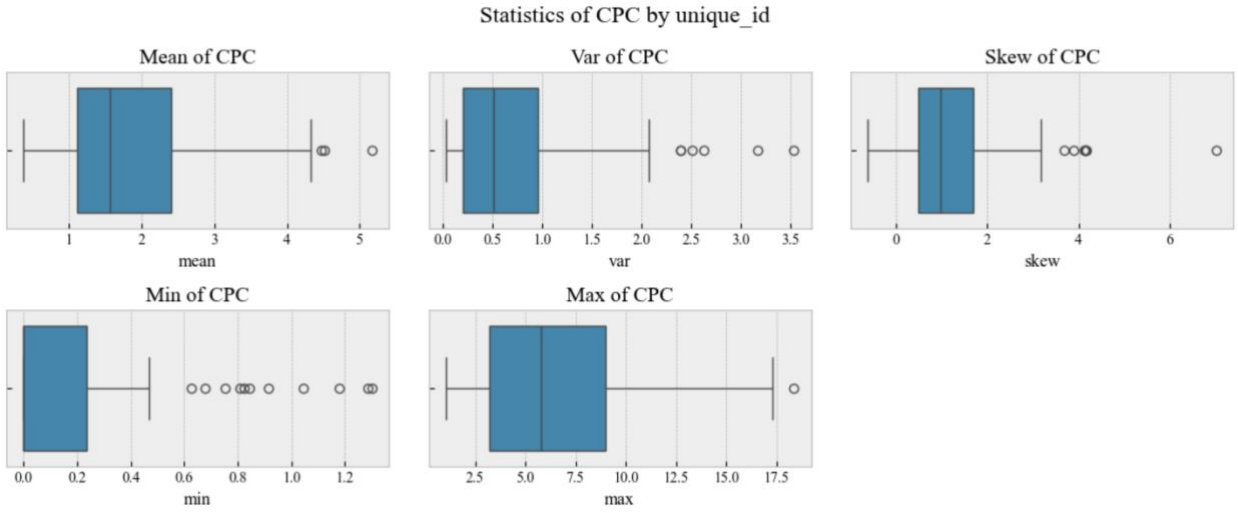

Figure 4: Box plots of different statistics across CPC time-series

Another way of analyzing our target is looking at the different distribution of statistics across our keyword time-series. Here it becomes apparent that statistics differ to a big extend between time-series (see Figure 4). This indicates that we are dealing with quite heterogenic data, which might make inference from one to another time-series less promising (Fujimoto et al. 2024).

While time-series forecasting static distributions are a good first indicator of unique characteristics, it is especially interesting how values fluctuate over time. Figure 5 illustrates the average CPC across all advertisers, offering a clearer view of these trends.

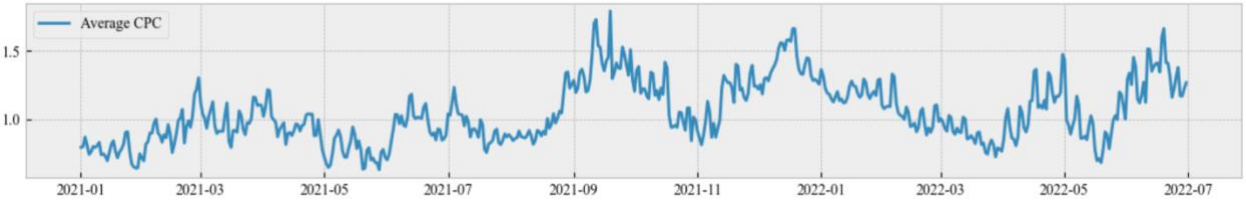

Figure 5: Average CPC per day across all domains

An upward trend in average CPC across the dataset can be observed in Figure 5. There are occasional spikes, however they are hard to attribute to specific market events.

To further investigate the composition of our time-series further we can perform additive time-series decomposition. Time-series decomposition aims to extract constituent latent subseries, for

example trend, seasonality and remaining noise, in order to better understand the underlying patterns in the data, isolate meaningful components, and thereby inform choice of models and parameters (West 1997). Time-series decomposition requires a parameter for the interval of seasonality. We experimented with different intervals like 7 (weekly), 4 (quarterly) or 12 (monthly) based on market knowledge and visual observations. The weekly interval was found to perform best. When applying it to our average CPC data, the following subseries are observed:

Trend

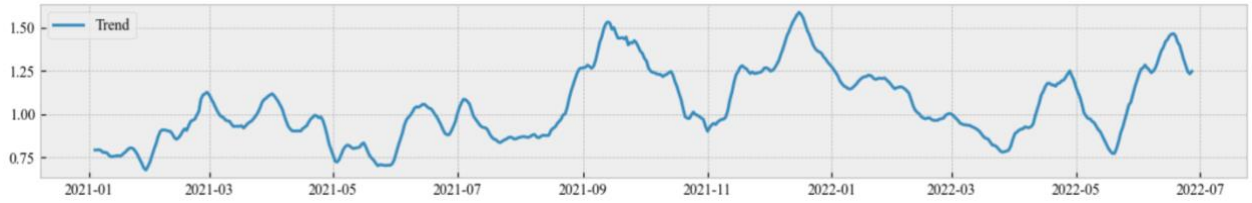

Figure 6: Trend in average CPC across all domains

The decomposition results in the trend seen in Figure 6, showing an increase of average CPC across the dataset. The trend is not linear and has quite a few local optima and minima in between.

Seasonality

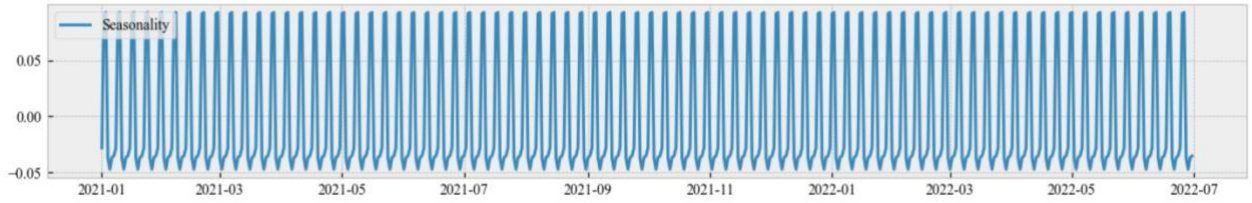

Figure 7: Seasonality in average CPC across all domains

A seasonality is picked up, oscillating between -0.05 and 0.10 in weekly intervals. This should be interpreted cautiously, as the additive time-series decomposition used in this study may detect seasonality even when none clearly exists. An indicator supporting this, might be the low scale of seasonal change visible in Figure 7.

Noise

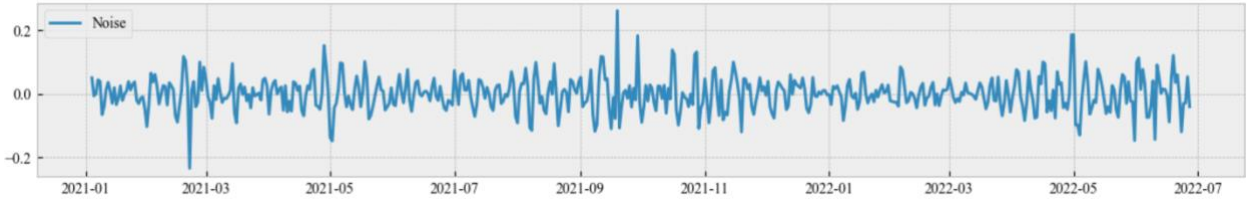

Figure 8: Noise in average CPC across all domains

After addition of trend and seasonality, Figure 8 shows that there is still a lot of noise ranging from less than -0.2 up to around 0.25 to explain the time-series values. Our models will aim to find patterns to accurately predict this noise.

Overall, it is important to note that the identified trend in CPC is very volatile. The noise is significantly larger in magnitude than the seasonality, suggesting that the seasonality likely constitutes only a small and insignificant portion of the overall series values. This suggests that the timeseries will be more difficult to predict as many techniques rely on identifying and leveraging components such as trend and seasonality.

3.4.2. Covariates

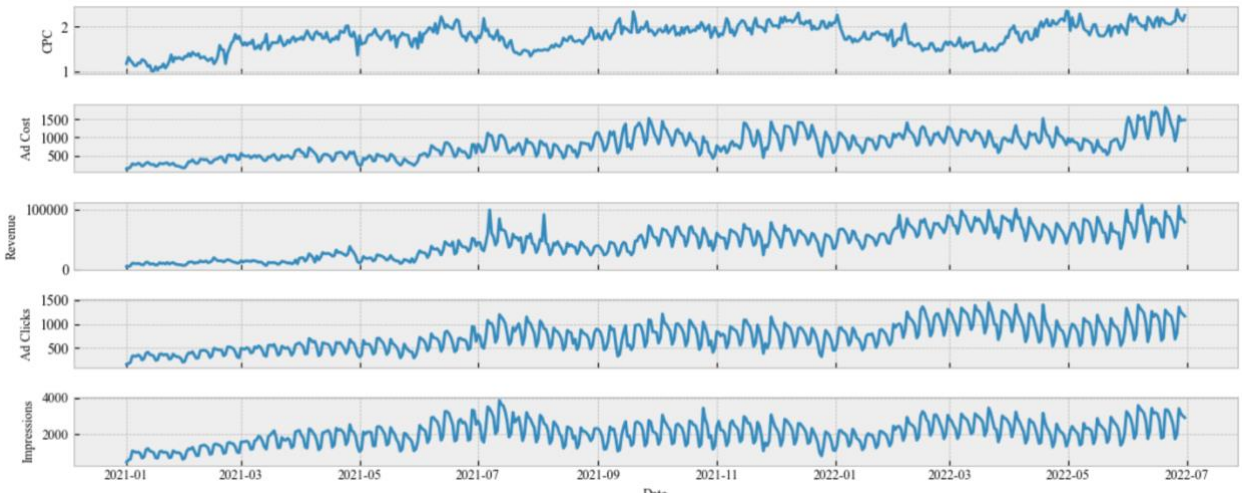

Figure 9: Average of CPC and Covariates

Next to CPC, our dataset contains additional numeric features, which we will refer to as numeric covariates. To better understand the covariates dynamics over time, we analyze their average compared to the average CPC in Figure 9.

Visually, the covariates seem to be following a similar trend as the one extracted from CPC in Figure 6. This is unsurprising for Ad Cost and Ad Clicks, as CPC is inherently derived from their ratio. However, there are many patterns visible in the covariates that are not picked up in our target variable. For example, the covariates have a more significant weekly seasonality when compared to CPC. This, while no definite proof, might indicate that the covariates are not significant in predicting future CPC, especially in short horizons where weekly seasonality does not match.

3.5. Correlation Analysis

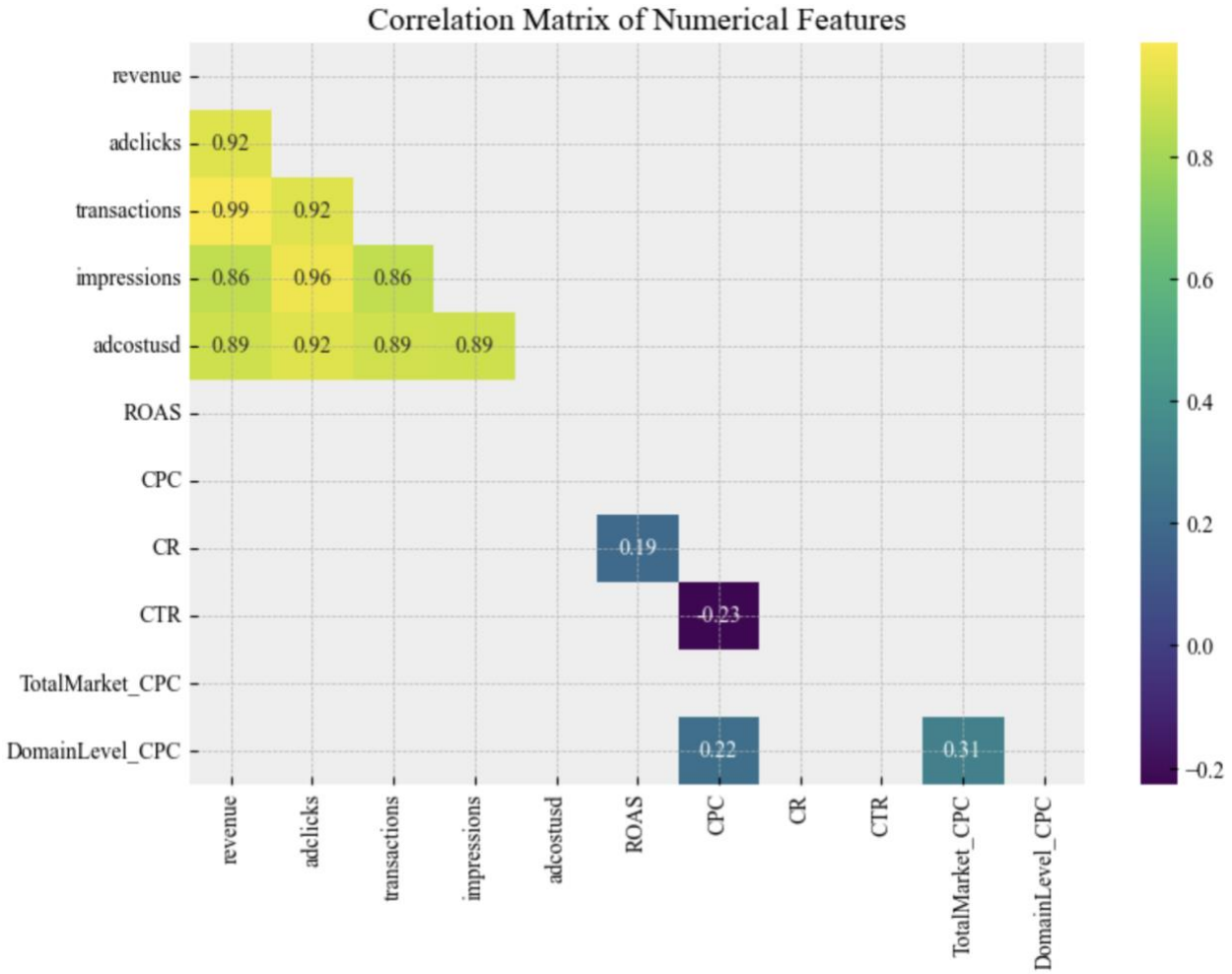

Figure 10: Correlation matrix of numerical features

Correlations serve as a valuable metric for assessing the interactions between variables and evaluating the predictive potential of features. We used Pearson's correlation coefficient on our full dataset to identify linear relationships. Our correlation matrix reveals very high correlation between adclicks, transactions, impressions and adcostusd. This is expected, as clicks require

impressions, transactions require clicks, and Google only charges costs based on the number of clicks. No clicks mean no cost for the advertiser. Apart from that Figure 10 reveals a moderate correlation of 0.19 between ROAS and CR, which is to be expected as higher conversion should ceteris paribus lead to higher returns. Our target CPC surprisingly has a moderately high negative correlation of (-0.23) with CTR, which might seem counterintuitive at first. Typically, one would expect that as CTR increases, advertisers might be willing to pay more per click, driving CPC higher. This might hint at the importance of the ad quality score in Google's algorithm, since ads with a higher CTR seem to have lower CPC.

To conclude, the correlation matrix does not provide a highly correlated feature to use as predictor for CPC when looking at the full dataset. In our experiment we will still try and see if models can pick up more subtle interplay, since low Pearson correlation does not mean there is no interplay or predictive value between variables at all. Relationships might still exist nonlinear or in the combined interaction of variables, both of which Pearson's correlation coefficient cannot measure.

3.6. Summary of Findings

3.6.1. Key Insights

In our EDA, we filtered our large dataset down to a subset of 78 complete keywords while keeping more than 54% of total revenue (see Figure 2) generated through paid search advertising across the 5 advertisers included. This step was taken to ensure that the time-series meet a 98% completeness threshold, which is crucial for minimizing the impact of missing data and improving the reliability and accuracy of the time-series forecasting results. The high data quality in this subset allows us to perform time-series forecasting on the most relevant keywords of the data.

The time-series decomposition of our target variable CPC showed a big amount of noise obstructing any clear seasonality in the data. Also, the variance is greater than the mean, indicating a wide spread of CPC values. There are no linear correlations in the data that are relevant for our predictive task. The big differences in individual CPC statistics of time-series indicates that we are

dealing with quite heterogenic data, which makes approaches relying on inference between time-series less promising.

3.6.2. Implications for Modeling

Given these insights, the focus of our research will be accessing how we can predict data that is probably hard to predict given the volatile trend, low seasonality, high amount of noise and high variance. Additionally, we will try to uncover nonlinear, hidden and or high-dimensional relationships between variables by combining them in nonlinear models.

4. Methodology

This chapter details the experimental setup used to evaluate the performance of various forecasting models on the CPC dataset. It outlines the standardized pipeline (used across all models to ensure consistency), the input configurations tested, and the forecast horizons chosen for evaluation. Additionally, we describe the cross-validation framework and evaluation metrics employed to ensure robust and reliable comparisons between the models.

4.1. Experimental Setup

This study aims to compare the performance of the different forecasting models mentioned in Chapter 2.3 on our dataset. For any experiment that aims to compare different approaches, having the same inputs and preprocessing is of utmost importance to ensure a fair comparison. We secure consistency in our experiment by using one joined data- and machine-learning-pipeline for all models we explored.

We structured this pipeline in a progressive way, starting with simpler univariate models and advancing to more complex architectures that incorporate additional features. This approach allows us to attribute changes in performance to either the inclusion of new features or architectural modifications. As illustrated in Figure 11, we tested four different sets of input features, selected based on the capabilities of each model:

(a) **Univariate**: Univariate models use only past values of the target variable to predict future values of the target.

(b) **Univariate w/ time features**: Our first progression from univariate models is including time features, namely as day-of-week and month-of-year.

(c) **Univariate w/ abs. keyword_KPI features**: Secondly, we added absolute exogenous features. This includes clicks, impressions, transactions and revenue which were included in the original dataset.

(d) **Univariate w/ rel. keyword_KPI features**: Lastly, we have added relative exogenous features, such as CTR, ROAS and CR, which were derived from the original data.

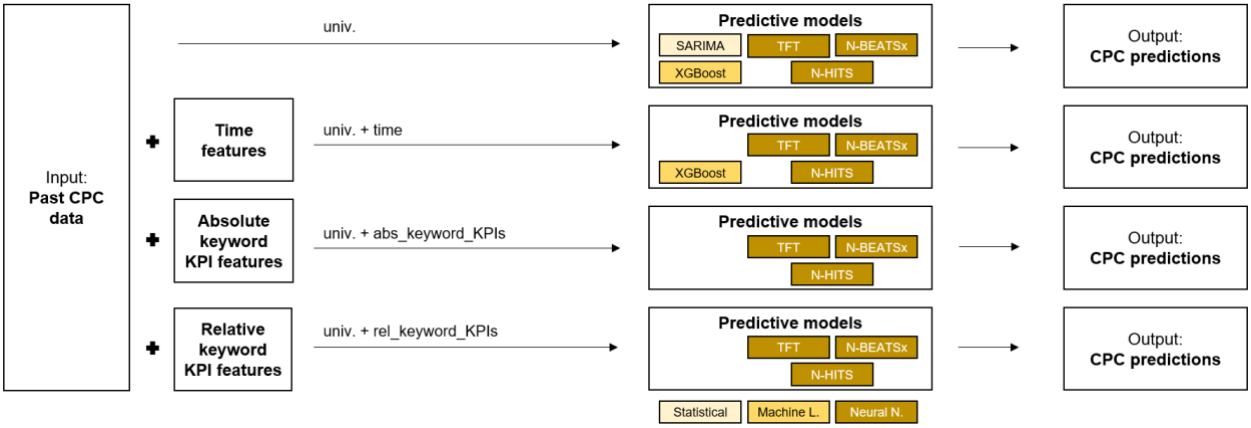

Figure 11: Overview of experimental setup

Separating absolute and relative features allows for a clearer understanding of their individual contributions to prediction performance. Absolute features capture raw activity levels inherent in the dataset, while relative features reflect proportional metrics derived from them. Testing them independently avoids information overlap and helps determine their unique impact. Hardware constraints hindered us from exploring both feature types simultaneously and combining them with individual feature selection was not feasible due to the significant computational burden of testing numerous feature combinations across multiple model types. However, if both configurations independently would have improved predictive performance, we would have further explored ways to overcome these limitations.

4.1.1. Forecast Horizons

We decided to conduct forecasts for 14-day, 30-day, and 60-day periods. These forecast horizons are chosen for their relevance in business decision-making, specifically in optimizing advertising budgets and campaign strategies, and are consistent with the methodology outlined in Oldenburg, Han, and Kaiser (2024). By covering short, medium, and longer-term horizons, this approach allows us to capture the temporal dynamics and challenges inherent in CPC prediction.

Each horizon offers unique value from a business perspective:

- **14-day horizon**: This short-term forecast is critical for managing immediate tactical adjustments to campaigns, such as responding to rapid changes in market conditions, seasonal trends, or competitor activity. Therefore, operational business units rely on accurate short-term forecasts to dynamically allocate budgets, refine bidding strategies, and ensure alignment with real-time performance goals.
- **30-day horizon**: The medium-term forecast of 30 days is in alignment with monthly planning. Consequently, it allows advertisers to evaluate and adjust campaign strategies based on expected performance over the upcoming month. This includes the establishment of monthly performance benchmarks as well as the synchronization of marketing efforts with broader business objectives.
- **60-day horizon**: Our longest forecast provides valuable insights for strategic decision-making, such as planning large-scale campaigns, adjusting quarterly budgets, and anticipating future market dynamics. It helps businesses identify emerging trends, mitigate risks, and optimize resource allocation for a sustained competitive advantage in the digital advertisement industry.

4.1.2. Error Metrics

For evaluation of the performance of different forecasting models across varying time horizons, we use MAE and SMAPE as our primary evaluation metrics. Both metrics were chosen for their complementary strengths in assessing forecasting accuracy, particularly in the context of CPC data, which exhibits dynamic and skewed value distributions.

- **MAE** was selected due to its simplicity and interpretability, as it measures the average absolute difference between predicted and actual values in the same units as the target variable CPC. This makes MAE highly interpretable and particularly useful for understanding the absolute error magnitude in CPC forecasts, which is directly relevant to

business decisions, such as budget allocation and bid optimization. Unlike percentage-based metrics, MAE is not influenced by the scale of the target values. This makes it robust when CPC magnitudes vary.

- **SMAPE** was chosen to complement MAE by introducing a relative error perspective, which accounts for the magnitude of both the actual and predicted values. Unlike the Mean Absolute Percentage Error (MAPE), SMAPE is symmetric, preventing the metric from disproportionately penalizing overpredictions or underpredictions. This property is especially important for CPC data, where small actual values could inflate errors in MAPE, leading to misleading evaluations. SMAPE therefore ensures that the metric remains fair across the range of CPC values.

4.1.3. Temporal Cross Validation

To ensure a robust evaluation of forecasting models, we implemented a tailored cross-validation framework created by Nixtla which is designed specifically for time-series data. Unlike traditional cross-validation, which assumes independence between data points, time-series cross-validation respects the inherent temporal dependencies, which ensures that information from the future does not influence past predictions.

Depending on the design of the cross-validation framework for the different modelling approaches, we apply it to both validation and test phases. For neural network and ML models, cross-validation is employed during training alongside early stopping to fine-tune hyperparameters, monitor performance, and mitigate overfitting. When testing on the unseen test set, three evaluation windows are used to assess generalization performance. In contrast, for SARIMA, cross-validation is applied exclusively during the test phase to evaluate predictive accuracy, as its parameter selection is based on internal statistical criteria rather than iterative training.

We employ an expanding window strategy for our ML and neural network models, where the size of the training set incrementally increases with each iteration. This strategy allows the model to

learn from progressively larger historical datasets. For validation, we adopt a sliding window method with a predefined window size set to half the input size. This choice ensures a scalable pipeline across various forecasting horizons with two validation sets for each configuration.

In the test phase, we implemented cross-validation to derive robust evaluation metrics across several test set folds. This way, we can achieve reliable and generalizable insights into model performances across different model architectures which all use the same folds for testing.

4.2. Implementation Details

This section outlines the implementation details of the models used in this study, categorized into statistical models, ML models, and neural network models. For most of these models, the Nixtla library was used, as it offers efficient and standardized tools for time-series forecasting. Each section below details the configurations, preprocessing steps, and hyperparameter tuning processes. This explains how the models were adapted to address the characteristics of the CPC dataset, including its low seasonality and high noise.

4.2.1. Statistical Models

For the implementation of the SARIMA model, we utilized Nixtla's AutoARIMA to automatically identify the best model parameters. While SARIMA traditionally requires manual specification of its autoregressive (p), differencing (d), moving average (q), and seasonal components (P, D, Q), the AutoARIMA implementation automates this process, ensuring an optimal parameter selection based on the dataset. The only parameter requiring manual adjustment is the season length, which was set to **7** to represent a weekly cycle. Although the EDA indicated that seasonality in CPC data is not strongly visible and is overshadowed by noise, slight weekly patterns driven by variations in user behavior, ad performance, and bidding strategies make this choice reasonable.

4.2.2. Machine Learning Models

For the XGBoost implementation, two configurations are used:

(a) relying solely on lagged features of the target variable and

(b) incorporating date-based variables.

Both configurations use lagged values (1, 7 and 14 days) as predictors to capture short-term and weekly patterns in the CPC data. A grid search is performed to optimize hyperparameters, such as learning rate and number of estimators, ensuring the model is well-suited to the dataset.

The first configuration, which excludes exogenous features, trains the model using only lagged predictors. The second configuration incorporates exogenous variables in the form of date-based features - day of the week and the month of the year - to capture calendar-driven patterns that influence CPC dynamics. However, due to XGBoost's limitations in Nixtla, only time-based features can be integrated as exogenous variables. This restriction prevents the inclusion of relative and absolute keyword KPIs.

The optimal parameters identified during the grid search are used to train the final models. This setup ensures the XGBoost models effectively capture potential temporal and calendar-driven seasonalities of the CPC dataset.

4.2.3. Neural Network Models

Our selected neural network models already outlined in Figure 11 are capable of handling historical and future exogenous variables and are therefore applied for all four approaches. To preprocess the input features, we apply robust scaling, which is the standard setting for Nixtla's neural network models. It normalizes our input features to reduce the influence of outliers and to ensure consistency across features. Since many of the settings in our experimental setup are uniformly applied to all neural network models, they are outlined in Table 2.

Parameter	Value	Explanation
Input size	4x the forecast horizon (2x for 60-day horizon)	Ensures sufficient data points for training and validation across all three horizons
Loss function	Mean Absolute Error (MAE)	Prioritizes the minimization of large prediction errors and aligns with our primary evaluation metric
Learning rate	{0.1, 0.01, 0.001}	3 different learning rates balances grid search efficiency and effectiveness
Batch size	2	Required due to GPU memory constraints
Early stopping	5 validation checks with no improvement	Prevents overfitting and reduces unnecessary computation
Validation checkpoints	Every 5 steps	Regular monitoring of validation loss to guide optimization
Number of samples	27	Exhaustive exploration of the hyperparameter grid (max grid size: 3 x 3 x 3)
Maximum training steps	50	Balances convergence and computational efficiency

Table 2: General settings for all neural network models

In addition to these uniform settings, model-specific hyperparameters were included in the grid search. These are detailed below for each model.

TFT

Since the TFT excels at capturing complex temporal relationships and interactions between features through its attention mechanism, our hyperparameter tuning focuses on the number of attention heads to explore varying levels of temporal dependency modeling. Additionally, we test different hidden sizes to strike a balance between the model's ability to learn detailed patterns and the risk of overfitting.

NHITS

For the NHITS, we tune two hyperparameters specific to its hierarchical architecture. The pooling kernel size is finetuned to explore trade-offs in down sampling efficiency and computational cost. Frequency downsampling is tested with different configurations to optimize the model for different temporal granularities to match different seasonality patterns.

NBEATSx

The NBEATSx model can utilize polynomial basis functions to model trends in the data. These functions enable the model to capture trends with varying levels of complexity. In our experiments, we test different values for the *n_polynomials* parameter, which decide if and to what degree the model can represent both simple linear trends and more complex higher-order dynamics, such as quadratic or cubic trends.

5. Results

In this chapter, we first evaluate and compare the performance of statistical, ML, and deep learning models across short-term (14 days), mid-term (30 days), and long-term (60 days) horizons. Specifically, we analyze the statistical model SARIMA, the ML model XGBoost, and deep learning models TFT, N-HITS, and NBEATSx. We consider scenarios with and without exogenous variables, including time features and keyword-specific KPIs (absolute and relative), to assess their impact on forecasting accuracy. Subsequently, we analyze the parametrization of SARIMA and feature importance of TFT to gain insights into how the best performing models use the data provided in their prediction.

5.1. Model Performance Comparison

The results, presented in detail in Table 3, reveal clear performance trends across the evaluated models and forecasting horizons. SARIMA achieves the best accuracy for short-term (14 days) and mid-term (30 days) forecasting, with an MAE of 0.491 and 0.529 and an SMAPE of 0.263 and 0.300, respectively. However, its performance declines significantly over the long-term (60 days) where it falls behind other models with an MAE of 0.645 and an SMAPE of 0.352.

For XGBoost, the inclusion of time features provides slight improvements on short-term horizons, with an MAE of 0.511 (14 days), in contrast to the generally negative impact observed for time features in neural network models. Despite these improvements, XGBoost still underperforms compared to other models across all horizons, with its long-term performance (MAE 0.721, SMAPE 0.397) showing the greatest gap.

In contrast, neural network models, particularly the TFT perform more stable over longer horizons. While TFT is slightly outperformed by SARIMA on the short-term horizon, TFTs performance declines only marginally over time. For instance, TFT achieves an MAE of 0.501 (14 days), 0.596

(30 days), and 0.614 (60 days), effectively closing the performance gap with SARIMA and surpassing it on longer horizons.

		Features Included								
			Keyword KPIs		14 Days		30 Days		60 Days	
	Model	Time	Absolute	Relative	MAE	SMAPE	MAE	SMAPE	MAE	SMAPE
Stat. Model	SARIMA	No	No	No	**0.491**	**0.263**	**0.529**	**0.300**	0.645	0.352
ML Models	XGB univ.	No	No	No	0.528	0.285	0.570	0.339	0.705	0.387
	XGB univ.	Yes	No	No	0.511	0.279	0.568	0.343	0.721	0.397
Neural Network Models	TFT univ.	No	No	No	0.501	0.265	0.596	0.346	0.614	0.336
	N-HITS univ	No	No	No	0.497	0.265	0.574	0.328	0.618	0.343
	N-BEATSx	No	No	No	0.502	0.268	0.582	0.330	0.637	0.352
	TFT univ.	Yes	No	No	0.655	0.326	0.863	0.431	0.775	0.396
	N-HITS univ	Yes	No	No	0.544	0.293	0.666	0.395	0.788	0.398
	N-BEATSx	Yes	No	No	0.548	0.303	0.763	0.468	0.822	0.442
	TFT univ.	No	Yes	No	0.511	0.272	0.597	0.340	0.594	0.331
	N-HITS univ	No	Yes	No	0.543	0.294	0.959	0.354	0.759	0.394
	N-BEATSx	No	Yes	No	0.541	0.284	0.668	0.356	0.754	0.393
	TFT univ.	No	No	Yes	0.522	0.275	0.581	0.331	**0.586**	**0.328**
	N-HITS univ	No	No	Yes	0.552	0.290	0.692	0.374	0.780	0.409
	N-BEATSx	No	No	Yes	0.531	0.282	0.679	0.357	0.777	0.405
	TFT univ.	No	Yes	Yes	0.507	0.272	0.581	0.331	0.599	0.334
	N-HITS univ	No	Yes	Yes	0.576	0.294	0.674	0.358	0.779	0.402
	N-BEATSx	No	Yes	Yes	0.561	0.288	0.687	0.361	0.793	0.407
	TFT univ.	Yes	Yes	Yes	0.542	0.306	0.634	0.367	0.617	0.349
	N-HITS univ	Yes	Yes	Yes	0.587	0.305	0.666	0.360	0.796	0.410
	N-BEATSx	Yes	Yes	Yes	0.584	0.300	0.709	0.371	0.842	0.412
	TFT univ.	Yes	No	Yes	0.520	0.289	0.648	0.360	0.596	0.337
	N-HITS univ	Yes	No	Yes	0.592	0.307	0.663	0.360	0.802	0.409
	N-BEATSx	Yes	No	Yes	0.584	0.306	0.680	0.384	0.818	0.434

0.123 best model per time horizon

Table 3: Forecasting results comparison – Baseline

The inclusion of absolute and relative keyword KPIs provides performance improvements exclusively for TFT. Notably, relative keyword KPIs alone yield the most significant gains, with TFT delivering the best overall performance on the long-term horizon (60 days), with an MAE of 0.586 and an SMAPE of 0.328. Interestingly, combining absolute and relative keyword KPIs leads to worse results than using relative KPIs alone.

5.2. Parametrization of SARIMA

SARIMA demonstrates strong performance on the 14-day and 30-day horizons, consistent with findings from prior studies, where statistical models have outperformed more advanced machine learning models. The parametrization of its different terms show in Table 4 can indicate possible reasons:

Parameter	Description	Value
p	The number of autoregressive terms (AR) in the non-seasonal part of the model	0
d	The degree of differencing in the non-seasonal part of the mode	2
q	The number of moving average terms (MA) in the non-seasonal part of the model	1
P	The number of autoregressive terms (AR) in the seasonal part of the model	1
D	The degree of seasonal differencing	7
Q	The number of moving average terms (MA) in the seasonal part of the model	1
constant	Indicating if a constant term is included in the model (0 means no constant)	0

Table 4: SARIMA parameter configuration

The auto-selected parameters indicate the model's reliance on second-order differencing (d = 2) to address non-stationarity and a high seasonal differencing order (D = 7) to smooth repeating patterns. While these parameter choices enable SARIMA to capture immediate dynamics effectively, they might introduce challenges for long-term forecasting. High D values can lead to excessive smoothing, which would erase critical seasonal information necessary for nuanced predictions over a 60-day horizon. This potential limitation aligns with the model's weaker performance on extended horizons, where the complexity of CPC dynamics exceeds the capacity of purely statistical methods.

5.3. Feature Importance of TFT Model

A key feature of our benchmarked neural network models is their capacity to integrate exogenous covariates. These features might provide contextual information, which SARIMA, restricted to historical CPC values, does not have access to. Our results for the 60-day horizon - where NNs beat SARIMA - show the TFT model performing best when leveraging the relative KPIs CTR, ROAS, and CR. The Variable Selection Networks (VSNs), which dynamically assign importance

to inputs across time steps, offer the possibility to analyze the TFTs feature importance. We are going to analyze this average feature importance as well as temporal attention to get a sense of the KPIs contribution to the forecasting objective.

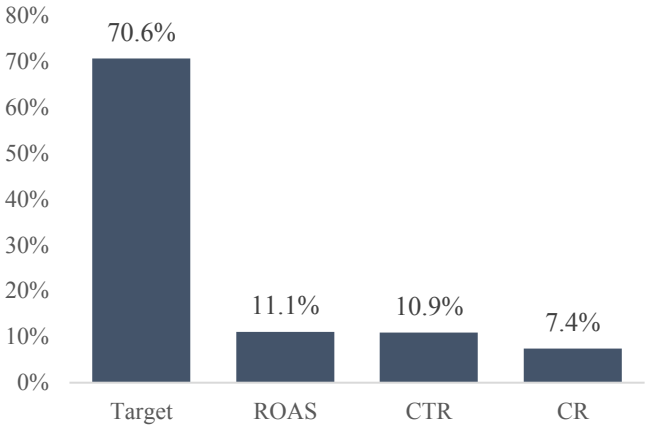

Figure 12: Average feature importance of TFT

The average feature importance across all time steps of the input sequence shows the TFT still heavily (70.6% importance) relies on historical CPC values for its predictions. The relative KPIs split the remaining importance with ROAS and CTR having very comparable values of around 11% and CR lacking behind with 7.4% average feature importance across all timesteps.

A prior study from Lim and colleagues (2020) suggests that external variables provide critical context for metrics, especially over extended horizons where historical patterns alone may lack sufficient explanatory power. While we cannot confidently attribute causality, our results align with this.

6. Discussion

The findings presented in the previous chapter provide valuable insights into the research questions regarding the accuracy of different modeling approaches and the impact of exogenous variables on keyword-level CPC prediction in competitive paid search advertising.

> *RQ_1: Which state-of-the-art modeling approach achieves the highest accuracy for keyword-level CPC prediction in competitive paid search advertising?*

The results show that model selection requires a nuanced approach specifically considering forecast horizon. SARIMA demonstrates strong performance for short-term (14 days) and mid-term (30 days) horizons. Its accuracy deteriorates significantly for long-term (60 days) forecasts, where it falls behind other approaches.

In contrast, neural network models, particularly the TFT model, display greater stability across all horizons. While less accurate than SARIMA for short- and mid-term predictions, TFT shows only a minimal performance decline as the forecast horizon extends. For long-term (60 days) predictions, TFT therefore outperforms SARIMA and other models and establishes itself as the most robust solution for long-term forecasting tasks.

The performance of XGBoost remains weaker compared to SARIMA and TFT, especially on longer horizons. Nevertheless, its results improve slightly when time features are included, in contrast to neural networks, where time features tend to have a negative impact.

To answer our research question RQ_1, SARIMA is most effective for short-term and mid-term predictions, while TFT, with its stable performance, emerges as the most reliable approach for long-term CPC forecasting.

> *RQ_2: Does the integration of relevant exogenous variables enhance the forecasting accuracy of keyword-level CPC predictions?*

The impact of exogenous variables differs depending on the model and type of features used, with notable effects observed exclusively for TFT among the neural networks. For mid-term and long-term horizons, the inclusion of relative keyword KPIs improves accuracy, which demonstrates their relevance as informative exogenous inputs. While the addition of absolute keyword KPIs also enhances performance compared to the baseline, the improvements are less significant than those achieved with relative keyword KPIs. However, for other neural networks, the inclusion of absolute and relative keyword KPIs, worsens performance.

Interestingly, time features show a mixed effect. While they slightly enhance performance for XGBoost on the short horizon, they generally have a negative impact on neural network models, including TFT. This highlights the importance of aligning exogenous variables carefully with the model architecture and the data characteristics to avoid introducing noise or redundancy.

In conclusion, relative keyword KPIs are the most beneficial exogenous variables, however exclusively for long-term forecasting with TFT. These results undermine the need for careful feature selection, as some exogenous variables can degrade performance instead of improving it.

7. Conclusion

7.1. Summary of Findings

The goal of the first part of our research was to understand the dynamics of CPC in paid search keyword auctions and benchmark different models to forecast keyword-level CPC on different horizons. Our EDA revealed CPC keyword time-series data to be highly heterogenous. The timeseries decomposition showed only low levels of trend and seasonality, which are further diluted by a high amount of noise in our data. This already indicated that we are dealing with hard to predict data.

Our benchmarks showed the statistical SARIMA outperforming the significantly more complex ML (XGBoost) and neural network models (TFT, NHITS, NBEATS) on 14-day and 30-day prediction horizon. This reinforces select studies showing that for many time-series datasets, especially on shorter horizons and when dealing with limited data, high quality statistical benchmarks should be preferred because of their simplicity, explainability and reliability. On the longest 60-day horizon, the dynamics shifted, with the TFT model leveraging exogenous variables—specifically, relative keyword KPIs derived from the original data—achieving the best performance in our prediction task.

7.2. Limitations

This study has faced several limitations, which we have categorized into methodological limitations and data limitations.

7.2.1. Methodological Limitations

Our filtered dataset focuses exclusively on the car rental industry because it benefits from a high degree of homogeneity in its product offerings. Although we suppose that this homogeneity makes keyword bidding patterns more stable and predictable, generalizability of the findings to the broader Google Ads market is limited. Additionally, only a selection of time-series forecasting models that had demonstrated promising results in prior research were used.

7.2.2. Data Limitations

Careful data curation was necessary to preprocess our data for the modeling stage. This involved excluding a substantial number of keywords that did not meet the required completeness threshold, potentially leading to an incomplete representation of market dynamics. Although most revenue-generating keywords were included, this narrowed down subset might be prone to overlooking valuable nuances in the broader dataset. As less complete keyword data was available for longer periods, the study was restricted to an 18-month timeframe.

Data limitations are also induced by Google, as its "black box" algorithms behind the CPC calculation obscures critical factors that influence CPC forecasting. Google's Quality Score significantly impacts CPC, yet its components are unavailable, which prevents the models from accounting for it. Additionally, Google's budget pacing algorithms introduce unpredictable fluctuations in daily spending patterns, as they dynamically optimize ad delivery, which induces noise that can hardly be captured by the forecasting models.

Last, advertisers' budget allocations and campaign durations directly influence CPC outcomes. Our dataset lacks information on individual campaign budgets and timelines, which limits our ability to incorporate these factors into our forecasting models.

7.3. Outlook

Building on the limitations identified in this study, several opportunities for future research can be explored to enhance the accuracy and applicability of keyword-level CPC predictions in competitive online advertising contexts.

While the car rental industry provided a homogeneous dataset with relatively stable keyword bidding patterns, future research should expand to more diverse industries. Examining sectors with varied product offerings and competitive dynamics could improve the generalizability of the findings and offer deeper insights into CPC behavior across the broader Google Ads market. Second, addressing the exclusion of incomplete keywords due to the completeness threshold

would allow for a more comprehensive representation of market dynamics. Techniques such as imputation strategies or advanced modeling methods capable of handling incomplete time-series could help incorporate these keywords, capturing additional nuances within the dataset.

Extending the analysis to a longer timeframe would enable the identification of long-term trends, seasonality, and potential structural changes in CPC dynamics. This could provide a more holistic understanding of temporal patterns and improve the robustness of forecasting models.

Future studies could explore additional forecasting approaches beyond those selected here. Evaluating hybrid models, which combine the strengths of statistical, ML, and deep learning techniques, may further improve prediction accuracy. Additionally, integrating more exogenous variables, such as budget allocations or broader market conditions, could provide valuable context and insights, particularly given the strong influence of budget on CPC dynamics.

Incorporating data related to Google's Quality Score as exogenous variables could enhance model performance. Since obtaining this data directly from Google may be challenging or unfeasible, proxy variables reflecting key components of the Quality Score could be carefully explored and developed to mitigate the uncertainty caused by the lack of direct data. As CPC is heavily influenced by the budget allocated to individual keywords or entire campaigns, access to data on advertisers' budget plans and campaign timeframes could potentially improve CPC forecasting. This approach appears more straightforward for an individual advertiser compared to the complex task of developing proxy variables for the key components of Google's Quality Score.

8. Motivation for Deep Dives

Inspired by Oldenburg, Han and Kaisers (2024) paper *"Interpretable Deep Learning for Forecasting Online Advertising Costs: Insights from the Competitive Bidding Landscape"* that showed competition as a relevant factor of advertiser-level CPC, we assume to be able to improve upon the benchmark results from this by incorporating keyword-level competition into our predictive models. When trying to do so, we are presented with the following challenge: The naive approach of identifying competition in auctions by looking at "co-bidding" of different advertisers on the same keyword (for example advertiser A and advertiser B both bidding on "car rental") is not possible, as only five keywords experience co-bidding in our dataset. Therefore, more advanced approaches to model competitive dynamics will be introduced and elaborated upon in the following deep dives:

Chapter A ("A Time-Series Clustering Approach") will adapt a similar methodology to Oldenburg, Han and Kaiser (2024) to assess if findings on advertisers-level CPC can be generalized to keyword level predictions. Evolving from that, Chapter B ("Semantic Keyword Clustering") clusters keywords semantically, assuming semantically similar keywords compete in the same keyword auctions. "Auction based Feature Extraction" (Chapter C) aims to leverage concrete evidence of auction participation to engineer and implement relevant features. In Chapter D ("Leveraging competitive dynamics using Graph Neural Networks") Graph Neural Networks are used adopted to predict using a graph structure representing competitive dynamics between keywords.

These four deep dives are meant to collectively answer the research question:

> *RQ_3: Can advertisers benefit from incorporating information on competition into their keyword-level CPC forecast?*

As the neural network models showed good results, especially on later horizons, and have the capability to leverage exogenous features, we will continue using them in our deep dive experiments.

After going into the competition-based approaches to improve baseline results, we will also experiment with the application and fine-tuning of foundational models for time-series prediction (E), as the success of foundational language models introduces questions about the transferability to time-series data, specifically to our hard-to-predict keyword level CPC data. This experiment is aimed to answer the question:

> *RQ_4: How does the forecasting accuracy of foundational models compare to state-of-the-art benchmarks for predicting daily keyword CPC in competitive online advertising markets?*

In Chapter 9 we will summarize our findings from deep dives A to E and provide our answers to research questions RQ_3 and RQ_4. We will also suggest implications for advertiser, informed by a combination of all results, in this last Chapte

A. A Time-Series Clustering Approach

A.1. Introduction

Oldenburg, Han, and Kaiser (2024) demonstrated that clustering advertisers based on their aggregated CPC time-series data allows for the detection and extraction of competition-informed patterns relevant for forecasting. These patterns were included as features into their predictive models and outperformed the benchmark across all tested horizons.

In this deep dive of our research, we want to answer RQ_3 - whether competition data can help advertisers to improve keyword-level CPC forecasts - by applying and finetuning Oldenburg, Han, and Kaiser's (2024) research approach to keyword level data. By clustering keywords that exhibit similar temporal patterns, we aim to directly capture competitive behaviours at the keyword-level (where competition occurs), using the same dataset and configuration used for the baseline models.

This research chapter applies two distinct methods in time-series clustering. The first method involves clustering time-series based on pairwise distances, with two variations: (Ia) using Euclidean distance as distance metric and (Ib) applies Dynamic Time Warping (DTW) as an alternative, more sophisticated distance metric for the distance calculation. The second method (II) clusters time-series based on extracted static characteristics, which capture the time-series' underlying properties.

Time-series clustering serves as a foundation to tune our baseline model configurations in three scenarios for each time-series clustering method. In the first scenario, our global baseline models are trained only on the data of each cluster instead of being trained on all data available. In the second scenario, features derived from the respective clustering method are added as exogenous variables to the models, but a global model trained on all 78 time-series is used. In the third scenario, both previous scenarios are combined.

A.2. Theoretical Background

A.2.1. Time-Series Clustering Overview

Clustering is a fundamental unsupervised learning method in machine learning, where data points within a cluster are more similar to each other than to points in other clusters. Traditional clustering algorithms, such as K-Means, operate on static data vectors by minimizing the sum of squared distances between those vectors and their assigned cluster centroids (Jin and Han 2010). The iterative process involves assigning data points or vectors to the nearest centroid (assignment step) and updating centroids as the mean of assigned points (update step). This happens until the assignments and therefore the mean of the assigned points converges (MacKay 2003).

However, when clustering time-series data, this straightforward method faces unique challenges due to the temporal nature of the data. Unlike static vectors, time-series data consists of sequential observations, where the order and temporal dependencies matter. These characteristics make traditional clustering methods less effective without appropriate adaptations. In this research chapter, we focus on two common time-series clustering techniques: distance-based time-series clustering and characteristics-based time-series clustering (Javed, Lee, and Rizzo 2020).

A.2.2. Distance-based Time-Series Clustering

Distance-based time-series clustering is a widely used method in time-series analysis because it considers the temporal ordering of observations. Two prominent distance metrics in this domain are Euclidean distance and Dynamic Time Warping (DTW). Euclidean distance is the most straightforward distance metric to measure similarity between time-series. It calculates the exact point-by-point differences between corresponding observations in two sequences. Therefore, it assumes that the sequences are perfectly aligned and of equal length. This introduces a significant limitation: Euclidean distance is highly sensitive to temporal misalignments or varying speeds within the sequences. For instance, if two series exhibit similar patterns but are shifted in time, the Euclidean distance will fail to recognize their similarity (Berndt and Clifford 1994).

To address the limitations of Euclidean distance, DTW was introduced as a more flexible similarity metric. DTW aligns sequences non-linearly by warping the time axis, which allows for comparisons even when sequences exhibit temporal shifts or varying speeds (Berndt and Clifford 1994). The DTW distance between two sequences is computed as the minimal cumulative cost required to align them. This is achieved by constructing a cost matrix and finding the optimal warping path that minimizes the total alignment cost. The centroid is computed using a DTW barycenter averaging method, which iteratively refines a representative sequence by minimizing the DTW distance to all series in the cluster (Tavenard 2017). While DTW provides a more robust measure of similarity, it is computational more complex than Euclidean distance, especially for large datasets or long sequences (Berndt and Clifford 1994).

A.2.3. Characteristics-based Time-Series Clustering

Characteristics-based clustering, in contrast, is a method that transforms time-series data into static representations. This is done by extracting a set of structural characteristics of each sequence that represent the statistical, structural, and dynamic properties of each time-series.

The research conducted by Bandara, Bergmeier and Smyl (2017) discovered that a smaller number of extracted features proved to be very effective for characteristics-based time-series clustering instead of exploiting all possible feature extractions. Their smaller number of extracted time-series features still provided a comprehensive representation of statistical, structural, and dynamic properties. A similar feature selection approach was also followed by Oldenburg, Han, and Kaiser (2024) and yielded favorable results for characteristics-based CPC time-series clustering on advertiser level.

Characteristics-based clustering can offer several advantages over dynamic clustering techniques. By abstracting time-series into static descriptors, computationally expensive temporal alignments such as DTW are avoided and a concise summary of each series' underlying structure that can be

more interpretable than distance-based clustering is provided (Huang et al. 2016; Bandara, Bergmeir, and Smyl 2017). However, this method disregards the temporal progression within the time-series, which can be a disadvantage for downstream tasks such as forecasting. If the temporal behavior of a competitor keyword time-series that is added as a covariate does not correlate with the CPC of the target time-series, its value as a covariate could be limited (Schiller et al. 2024; Shering, Alonso, and Apostolopoulou 2024).

A.2.4. Evaluation Metrics for Optimal Amount of Clusters

To determine the optimal number of clusters, the two most widely used direct validation methods are the Silhouette Score and the Elbow Method. The Silhouette Score quantifies the quality of clustering by measuring cohesion within clusters and separation between clusters. The overall Silhouette Score is the mean of the Silhouette Score across all points, with values ranging from −1 (poor clustering) to +1 (perfect clustering) (Rousseeuw 1987). The Elbow Method evaluates the distortion metric, also known as the within-cluster sum of squared distances (WCSS), for a range of cluster numbers. By plotting WCSS against the number of clusters, the Elbow Method identifies the "elbow point" where adding more clusters results in diminishing returns in reducing WCSS (Thorndike 1953). Combining both techniques supports the selection of the optimal number of clusters that balances compactness and separability (Patel, Sivaiah, and Patel 2022).

A.2.5. Comparison of Cluster Assignments

When multiple clustering methods are applied, it is of interest to compare how similar they assign data points or time-series to clusters. The Adjusted Rand Index (ARI) and Normalized Mutual Information (NMI) are robust metrics for this purpose. ARI measures the agreement between clustering assignments and adjusts for random assignment, with values ranging from -1 (complete disagreement) to 1 (perfect alignment). NMI quantifies shared information between clustering assignments, normalized to account for differences in cluster sizes or entropy. It ranges from 0 (no overlap) to 1 (identical clusters) (Vinh, Epps, and Bailey 2009).

A.2.6. Relevance of Clustering for Time-Series Forecasting

When time-series clustering is combined with time-series forecasting, it is commonly applied to select which time-series to feed into global forecasting models together. These global models compute global weights based on all intra-cluster time-series, which allows the model to learn relationships specific to each cluster. This approach reduces the risk of noise induction from unrelated time-series and mitigates the risk of overfitting compared to local models, as the weights are not adjusted to each individual time-series (Oriona, Manso, and Fernández 2023; Laurinec and Lucká 2017).

Another application of time-series clustering for time-series forecasting is the extraction of features from clustering which capture shared patterns of the cluster's time-series. This can be particularly useful in highly interdependent environments, where individual trends often follow group behaviors. Oldenburg, Han and Kaiser (2024) discovered that extracting covariates based on clustering for CPC forecasting, specifically the CPC of other advertisers, resulted in improved model performance in predicting advertiser-level CPCs.

A.2.7. Hypotheses and Research Gaps

While Oldenburg, Han and Kaiser (2024) found out that CPC time-series of advertisers in the same cluster helped to improve model performances on advertiser level, clustering-based covariate extraction on keyword level for CPC forecasting is still to be explored. Moreover, training global models only on the time-series of the same cluster was not explored in the specific context of CPC forecasting on keyword level. To address these gaps, this study proposes the following hypotheses:

> $H_{A.1}$: Training the models only on the time-series of each cluster derived from the respective clustering method enhances the forecasting accuracy of CPC at keyword level.
>
> $H_{A.2}$: Competition covariates derived from the respective clustering method enhance the forecasting accuracy of CPC at keyword level.

A.3. Methodology

To ensure comparability and to answer our research hypotheses, we adhere to the ceteris paribus principle by maintaining the model architecture, forecasting horizons, input data and evaluation criteria the same as in the baseline methodology described in Chapter 4 across all experimental setups. This also includes the settings outlined in Table 2 and the hyperparameter grid specified in Chapter 4.2.3. All clustering and feature engineering approaches are exclusively applied to the training dataset to prevent data leakage.

Given the ability of the TFT, N-HITS, and N-BEATSx models to handle historical exogenous features as well as being trained globally with only intra-cluster time-series, these model architectures are selected to evaluate our research hypotheses. This way, we ensure that state-of-the-art neural networks are used, which have demonstrated superb performance for time-series forecasting with exogenous variables (Apte and Haribhakta 2024). These neural network models forecasted promising baseline results, especially for longer horizon forecasting.

Once clusters are established, three model configurations are executed for each of the three time-series clustering methods (Ia, Ib and II): (1) with cluster-specific global models that are only trained on the cluster's time-series data, (2) with the inclusion of relevant covariates but trained on all time-series, and (3) both combined. This approach is schematically visualized in Figure 13.

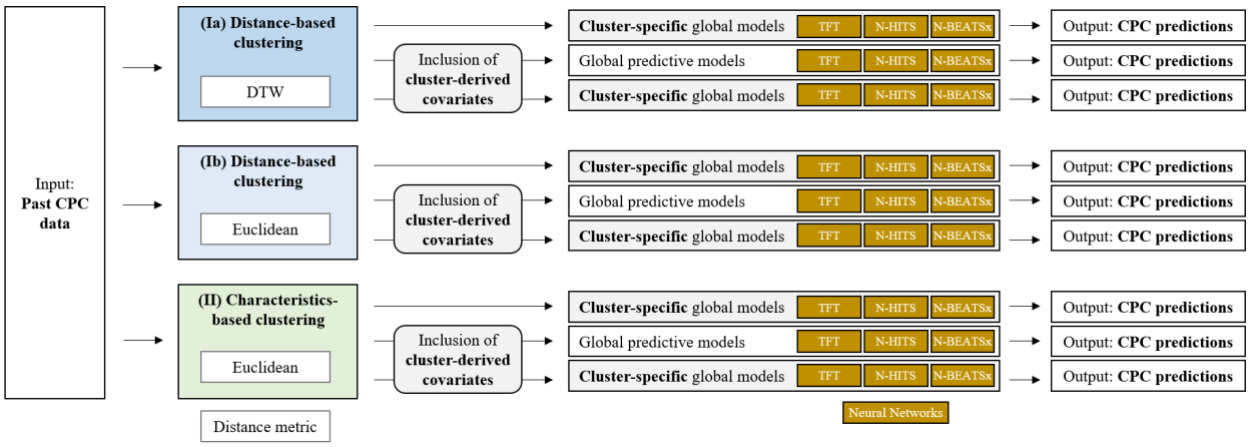

Figure 13: Methodological workflow of time-series forecasting configurations derived from time-series clustering

A.3.1.1. Distance-based Clustering

To cluster the time-series distance-based, we leveraged the TimeSeriesKMeans algorithm from the *tslearn* library. This is a clustering algorithm designed specifically for time-series data. It supports both Euclidean distance and DTW as distance metrics (Tavenard 2017).

Before applying each of the distance metrics within TimeSeriesKMeans, we first transformed the data to address skewness and differences in magnitude. A logarithmic transformation was applied to mitigate the positive skewness inherent in the time-series. This ensures that extreme values do not disproportionately influence the clustering process. Subsequently, the time-series were standardized using z-score normalization, which standardizes the data to have a mean of zero and a standard deviation of one. This is a commonly used standardization technique for time-series clustering and helps to standardize raw CPC time-series of different magnitudes (Ruiz et al. 2020).

To finetune the DTW calculation, we applied a Sakoe-Chiba constraint with a radius of 30. The chosen radius aligns with the research conducted by Oldenburg, Han and Kaiser (2024). The Sakoe-Chiba constraint is a global band constraint that limits the warping path to a diagonal band around the main diagonal of the cost matrix. Based on the chosen radius, this allows for a maximum temporal shift of 30 time-steps during the warping process. Limiting the path helps to maintain an alignment that better reflects the true temporal relationship between sequences, as it avoids unrealistic alignments caused by unconstrained flexibility. In comparison to other constraining techniques, such as the Itakura parallelogram, the Sakoe-Chiba band has been found to perform best (Rakthanmanon et al. 2013).

For both distance-based clustering methods, 3-, 7- and 14-day moving average smoothing and exponential smoothing techniques were tested in an experimental step to reduce noise before applying TimeSeriesKMeans. Average smoothing and exponential smoothing are common ways to extract patterns and the underlying structure of time-series, which can help to derive more meaningful clusters (Abhishek and Khullar 2024). However, both smoothing techniques did not

result in clusters that improved the final forecasting accuracy. Thus, all described distance-based time-series clustering experiments and results that follow use the unsmoothed time-series data.

A.3.1.2. Characteristics-based Clustering

To cluster the time-series based on their characteristics, we first calculated a set of structural features, which were then used to create a feature vector for each time series. The selection of characteristics builds upon the research of Bandara, Bergmeier and Smyl (2017). An overview of all characteristics used as static features can be found in Table 5.

Extracted characteristic	Description
mean	The average of all values in the time-series
median	The middle value of the series when arranged in order
variance	The extent to which values deviate from the mean
iqr	The range between the 75^{th} and 25^{th} percentiles
acf_1	The correlation of the series with itself at a lag of 1
acf_7	The autocorrelation at a defined seasonal lag of 7
trend	The slope of a linear regression line fitted to the time-series
linearity	The R^2 value of the linear fit (adherence to a straight line)
curvature	The R^2 of a quadratic polynomial fit (indicating the degree of curvature)
peak	The proportion of local maxima in the series
trough	The proportion of local minima in the series (negative peaks)
entropy	The complexity of the series, measured by normalized spectral entropy
lumpiness	The variability in rolling variances of the series over a fixed window of 10
spikiness	The variance of residuals after removing a fitted linear trend
f_spots	The count of flat regions in the series (histogram bins with zero values)
c_points	The number of times the series crosses its mean

Table 5: Overview and description of extracted characteristics

The extracted features are standardized using z-score normalization. This ensures that the extracted features, which differ significantly in scale, are standardized with a mean of zero and standard deviation of 1. As a result, each characteristic can contribute equally to the clustering outcome.

Once these characteristics are computed and standardized, they are combined into a single vector, so that each time-series is represented as a static feature vector. The vectors are then clustered

using the standard K-Means algorithm from scikit-learn, which employs the Lloyd algorithm to assign each vector to the nearest cluster based on the Euclidean distance (MacKay 2003).

A.3.2. Extracting Cluster-based Covariates for Forecasting

Building upon the work of Oldenburg, Han, and Kaiser (2024), who introduced competing advertisers as covariates, we propose a novel approach by incorporating only the three closest time-series of the same cluster as covariates that do not belong to the same advertiser. Including time-series from the same advertiser would not reflect competition but rather internal dynamics or correlations specific to that advertiser's own performance. By including time-series from *other* advertisers in the same cluster as covariates into our forecasting models, we are aiming at capturing competitive forces that influence the target time-series. Our approach with a limited amount of competitor's time-series ensures comparability across clusters of varying sizes, enables pipeline automation, optimizes the feature space by prioritizing the most relevant covariates, and reduces the risk of multicollinearity.

To capture the shared dynamics and contextual relevance within a cluster, we additionally included aggregated cluster-level statistics in an experimental setup. Specifically, we added the daily CPC mean, median, 25^{th} percentile, and 75^{th} percentile of the cluster as covariates. These statistics capture the collective behavior and variability within each cluster. Therefore, they provide a higher-level context and enrich the training dataset with broader cluster patterns of potentially competing keywords. However, adding those aggregated, daily cluster-level statistics as covariates did result in worse forecasting results – both smoothed and unsmoothed. Therefore, the final forecasting models discussed in the results part were not fed any daily aggregated cluster statistics as covariates.

A.4. Results

This section first presents the clustering results, followed by the analysis of the impact of the cluster-specific global models and the clustering-derived features on the forecasting performance.

A.4.1. Clustering Results

A.4.1.1. Number of Clusters per Method

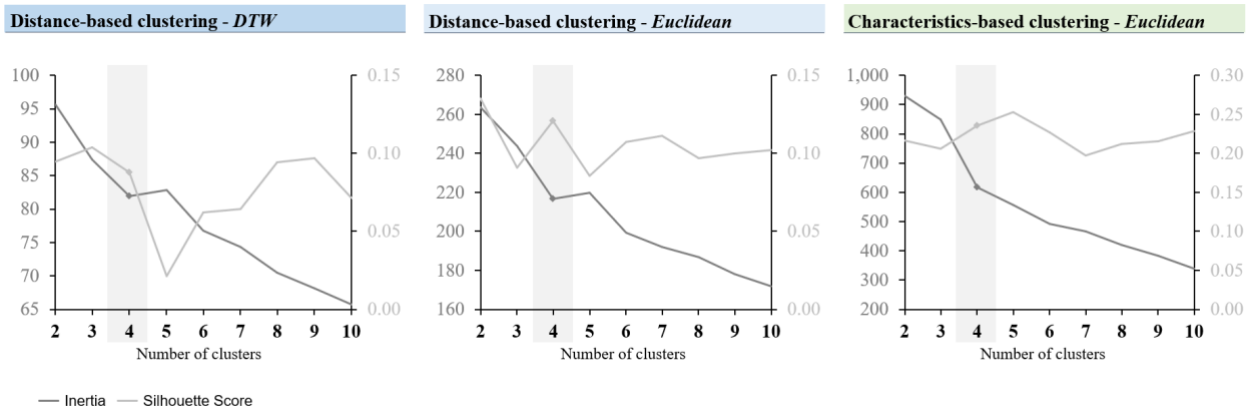

Figure 14: Selection of number of clusters per clustering method based on Elbow Method and Silhouette Score

Based on the Silhouette Score and the Elbow Method, the optimal number of clusters was determined to be four across all clustering methods. Figure 14 illustrates that four clusters achieve the best balance between inertia (Elbow Method) and the Silhouette Score across all clustering methods. Hence, TimeSeriesKMeans for distance-based methods and the standard k-means for the characteristics-based time-series clustering method was initialized with $k = 4$.

A.4.1.2. Cluster Assignments

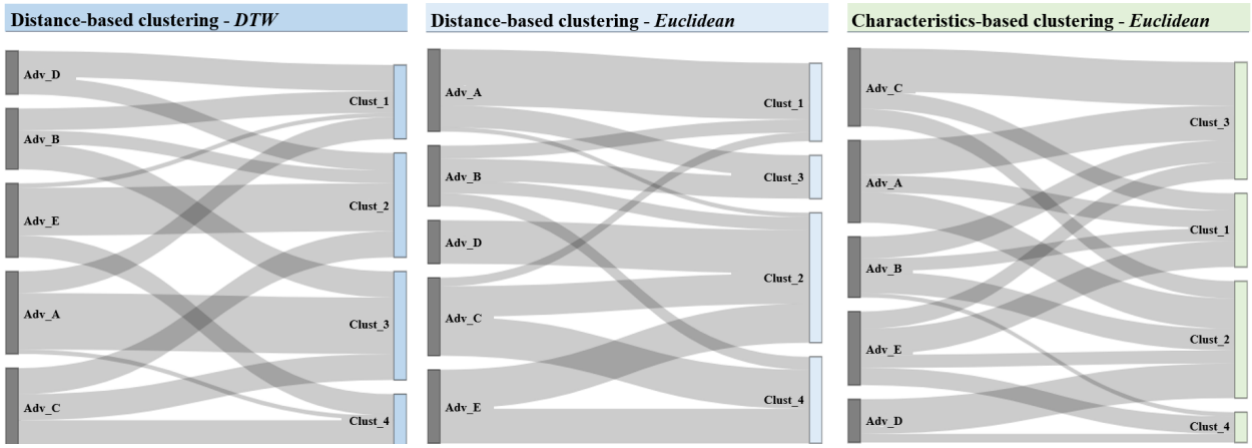

Figure 15: Sankey diagrams for cluster assignments of each keyword compared to the advertiser of the keyword

We have found the clustering assignment results being different depending on the clustering method used as visualized in the Sankey diagrams in Figure 15. The DTW distance-based method produces clusters with more evenly distributed sizes, with each cluster containing time-series from 3 to 4 different advertisers. The Euclidean distance-based method results in one dominant cluster containing time-series from all five competitors, alongside smaller, more exclusive clusters. Finally, the characteristics-based clustering method creates one cluster containing time-series from all competitors, two mid-sized clusters with relatively equal distribution among four competitors, and one small cluster dominated by advertiser E.

Adjusted Rand Index (ARI)

	Distance-based clustering – DTW	Distance-based clustering – Euclidean	Characteristics-based clustering – Euclidean
Distance-based clustering – DTW			
Distance-based clustering – Euclidean	0.44		
Characteristics-based clustering – Euclidean	0.12	0.11	

Normalized Mutual Information (NMI)

	Distance-based clustering – DTW	Distance-based clustering – Euclidean	Characteristics-based clustering – Euclidean
Distance-based clustering – DTW			
Distance-based clustering – Euclidean	0.51		
Characteristics-based clustering – Euclidean	0.18	0.23	

Figure 16: ARI and NMI scores to measure similarity of cluster assignments between clustering methods

As shown in Figure 16, the clusters generated by the two time-series distance-based clustering methods show a decent degree of similarity based on the ARI and MRI scores. In contrast, the clusters from the characteristics-based clustering method share only minimal similarity with those derived from the distance-based methods.

A.4.1.3. Correlation of Keyword Covariates with Target CPC

Covariate correlation analysis

	Closest_1	Closest_2	Closest_3
Distance-based clustering – DTW	0.53	0.41	0.34
Distance-based clustering – Euclidean	0.64	0.58	0.53
Characteristics-based clustering – Euclidean	0.46	0.46	0.38

Figure 17: Average correlation of closest keyword time-series from competitors with target CPC for each clustering method

Analyzing the correlation between the target time-series' CPC and the closest CPC time-series from competing advertisers, grouped by clustering method, reveals that these correlations are, on

average, moderately positive (see Figure 17). As expected, the strongest correlation is observed with the closest competitor, followed by the second closest. Notably, the distance-based Euclidean clustering method consistently yields higher correlations compared to the other two methods.

A.4.2. Forecasting Results

	Model		14 days		30 days		60 days	
			MAE	SMAPE	MAE	SMAPE	MAE	SMAPE
Baseline	SARIMA		0.491	0.263	**0.529**	**0.300**	0.645	0.352
	TFT univ. + rel_keyword_KPIs		0.522	0.275	0.581	0.331	**0.586**	**0.328**
	N-HITS univ.		0.497	0.265	0.574	0.328	0.618	0.343
	NBEATSx univ.		0.502	0.268	0.582	0.330	0.637	0.352
Distance-based clusting - DTW	TFT univ.	cluster-specific global models	0.504	0.268	*0.558*	*0.326*	0.624	0.347
	N-HITS univ.	cluster-specific global models	0.516	0.275	0.646	0.356	0.658	0.360
	NBEATSx univ.	cluster-specific global models	0.510	0.273	0.619	0.352	0.709	0.383
	TFT univ.	competition covariates	0.520	0.275	0.566	0.327	*0.587*	**0.328**
	N-HITS univ.	competition covariates	0.525	0.284	0.675	0.364	0.736	0.396
	NBEATSx univ.	competition covariates	0.530	0.283	0.679	0.366	0.711	0.394
	TFT univ.	both combined	0.537	0.285	0.586	0.338	0.626	0.348
	N-HITS univ.	both combined	0.557	0.295	0.686	0.366	0.821	0.431
	NBEATSx univ.	both combined	0.594	0.307	0.681	0.365	0.751	0.414
Distance-based clustering - Euclidean	TFT univ.	cluster-specific global models	0.519	0.276	0.587	0.347	0.633	0.354
	N-HITS univ.	cluster-specific global models	0.533	0.279	0.622	0.343	0.648	0.348
	NBEATSx univ.	cluster-specific global models	0.531	0.282	0.638	0.348	0.648	0.357
	TFT univ.	competition covariates	0.524	0.280	0.596	0.338	0.593	0.331
	N-HITS univ.	competition covariates	0.538	0.292	0.684	0.369	0.713	0.380
	NBEATSx univ.	competition covariates	0.509	0.279	0.662	0.364	0.747	0.390
	TFT univ.	both combined	0.527	0.281	0.614	0.356	0.631	0.356
	N-HITS univ.	both combined	0.546	0.290	0.657	0.355	0.781	0.417
	NBEATSx univ.	both combined	0.556	0.297	0.671	0.364	0.765	0.400
Characteristics-based clustering - Euclidean	TFT univ.	cluster-specific global models	0.493	0.264	0.579	0.334	0.626	0.374
	N-HITS univ.	cluster-specific global models	0.503	0.267	0.613	0.344	0.664	0.366
	NBEATSx univ.	cluster-specific global models	**0.487**	**0.262**	0.577	0.343	0.711	0.372
	TFT univ.	competition covariates	0.525	0.275	0.581	0.332	0.589	0.329
	N-HITS univ.	competition covariates	0.586	0.311	0.659	0.359	0.747	0.389
	NBEATSx univ.	competition covariates	0.519	0.277	0.704	0.374	0.758	0.398
	TFT univ.	both combined	0.523	0.278	0.583	0.342	0.632	0.354
	N-HITS univ.	both combined	0.535	0.291	0.684	0.378	0.726	0.393
	NBEATSx univ.	both combined	0.542	0.297	0.682	0.378	0.772	0.412

0.123 best forecasting model within specific clustering method
0.123 best forecasting model across all time-series clustering methods
0.123 best forecasting model across all time-series clustering methods + equal or better performance than best baseline model

Table 6: Forecasting results comparison – Chapter A

As visible in Table , all 3 configurations across the 3 different clustering methods did not yield consistent model improvements in comparison to the baseline.

Training the models only on intra-cluster time-series (*"cluster-specific global models"*) without exogenous features improved one single model configuration compared to the baseline, namely the univariate NBEATSx model on the 14-day horizon with clusters formed by the characteristics-based clustering method. However, this improvement is minimal with a 3.0% MAE (2.2% SMAPE) improvement compared to the baseline NBEATSx, and a 0.8% MAE (0.3% SMAPE) improvement compared to the best-performing 14-day horizon baseline model, the SARIMA.

The incorporation of the closest time-series of the same cluster that do not belong to the same advertiser (*"competition covariates"*) did worsen the model results compared to the univariate baseline without exogenous variables, except for the TFT on the 60-day forecasting horizon with competition covariates derived from distance-based clustering using DTW. Here, baseline TFT results could be matched based on the SMAPE evaluation metric (0.328), while performing slightly worse based on the MAE (0.587 versus 0.586).

Combining global models trained exclusively on intra-cluster data with the inclusion of competition covariates (*"both combined"*) did not enhance model performances. Instead, these model configurations declined in performance, as none of the three clustering methods yielded the best forecasting results when both strategies were included in the models at the same time.

With a few exceptions, the TFT demonstrates the best performance among the neural network forecasting models. This is in line with the findings of our baseline models. Exceptions include the distance-based Euclidean method on a 14-day prediction horizon with competition covariates, as well as the characteristics-based clustering method on a 14-day prediction horizon with competition covariates or intra-cluster global models (but not both combined), and the 30-day prediction horizon with intra-cluster global models. In all these cases, the NBEATSx outperformed the TFT based on the MAE evaluation metric.

A.5. Discussion

A.5.1. Clustering Assignments

Our evaluation of the clustering assignments revealed that the two distance-based time-series clustering methods produced rather similar results, while significantly differing from the clustering assignments of the characteristics-based method. This is expected from a methodological perspective, as distance-based methods and characteristics-based methods use fundamentally different techniques to derive clusters. There was no clustering technique that consistently outperformed or underperformed the others in the downstream task of forecasting CPC time-series.

These results highlight that clustering CPC time-series to identify competing keywords is not a trivial task. Instead, the clusters only reveal which time-series are similar based on the chosen technique for defining similarity. This limitation is further outlined in Chapter A.6.1.

A.5.2. Forecasting Results

When analyzing the results of the final time-series forecasting task, our adapted methodology at the keyword level did not yield covariates that could improve model performances, which contrasts with the research of Oldenburg, Han, and Kaiser (2024), who achieved promising results on the aggregated advertiser level. Moreover, training global models only on time-series of the same cluster did not help either to improve forecasting accuracy, except for marginal improvements for one single model out of 27 model configurations (NBEATSx with cluster-specific global models based on characteristics-based clustering on the 14-day horizon). On average, combining both strategies produced worse results than applying each strategy individually. Consequently, we reject both of our research hypotheses outlined for this research chapter. Neither global models only trained on intra-cluster time-series, nor covariates derived from clustering consistently improve the accuracy of CPC forecasts at the keyword level.

A.6. Conclusion

This chapter concludes with a negative response to our research question: *Can advertisers benefit from incorporating information on competition into their keyword-level CPC forecast?*. Our findings indicate that clustering time-series to train cluster-specific global models as well as utilizing the closest intra-cluster time-series from competitors as covariates did not improve forecasting results accuracy. The limitations this chapter is subject to are discussed below.

A.6.1. Limitations

To avoid repetition, limitations already included in Chapter 7.2 will not be covered in this section. While we adopted a methodology that resulted in promising results at the advertiser level and covered three common time-series clustering techniques leading to 27 forecasting model configurations, it is not exhaustive. Future research could explore additional clustering methods, as well as alternative warping constraints for DTW or new feature extraction strategies for characteristics-based clustering. In the forecasting task, exploring other neural network architectures, such as graph-based models, could further enhance performance.

As introduced in Chapter 2.1.3, Google's quality score makes it impossible to identify the exact parameters influencing the final CPC. Keywords from different advertisers bidding on the same search queries may have vastly different CPC time-series. Variations in the quality score, or even Google's untransparent daily distribution of a campaign's monthly budget, contribute to these discrepancies. As a result, competing keywords with potentially different CPC time-series may not be clustered together. Therefore, any potential improvements in forecasting accuracy that would originate from time-series clustering may not necessarily reflect competitive dynamics.

A.6.2. Outlook

To explore an alternative perspective on clustering CPC time-series, the next chapter will examine whether clustering based on the semantics of keywords offers a more effective approach to derive competition covariates that help to improve the keyword-level CPC forecasting accuracy.

B. A Semantic-based Clustering Approach

B.1. Introduction

Initial approaches, as discussed in Chapter A, involved forecasting CPC at the keyword level by clustering time-series based on their distances and characteristics. However, this methodology does not fully account for the competitive dynamics inherent in keyword bidding. To address this limitation, this chapter leverages the latest Natural Language Processing (NLP) techniques to enhance CPC time-series predictions by employing a semantic clustering approach. This approach identifies similar keywords under the assumption that keywords with similar meanings are more likely to compete in the same auction.

This approach aligns with the overarching goal of addressing RQ_3: *Can advertisers benefit from incorporating information on competition into their keyword-level CPC forecast?*

By incorporating competition-informed covariates from semantic clustering, the study aims to enhance prediction accuracy and bridge traditional clustering methods with the competitive factors influencing keyword-level bidding strategies. Therefore, the leading hypothesis for this chapter presents itself as:

> *$H_{B.1}$: Utilizing semantic-based clustering of keywords to replicate cobidding dynamics and enhance the accuracy of time-series forecasting at the keyword-level for CPC.*

This chapter is organized as follows: Chapter 2 reviews literature on semantic clustering in keyword analysis and time-series forecasting. Chapter 3 details the methodology, including keyword grouping, cluster variable calculation, and model integration. Chapter 4 presents results, assessing the impact on model performance. Chapter 5 discusses implications and limitations, and Chapter 6 concludes with key findings and future research directions.

B.2. Literature Review

B.2.1. Advanced Semantic Clustering Techniques for Keyword Analysis

Semantic clustering has emerged as a technique in marketing, particularly in keyword analysis, where understanding the nuanced relationships between terms is essential for effective targeting and content strategy (Liu and Toubia 2018). To effectively group semantically related keywords, a variety of clustering methodologies can be employed.

Traditional clustering methods, such as K-Means, have been extensively utilized due to their simplicity and efficiency. K-Means partitions data into non-overlapping clusters based on distance metrics, effectively grouping similar keywords together (Sinaga and Yang 2020). However, its strict cluster assignments often overlook the natural ambiguities and overlaps in language, which can lead to potential misclassification of semantically rich terms.

To address these limitations, fuzzy clustering methods, notably Fuzzy C-Means (FCM), have been introduced. FCM allows data points to belong to multiple clusters with varying degrees of membership, reflecting the real-world scenario where a keyword may pertain to multiple themes or topics (Khang et al. 2020). This flexibility enhances the granularity of keyword analysis, enabling marketers to capture subtle semantic relationships and tailor strategies accordingly. For instance, in analyzing consumer reviews, a keyword like "affordable" might relate to clusters concerning both "price" and "value." The adaptability of FCM in handling such nuances makes it preferable for capturing the multifaceted nature of marketing data (Russell and Lodwick 1999).

However, before utilizing these clustering algorithms, structured input data is required. Since the keywords we aim to cluster are unstructured text, they must first be transformed into structured embeddings. This transformation is achieved using a sentence transformer, which generates dense vector representations (embeddings) of sentences. These embeddings capture semantic meaning, ensuring that similar sentences are represented by closer vectors in the embedding space.

Therefore, we review three state-of-the-art embedding models that could potentially be of interest for analyzing the Google Ad Keywords dataset:

text-embedding-3-large (OpenAI): Developed by OpenAI, this model generates 3,072-dimensional embeddings that capture complex semantic relationships within textual content. Its high dimensionality enhances keyword analysis and contextual understanding, and it demonstrates strong performance in multilingual retrieval tasks, making it suitable for diverse datasets (OpenAI 2024).

WordLlama: WordLlama utilizes the token embedding codebook from large language models like Llama 3 70B to create efficient and compact word representations. Optimized for CPU inference with minimal dependencies, it excels in tasks such as similarity computations and clustering. It balances computational efficiency with accurate details (Miller 2024).

valurank/MiniLM-L6-Keyword-Extraction: This sentence-transformers model generates efficient 384-dimensional embeddings for tasks like clustering, semantic search, and keyword extraction. Fine-tuned on over 1 billion sentence pairs using contrastive learning, it captures semantic relationships effectively (Valurank 2024).

B.2.2. Influence of Semantic Clustering on Time-Series Prediction

When clustering techniques are applied in time-series contexts, they often serve to uncover shared temporal patterns or structural similarities among series, which could subsequently inform forecasting models (Aghabozorgi et al. 2015). This process allows for the generation of covariates that represent interdependencies within the dataset, effectively serving as higher-level abstractions of the underlying data dynamics. Such features have been shown to reduce noise, enhance model generalization, and provide a more robust foundation (Oriona, Manso, and Fernández 2023; Laurinec and Lucká 2017).

Furthermore, Oldenburg, Han, and Kaiser (2024) demonstrated that forecasting advertising costs benefits from multivariate models using covariates from competitors' CPC development identified through time-series clustering, resulting in more accurate CPC predictions.

In conclusion, with regard to this chapter, leveraging semantic-based clustering to replicate cobidding dynamics and integrating the resulting exogenous variables into time-series forecasting models could potentially enhance predictive accuracy at the keyword level for CPC.

B.3. Methodology

This chapter outlines the methodological framework adopted to study the prediction of CPC developments over time for selected keywords, utilizing a semantic cluster-based approach. The methodology is structured into three sequential components: (1) keyword grouping through semantic clustering, (2) calculation of a CPC-weighted metric averaged per day and per cluster based on the clusters determined in the first component, and (3) integration of the newly generated exogenous variable into predictive models for CPC trends.

This three-part methodology seeks to enhance the understanding of CPC dynamics by integrating semantic analysis.

B.3.1. Keyword Grouping Process

The keyword grouping process is schematically outlined in Figure 18.

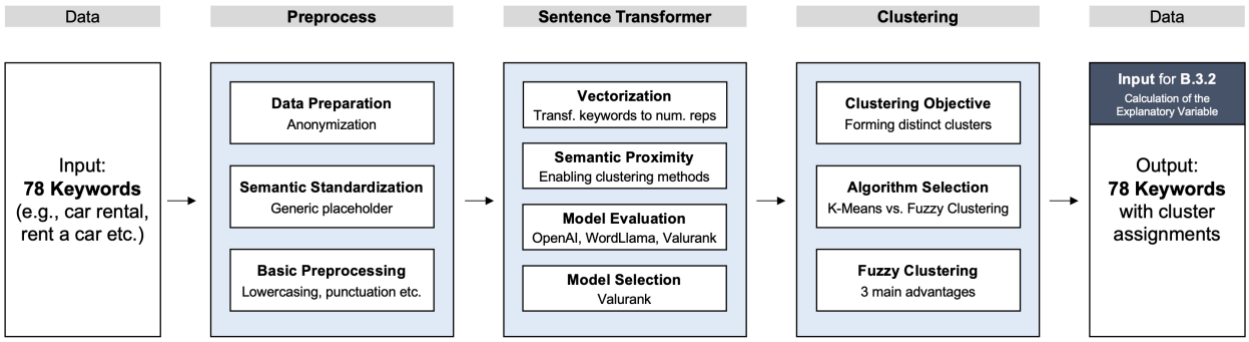

Figure 18: Schematic visualization of semantic clustering for time-series forecasting (1/2)

B.3.1.1. Preprocessing Procedures for Keyword Data

The dataset consists of 78 Google keywords (73 unique keywords), identical to those detailed in chapter 3, to maintain consistency and comparability across different models. Unique keywords mean that while the dataset contains 78 entries, some of these keywords are identical, such as when two companies bid on the exact same term (e.g., 'car rental'), resulting in separate entries with individual time-series developments.

To preserve confidentiality, car rental company names have been anonymized as "Company_A," "Company_B," and so forth. A sample of these keywords includes "Company_A car rental," "Company_B rent a car," "car rental," "car hire," and "www Company_C car rental."

Given that 89% of the keywords are branded in our final dataset, a targeted approach is necessary to link semantically similar keywords that may not appear directly comparable.

For example, keywords such as "Company_A *car rental*" and "Company_B *car rental*" are semantically similar and exhibit analogous trends in CPC over time. To prevent the clustering algorithm from grouping only those keywords that begin with the same company name (domain-based clustering), an adjustment is made wherein each company name is replaced with a generic placeholder like "branded". Consequently, "Company_A car rental" is transformed into "branded car rental." This modification ensures that specific brand names do not disproportionately influence the clustering process. As a result, the approach enables a more accurate grouping of semantically similar keywords (Suntwal et al. 2020).

Subsequent to this adjustment, standard preprocessing steps are applied to the keywords. These steps include converting all text to lowercase, removing punctuation, and standardizing whitespace. Such preprocessing ensures that semantically similar keywords, like "+ car rental", "Car Rental", and "car rental" are correctly aligned and grouped.

More advanced preprocessing techniques, such as stemming or lemmatization, are intentionally omitted. Given the relatively short length of the keywords - averaging 13 characters - such

techniques could excessively alter their semantic meaning, thereby compromising the integrity of the analysis (Chai 2022). In conclusion, the outlined preprocessing steps play a key role in ensuring the effectiveness of the subsequent sentence transformer and clustering methods

B.3.1.2. Utilizing Sentence Transformers for Keyword Encoding

For the steps mentioned above, an appropriate sentence transformer is required. We tested three distinct embedding models to analyze the Google Ad Keywords dataset, as detailed in chapter B2. To evaluate the sentence transformers, each keyword was first transformed into its corresponding vector. Next, these vectors were clustered into groups using K-Means Clustering.

The evaluation of embedding models for clustering analysis did not focus on identifying the model with the closest-distance-based clustering performance (as in Chapter A) but rather on assessing the semantic appropriateness of the keyword groupings. This process involved manual evaluation, where the resulting clusters were reviewed to determine how well the keywords within each group were semantically related and aligned with their real-world contextual usage.

Among the models tested, the valurank/MiniLM-L6-Keyword-Extraction model produced the most semantically coherent and contextually relevant groupings. Its specialization in keyword extraction and computational efficiency further reinforced its suitability for this study.

B.3.1.3. Techniques for Clustering Semantic Keyword Embeddings

Following the encoding of keywords with the valurank/MiniLM-L6-Keyword-Extraction model into numerical vector representations, the subsequent phase involves selecting and implementing an appropriate clustering algorithm to group semantically similar keywords. The primary objective of the clustering process is to form distinct clusters, each comprising keywords that are closely aligned in meaning while being well-separated from those in other clusters (Sinaga and Yang 2020). This semantic grouping aims to evaluate whether clustering keywords semantically can replicate cobidding dynamics and improve keyword-level CPC forecasting accuracy.

Considering the extensive array of available clustering methodologies, this study concentrates on two prominent algorithms: K-Means Clustering and Fuzzy Clustering (an extension of K-Means Clustering). The selection of these algorithms is driven by their respective strengths and their suitability for addressing the specific characteristics of the Google Ad Keywords dataset employed in this research (see chapter B2). Ultimately, the model will be trained using the Fuzzy Clustering approach due to two major reasons:

Flexible Membership Assignments: Fuzzy Clustering allows keywords to have partial memberships in multiple clusters, accurately capturing the inherent semantic overlaps among keywords. This flexibility is particularly beneficial for the Google Ad Keywords dataset. For example, keywords such as "Company_A car rental" and "Company_B rent a car" may predominantly belong to a "car rental" cluster but also show partial membership in a "branded" cluster as they include the names of particular companies.

Individual Influence based on Membership Assignments: In Fuzzy Clustering, the degree of membership determines how much influence a keyword exerts on each cluster. This ensures that semantically distant keywords, with minimal membership (e.g., <1%), have little impact, while closely related keywords exert a stronger influence on their relevant clusters. Since the membership assignments differ for each keyword, this approach ensures that every individual keyword reflects its influences as accurately as possible.

Cluster Name	Branded		Branded "Car Rental"		Branded "Rental"		Branded ".com"		Non-Branded "Rental Car"	
Count of Keywords (73 total)	22		25		10		8		8	
Keyword List	Company_G	97%	Company_E car rental	88%	Company_F rental	91%	Company_F com	79%	rental car	79%
	Company_C	97%	Company_G car rental	88%	Company_G rental	91%	Company_E com	79%	rental car	79%
	Company_D	97%	Company_A car rental	88%	+Company_E +rental	91%	+Company_G +com	79%	car rental	79%
	Company_I	97%	Company_D car rental	88%	Company_E rental	91%	www Company_E com	71%	rental cars	74%
	Company_H	97%	Company_H car rental	88%	Company_D rental	91%	www Company_F com	71%	rent car	66%
	Company_B	97%	Company_F car rental	88%	Company_F rentals	73%	www Company_E	53%	car rentals	63%
	Company_E	97%	Company_B car rental	88%	Company_E rentals	73%	Company_Abookings com	33%	rent a car	56%
	Company_F	97%	Company_G rental car	69%	Company_F rent	55%	Company_J.com	30%	car hire	36%
	+Company_E	97%	Company_E rental car	69%	+Company_E +rent	55%				
	Company_D usa	39%	Company_F rental car	69%						
	Company_D near me	33%	Company_D rental car	69%						
	Company_E near me	33%	Company_E car rentals	68%						
	Company_G car	30%	Company_F car rentals	68%						
	Company_E car	30%	Company_D rental cars	67%						
	+Company_E +car	30%	Company_E rental cars	67%						
	Company_F car	30%	Company_F rental cars	67%						
	Company_D miami	29%	Company_A rent a car	58%						
	Company_Gh	28%	Company_G rent a car	58%						
	Company_Abookings	27%	Company_D rent a car	58%						
	Company_F military discount	26%	Company_E rent a car	58%						
			Company_F rent a car	58%						
			www Company_F car rental	41%						
			Company_D rental car near me	40%						
			Company_E car rental locations	39%						
			Company_Gh car rental	32%						

Figure 19: Keyword Assignment to Clusters Based on Highest Membership

The Figure above illustrates each unique keyword and its assignment to the cluster where its membership is highest.

B.3.2. Calculation of the Explanatory Variable

The calculation of the explanatory variable is schematically outlined in Figure 20.

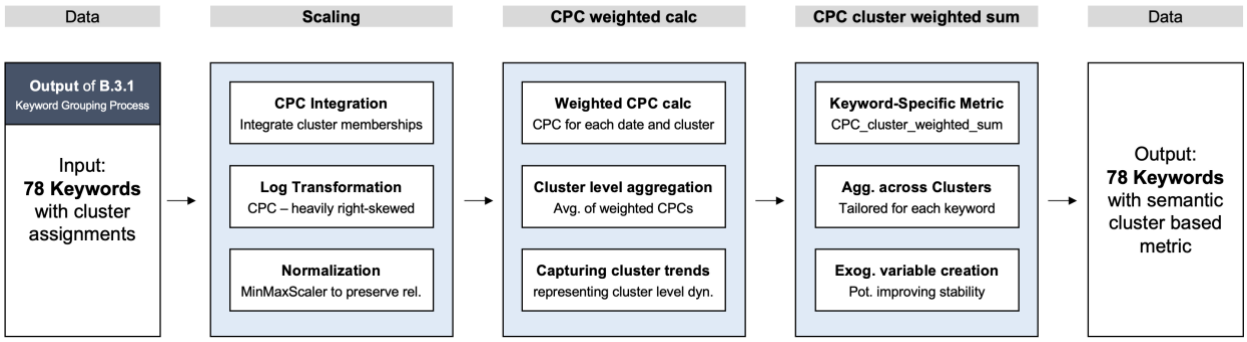

Figure 20: Schematic visualization of semantic clustering for time-series forecasting (2/2)

B.3.2.1. Scaling Technique for CPC Cluster Values

After determining the corresponding membership degrees for each unique keyword (see Figure 20), the clustering results were integrated with the keyword-level CPC metric to compute a relevant exogenous variable. This clustering-based exogenous variable represents a novel approach to integrating semantic clustering with the aim of enhancing CPC time prediction accuracy within the predictive models.

Given that the CPC metric exhibits a right-skewed distribution, as identified in the exploratory data analysis (Chapter 3), a logarithmic transformation was applied. This transformation

effectively reduces the impact of large outliers and renders the data distribution more symmetric, thereby enhancing the robustness of subsequent analyses (Würfel et al. 2021). Following the log transformation, a MinMaxScaler was employed to normalize the CPC values to a fixed range of [0, 1]. This scaling method was chosen over alternatives such as StandardScaler or RobustScaler because it preserves the relative relationships between data points, which is crucial for accurately calculating the weighted average CPC values based on membership degrees.

B.3.2.2. Computation of Weighted CPC Metric

Building upon the scaled CPC metrics and the established cluster memberships, the subsequent step involves calculating the Weighted CPC per date for each cluster. From this point forward in this chapter, any reference to CPC will refer to the scaled CPC metric.

To achieve this, the CPC value of each keyword is multiplied by its corresponding membership degree to a specific cluster. Mathematically, for each keyword *k* on a given date *d* and cluster *c*, the weighted CPC is computed as:

$$Weighted\ CPC_{k,c,d} = CPC_{k,d} \times \mu_{k,c}$$

Equation 3: Weighted CPC

where $\mu_{k,c}$ represents the membership degree of keyword k to cluster c.

Subsequently, the mean of these weighted CPC values is calculated for each cluster on each date, resulting in the metric termed *CPC_cluster_weighted*. This aggregation ensures that keywords with higher membership degrees exert a greater influence on the cluster-level CPC, thereby providing a more accurate and representative measure of CPC trends within each semantic group. The final equation for *CPC_cluster_weighted* for cluster *c* on date *d* is:

$$CPC\_cluster\_weighted_{c,d} = \frac{1}{N_c} \sum_{k=1}^{N} Weighted\ CPC_{\kappa,d,c}$$

Equation 4: CPC_cluster_weighted

where N_c is the number of keywords associated with cluster c.

This weighted aggregation method ensures that each cluster has a single value for a given date, acting as a stabilizer to indicate whether the cluster's CPC trends upward or downward. By reducing the influence of individual keywords, it aims to minimize the risk of overfitting.

B.3.2.3. Calculating the Weighted Sum of CPC Clusters

To apply the clustering results at the keyword level, this study calculates the *CPC_cluster_weighted_sum* metric. This metric is meant to capture the influence of the semantic clusters on individual keyword CPC trends. The computation process is as follows:

For each keyword *k* on a given date *d*, the membership degrees $\mu_{k,c}$ to each cluster *c* are multiplied by the corresponding *CPC_cluster_weighted* value of that cluster on the same date. Mathematically, this is expressed as:

$$CPC\_cluster\_weighted\_sum_{k,d} = \sum_{c=1}^{C} \mu_{k,c} \times CPC\_cluster\_weighted_{c,d}$$

Equation 5: CPC_cluster_weighted_sum

where C denotes the total number of clusters. This summation aggregates the weighted CPC contributions from all relevant clusters, tailored to the specific membership profile of each keyword.

The resulting *CPC_cluster_weighted_sum* provides a nuanced, keyword-specific metric that reflects the aggregated CPC dynamics of its associated semantic clusters. As an exogenous variable, it is designed to stabilize and optimize CPC predictions by leveraging the competitive behavior of semantically related keywords.

B.3.3. Incorporation into Forecasting Models

To augment the baseline models, the *CPC_cluster_weighted_sum* is integrated as an exogenous variable. In time-series forecasting, exogenous variables provide external context that can enhance predictive performance by capturing additional influences on the target variable.

Building upon this foundation, as outlined in Chapter A.2.6, clustering is often applied in time-series forecasting to group series with similar patterns, enabling global models to compute shared weights for intra-cluster series. This reduces noise, mitigates overfitting, and captures relationships specific to each cluster (Oriona, Manso, and Fernández 2023; Laurinec and Lucká 2017).

Furthermore, looking at the data and as discussed in chapter 3 the CPC data is right skewed with substantial outliers and fluctuations, posing challenges for prediction. While some keywords show temporal patterns, others lack clear trends or seasonality.

Another critical component in constructing the semantic-based clustering forecasting model for online advertising lies in examining the relationship between the target variable (CPC) and the exogenous variable (cluster CPC). A correlation analysis between the target CPC and its corresponding cluster-weighted CPC sum reveals varying distributions of correlations across keywords. Specifically, approximately 10% of the keywords demonstrate a strong time-series correlation ($\geq$ 60%), whereas the remaining 90% exhibit weaker correlations ($<$ 60%). This variability suggests that, while the exogenous variable (cluster CPC) may influence CPC predictions for certain keywords, its predictive power might be limited for others.

B.4. Results

As shown in Table 7, the incorporation of semantic clustering across three configurations did not consistently improve model performance compared to the univariate baseline across any forecasting horizon. Three key findings emerged:

TFT Model Performs Best Overall: The TFT model outperformed NHITS and NBEATSx across all horizons. One major reason for this could be its advanced attention mechanism, which allows it to better assess feature importance over time and adapt to complex temporal patterns.

Combining Configurations Enhances Performance: For the TFT model, which performed best overall, the combined configuration (global models trained on semantic clusters with competition covariates) yielded the best results on the 14-day and 60-day horizons, with MAE of 0.503 and 0.621 and SMAPE of 0.267 and 0.337, respectively. On the 30-day horizon, the cluster-specific global models performed slightly better (MAE 0.572, SMAPE 0.323).

Prediction Accuracy Declines with Longer Horizons: Accuracy decreased as the horizon lengthened, with TFT's MAE increasing from 0.503 on the 14-day horizon to 0.621 on the 60-day horizon in the combined configuration. This highlights the growing uncertainty and variability inherent in long-term forecasting.

These findings underscore the importance of model selection, feature integration, and horizon-specific considerations in achieving optimal performance for univariate time-series forecasting with exogenous variables.

	Model		14 days		30 days		60 days	
			MAE	sMAPE	MAE	sMAPE	MAE	sMAPE
Baseline	SARIMA		0,491	0,263	**0,529**	**0,300**	0,645	0,352
	TFT univ. + rel_keyword_KPIs		0,522	0,275	0,581	0,331	**0,586**	**0,328**
Semantic Clustering	TFT univ.	cluster-specific global models	0,521	0,271	0,572	0,323	0,625	0,337
	N-HITS univ.	cluster-specific global models	0,510	0,269	0,626	0,344	0,684	0,372
	NBEATSx univ.	cluster-specific global models	0,533	0,277	0,625	0,341	0,681	0,358
	TFT univ.	competition covariates	0,535	0,277	0,583	0,329	0,640	0,340
	N-HITS univ.	competition covariates	0,539	0,288	0,611	0,348	0,699	0,379
	NBEATSx univ.	competition covariates	0,513	0,269	0,576	0,332	0,726	0,372
	TFT univ.	both combined	0,503	0,267	0,575	0,325	0,621	0,337
	N-HITS univ.	both combined	0,562	0,288	0,682	0,371	0,712	0,373
	NBEATSx univ.	both combined	0,556	0,290	0,672	0,369	0,736	0,381

0.123 best model within specific clustering method
0.123 best model across all time-series clustering methods
0.123 best model across all time-series clustering methods + equal or better performance than best baseline model

Table 7: Forecasting results comparison – Chapter B

B.5. Discussion and Limitations

The integration of the *CPC_cluster_weighted_sum* as an exogenous variable was hypothesized to enhance the accuracy of time-series forecasting at the keyword-level CPC. However, empirical results from this particular setup indicate that the inclusion of this exogenous variable did not improve, and in some cases slightly worsened, the predictive accuracy of the models. Specifically, models augmented with semantic clustering exhibited higher MAE and SMAPE values compared to their univariate counterparts, suggesting that the exogenous variable may not have effectively captured the intended competitive cobidding influences. This unexpected outcome can be attributed to several interrelated factors, which can be categorized into Methodological Limitations, Data Quality Concerns, and Hypothesis Framing.

Methodological Limitations: The utilization of a weighted CPC sum as the exogenous variable may have oversimplified the complex relationships inherent in the data. This approach potentially failed to capture cobidding interactions between keywords, especially when examining distinctions between clusters. For instance, in the "Branded Car Rental" cluster, keywords like "Company_G rental car" or "Company_E car rentals" are influenced by direct competition among brands targeting the same audience. In contrast, the "Non-Branded Car Rental" cluster, including terms like "rent a car" and "car hire," exhibits potentially broader seasonal and regional trends, such as increased demand during holiday periods. By aggregating the CPC dynamics of keywords within a cluster, the model may miss inter-cluster relationships, such as how branded terms interact with non-branded keywords in shared auctions. While the weighted CPC sum approach appeared to stabilize trends within clusters it overlooked these further competitive dynamics.

Additionally, the current methodology did not account for lagged effects in CPC dynamics, which may have restricted the models' ability to capture delayed market influences. Incorporating temporal alignment and considering lagged variables could enhance the models' capacity to reflect the temporal dependencies present in CPC trends.

Data Quality Concerns: Our semantic clustering process did not encompass all keywords due to the completeness threshold (see Chapter 3), resulting in a representation of market dynamics that is somewhat not fully complete. While the dataset included primary keywords, such as variations of "Company_A," "Company_B," "Company_C," and general "car rental" terms, numerous location-specific keywords - such as "Company_A car hire UK," "Company_B car rentals near me," and "Company_C car rental in New York" – were excluded due to data quality issues and the application of the 98% completeness threshold for time-series data. This exclusion likely reduced the effectiveness of competition data as an exogenous variable, as omitted keywords could have influenced CPC through localized competition and diverse market conditions. Consequently, regional market trends and competitive actions within Google's auction mechanism were not fully captured, potentially skewing CPC predictions and diminishing the models' predictive power. Future research should aim to incorporate a more exhaustive set of keywords while still maintaining a high completeness threshold to better reflect the nuanced competition dynamics across different market segments.

Hypothesis Framing: The semantic relatedness of the selected keywords may not have accurately reflected shared market dynamics, potentially introducing noise rather than meaningful signals into the models. This relatedness was based on a manual evaluation process described in chapter B3, where embedding models were assessed for the semantic appropriateness of keyword groupings. Although this manual review aimed to ensure that keywords within each cluster were contextually aligned, it may not have fully captured all relevant nuances. For instance, similar keywords such as "Company_D" and " Company_D usa" were grouped together based on semantic similarity, yet their distinct geographic contexts can lead to different search behaviors and competitive dynamics. Consequently, refining the selection criteria for related keywords to ensure greater relevance and contextual accuracy could improve the signal-to-noise ratio and enhance the forecasting models' performance.

In summary, the findings highlight the complexity of integrating semantic clustering into CPC forecasting models.

B.6. Conclusion

B.6.1. Summary of Findings

This chapter investigated the impact of incorporating an exogenous variable, derived from semantically related keywords' cluster CPC metric, on forecasting models for CPC. The results indicated that the inclusion of these exogenous variables did not enhance model performance; in fact, baseline models outperformed the enhanced versions. Furthermore, the discussion highlighted that methodological limitations, such as the oversimplification introduced by the *Cluster Weighted Sum* approach, data quality issues including incomplete keyword coverage, and hypothesis framing concerns related to the semantic relatedness of the selected keywords, could have potentially contributed to the underperformance of the enhanced models. These factors likely introduced noise and obscured critical competitive dynamics, thereby diminishing the predictive power of the exogenous variables. Consequently, we reject the research hypothesis ($H_{B.1}$) outlined for this chapter. Neither global models only trained on intra-cluster time-series, nor covariates derived from clustering consistently improved the accuracy of daily CPC forecasts at the keyword level.

B.6.2. Recommendations and Future Research

Building on the insights from Chapter B.5, future research should prioritize integrating user search queries into CPC forecasting models, as they offer critical insights into consumer intent and behavior. By combining detailed search query data with existing keyword metrics, models could better capture competitor bidding strategies and market dynamics. Additionally, employing advanced NLP techniques to analyze search queries could enhance the accuracy and depth of predictive analytics.

C. An Auction-based Covariate Extraction Approach

C.1. Introduction – Decoding the Google Ads Auction Process

The prior chapter introduced a semantic clustering approach to identify similar keywords, assuming that keywords with related meanings compete in the same auction. Building on this, the following section incorporates newly engineered features to reflect auction dynamics and competition. Until now, competition-based features for keywords entering the same auction have not been explored.

As described in Chapter 2.1.3, the Google Ads auction operates as a Generalized Second Price (GSP) auction, where advertisers bid a maximum CPC on keywords, and auctions are triggered by user search queries. The Grips dataset provides an unique opportunity to study this process in depth. Unlike traditional datasets, it includes the individual user search queries, represented by the *admatchedquery* column, that triggers a paid search auction and reveals the unique domains and keywords that were included in the auction.

We introduce four new model configurations that incorporate auction-based features to capture underlying auction dynamics. These models are compared against the baseline models to determine whether auction-based competition data improves predictive performance for deep learning models across short-, medium-, and long-term horizons, surpassing the SARIMA model on short- and medium-term horizons and the deep learning models on the long-term horizon.

Based on this, we propose the following hypothesis:

> $H_{C.1:}$ "Incorporating auction-based competition-based features will enable us to better replicate auction dynamics, resulting in improved predictive performance."

In this deep dive of our research, we want to answer RQ_3: *Can advertisers benefit from incorporating information on competition into their keyword-level CPC forecast?*

The remainder of this chapter is organized as follows. First, the theoretical foundations of bidding behavior in the GSP auction system and the implications of different keyword types are examined. Second, the characteristics and role of the *admatchedquery* column are analyzed, highlighting its unique implications for detecting co-bidding. Third, the model development process, including feature selection, is reviewed, with the rationale behind each model explained. Finally, the results are presented, evaluating the impact of competitive auction data on model performance.

C.2. Theoretical Background

To comprehensively understand the dynamics of Google Ads auctions, it is crucial to examine their structural framework, operational mechanisms, and the influence of bidding strategies and keyword types on auction outcomes.

C.2.1. Auction Dynamics

An auction, as defined by Wolfstetter (1996), is "a bidding mechanism, described by a set of auction rules that specify how the winner is determined and how much he has to pay. In addition, auction rules may restrict participation and feasible bids and impose certain rules of behavior." Auctions are typically categorized into oral and sealed-bid formats, with our research focusing on sealed-bid and generalized second price (GSP) auctions, as this format is mostly adapted in the domain of online advertising (Edelman, Ostrovsky, and Schwarz 2007). The GSP auction operates differently from traditional auctions, where the winner pays their highest bid. Instead, in a GSP auction, the highest bidder secures the top position but pays the amount of the second-highest bid Wolfstetter (1996). In the context of Google Ads, and specifically the ad rank formula introduced in Chapter 2.1.3, the auction functions as a GSP auction under the assumption that all ads have the same relevance and click-through rates conditional on position. In this framework, ad placements are determined by bid rankings, with winners paying the next highest bid. This design creates a strategic dynamic among bidders, as their bids influence both their ranking and the prices paid by competitors (Edelman, Ostrovsky, and Schwarz 2007). Past studies have tried to explore this

relationship, for example Athey and Nekipelov (2010) showed how incorporating competitor's bidding strategies, actually enhanced the interpretability of bidding outcomes. To include this strategic interplay various features will be introduced, that try to capture the bidding dynamics.

C.2.2. Keyword Match Type

A significant determinant in this process is the type of keyword match between the user search intent and the advertisers ad. The keyword match type is set by each individual advertiser on the keyword level and affects how frequently keywords are included in auctions. Each of the three different match types: broad match, phrase match and exact match have different business implications, as well as impacts on relevant marketing KPIs and are compared in Table 8.

Match Type	Symbol	Keyword	Trigger	Search	KPI Impact
Broad Match	+keyword	+car +rental	Modified term (or closer variations, but not synonyms), in any order.	rental car lisbon	Reach: High CPC: Lower, due to broader targeting
Phrase Match	"keyword"	"car rental"	Phrase and a close variation of that phrase.	luxury car rental Lisbon	Reach: Moderate CPC: Higher, due to more precise targeting
Exact Match	[keyword]	[car rental]	Are an exact term and close variations of the exact term.	car rental	Reach: Low CPC: Highest, strong alignment with user intent

Table 8: Overview of the different Keyword Match Types

The broad match keyword type is the most general and inclusive, being triggered when the user's search query contains the keyword, any variation of it, or related terms, regardless of order. Broad match keywords offer the highest reach among all match types, as they are eligible to enter many auctions (Hopkins 2023). However, according to a study by Hopkins (2023) for Optmyzr, broad match keywords typically experience lower CPC, lower conversion rates (CR), and, on average,

lower return on ad spend (ROAS) due to targeting that often fails to align with user search intent. Despite this, the broad match is employed as an aggressive bidding strategy to maximize visibility.

The phrase match keyword type, by contrast, is only triggered when the user's search query contains the keyword or a close variation of it in the specified order. This type has a moderate reach compared to the broad match, entering fewer auctions because of its stricter matching requirements. The more precise targeting of phrase match keywords results in higher CPC and CR than broad match, as the search queries they match are more closely aligned with the user's intent (Hopkins 2023).

Finally, exact match keywords are the most restrictive and specific of the match types. They are only triggered when there is an exact match or close variation between the keyword and the user's search query. As a result, exact match keywords have the lowest reach, entering fewer auctions due to their strict requirements. However, this precision often results in higher conversion rates (CR) and better alignment with user search intent, which drives stronger ROAS compared to broad match, despite typically having higher CPC. According to Hopkins (2023), exact match consistently outperforms broad match for most accounts.

Due to the previous filtering steps described in Chapter 3.2.4 of the paper, the final dataset only includes, phrase matches, and broad matches.

C.2.3. Co-Bidding

To analyze competition dynamics, it's important to first understand the concept of co-bidding. In the context of Google Ads, an auction is triggered whenever a user searches for something online. Co-bidding occurs when multiple advertisers compete in the same auction to display their ads for that search query. This competition can involve either the same keyword being bid on by different advertisers or multiple keywords, potentially from the same advertiser or different ones.

Figure 21 illustrates co-bidding behavior among the five advertisers across 78 keywords. In this Figure, red nodes represent shared keywords, indicating co-bidding between advertisers on the

keyword level. Only five keywords are shared across different domains, revealing minimal co-bidding at the keyword level. This suggests that keyword selection is largely independent across advertisers, limiting the potential to extract competition or auction-based features from there.

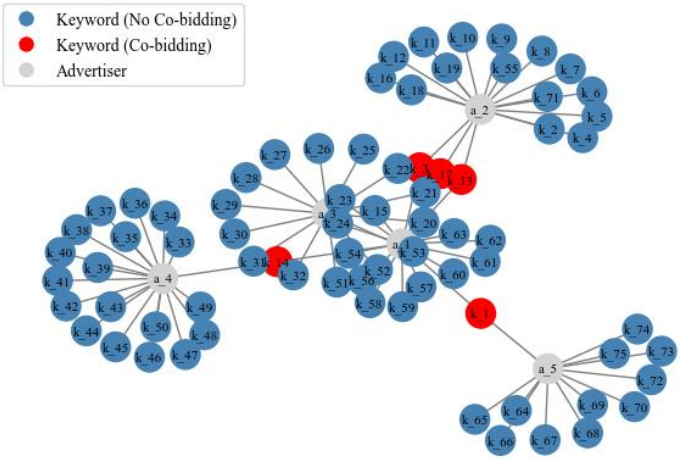

Figure 21: Network Graph showing the co-bidding on the keyword level

Another way to identify co-bidding is to consider how the Google auction process operates. When a user performs a search, Google triggers an auction by selecting relevant keywords based on various factors. In our dataset, this process is represented by the *admatchedquery* column, which captures the user's search term and provides insights into which advertisers and keywords participated in each individual user search.

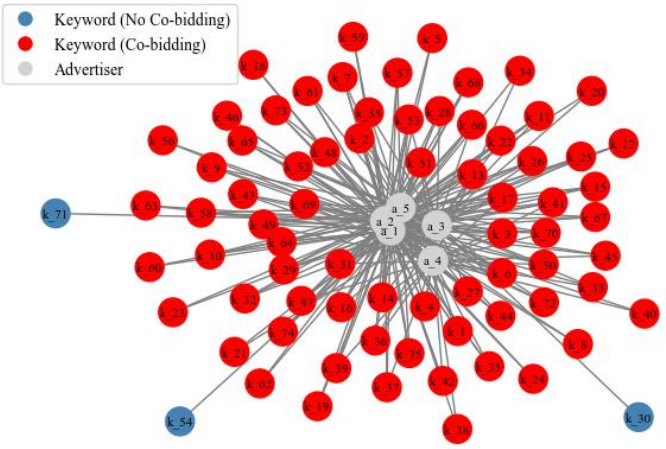

Figure 22: Graph chart showing co-bidding on the keyword via the admatchedquery column

Figure 22 visualizes co-bidding behavior at this level, showing that all advertisers experience co-bidding, with only three keywords lacking shared bids. This presents a stark contrast to the minimal co-bidding observed at the keyword level.

These findings reveal that competition occurs primarily at the users search level and at the keywords that are pulled into the auctions, rather than directly on the keywords itself. However, predicting CPCs at the keyword level remains viable, as advertisers set maximum CPCs per keyword and have limited control over their inclusion in auctions. The users search term thus provides crucial competitive insights that could enhance model performance.

C.3. Methodology

C.3.1. Characteristics of Auction Based Features

As now established, co-bidding is present on the users search query level, a few new features are introduced with different characteristics to catch the auction dynamics and the competition.

C.3.1.1. Keyword Match Type

As established, keyword match types significantly influence KPIs and have distinct implications. Table 9 presents clear differences between phrase match and broad match keywords, aligning with findings from Hopkins (2023). The dataset consists of 68 phrase match, 10 broad match keywords and no exact match keywords, which is due to our filtering methods described in Chapter 3.2.4.

Phrase match keywords demonstrate higher average and median CR, CTR, CPC, and ROAS compared to broad match keywords. However, the ROAS values for both match types show substantial variability, with outliers skewing the mean. Analyzing the absolute KPIs reveals anticipated patterns: phrase match keywords generate higher average transactions, ad clicks, and revenue, while broad match keywords exhibit lower average ad costs and a greater number of impressions, reflecting their broader reach. This suggests that broad match keywords are effective for achieving wider reach, whereas phrase match keywords are more suitable for precise targeting and conservative bidding strategies (Du et al. 2017).

		Phrase Match		Broad Match	
		Mean	*Median*	Mean	*Median*
Relative KPIs	CR	0.141	*0.116*	0.098	*0.060*
	CTR	0.767	*0.750*	0.392	*0.429*
	CPC	2.251	*2.228*	1.743	*1.434*
	ROAS	356.71	*17.45*	54.90	*13.78*
Absolute KPIs	Transactions	36.91	*2.00*	4.91	*1.00*
	Adclicks	282.60	*25.00*	35.40	*17.00*
	Revenue (USD)	17258.80	*540.45*	2186.02	*771.32*
	Adcost (USD)	291.70	*42.78*	85.80	*40.01*
	Impressions	54.10	*21.00*	786.96	*96.00*

Table 9: Difference in relative and absolute KPIs between Phrase Match and Broad Match

Notably, broad match keywords display higher visibility despite their limited share in the dataset. Representing only 12.82% of the total keywords, they account for 15.81% of the auctions entered, further reinforcing their role in expanding reach. These findings are consistent with prior research indicating that broad match keywords achieve wider reach but are associated with lower CR, CPC, and CTR, whereas phrase match keywords demonstrate narrower reach with higher CR, CPC, and CTR. However, caution is warranted in interpreting these results due to the limited sample size of broad match keywords, where a small number of outliers may disproportionately affect the analysis.

C.3.1.2. User Search Terms - Admatchedquery

The *admatchedquery* column represents user search terms and is essential for analyzing co-bidding dynamics as it connects the users search activity to auction outcomes. Co-bidding can be understood at two levels. At the keyword level, multiple keywords enter the same auction, which can originate either from the same advertiser or from different advertisers. This type of co-bidding occurs in all auctions classified as involving co-bidding. At the advertiser level, competition arises when different advertisers enter through their keywords into the same auction.

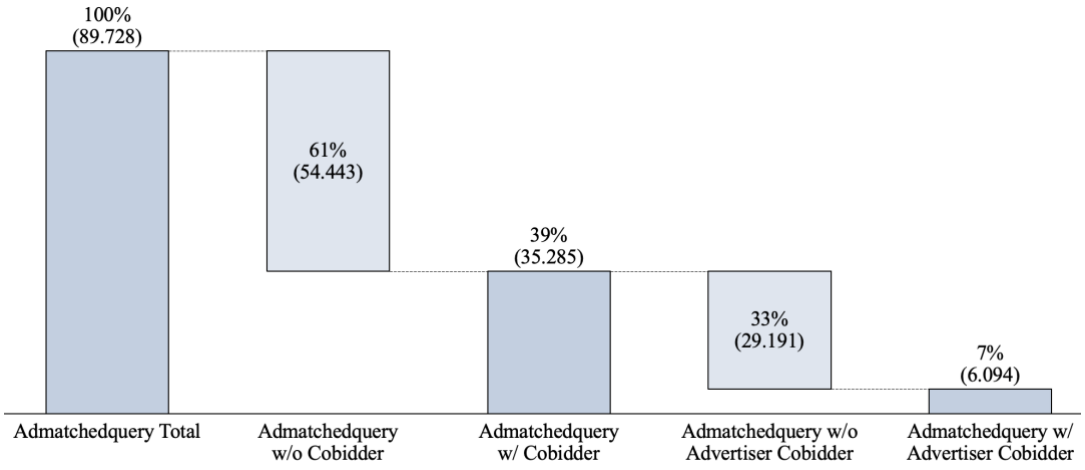

Figure 23: Visualization of the number of admatchedqueries and how co-bidding on keyword and advertiser level is present.

Our dataset contains 89,728 recorded search queries, of which 35,285 (39 percent) involve co-bidding at either the keyword level, advertiser level, or both, as illustrated in Figure 23. However, only 6,094 search queries (7 percent) reflect competition specifically between multiple advertisers. While the proportion of advertiser-level co-bidding appears relatively low, its significance remains noteworthy. This reduced number of advertiser-level co-bidding instances is primarily due to the filtering methods detailed in Chapter 3.2.4, where many keywords were excluded, leaving a dataset primarily composed of likely auction winners.

As Google's auction mechanism can lead to internal competition when the same advertiser enters a single auction with multiple keywords targeting the same user search. Analyzing the average number of keywords and unique advertisers participating in each auction reveals that on average, 5.21 keywords enter each auction. But these often come from a limited number of advertisers—an average of just 1.07 unique advertisers per auction. Proofing that, in most cases, most of the keywords competing in an auction belong to the same advertiser, meaning that the advertisers bid against themselves. Such internal competition has the same effect on CPC than external competition and suggests that incorporating this information as a feature in CPC prediction models could improve their accuracy.

C.3.2. Introduction of Competition Based Features

Building on the findings from the previous chapter, we now incorporate auction-based competition features into the modeling process. Figure 24 illustrates the feature engineering process designed to replicate the Google Ads auction environment.

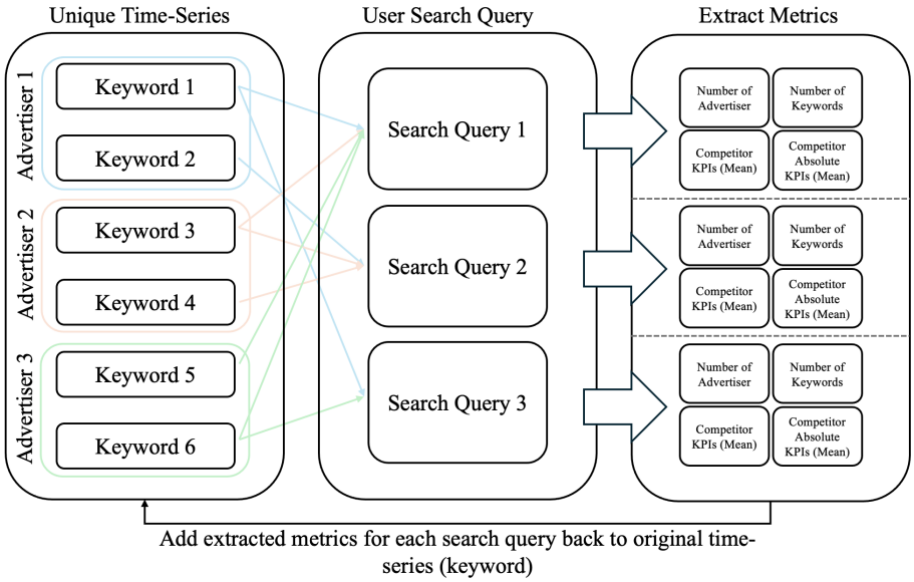

Figure 24: Representation of the google auction process and how the admatchedquery is utilized to include co-bidding

Each auction is triggered at the *admatchedquery* level, where different keywords are pulled in based on their relevance to the user's search query. For instance, if the search query is "rental car Lisbon," Google might pull in keywords such as "car rental" (Keyword 1 from Advertiser 1) and "rental car" (Keyword 3 from Advertiser 2) into the same auction. Then the next step is, to calculate both relative and absolute features for each keyword by aggregating data from all auctions the keyword participates in, throughout the day. These features are then integrated as new features into the keyword's original time-series data. A comprehensive summary of these features, along with their definitions and summary statistics, is presented in Table 10. Table 10 highlights significant variability across several features, notably auction revenue and impressions, where large differences between the mean and median indicate high standard deviations in the new feature set. This suggests that many features are right-skewed, as evidenced by the median values being lower than the mean. The substantial spread in features such as auction revenue and impressions could introduce challenges during the modeling phase, potentially hindering the

model's ability to generalize effectively. To address this, a log transformation will be applied to these features.

Extracted Characteristics	Description	Mean	Median	Std.
Auction Adclicks	Total number of ad clicks recorded in auctions for the specific keyword on a given day.	147.06	11.45	705.16
Auction Adcost (USD)	Total ad cost incurred in auctions for the specific keyword on a given day.	146.08	16.41	646.35
Auction CPC	Average cost-per-click for all advertisers competing in the auction for the specific keyword on a given day.	1.59	1.33	1.11
Auction CR	Average conversion rate across all advertisers participating in the auction for the specific keyword on a given day.	0.11	0.08	0.24
Auction CTR	Average cost-per-click for all advertisers competing in the auction for the specific keyword on a given day.	0.41	0.43	0.23
Auction Impressions	Total number of impressions recorded in auctions for the specific keyword on a given day.	423.88	43.67	1702.23
Auction Revenue (USD)	Total revenue generated in auctions for the specific keyword on a given day.	9825.55	339.61	50506.55
Auction ROAS	Average conversion rate across all advertisers participating in the auction for the specific keyword on a given day.	69.77	23.40	832.91
Auction Tansactions	Total number of transactions recorded in auctions for the specific keyword on a given day.	21.53	1.00	110.33
Num Auctions	Total count of auctions in which the specific keyword participated on a given day.	11.04	4.00	24.07
Num Advertisers in Auction	Total count of unique advertisers competing in auctions involving the specific keyword on a given day.	1.07	1.00	0.26
Num Keywords in Auction	Total number of keywords competing in the auction on a given day.	5.21	4.33	6.32

Table 10: Engineered features with the corresponding mean, median and standard deviation

C.3.3. Model Development

With the newly created features established, we now transition to the model development phase, examining the differences among the four developed models. A summary of all model

configurations is presented in Table 11. Due to hardware limitations, exploring all feature types simultaneously was not possible, and incorporating individual feature selection was infeasible given the significant computational demands of testing numerous feature combinations across multiple model types.

C.3.3.1. Model 1: Keyword Type and Numerical Co-bidding Features

The first model configuration integrates eight features, by building upon the best performing model from the previous group chapter, which included relative KPIs. Now three additional features are included, to add information about the competitive environment within the auction: *NumKeywords*, *NumAuctions*, and *NumAdvertisersInAuction*. Additionally, two binary variables, *MatchTypePhraseMatch* and *MatchTypeBroadMatch*, are added to enhance the model by capturing underlying information associated with each match type.

C.3.3.2. Model 2: All Features

Building upon the foundational features of the first model, eleven more features are added by introducing both relative and absolute KPIs derived from auction level data, along with a categorical feature distinguishing between branded and generic search types. The relative KPIs include metrics such as *Auction CTR*, *Auction CR*, *Auction CPC* and *Auction ROAS*, which are calculated values for each keyword for each day from the auctions that the keyword entered on that day. In addition to relative KPIs, absolute KPIs, such as *Auction Transactions*, *Auction Adcost*, *Auction Impressions*, and *Auction Revenue*, are included, as the sum of other keywords that participated in the same auction as the original keyword. Together, these KPIs should help the model to get a better understanding of the overall auction dynamics and get a better idea of how the CPC evolves, with other advertisers or keywords participating in the same auction.

C.3.3.3. Model 3: Most Important Features Based on the Feature Importance

The third model builds upon the previous configuration by focusing on a reduced feature set, selected based on the feature importance results derived from the TFT results in Model 2. Given

that feature importance varies across different prediction horizons, the top features were identified by averaging the importance scores of each feature across the horizons. The features that emerged as the most important include: *MatchTypePhraseMatch*, *NumKeywordsInAuction, CTR, CR, ROAS, NumAdvertisersInAuction, AuctionCPC, AuctionTransactions, AuctionImpressions.*

Extracted Characteristics	Metric/Unit	Model 1	Model 2	Model 3	Model 4
Match Type Phrase Match	Boolean	Yes	Yes	Yes	-
Num Keywords in Auction	Count	Yes	Yes	Yes	-
CTR	Ratio	Yes	Yes	Yes	Yes
CR	Ratio	Yes	Yes	Yes	-
ROAS	Decimal	Yes	Yes	Yes	-
Num Auctions	Count	Yes	Yes	-	-
Num Advertisers in Auction	Count	Yes	Yes	Yes	-
Match Type Broad Match	Boolean	Yes	Yes	-	-
Auction CPC	USD	-	Yes	Yes	Yes
Auction CTR	Ratio	-	Yes	-	-
Auction CR	Ratio	-	Yes	-	-
Auction ROAS	Decimal	-	Yes	-	-
Search Type Generic	Boolean	-	Yes	-	-
Search Type Branded	Boolean	-	Yes	-	-
Auction Tansactions	Count	-	Yes	Yes	-
Auction Adcost (USD)	USD	-	Yes	-	-
Auction Impressions	Count	-	Yes	Yes	-
Auction Adclicks	Count	-	Yes	-	-
Auction Revenue (USD)	USD	-	Yes	-	-
Number of Features		8	19	9	2

Table 11: Feature overview

C.3.3.4. Model 4: Model with the Two Most Important Features

The fourth and final model narrows the feature set even further, incorporating only the top two features identified from model 3 based on the TFT feature importance: *CTR* and *AuctionCPC*. The goal is to create a similarly performing model with fewer features, making it more computationally efficient and easier to use by reducing reliance on exogenous variable.

C.4. Results

Table 12 presents the results across the four models, showing corresponding forecasting errors (SMAPE and MAE) across the 14-, 30-, and 60-day horizons and how they compare to the baseline models.

For the 14-day and 30-day horizons, SARIMA delivers the best performance, achieving the lowest SMAPE values of 0.263 and 0.300, respectively. Model 1 performs competitively with a SMAPE of 0.267 (14 days) and 0.325 (30 days), outperforming the baseline deep learning models on the 30-day prediction horizon, but not surpassing the SARIMA model.

At the 60-day horizon, the performance dynamics shift in favor of deep learning-based models. Model 1 achieves a SMAPE of 0.331, closely followed by Model 2 with 0.334. Importantly, Model 1 accomplishes this with a reduced feature set of only 8 features compared to 19 features in Model 2. Next, models were evaluated with features selected based on feature importance scores.

On the 14-day and 30-day horizons, SARIMA remains the top-performing model. Notably, Model 3, trained with fewer features, shows a decline in performance on the 14-day and 30-day prediction horizon, compared to the baseline, with SMAPE values of 0.279 and 0.337. In contrast, on the 60-day horizon, again deep learning models outperform the SARIMA. Model 3, using the reduced feature set, achieves a SMAPE of 0.328, matching the best performance observed from Chapter 1.

When reducing the feature set to only the two most important features from model 3, CTR and Auction CPC (as highlighted in Table 12), the results are mixed. On the 14-day and 30-day horizons, this minimal feature model performs worse than SARIMA but still surpasses the performance of the feature-rich Model 3, with SMAPE values of 0.274 and 0.331, respectively. For the 60-day horizon, the reduced model outperforms SARIMA with a SMAPE of 0.330, although it falls just short of Model 3, which remains the top performer for this horizon.

Overall, SARIMA still provides the most accurate forecasts for shorter horizons of 14 and 30 days. For the 60-day horizon, Model 3 achieves the best performance with a SMAPE of 0.328,

demonstrating the effectiveness of deep learning models, but also showing that the inclusion of competition-based features does not improve the overall performance on the short- and medium-term horizon.

		14 Days		30 Days		60 Days	
	Model	MAE	SMAPE	MAE	SMAPE	MAE	SMAPE
Baseline	SARIMA	0.491	**0.263**	**0.529**	**0.300**	0.645	0.352
	TFT Univ. + rel_keyword_KPIs	0.522	0.275	0.581	0.331	0.586	**0.328**
Model 1	TFT univ. + auction features	**0.482**	*0.267*	0.547	*0.325*	0.582	0.331
	N-HITS univ + auction features	0.547	0.295	0.661	0.361	0.776	0.403
	N-BEATSx + auction features	0.558	0.293	0.663	0.365	0.773	0.404
Model 2	TFT univ. + auction features	0.495	0.274	*0.533*	0.328	0.585	0.334
	N-HITS univ + auction features	0.596	0.346	0.786	0.409	0.845	0.438
	N-BEATSx + auction features	0.661	0.362	0.793	0.415	0.987	0.531
Model 3	TFT univ. + auction features	0.513	0.279	0.573	0.337	**0.575**	**0.328**
	N-HITS univ + auction features	0.703	0.328	0.811	0.422	0.901	0.464
	N-BEATSx + auction features	0.608	0.324	0.865	0.393	0.886	0.448
Model 4	TFT univ. + auction features	0.493	0.274	0.562	0.331	0.580	0.330
	N-HITS univ + auction features	0.523	0.281	0.647	0.356	0.693	0.375
	N-BEATSx + auction features	0.502	0.272	0.637	0.359	0.697	0.382

0.123 best model within specific time horizon from the new models

0.123 best model across all new models + equal or better performance than best baseline model

Table 12: Forecasting results comparison – Chapter C

C.5. Discussion

Contrary to the initial hypothesis, incorporating competition-related features did not enhance forecasting performance on short- and medium-term horizons. In these scenarios, the baseline SARIMA consistently outperformed both deep learning and ML models. Only on the longer 60-day horizon did the deep learning models show some advantage by incorporating external features; however, they failed to surpass the previously best-performing models, yielding similar results.

As a result, we reject the hypothesis that including auction- and bidding-related features improves performance across all prediction horizons. One potential explanation lies in the dataset's limited scope. Although the dataset captures advertiser-level bidding competition, it omits a substantial portion of the market. The analysis is restricted to four major car rental providers and a single comparison website, excluding smaller local rental agencies that likely participate in auctions, as well as the majority of price comparison websites, which are increasingly significant booking channels for consumers with the OTA market showing an expected growth of 9.9% CAGR till 2030 (Statista 2023). This limitation hinders the ability to fully replicate auction dynamics, particularly in GSP auctions, where interactions between bidders play an important role.

C.6. Conclusion

Reflecting on this section, the study demonstrates that including competition-related features does not enhance results for CPC prediction. In addressing the research question, “Can advertisers benefit from incorporating information on competition into their keyword-level CPC forecast?” the findings indicate that these features do not improve overall model performance on any prediction horizon. While competition features matched the best previous results, they failed to surpass them. Consequently, we recommend advertisers to rely on baseline models, particularly SARIMA, which produced the best results for the 14- and 30-day prediction horizons. Beyond superior accuracy, SARIMA is also computationally efficient, relying solely on historical target variable data and eliminating the need for external market information that may not always be available.

C.6.1. Limitations

While the models did not meet expectations, it is important to acknowledge certain limitations. The primary constraint lies in the dataset's scope and completeness. First, only a limited number of complete time-series were available due to pre-processing steps and the 98% completeness threshold, which resulted in the exclusion of many time-series. As a result, the dataset focuses

primarily on auction winners, leaving out unsuccessful participants who attempted to enter the auction. This omission makes it harder to see the full picture of the competition, making it difficult to analyze how bidders interact and understand the overall bidding dynamics. Second, accurately modeling bidding competition requires data that includes all advertisers participating in the auctions. Although the Grips dataset offers a broad market overview and covers companies with 65% of the market share in the car rental space, it still excludes 35% of the market. This limitation makes it difficult to capture the underlying dynamics between bidders and the overall competition, as the dataset lacks the comprehensive coverage needed to fully represent auction behavior.

C.6.2. Outlook

Future research could tackle these limitations by incorporating more comprehensive competition data or using datasets with broader market coverage. Including information like the actual ad rank achieved for each keyword, alongside user search queries, could also offer valuable insights into auction dynamics. This would provide a clearer understanding of quality scores and serve as an external benchmark for evaluating keyword performance.

Another approach worth exploring is modeling auction dynamics with graph neural networks (GNNs). GNNs could capture patterns of bidding competition reflected in the *admatchedquery* column without requiring data on all auction participants. By representing the relationships between keywords as a graph, GNNs could better capture auction-level competition and reveal interactions that are often missed by traditional methods. The next chapter will take a closer look at this approach.

D. Leveraging Competitive Dynamics using Graph Neural Networks

D.1. Introduction

Google's method to determine CPC by running auctions on advertisers' keyword-level bids was described in detail in Chapter 2.1.3. In these auctions, when a user searches for a given query e.g. "car rental" many keywords from different advertisers like "car rental EasyCar", "EasierCar car rental" or "EasiestCar rent a car" will be considered based on undisclosed Google matching algorithms.

In our goal to answer the research question *RQ_{3}: Can advertisers benefit from incorporating information on competition into their keyword-level CPC forecast?* we assume that knowing who is competing in which auctions should be valuable. That is because, by the nature of auctions reviewed by Klemperer (Klemperer 1999), CPC of one advertiser should be driven by other advertisers CPCs competing for the same search query.

The naive approach of identifying competition in auctions by looking at "co-bidding" of different advertisers on the exact same keyword is not possible, because of the lack of this "co-bidding" in our dataset (only existent for 5 out of 78 keywords). In Chapter C "admatchedquery" was introduced for deriving keywords participation in auctions and additional features were engineered to use this knowledge in our benchmark models. While models incorporating these features did not beat univariate baselines, we are going one step further by using the same "admatchedquery" information to model our time-series in a graph structure capturing competition on auction level. Graph neural networks will then be applied to leverage this structural embedding within our predictive models. The leading hypothesis for this chapter presents itself as:

> *$H_{D.1}$: Leveraging graph structures representing auction competition improves accuracy in time-series forecasting of keyword-level CPC.*

This part will first discuss the theoretical background of graph neural networks for competition modeling in time-series. Graph convolution is introduced as our mechanism of choice to leverage

graph structured data. We subsequently show how we constructed a meaningful competition graph for the time-series in our dataset. Different graph-based models are constructed and benchmarked, and finally results are discussed to accept or reject the hypothesis.

D.2. Theoretical Background

D.2.1. Graph Neural Networks for Time-Series Forecasting

Graph-based systems and Graph Neural Networks (GNNs) have emerged as powerful tools for modeling complex relationships in time-series forecasting of intercorrelated time-series, where dependencies often extend beyond simple sequential order. By representing time-series data as graph structures, these methods capture intricate connections, such as similarities across variables or temporal correlations between time steps (M. Jin, Koh, et al. 2024). GNNs process these graphs by learning embeddings that integrate both the graph topology and node-specific features, enabling accurate predictions in dynamic systems (Hamilton, Ying, and Leskovec 2018). This approach has proven particularly effective in domains like traffic, supply chain and energy grid optimization, where interconnected components influence each other (Cao et al. 2020). In their paper *"Spatio-Temporal Graph Convolutional Networks: A Deep Learning Framework for Traffic Forecasting"* Yu, Yin and Zhu (2018) and colleagues successfully implemented a graph neural network architecture to enhance accuracy in traffic forecasting by leveraging a graph representation of the road network. Their results inspire us to adapt the concept to model paid search keywords with a graph representing competitive interactions between keywords. A detailed approach to building this graph is described in D.3. "Methodology".

D.2.2. Graph Convolution

Graph Convolutional Networks (GCNs) extend the concept of Convolutional Neural Networks (CNNs) (Bengio and Lecun 1997) to graph-structured data, making them suitable for applications involving relational and non-Euclidean data structures (Wu et al. 2021). Unlike traditional CNNs, which apply convolutional operations on regular grid-like data such as images, GCNs operate on

graphs by generalizing the convolution operation (as shown in Figure 25) to aggregate and transform information from neighboring nodes, capturing both local and global graph features (Wu et al. 2021). This aggregation mechanism enables GCNs to learn embeddings that represent the structure and properties of graphs (Hamilton, Ying, and Leskovec 2018).

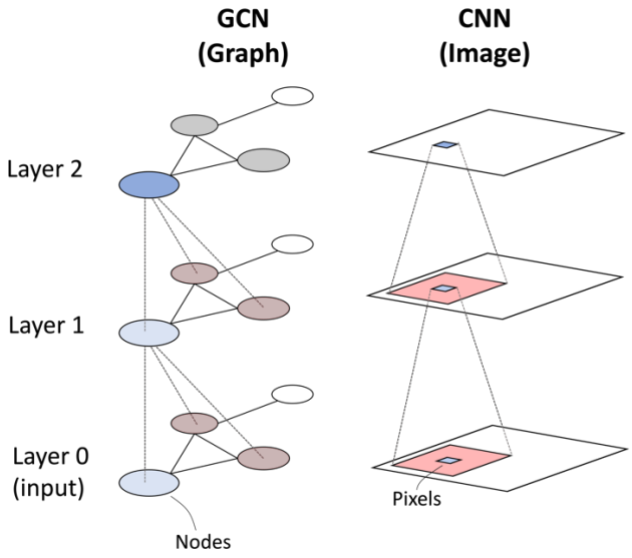

Figure 25: Comparison of convolution operation on graph vs image data (Bernstein 2023)

D.2.3. Graph Analysis

Graph analysis is the study of networks to understand their structure, relationships, and key properties. One fundamental measure in graph analysis is the degree of a node, which represents the number of edges connected to it. Nodes with high degrees often act as hubs or critical points of interaction within the network (Freeman 1978). Another essential concept is path length, the shortest number of edges between two nodes. This helps identify how easily information or influence can travel through the graph (Even 1979).

The diameter of a graph is the longest shortest path between any two nodes, reflecting the network's overall reach or extent. Networks with small diameters are often described as "small-world" networks due to their efficiency in connectivity (Watts and Strogatz 1998). The clustering coefficient measures the tendency of a node’s neighbors to connect with each other, indicating the presence of tightly-knit groups or communities (Schank and Wagner 2005). Lastly, density

quantifies how interconnected a graph is by comparing the number of existing edges to the maximum possible number of edges, offering a sense of how sparse or dense the network is (Scott 2017). We will use these metrics in this research to analyze our graph representation of keywords and study the impact of node connectiveness on predictive accuracy of GNNs.

D.2.4. Long Short-Term Memory Networks

Long Short-Term Memory (LSTM) networks are a type of recurrent neural network (RNN) architecture, specifically designed to handle sequential data and overcome the vanishing gradient problem associated with traditional RNNs (Hochreiter 1997). LSTMs achieve this through specialized "gates" within their memory cell structure - namely, the input, forget, and output gates - that control information flow and allow the network to retain information over longer sequences (Hochreiter 1997). Each gate operates through learned weights and biases, adjusting the network's "memory" based on input at each time step. Consequently, LSTMs are well-suited for tasks such as language modeling, speech recognition, and time-series forecasting, where data dependencies span across long sequences (Sundermeyer, Schlüter, and Ney 2012) (Fischer and Krauss 2018).

LSTMs are designed to manage lagged effects and seasonal components by selectively retaining relevant information through their gated structure, which helps prevent the loss of important historical context (Sezer, Gudelek, and Ozbayoglu 2020). Moreover, their flexible architecture allows them to adapt to both short- and long-term dependencies in time-series data, making them preferable in fields like finance, meteorology, and industrial process monitoring, where accurate forecasting can have significant impacts (Brownlee 2018). We use LSTM because of the possibility to construct simple models, making it easier to contribute increases in accuracy to our graph convolution component.

D.3. Methodology

D.3.1. Building a Graph Structure for Keywords and Queries

To enable the use of graph convolution, data must be transformed into a graph structure. Bipartite graphs frequently appear in various complex systems because they represent interactions between two distinct types of entities, such as genes and proteins, metabolites and enzymes, authors and publications, or products and customers. A standard method for examining these graphs involves constructing a network among the nodes on one side based on their connections to the nodes on the opposite side. This process, known as one-mode projection, serves as a foundational step for subsequent analyses (Zweig and Kaufmann 2011) and is illustrated in Figure 26.

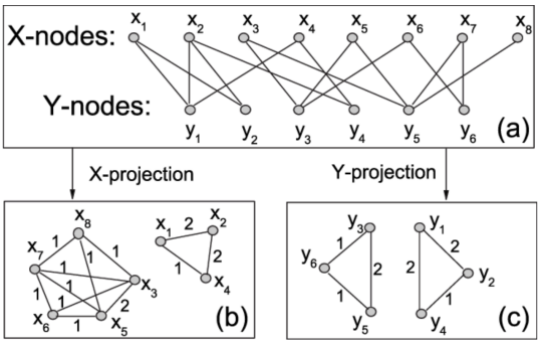

Figure 26: Illustration of a bipartite network (a), as well as its X projection (b) and Y projection (c) (Zhou et al. 2007)

Translating the concept of bipartite graphs to our use case, we have the two node types "advertiser-keywords" and "user queries" (data field: "admatchedquery"). Advertiser-keywords represent the combination of an advertiser and a keyword we want to predict CPC for. User queries represent exact search terms users put into Google search before they saw and clicked paid search results.

Relationships between the node types can be established by creating edges between advertiser-keywords and the user queries they were matched to by the Google algorithm (Figure 27, Step A). User queries are only connected to advertiser-keywords if there are more than 10 occurrences of the keyword being matched to the query. This results in a sparser graph, ensuring focus on relevant, more frequent connections.

We decided for a static graph aggregating the interactions across the complete timeframe of the training dataset, instead of a dynamic graph reflecting daily interactions, to account for the dynamics of Google paid search auctions. Since there can only be a limited number of winners of the auction in each day, only looking at the winners of the day would suppress a lot of competition. We expect that regular winners across the whole dataset are in constant competition for the same user queries. That is why using a static graph offers a more wholistic model of competitive dynamics. Only looking at interactions inside the training set prevents data-leakage.

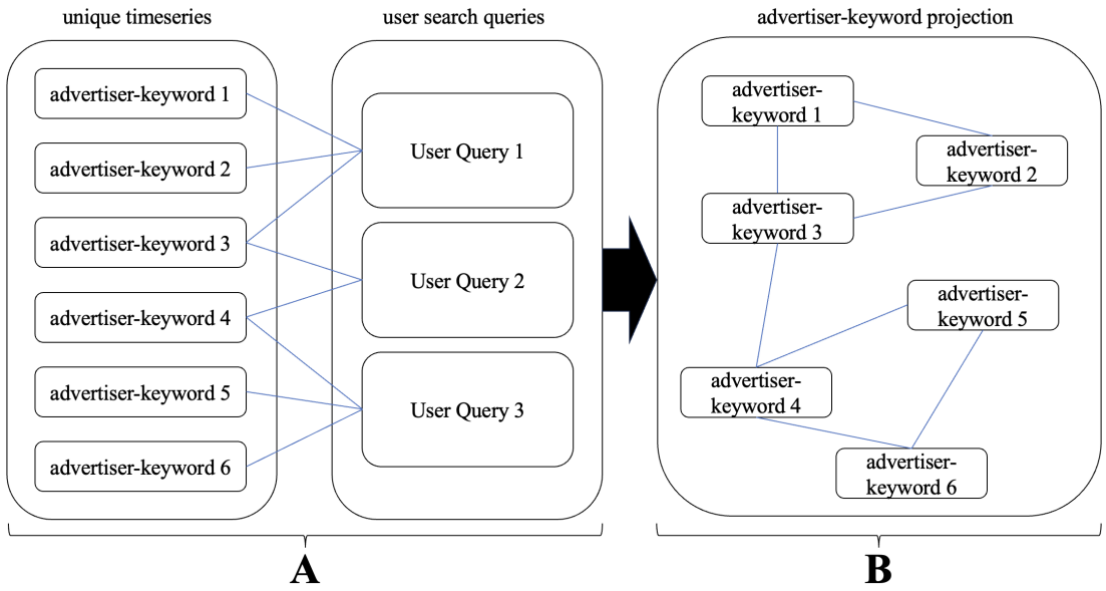

Figure 27: Creation of domain-keyword graph by projecting from bipartite graph of domain-keywords and user queries

The graph resulting from Step A shows interactions between advertiser-keywords and user queries. This representation can only be leveraged for further analysis in a limited way. In earlier experiments (Chapter C) we were not able to beat univariate benchmarks using statistics aggregated on user query level. Since we hope to gain advantage in CPC forecasting by including competing CPC bids, we need a representation of competitive interactions between advertiser-keywords. This is the motivation for creating a one-mode projection on the advertiser-keyword space based on shared user queries (Figure 27, Step B). A connection between advertiser-keywords in this space represents the advertiser-keywords regularly competing in the same auctions, therefore potentially influencing each other's CPC.

D.3.2. Analysing the Resulting Graph

The advertiser-keyword projection has 78 (number of unique advertiser-keywords) nodes and 397 edges. After excluding 5 nodes that collectively only have 1 edge and are not connected to the rest of the graph, we can calculate more metrics to describe the graph and gain additional insights. The diameter of the remaining connected graph is 7 and the average path length is 2.65. This shows that, while the most distance advertiser-keywords are only reached after 7 hops on the graph, most keywords are more closely connected. A rather high clustering coefficient of 0.675 in combination with a moderate density of 0.132 suggest a meaningful structure with specialized clusters that connect closely but are still sufficiently separated from other clusters (otherwise density would be higher). The full graph has an average degree of 5.09, so an average node would have 5 edges. The degree distribution in Figure 28 provides more detailed insights into the connectiveness of the graph:

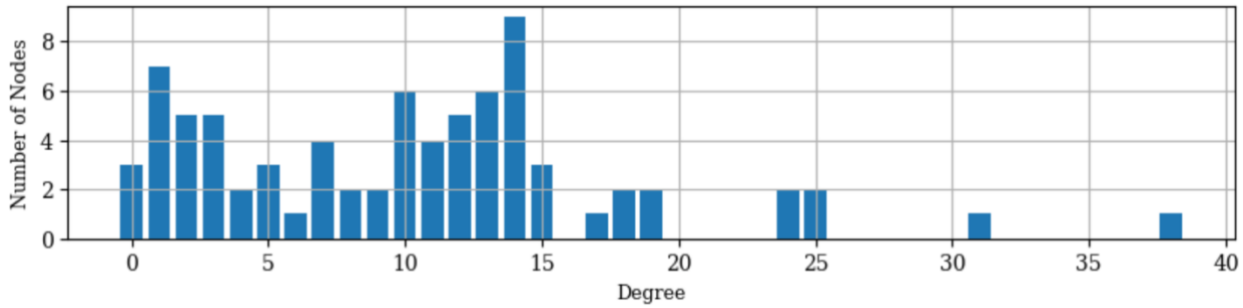

Figure 28: Degree Distribution of Advertiser-Keyword Projection

The higher degree (>15) nodes are especially interesting to analyze, to get insights into how different types of advertiser-keywords interact in the graph. Here we can see a clear pattern: 64% of these keywords are unbranded. In contrast 94% of the low to mid degree (<=15) keywords are branded, containing a car rental company's name. We can deduct that unbranded keywords are present in more different auctions and therefore probably experience more complex competitive dynamics.

D.3.3. Model Architecture

As a result of our chosen benchmarked models by Nixtla being closed source, we are not able to train models with graph convolution leveraging the exact same forecasting setup as in Chapter 4. We will substitute with custom TFT and LSTM based forecasting architectures. All GNN models used, will be comprised of two main building blocks:

1) Graph processing Layer: We are going to use a graph convolutional layer to gather, aggregate and transform CPC information from connected advertiser-keywords. This layer is supposed to model the auction dynamics which we hypothesize to be a competitive force driving CPC.

2) Forecasting Layer: The forecasting layer uses the features created by the graph convolutional layer to predict CPC for the given forecast horizon. We will adapt a LSTM and a TFT architecture (introduced in Chapter 2) as possible forecasting layers. The possibility to build very simple LSTM versions, especially when compared to benchmarked TFT or NBEATSx models, enables us to develop a study setup where we can easily measure the impact of our graph convolution. The high dimensional TFT should make sure our models' accuracy is not hindered by lack of representational power.

The code implementation of our model architectures was inspired by a proposed implementation of Spatio-Temporal Graph Convolutional Network (STGCN) by Khodadadi in the TensorFlow/Keras documentation (Khodadadi 2023).

D.3.4. Experimental Setup

To determine the effect of adding graph convolution we are going to compare different model architectures once with and once without using a graph convolutional layer. The idea is to keep the predictive layers untouched, therefore being able to confidently attribute differences in performance to the added graph convolution only. We are going to stick with the same forecast horizons (14, 30, 60) and error metrics (MAE, SMAPE) as in Chapter 4 to ensure comparable results. We will also include the high performing SARIMA model in the comparison.

D.4. Results

D.4.1. Model Performance Comparison

The final scores for every model configuration and each of the three forecasting horizons are demonstrated in Table 13. A valuable metric for evaluating the effectiveness of the graph-based modeling approach presented in this part is the delta between errors of LSTM and TFT with and without the graph convolutional layer:

$$\Delta Error_{GC} = \frac{Error_{LSTM} - Error_{GC+LSTM} + Error_{TFT} - Error_{GC+TFT}}{Error_{GC+LSTM} + Error_{GC+TFT}}$$

Equation 6: Delta Error Graph Convolution

For MAE (SMAPE) this delta is 49.4% (53.0%) on the 14-day horizon, 21.5% (17.2%) on the 30-day horizon and -1.2% (-2.4%) on the 60-day horizon.

This can be interpreted as the addition of graph convolution significantly improving results on short horizons but decreasing in effectiveness with the length of the forecast period as much as showing slightly negative effects on the 60-day horizon.

Model	**14 days**		**30 days**		**60 days**	
	MAE	SMAPE	MAE	SMAPE	MAE	SMAPE
SARIMA	0.491	0.263	0.529	0.300	0.645	0.352
LSTM	0.843	0.444	0.811	0.439	0.810	0.429
GC + LSTM	0.598	0.294	**0.651**	**0.366**	0.823	0.442
TFT	0.896	0.445	0.807	0.439	**0.802**	**0.428**
GC + TFT	**0.566**	**0.287**	0.681	0.383	0.809	0.436

0.123 best model excluding SARIMA baseline

Table 13: Forecasting results comparison – Chapter D

D.4.2. Analysing Error by Node Degree

Because our Graph Neural Networks leverage our graph structure to create advanced features ultimately improving performance, it is interesting to analyze the increase in performance in dependence of a time-series' node degree centrality. We are going to analyze errors of the TFT based models from the 14-day horizon, since the delta in error was most significant there.

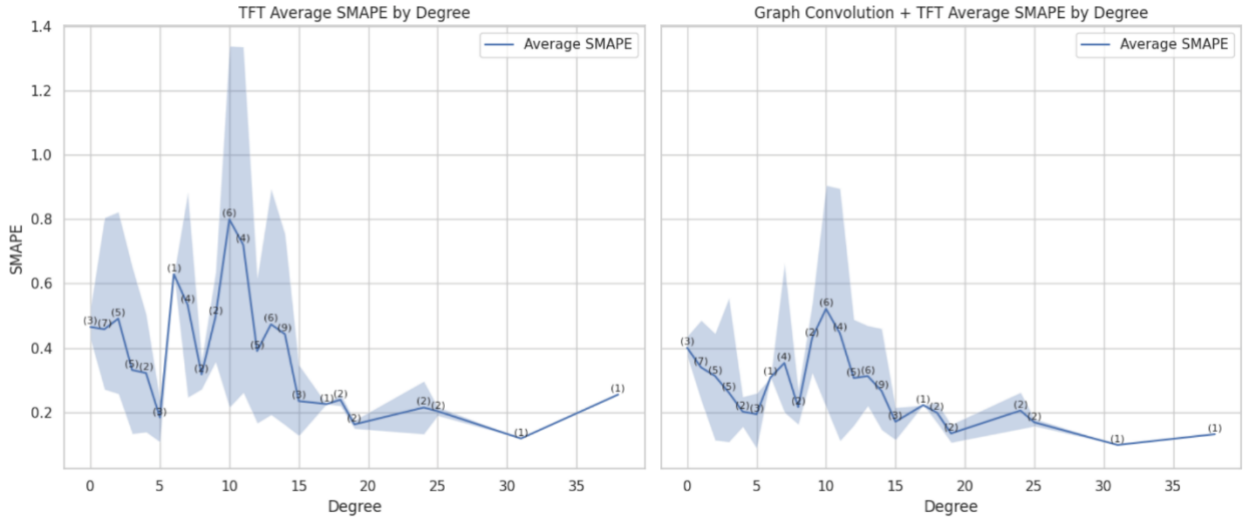

Figure 29: Comparison of average SMAPE of TFT models by degree on 14-day horizon

Most significant performance increases can be seen in the degree range from 5 to 15 (see Figure 29; the blue area shows the range of SMAPE values for the given degree). For nodes below and above this degree threshold, increases in performance are minor, with the highest degree node (degree 38) as a lone outlier with significant improvement.

D.4.3. Parametrisation of Graph Convolutional Layer

To maximize the effectiveness of our graph convolutional layer, we applied tuning via grid search to the hyperparameters aggregation-type, combination-type, activation function and number of output features. The most successful aggregation type was "mean", calculating a mean of neighbors CPC values, and concatenating this value with the original nodes CPC value (combination type "concat" preferred to "add") while using linear activation as supposed to a non-linear Rectified Linear Unit (ReLU) alternative. It also turned out to be most effective to create just one additional feature next to original CPC.

D.5. Discussion

This study establishes the approach of modeling competitive dynamics using a keyword graph and subsequently using this graph in deep neural networks, specifically via graph convolution, as a viable method to improve short term predictive performance of keyword-level CPC.

Because this method seems to have diminishing returns with increasing horizons, our hypothesis $H_{D.1}$ – stating that GNN based models improve accuracy in time-series forecasting of keyword-level CPC – can only be accepted when limited to time horizons smaller or equal to 30 days and must be rejected for longer time horizons.

The analysis in error by node degree indicates that advertiser-keywords with 5 to 15 neighbors can draw the most information from message passing of their neighbors. A possible reason could be that 5 to 15 neighbors are enough to build a robust competition feature, but not so many as to dilute relevant patterns during message passing via graph convolution.

From the optimal parametrization of the graph convolutional layer (aggregation="mean", combination="concat", activation="None"), we can derive that next to the original CPC value, models perform best when using information informed by the average CPC bid of surrounding keywords without applying nonlinear transformation.

The more basic LSTM architecture performing very similar to TFT and even outperforming it on some horizons indicates the TFTs complexity making it prone to overfitting the training data. Even though the graph convolution caused significant improvements in the short horizons, it needs to be acknowledged, that models using it still do not beat SARIMA from Chapter 4. Gains in accuracy might be related to limitations of the custom models build for this part and possibly will not translate when combining them with architectures that are more powerful by default.

D.6. Conclusion

D.6.1. Summary of Findings

Our results show that for our dataset adding a graph convolutional layer significantly improves predictive performance of very simple forecasting components (LSTM, 64 units) as well as more advanced TFT forecasting architectures on short to mid-length time-horizons (14-day, 30-day). This can be attributed to successful modeling of competitive bidding dynamics and their impact on CPC which was prior discussed on a similar dataset in Oldenburg, Han and Kaiser's (2024) paper "*Interpretable Deep Learning for Forecasting Online Advertising Costs: Insights from the Competitive Bidding Landscape"*. While this past paper analyzed competitive effects on CPC at advertiser level, we were able to propose a different way to uncover and use these effects on keyword level.

D.6.2. Limitations

Because of the closed source nature of our chosen benchmarked models by Nixtla, we were not able to train models with graph convolution leveraging the exact same forecasting setup as in Chapter 4. We substituted with custom TFT and LSTM architectures, that did not perform as well as the Nixtla variants. While significant increases in forecast accuracy could be observed when comparing our architectures with and without a graph convolutional layer, this must not be translated to the more powerful Nixtla models without explicit testing. We also did not look at combinations of other powerful models like NHITS or NBEATS that yielded good results in our benchmark in combination with upstream graph convolution. Therefore, we do not have data to evaluate whether our proposed graph-based approach could beat high performing baselines constructed in Chapter 4 when combined with more powerful models.

Additionally, our research is limited by the small dataset only encompassing one year of data for 78 advertiser-keywords in the training set. The focus on car-rental industry limits the transferability to other advertising domains without further validation.

D.6.3. Outlook

Drawing from the promising first results of combining a graph convolutional layer with downstream forecasting architectures – more precisely LSTM and TFT – future research should explore more different combinations of graph convolution and powerful models. Additionally, an application of this approach to keyword-graphs of different sizes could yield a more robust estimation of effectiveness. With more precises data of keyword auction participants and bids, it would also be possible to create a dynamic graph representation of daily auction interactions instead of relying on a static representation of interactions across the complete training timeframe. The highly versatile (see foundational models) attention mechanism used in TFT could also be made use of in a graph attention layer replacing the graph convolution layer. Lastly, it would also be interesting to see whether our approach can be leveraged in similar domains, especially where competitive dynamics of auctions need to be modeled.

E. Evaluating and Fine-Tuning Foundational Models

E.1. Introduction

Deep learning models have recently gained prominence as a powerful tool for time-series forecasting, as highlighted in Chapter 2. At the same time, the rapid progress in Natural Language Processing (NLP) has demonstrated the transformative potential of large foundational models (FM), particularly in their ability to generalize across tasks and domains. Large language models (LLMs), such as GPT, have shown remarkable capabilities in generating text, translating languages, and handling diverse creative and analytical tasks (Brown et al. 2020). This success has inspired research into whether similar FMs, trained on vast time-series datasets, can effectively learn temporal patterns for forecasting on unseen data, enabling zero-shot learning.

Recent advancements have introduced FM for time-series forecasting which leverage large-scale pretraining to deliver promising results. These models eliminate the need for extensive task-specific retraining by offering efficient and adaptable solutions across domains and applications.

This chapter applies these three state-of-the-art zero-shot foundational models - TimeLLM, TimesFM, and TimeGPT - to the specific research question:

> *RQ_4: How does the forecasting accuracy of foundational models compare to state-of-the-art benchmarks for predicting daily keyword CPC in competitive online advertising markets?*

This part of the study begins with a theoretical framework outlining the principles and paradigms of foundational models, followed by an introduction to the selected models and their capabilities. The methodology details the experimental setup, covering baseline configurations, fine-tuning approaches, and testing with exogenous variables, with particular emphasis on TimeGPT's unique ability to integrate additional features. Finally, the results and discussion evaluate the models' performance across two scenarios - baseline and fine-tuned configurations - assessing their ability to capture CPC dynamics and their overall effectiveness in addressing the research question.

E.2. Theoretical Background

This chapter provides a comprehensive overview of FMs and their application in time-series forecasting. It explores their underlying principles, key paradigms, and relevance for predicting keyword-level CPC in competitive online advertising markets.

E.2.1. Principles and Applications of Foundational Models

Foundational models (FMs) are large pre-trained models designed to leverage vast amounts of data to learn general-purpose representations that can be adapted for a variety of downstream tasks with no or minimal additional training. Originally popularized in fields like natural language processing (e.g., GPT-3) FMs have revolutionized how tasks are approached by emphasizing scalability, versatility, and efficiency (Bommasani et al. 2022).

In time-series analysis, FMs aim to generalize across diverse temporal datasets, which helps to capture time-series specific dependencies such as trends and seasonality. By employing architectures like transformers, these models benefit from capabilities such as long-range sequence modeling and parallelization, which make them particularly suited for handling large-scale sequential data (Vaswani et al. 2023).

FMs can be applied directly in zero-shot scenarios, where the model generates predictions for tasks it has not been explicitly trained on by only using its pre-existing knowledge gained during pretraining. This requires no additional task-specific training or fine-tuning, which makes zero-shot learning highly efficient for quickly deploying models in new domains or tasks. For example, a FM trained on diverse datasets can forecast time-series data in a new domain without requiring labeled examples from that domain.

In contrast, few-shot scenarios involve providing the model with a limited number of labeled examples or task-specific demonstrations. These examples, often included in the input as part of the prompt or during minimal fine-tuning, guide the model in adapting its general knowledge to the specific requirements of the task. Few-shot learning bridges the gap between zero-shot and full

fine-tuning, enabling models to perform well with minimal additional data and training, while still benefiting from their pre-trained versatility.

E.2.2. Paradigms of Foundational Models for Time-Series Forecasting

The recent advancements in introducing FMs for time-series forecasting can be split into three categories:

(a) **Prompting Large Language Models (LLMs):** Pre-trained LLMs can be used for numerical time-series forecasting by encoding the data as text and generating predictions in a natural language format, without requiring fine-tuning or architectural modifications tailored for time-series tasks. While LLMs excel at recognizing general patterns in data, directly prompting them without adapting their architecture to account for time-series-specific characteristics limits their forecasting effectiveness. Recent studies show that even with extensive prompt engineering and the use of dataset-specific templates, models like LLMTime still exhibit constrained performance. This is especially apparent when compared to purpose-built models or frameworks specifically adapted for time-series forecasting, as demonstrated in paradigm (b) (Ansari et al. 2024). For this reason, paradigm (a) will not be considered for this study.

(b) **Fine-tuning pre-trained LLMs**: This approach adapts pre-trained language models for time-series forecasting by introducing specialized adapter layers and applying task-specific fine-tuning. Fine-tuning involves modifying certain components of the pre-trained LLM to align its general-purpose architecture with the unique characteristics of time-series data, such as sequential dependencies, temporal patterns, and seasonality. Recent studies highlight the effectiveness of this approach, exemplified by TimeLLM, which surpasses specialized forecasting models on selected benchmarks and performs exceptionally well in few-shot and zero-shot scenarios (M. Jin, Wang, et al. 2024).

(c) **Pretraining transformer-based models from scratch:** This approach involves training transformers explicitly designed for time-series forecasting, with architectures tailored to

capture temporal dependencies and adapt to varying forecasting horizons effectively. Unlike LLM-based models, which adapt general-purpose architectures not optimized for time-series data, models like TimeGPT and TimesFM are purpose-built for this domain, aligning more closely with the unique requirements of time-series tasks. Studies demonstrate that this approach outperforms established models such as TFT and DeepAR on diverse datasets (Azul Garza et al. 2024, 6). These findings highlight the potential of pre-trained transformers as scalable, specialized solutions for time-series forecasting (Das et al. 2024).

E.2.3. Selected Foundational Models for CPC Prediction

For this study, TimeLLM, TimeGPT, and TimesFM have been selected as state-of-the-art FMs for time-series forecasting. These models exemplify two of the most promising paradigms in FM development: fine-tuning pre-trained LLMs and pretraining transformer-based models explicitly for time-series tasks. Their architectures, methodologies, and applications will be explored in detail in the following sections.

E.2.3.1. TimeLLM

TimeLLM, developed as part of a collaborative research effort, represents a novel application of LLMs to time-series forecasting (see paradigm b). Unlike models specifically pre-trained on time-series data, TimeLLM reprograms existing LLMs, such as GPT-2, to handle time-series tasks while preserving their original architectures. This is achieved through techniques like Prompt-as-Prefix (PaP), which integrates task-specific instructions and domain knowledge into input prompts. Additionally, patch reprogramming transforms structured time-series data into text-based formats compatible with LLM processing. This enables the seamless application of these models to sequential data tasks (M. Jin, Wang, et al. 2024).

The reprogramming layer acts as a bridge between the numerical time-series data and the text-based processing capabilities of the pre-trained LLM (M. Jin, Wang, et al. 2024). This ensures that the model can make sense of time-series-specific patterns such as seasonality and temporal

dependencies. However, this layer requires task-specific training to tailor the prompts and data representation to the unique characteristics of each dataset. This dependency makes zero-shot forecasting infeasible, as the model cannot directly generalize from its pre-trained knowledge without the additional context and adaptations provided by the reprogramming layer.

TimeLLM does not modify the pre-trained LLM's parameters but relies on lightweight transformations at the input and output layers (M. Jin, Wang, et al. 2024). This approach supports various use cases by leveraging the pre-existing knowledge of LLMs while introducing minimal task-specific adaptations. Unlike models designed specifically for time-series data, TimeLLM's reliance on text-based representations introduces a different approach by combining natural language and numerical reasoning capabilities (M. Jin, Wang, et al. 2024).

E.2.3.2. TimeGPT

Unlike existing LLMs repurposed for numerical tasks, TimeGPT – developed by Nixtla - is specifically designed for time-series data. Its architecture is transformer-based, which helps incorporating self-attention mechanisms to capture diverse temporal patterns. It features an encoder-decoder structure with multiple layers, each employing residual connections and layer normalization, and a final linear layer mapping outputs to the forecasting window (Garza, Challu, and Mergenthaler-Canseco 2024). The parameter size of TimeGPT has not been disclosed.

Following the principle of training large transformer models on extensive datasets, TimeGPT was developed using the largest publicly available collection of time-series data. This collection contains over 100 billion data points across domains such as finance, demographics, and energy. The given data diversity equips TimeGPT to handle a wide range of temporal patterns, including trends, seasonality, and varying frequencies which makes it effective for complex forecasting scenarios (Garza, Challu, and Mergenthaler-Canseco 2024). TimeGPT also supports the integration of exogenous variables, including static, historical, and future covariates. This enhances its ability to model real-world scenarios where external factors impact forecasts. The

model offers an integrated possibility to train the model on specific data, which enables a few-shot learning approach.

E.2.3.3. TimesFM

TimesFM, developed by Google Research, also is a FM specifically designed for time-series forecasting. Unlike models such as TimeGPT with encoder-decoder architectures, TimesFM employs a decoder-only transformer architecture optimized for efficiency in sequence modelling (Das et al. 2024). This design provides linear complexity, compared to quadratic complexity of encoder-decoder models, which enables the model to handle long time-series data more efficiently. Its architecture is tailored to capture both global patterns across multiple series and localized temporal dynamics (Das et al. 2024).

TimesFM has a parameter size of 200 million and like TimeGPT, it was trained on approximately 100 billion data points (Das et al. 2024). These datasets include real-world sources such as Google Trends and Wikipedia page views, as well as synthetic data generated using statistical models (Das et al. 2024). This diverse training corpus equips TimesFM to generalize effectively to unseen data which can enable robust zero-shot performance across a variety of forecasting horizons. TimesFM also supports the integration of exogenous variables, including static and future covariates. However, it does not support historical exogenous variables, differentiating it from other FM like TimeGPT. Also, the TimesFM library lacks a built-in functionality for task-specific fine-tuning.

	TimeGPT	**TimesFM**	**TimeLLM (GPT-2)**
Parameter Size	Not disclosed	~200 million	~1,5 billion
Training data size	100B data points	100B data points	8M web pages
Historical Exogenous Features Support	✅	❌	❌
Static Exogenous Features Support	✅	✅	✅
Future Exogenous Features Support	✅	✅	✅
Fine-Tuning Capability	Task-specific training	None	Prompt-as-Prefix

Table 14: Characteristics of the selected Foundational Models

E.2.4. Hypotheses and Research Gaps

The application of FMs in the domain of CPC prediction remains largely unexplored. While foundational models such as TimeGPT, TimesFM, and TimeLLM have shown promise in time-series forecasting tasks across various domains, there is no documented research examining their effectiveness for predicting CPC, particularly at the keyword level, in the context of dynamic and competitive paid search markets. This gap raises questions about their suitability for capturing the temporal and competitive dynamics of CPC, a critical metric in digital advertising.

Moreover, the role of exogenous variables, such as keyword-specific KPIs (e.g., click-through rates, impressions) and time features (e.g., day of the week), has not been comprehensively analyzed in the context of foundational models for CPC forecasting.

To address these gaps, the study proposes the following hypotheses:

> *$H_{E.1}$: Foundational Models can predict Keyword CPC with greater accuracy than the established statistical, machine learning and neural network approaches*
>
> *$H_{E.2}$: Incorporating time features and keyword KPIs improves the forecasting accuracy of Keyword CPC predictions made by foundational models*

By addressing these hypotheses, this study aims to contribute to both theory and practice by exploring the applicability of foundational models in CPC prediction and examining the impact of exogenous variables in enhancing forecasting accuracy. These insights are expected to fill a critical gap in the literature and provide actionable recommendations for advertisers.

E.3. Methodology

E.3.1. Experimental Setup

To ensure comparability with the baseline results established in the group project, we will maintain consistent experimental conditions, as detailed in Chapter 4. This includes using the same dataset, evaluating performance over 14-, 30-, and 60-day horizons, and assessing models with MAE and SMAPE, prioritizing MAE for fine-tuning. A sliding window validation ensures robust testing across temporal segments

For each model, we will first establish

- a **baseline** using the model's default configuration.
- a **finetuned model**, if applicable. The individual finetuning approaches of each model will be presented in chapter E3.2.
- a **finetuned model incorporating exogenous variables** as input. Since TimeGPT is the only FM of this study that supports the integration of historical exogenous variables, this approach will be exclusively applied to TimeGPT.

Even though exogenous features did not consistently improve forecasting accuracy in our previous comparison of statistical, ML, and neural network models, this does not imply that they cannot perform well in foundational models. According to the No Free Lunch theorem, no single model or approach is universally superior across all problems; performance depends on the specific data and context (Wolpert and Macready 1997). Thus, the inclusion of exogenous features may still enhance forecasting accuracy in foundational models if the data and task align well with their strengths.

Therefore, we will test whether relative keyword KPIs, absolute keyword KPIs, and time features individually enhance TimeGPT's performance. If multiple features demonstrate improvement, we

will combine them to evaluate their joint impact on the model's accuracy. The specific setup for each FM is detailed in the following implementation details.

E.3.2. Implementation Details

E.3.2.1. TimeLLM

We will implement TimeLLM using the version available in the Nixtla library, which simplifies the integration process. Due to the nature of the reprogramming layer, it only allows training the model on specific data before forecasting, meaning a zero-shot scenario is not possible, and we will rely on a few-shot approach. The underlying LLM for this implementation will be *GPT-2* by default. As described in the theoretical background, TimeLLM utilizes the PaP functionality, which provides context to the model and improves forecast quality. The baseline configuration won't involve a PaP setup with additional business or data context. For the fine-tuned version, we will use a detailed and contextually rich prompt, incorporating insights from the EDA, such as the positive upward trend in CPC data, to improve the model's forecasting accuracy.

E.3.2.2. TimeGPT

We implemented TimeGPT using the API provided by Nixtla. The model supports both zero-shot and few-shot scenarios. The zero-shot prediction serves as the baseline. For the fine-tuned version, we trained the model on our specific dataset to adapt it to our forecasting context. Nixtla's API enables customization of two key fine-tuning parameters: finetuning depth and finetuning steps.

- **Finetuning depth** controls the number of parameters adjusted during fine-tuning, with options ranging from 1 (minimal adjustments) to 5 (adjusting all parameters). Higher depths increase both computational requirements and the risk of overfitting.
- **Finetuning steps** determine the number of training iterations on our dataset. We will use the proposed setting in Nixtla's examples of 10 steps to ensure adequate learning.

To identify the optimal finetuning depth, we conducted a grid search. This approach ensures we select the configuration for each forecast horizon that delivers the best results in terms of forecasting accuracy.

Nixtla provides two model versions: TimeGPT-1 and TimeGPT-1-Long-Horizon. For this study, we will use the Long-Horizon version in all configurations, as Nixtla specifies it is designed for forecasting horizons of 14 days or longer with daily data (Nixtla 2024). Since our minimum forecast horizon is 14 days, the Long-Horizon version aligns with the requirements of our analysis.

E.3.2.3. TimesFM

TimesFM was used directly through the TimesFM library as no integration in Nixtla is available. We utilized the pretrained model weights from the "timesfm-1.0-200m" checkpoint (Google 2024b). Unlike the other models, we established only a baseline model without additional fine-tuning. This decision was due to limitations in the library and resource constraints, as the TimesFM library lacks built-in fine-tuning functionality for specific datasets. While a proposed notebook on their GitHub repository outlines a method for fine-tuning, it cannot be directly used in Google Colab due to conflicts between required libraries and the Colab environment.

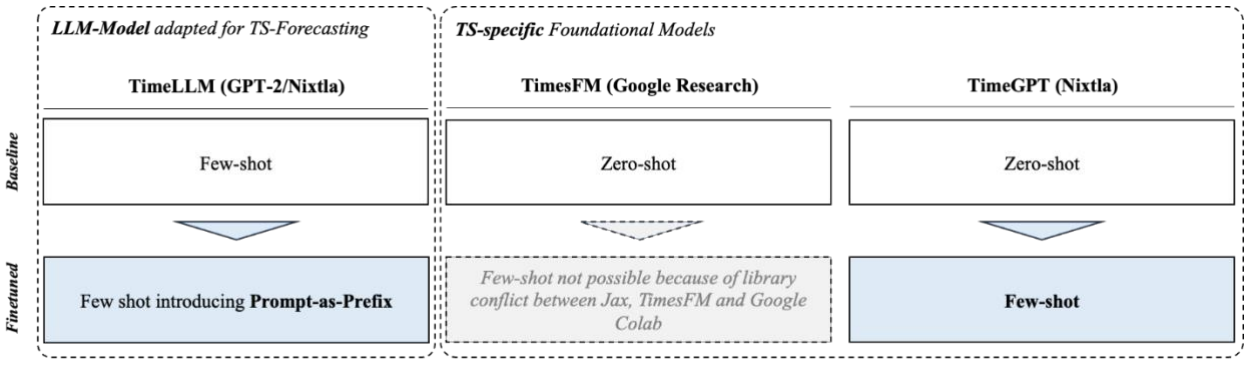

Figure 30: Selected Foundational Models and their finetuning capabilities

E.4. Results

The analysis provides several key insights into the performance of the models across different prediction horizons. It examines Zero-Shot and Few-Shot learning configurations for TimeGPT, as well as the introduction of the PaP method for TimeLLM. Additionally, it explores the impact of exogenous features on TimeGPT's forecasting accuracy. Table 15 includes a detailed breakdown of the results of all foundational models in all different configurations, alongside the best-performing models from the previously established baseline, which are SARIMA for the short- and mid-term horizons (14 and 30 days) and TFT with relative keyword KPIs for the 60-day horizon.

	Model	14 days		30 days		60 days	
		MAE	SMAPE	MAE	SMAPE	MAE	SMAPE
Group Part Baseline	SARIMA	0.491	0.263	**0.529**	**0.300**	0.645	0.352
	TFT univ. + rel_keyword_KPIs	0.522	0.275	0.581	0.331	**0.586**	**0.328**
TimeLLM	TimeLLM_few_shot	0.928	0.455	0.934	0.503	0.821	0.454
	TimeLLM_few_shot + prompt_as_prefix	0.707	0.354	0.862	0.468	0.778	0.434
TimesFM	TimesFM_zero_shot	0.536	0.265	0.557	0.308	0.616	0.354
TimeGPT	TimeGPT_zero_shot	**0.485**	**0.260**	0.530	0.304	0.643	0.346
	TimeGPT_few_shot	0.487	0.261	0.533	0.304	0.635	0.342
	TimeGPT_few_shot + rel_keyword_KPIs	0.656	0.366	0.757	0.417	1.060	0.521
	TimeGPT_few_shot + abs_keyword_KPIs	0.763	0.420	1.009	0.498	1.664	0.666
	TimeGPT_few_shot + time	0.626	0.339	0.646	0.349	0.841	0.413

0.123 best model within all foundational models
0.123 best model across all previously established state-of-the-art + foundational models

Table 15: Forecasting results comparison – Chapter E

Zero-Shot models, particularly TimesFM Zero Shot and TimeGPT Zero Shot, demonstrate strong performance across all horizons. TimeGPT Zero Shot achieves the lowest MAE (0.485) and SMAPE (0.260) on the 14-day horizon, outperforming all other models, including the previously

established baseline models. For the 30-day horizon, TimeGPT Zero Shot maintains the lowest MAE (0.530), while both TimeGPT and TimesFM achieve the lowest SMAPE (0.304) among foundational models, though SARIMA outperforms both overall. On the 60-day horizon, TimesFM Zero Shot achieves the best MAE (0.616) and TimeGPT the best SMAPE (0.354) among foundational models, though both are outperformed by TFT with exogenous covariates.

Few-Shot configurations of TimeGPT generally perform slightly worse than Zero-Shot, especially on shorter horizons, with minimal improvements for the 60-day horizon. TimeLLM shows consistently poor performance across all horizons, even with the PaP method. The inclusion of exogenous features, such as keyword KPIs, further degrades performance, highlighting their limited utility in this context.

Overall, TimeGPT Zero Shot performs best on short and medium horizons, while TimesFM excels in long-term predictions among foundational models. Few-Shot learning, exogenous features, and TimeLLM show limited suitability for CPC forecasting at the keyword level.

E.5. Discussion

The study sheds light on the potential of FM for time-series forecasting, specifically for predicting daily CPC of keywords in competitive paid search advertising markets. The findings reveal key insights into the performance and practicality of FM models.

FM models specialized for time-series forecasting, in the case of this study TimesFM and TimeGPT, demonstrated very good performance. TimeGPT outperformed individually trained baseline models on the short horizon of 14 days, providing partial confirmation of $H_{E.1}$ that foundational models can predict CPC with greater accuracy than established baseline approaches. Remarkably, the zero-shot configurations of TimeGPT and TimesFM performed best on the shorter horizons (14 and 30 days). This suggests that, for short-term forecasting, the zero-shot approach is not only simpler but also more effective. TimeGPT Zero Shot emerged as the best-performing model for the shortest horizon, while its performance on longer horizons converged

with that of TimesFM. Interestingly, for long-term forecasting TimeGPT's few-shot configuration slightly surpassed its zero-shot counterpart, though the margin was minimal. These findings suggest that TimeGPT is the optimal choice for short-term predictions, while both TimeGPT and TimesFM perform comparably for medium- and long-term forecasting.

The study also highlighted the limited suitability of TimeLLM for CPC prediction. While the PaP method significantly improved its performance, TimeLLM's results remained considerably worse than those of other models. This makes it a less favorable choice for our research purpose.

The inclusion of exogenous features significantly reduced the forecasting performance of TimeGPT. This leads to the rejection of $H_{E.2}$, which hypothesized that incorporating exogenous variables, such as time features and keyword KPIs, would improve the forecasting accuracy of keyword CPC predictions made by foundational models. These features appear to introduce additional complexity or noise, which reduces the effectiveness of the model. Similarly, the few-shot scenario failed to provide meaningful additional value and, in some cases, slightly worsened performance, particularly on shorter horizons. While there was a slight improvement for the 60-day horizon, it was not enough to justify the added complexity of fine-tuning. These findings emphasize the need for simpler, well-generalized configurations for CPC prediction of keywords. This may be attributed to the presence of time-series with limited seasonality or inherently high unpredictability, which make it challenging for the model to generalize effectively.

While previous research positioned foundational models as practical, this study demonstrates their potential as a powerful and reliable solution for predicting CPC at the keyword level. One of the key advantages of foundational models is their ease of use, as they require no complex data pipelines or model architecture modifications. Additionally, as pretrained models, they demand significantly less computational power which makes them faster and more accessible compared to traditional methods. These strengths position foundational models as a practical and efficient

choice for time-series forecasting, particularly in scenarios where computational resources are limited or rapid deployment is required.

E.6. Conclusion

E.6.1. Summary of Findings

This study demonstrates the potential of foundational models as effective tools for time-series forecasting in the context of CPC prediction. In answering the research question "*How does the forecasting accuracy of foundational models compare to state-of-the-art benchmarks for predicting daily keyword CPC in competitive online advertising markets?"* the findings reveal that foundational models, particularly in their zero-shot configurations, perform exceptionally well for short-term horizons. TimeGPT and TimesFM, when used without task-specific fine-tuning, delivered strong performance, outperforming traditional models in short-term forecasting. On longer horizons, foundational models, while slightly behind the best baseline approaches such as TFT with exogenous covariates, performed nearly as well. This highlights their ability to generalize across data with minimal customization, making them practical and efficient solutions for forecasting CPC in competitive markets.

However, task-specific adaptations, such as few-shot learning and the inclusion of exogenous variables, failed to enhance accuracy and, in some cases, worsened performance. This was particularly evident in our CPC dataset, characterized by high noise and low seasonality, where added complexity amplified overfitting risks. Despite these limitations, foundational models proved scalable, computationally efficient, and well-suited to address the inherent challenges of keyword-level CPC prediction, positioning them as valuable tools for diverse forecasting applications in dynamic online advertising environments.

E.6.2. Limitations

This study encountered several constraints that impacted the scope and findings. First, the few-shot scenario for TimesFM could not be implemented due to library conflicts. The fine-tuning

notebook provided in the TimesFM repository was incompatible with the chosen environment, which prevented an evaluation of the model's few-shot learning capabilities.

Second, developing an effective PaP for TimeLLM proved to be challenging. The model's performance was highly sensitive to minor changes in the prompt, with slight modifications either significantly improving or worsening results. This variability introduced uncertainty, making it difficult to fully leverage the model's potential.

Finally, exogenous variables were tested only with TimeGPT, as the functionality to include exogenous inputs is not yet available for the other foundational models. Even with TimeGPT, these features did not improve forecasting accuracy, possibly due to the characteristics of the CPC dataset, which is marked by high noise and low seasonality.

E.6.3. Outlook

Building on these limitations, future research can address several opportunities.

First, the few-shot learning capabilities of TimesFM should be explored by resolving library conflicts and experimenting with alternative environments or setups. This would allow for a more comprehensive evaluation of its adaptability and performance under few-shot conditions.

Second, systematic methods for optimizing prompts in TimeLLM should be developed to enhance consistency and forecasting accuracy. Techniques such as adaptive or automated prompt-tuning could further reduce variability and improve results.

Third, the impact of exogenous variables should be revisited, either by testing models that already support such inputs or by extending this functionality to other foundational models like TimesFM. Moreover, refining the selection and engineering of exogenous variables tailored to datasets with high noise and low seasonality, such as CPC, could yield better insights. Finally, future studies should evaluate foundational models across a broader range of datasets and forecasting tasks to assess their scalability, adaptability, and performance under varying data characteristics.

9. Summarizing Conclusion and Practical Implications

This chapter will first give a summary of results from the first benchmark study and our subsequent deep dives. Consequently, we will use our insights to discuss practical implications for advertisers in paid search advertising wanting to gain an advantage by establishing keyword CPC predictions.

9.1. Summary of Findings

Our research can be split into three main sections. First, we conducted a literature review to identify and specify the research gap in keyword-level CPC forecasting. We adapted our methodology to best fit the findings of this theoretical review. Through our EDA, we gathered key insights that complemented findings of the literature review and informed the model setup. Statistical, ML and neural network models that achieved excellent results in time-series benchmark competitions were implemented, and their forecasting results compared using the average MAE and SMAPE to evaluate their suitability across short-, mid-, and long-term horizons using different sets of features were applicable. This section aimed to answer our research questions:

> *RQ_1: Which state-of-the-art modeling approach achieves the highest accuracy for keyword-level CPC prediction in competitive paid search advertising?*
>
> *RQ_2: Does the integration of select exogenous variables enhance the forecasting accuracy of keyword-level CPC predictions?*

We established the statistical SARIMA with hyperparameters finetuned to our data as the best performing model for 14-day and 30-day prediction horizon. On the longest 60-day horizon, the TFT was able to slightly outperform it. It did so leveraging exogenous variables, more specifically relative keyword KPIs we derived from the original data. In all other models and horizons, exogenous variables did not enhance forecasting accuracy beyond baseline results.

In a second step covering the deep dive chapters A to D, multiple approaches to improve upon these benchmarks were developed. The approaches focus on how to incorporate competition into

the forecasting models. Incorporating competition data was hypothesized to support the improvement of forecasting accuracy based on a prior study by Oldenburg, Han and Kaiser (2024). The guiding research question for these deep dives was set as:

> *RQ_3: Can advertisers benefit from incorporating information on competition into their keyword-level CPC forecast?*

To answer this, we explored different methods of incorporating information on competition in our predictive models. We applied time-series clustering with distance- and characteristics-based methods (Chapter A) and keyword semantics (Chapter B), assuming that these clusters could represent groups of competing keywords. In Chapter C admatchedquery was introduced for deriving keywords participation in auctions and additional features were constructed to use this knowledge in our models. Chapter D went one step further using this admatchedquery information to derive a graph representation of intercorrelated time-series and leveraging this for time-series predictions with graph neural networks. Besides minor exceptions, the results showed no improvements of forecasting accuracy beyond the prior established baselines. Therefore, our findings suggest that incorporating competition data into keyword-level CPC forecasting does not lead to meaningful improvements in forecasting accuracy.

In the third section (deep dive chapter E), we explored the emerging trend of Foundational Models, including those derived from LLMs and time-series-specific approaches, to explore the research question:

> *RQ_4: How does the forecasting accuracy of foundational models compare to state-of-the-art benchmarks for predicting daily keyword CPC in competitive online advertising markets?*

Our analysis demonstrated promising results for time-series-specific FMs in zero-shot and few-shot inference, partially outperforming the benchmark models. Task-specific training did not consistently improve the results, and the inclusion of time-based features or keyword KPIs even

worsened them, possibly due to overfitting and the added complexity introducing additional noise. These findings are particularly noteworthy given the challenging nature of our CPC data, characterized by low seasonality and high noise.

9.2. Limitations and Outlook

This study exhaustively outlined overall limitations in Chapter 7.2, along with those specific to each deep dive in their respective limitation chapters. In this section, we will focus only on the most crucial constraints that are relevant across all parts of our study and discuss how future research can address them.

The main factor limiting all our results is our filtered dataset. We filtered for keywords time-series that are at least 98% complete. This resulted in a small dataset of only 78 keywords, which is at risk of showing an incomplete representation of the market. We also only used 12 months of data for training, restricting the analysis of long-term trends and seasonality, which would need to be present more frequently to be best picked up by our forecasting models. Furthermore, we missed features like budget, which was a highly important variable in prior research on advertiser-level (Oldenburg, Han and Kaiser 2024) and quality score, which is currently not disclosed by Google but would be promising to better model keyword auctions. The dataset's focus on the car rental industry, while benefiting from small amount of market participants, limits applicability to other industries with more diverse competition.

Looking forward, future research should expand our analysis using a more extensive dataset with more keywords from multiple industries, to ensure a comprehensive representation of market dynamics in different domains. Extending the temporal scope of the training set would allow improved modeling of long-term trends. Additional features like budget and quality score should be explored for their predictive value on keyword-level CPC. Lastly, further research might experiment with new forecasting models and approaches to include competition.

9.3. Practical Implications

Being aware of these limitations, our advice to practitioners in paid search advertising, seeking to gain an advantage by establishing keyword-level CPC forecasts, emphasizes both accuracy and practicality. Based on our experiments, zero-shot inference using TimeGPT offers the best balance of both qualities. It achieved best or highly competitive accuracy on all time horizons while requiring minimal implementation effort, relying solely on Nixtla's API. Moreover, no additional parameterization or fine-tuning is necessary, as it demonstrated robust performance out of the box.

For advertisers who are unwilling to adapt a pay per use model of TimeGPT, our advice must be differentiated based on the context: For short to medium forecasting horizons (14-day to 30-days), we recommend using SARIMA as a computationally inexpensive - but in our experiments most accurate – model. Forecasts across these horizons support immediate campaign adjustments, such as budget reallocation and bid optimization in response to market changes, seasonal trends or competitor activity. They also inform optimized monthly keyword and budget planning. For long-term forecasting (60 days or more), exogenous features should be incorporated into neural network models, preferably TFT, to achieve optimal results for strategic decisions like quarterly budgeting and long-term resource allocation. As we could not find ways to benefit from incorporating competition data on any time-horizon or model setup, our advice for advertisers is to focus on their own historical data when trying to improve keyword-level CPC forecasts.

In line with Bill Gates' prediction, online advertising is steadily continuing its growth in relevance, reinforcing its role as the future of advertising. Advertisers that adopt our recommendations should be able to leverage their keyword-level CPC predictions to more efficiently allocate budgets, ultimately increasing ROAS in paid search advertising – one of the cornerstones of online advertising.

List of Figures

List of Tables

List of Equations

List of Abbreviations

Abbreviation	Definition
API	Application Programming Interface
ARI	Adjusted Rand Index
ARIMA	Autoregressive Integrated Moving Average
CNN	Convolutional Neural Network
CPC	Cost-per-Click
CR	Conversion Rate
CTR	Click-Through Rate
DTW	Dynamic Time Warping
EDA	Exploratory Data Analysis
FCM	Fuzzy C-Means
FM	Foundational Models
GC	Graph Convolution
GCN	Graph Convolutional Network
GNN	Graph Neural Network
GPU	Graphics Processing Unit
GRU	Gated Recurrent Unit
GSP	Generalized Second Price
IQR	Interquartile Range
KPI	Key Performance Indicator
LLM	Large Language Model
LSTM	Long Short-Term Memory
MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Error
ML	Machine Learning
MSE	Mean Squared Error
NBEATS	Neural Basis Expansion Analysis for Interpretable Time Series Forecasting
NHITS	Neural Hierarchical Interpolation for Time Series Forecasting
NLP	Natural Language Processing
NMI	Normalized Mutual Information
PaP	Prompt-as-Prefix
ReLU	Rectified Linear Unit
RNN	Recurrent Neural Network
ROAS	Return on Ad Spend
SARIMA	Seasonal Autoregressive Integrated Moving Average
SEO	Search Engine Optimization
SERP	Search Engine Results Page
SMAPE	Symmetric Mean Absolute Percentage Error
STGCN	Spatio-Temporal Graph Convolutional Network
TFT	Temporal Fusion Transformer
WCSS	Within-Cluster Sum of Squared distances

References

Abhishek, and Vikas Khullar. 2024. “Exploring Techniques and Approaches for Time Series Analysis and Its Attacks.” *2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, 1–6. https://doi.org/10.1109/ICCCNT61001.2024.10725845.

Aghabozorgi, Saeed, Ali Seyed Shirkhorshidi, and Teh Ying Wah. 2015. “Time-Series Clustering – A Decade Review.” *Information Systems* 53 (October):16–38. https://doi.org/10.1016/j.is.2015.04.007.

Ansari, Abdul Fatir, Lorenzo Stella, Caner Turkmen, Xiyuan Zhang, Pedro Mercado, Huibin Shen, Oleksandr Shchur, et al. 2024. “Chronos: Learning the Language of Time Series.” arXiv. https://doi.org/10.48550/arXiv.2403.07815.

Apte, Mohit, and Yashodhara Haribhakta. 2024. “Advancing Financial Forecasting: A Comparative Analysis of Neural Forecasting Models N-HiTS and N-BEATS,” August. https://research.ebsco.com/linkprocessor/plink?id=d6a45e63-f9c6-3a97-8573-54c68cba2ca8.

Athey, Susan, and Denis Nekipelov. 2010. “A Structural Model of Sponsored Search Advertising Auctions https://www.researchgate.net/publication/265024742_A_Structural_Model_of_Sponsored_Search_Advertising_Auctions.

Baluch, Anna. 2023. “60 SEO Statistics For 2024.” Forbes Advisor. November 28, 2023. https://www.forbes.com/advisor/business/software/seo-statistics/.

Bandara, Kasun, Christoph Bergmeir, and Slawek Smyl. 2017. “Forecasting Across Time Series Databases Using Recurrent Neural Networks on Groups of Similar Series: A Clustering Approach,” October. https://research.ebsco.com/linkprocessor/plink?id=7dfeddeb-b4bb-3bb8-a4e7-53f63e2abdbc.

2020. “Forecasting across Time Series Databases Using Recurrent Neural Networks on Groups of Similar Series: A Clustering Approach.” *Expert Systems with Applications* 140 (February):112896. https://doi.org/10.1016/j.eswa.2019.112896.

Bengio, Y., and Yann Lecun. 1997. “Convolutional Networks for Images, Speech, and Time-Series,” November.

Berndt, Donald J., and James Clifford. 1994. “Using Dynamic Time Warping to Find Patterns in Time Series.” In *Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining*, 359–70. AAAIWS’94. Seattle, WA: AAAI Press.

Bernstein, Matthew N. 2023. “Graph Convolutional Neural Networks.” Matthew N. Bernstein. September 24, 2023. https://mbernste.github.io/posts/gcn/.

Bommasani, Rishi, Drew A. Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, et al. 2022. “On the Opportunities and Risks of Foundation Models.” arXiv. https://doi.org/10.48550/arXiv.2108.07258.

Brown, Tom B., Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, et al. 2020. “Language Models Are Few-Shot Learners.” arXiv. https://doi.org/10.48550/arXiv.2005.14165.

Brownlee, Jason. 2018. “Deep Learning for Time Series Forecasting.” *MachineLearningMastery.Com* (blog). 2018. https://www.machinelearningmastery.com/deep-learning-for-time-series-forecasting/.

Bughin, Jacques, Laura Corb, James Manyika, Olivia Nottebohm, Michael Chul, Borja de Muller Barbat, and Remi Said. 2011. “The Impact of Internet Technologies: Search.” High Tech Practice. McKinsey & Company. https://www.mckinsey.com/~/media/mckinsey/dotcom/client_service/high%20tech/pdfs/impact_of_internet_technologies_search_final2.ashx.

Cao, Defu, Yujing Wang, Juanyong Duan, Ce Zhang, Xia Zhu, Congrui Huang, Yunhai Tong, et al. 2020. “Spectral Temporal Graph Neural Network for Multivariate Time-Series Forecasting.” In *Advances in Neural Information Processing Systems*, 33:17766–78. Curran Associates, Inc. https://proceedings.neurips.cc/paper/2020/hash/cdf6581cb7aca4b7e19ef136c6e601a5-Abstract.html.

Cerqueira, Vitor, Luis Torgo, and Carlos Soares. 2019. “Machine Learning vs Statistical Methods for Time Series Forecasting: Size Matters.” arXiv. http://arxiv.org/abs/1909.13316.

Chai, Christine P. 2022. “Comparison of Text Preprocessing Methods.” *Natural Language Engineering* 29 (3): 509–53. https://doi.org/10.1017/S1351324922000213.

Chen, Tianqi, and Carlos Guestrin. 2016. “XGBoost: A Scalable Tree Boosting System.” In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785–94. San Francisco California USA: ACM. https://doi.org/10.1145/2939672.2939785.

Dan, Ovidiu, and Brian D. Davison. 2016. “Measuring and Predicting Search Engine Users’ Satisfaction.” *ACM Comput. Surv.* 49 (1): 18:1-18:35. https://doi.org/10.1145/2893486.

Das, Abhimanyu, Weihao Kong, Rajat Sen, and Yichen Zhou. 2024. “A Decoder-Only Foundation Model for Time-Series Forecasting.” arXiv. https://doi.org/10.48550/arXiv.2310.10688.

Dentsu. 2024. “Global Ad Spend Spend Share by Channel 2022-2026.” Statista. May 2024. https://www-statista-com.eu1.proxy.openathens.net/statistics/245440/distributuion-of-global-advertising-expenditure-by-media/.

Du, Xiaomeng, Meng Su, Xiaoquan (Michael) Zhang, and Xiaona Zheng. 2017. “Bidding for Multiple Keywords in Sponsored Search Advertising: Keyword Categories and Match Types.” *Information Systems Research* 28 (4): 711–22.

Edelman, Benjamin, Michael Ostrovsky, and Michael Schwarz. 2007. “Internet Advertising and the Generalized Second-Price Auction: Selling Billions of Dollars Worth of Keywords.” *American Economic Review* 97 (1): 242–59. https://doi.org/10.1257/aer.97.1.242.

Evans, David S. 2009. “The Online Advertising Industry: Economics, Evolution, and Privacy.” *Journal of Economic Perspectives* 23 (3): 37–60. https://doi.org/10.1257/jep.23.3.37.

Even, Shimon. 1979. “Path Lengths.” In *Graph Algorithms*. Cambridge University Press.

Fischer, Thomas, and Christopher Krauss. 2018. “Deep Learning with Long Short-Term Memory Networks for Financial Market Predictions.” *European Journal of Operational Research* 270 (2): 654–69. https://doi.org/10.1016/j.ejor.2017.11.054.

Freeman, Linton C. 1978. “Centrality in Social Networks: Conceptual Clarification.” *Social Network: Critical Concepts in Sociology. Londres: Routledge* 1:238–63.

Fujimoto, Ted, Joshua Suetterlein, Samrat Chatterjee, and Auroop Ganguly. 2024. “Assessing the Impact of Distribution Shift on Reinforcement Learning Performance.” arXiv. https://doi.org/10.48550/arXiv.2402.03590.

Garza, Azul, Cristian Challu, and Max Mergenthaler-Canseco. 2024. “TimeGPT-1.” arXiv. http://arxiv.org/abs/2310.03589.

Geddes, Brad. 2014. *Advanced Google AdWords*. 3rd edition. Indianapolis, IN: Sybex.

Google. 2024a. “Actual Cost-per-Click (CPC): Definition - Google Ads Help.” Actual Cost-per-Click (CPC): Definition. 2024. https://support.google.com/google-ads/answer/6297?hl=en.

Google. 2024b. “Google/Timesfm-1.0-200m · Hugging Face.” December 13, 2024. https://huggingface.co/google/timesfm-1.0-200m.

Hamilton, William L., Rex Ying, and Jure Leskovec. 2018. “Inductive Representation Learning on Large Graphs.” arXiv. https://doi.org/10.48550/arXiv.1706.02216.

Hamoudia, Mohsen, Spyros Makridakis, and Evangelos Spiliotis, eds. 2023. *Forecasting with Artificial Intelligence: Theory and Applications*. Palgrave Advances in the Economics of Innovation and Technology. Cham: Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-35879-1.

Hochreiter, S. 1997. “Long Short-Term Memory.” *Neural Computation MIT-Press*. https://sophieeunajang.wordpress.com/wp-content/uploads/2020/10/lstm.pdf.

Hopkins, Navah. 2023. “Optmyzr Study: Is It Time to Re-Evaluate Your Google Ads Strategies.” Optmyzr Google Ads Optimization. 2023. https://www.optmyzr.com/blog/optmyzr-study-broad-match-bidding/.

Huang, Xiaohui, Yunming Ye, Liyan Xiong, Raymond Y. K. Lau, Nan Jiang, and Shaokai Wang. 2016. “Time Series *k*-Means: A New *k*-Means Type Smooth Subspace Clustering for Time Series Data.” *Information Sciences* 367–368 (November):1–13. https://doi.org/10.1016/j.ins.2016.05.040.

Jansen, Bernard J., Zhe Liu, and Zach Simon. 2013. “The Effect of Ad Rank on the Performance of Keyword Advertising Campaigns.” *Journal of the American Society for Information Science and Technology* 64 (10): 2115–32. https://doi.org/10.1002/asi.22910.

Javed, Ali, Byung Suk Lee, and Donna M. Rizzo. 2020. “A Benchmark Study on Time Series Clustering.” *Machine Learning with Applications* 1 (September):100001. https://doi.org/10.1016/j.mlwa.2020.100001.

Jin, Ming, Huan Yee Koh, Qingsong Wen, Daniele Zambon, Cesare Alippi, Geoffrey I. Webb, Irwin King, and Shirui Pan. 2024. “A Survey on Graph Neural Networks for Time Series: Forecasting, Classification, Imputation, and Anomaly Detection.” *IEEE Transactions on Pattern Analysis and Machine Intelligence* 46 (12): 10466–85. https://doi.org/10.1109/TPAMI.2024.3443141.

Jin, Ming, Shiyu Wang, Lintao Ma, Zhixuan Chu, James Y. Zhang, Xiaoming Shi, Pin-Yu Chen, et al. 2024. “Time-LLM: Time Series Forecasting by Reprogramming Large Language Models.” arXiv. http://arxiv.org/abs/2310.01728.

Jin, Xin, and Jiawei Han. 2010. “K-Means Clustering.” In *Encyclopedia of Machine Learning*, edited by Claude Sammut and Geoffrey I. Webb, 563–64. Boston, MA: Springer US. https://doi.org/10.1007/978-0-387-30164-8_425.

Khang, Tran Dinh, Nguyen Duc Vuong, Manh-Kien Tran, and Michael Fowler. 2020. “Fuzzy C-Means Clustering Algorithm with Multiple Fuzzification Coefficients.” *Algorithms* 13 (7): 158. https://doi.org/10.3390/a13070158.

Khashei, Mehdi, Mehdi Bijari, and Seyed Reza Hejazi. 2012. “Combining Seasonal ARIMA Models with Computational Intelligence Techniques for Time Series Forecasting.” *Soft Computing* 16 (6): 1091–1105. https://doi.org/10.1007/s00500-012-0805-9.

Khodadadi, Arash. 2023. “Keras Documentation: Traffic Forecasting Using Graph Neural Networks and LSTM.” November 22, 2023. https://keras.io/examples/timeseries/timeseries_traffic_forecasting/.

Klemperer, Paul. 1999. “Auction Theory: A Guide to the Literature.” *Journal of Economic Surveys* 13 (3): 227–86. https://doi.org/10.1111/1467-6419.00083.

Kontopoulou, Vaia I., Athanasios D. Panagopoulos, Ioannis Kakkos, and George K. Matsopoulos. 2023. “A Review of ARIMA vs. Machine Learning Approaches for Time Series Forecasting in Data Driven Networks.” *Future Internet* 15 (8): 255. https://doi.org/10.3390/fi15080255.

Laffey, Des. 2007. “Paid Search: The Innovation That Changed the Web.” *Business Horizons* 50 (3): 211–18. https://doi.org/10.1016/j.bushor.2006.09.003.

Lanier, Saphia. 2023. “Prevent Google Ads Daily Budget Overspend: Best Practices.” HawkSEM. September 15, 2023. https://hawksem.com/blog/google-ads-daily-budget-overspend/.

Laurinec, Peter, and Mária Lucká. 2017. “New Clustering-Based Forecasting Method for Disaggregated End-Consumer Electricity Load Using Smart Grid Data.” In *2017 IEEE 14th International Scientific Conference on Informatics*, 210–15. https://doi.org/10.1109/INFORMATICS.2017.8327248.

Lim, Bryan, Sercan O. Arik, Nicolas Loeff, and Tomas Pfister. 2020. “Temporal Fusion Transformers for Interpretable Multi-Horizon Time Series Forecasting.” arXiv. http://arxiv.org/abs/1912.09363.

Lithgow, Tony. 2005. “Gates Reveals Vision for Advertising in the Future.” Campaign. 2005. https://www.campaignlive.co.uk/article/gates-reveals-vision-advertising-future/525242?utm_source=website&utm_medium=social.

Liu, Jia, and Olivier Toubia. 2018. “A Semantic Approach for Estimating Consumer Content Preferences from Online Search Queries.” *Marketing Science* 37 (6): 930–52. https://doi.org/10.1287/mksc.2018.1112.

MacKay, David. 2003. *Information Theory, Inference and Learning Algorithms*. Cambridge University Press.

Makridakis, Spyros, Evangelos Spiliotis, and Vassilios Assimakopoulos. 2018. "Statistical and Machine Learning Forecasting Methods: Concerns and Ways Forward." Edited by Alejandro Raul Hernandez Montoya. *PLOS ONE* 13 (3): e0194889. https://doi.org/10.1371/journal.pone.0194889.

Martens, Bertin. 2024. "The Impact of Search Engine Data Sharing on Competition and Consumer Welfare." *European Competition Journal* 20 (2): 537–54. https://doi.org/10.1080/17441056.2024.2313399.

Miller. 2024. "WordLlama: Recycled Token Embeddings from Large Language Models." 2024. https://github.com/dleemiller/wordllama.

Najafi-Asadolahi, Sami, and Kristin Fridgeirsdottir. 2014. "Cost-per-Click Pricing for Display Advertising." *Manufacturing & Service Operations Management* 16 (4): 482–97. https://doi.org/10.1287/msom.2014.0491.

Nixtla. 2024. "Long-Horizon Forecasting." TimeGPT Foundational Model for Time Series Forecasting and Anomaly Detection. 2024. https://docs.nixtla.io/docs/tutorials-long_horizon_forecasting.

Oldenburg, Fynn, Qiwei Han, and Maximilian Kaiser. 2024. "Interpretable Deep Learning for Forecasting Online Advertising Costs: Insights from the Competitive Bidding Landscape." arXiv. https://doi.org/10.48550/arXiv.2302.05762.

OpenAI. 2024. "New Embedding Models and API Updates." 2024. https://openai.com/index/new-embedding-models-and-api-updates/.

Oreshkin, Boris N., Dmitri Carpov, Nicolas Chapados, and Yoshua Bengio. 2020. "N-BEATS: Neural Basis Expansion Analysis for Interpretable Time Series Forecasting." arXiv. http://arxiv.org/abs/1905.10437.

Oriona, Ángel López, Pablo Montero Manso, and José Antonio Vilar Fernández. 2023. "Time Series Clustering Based on Prediction Accuracy of Global Forecasting Models." arXiv. https://doi.org/10.48550/arXiv.2305.00473.

Patel, Punyaban, Borra Sivaiah, and Riyam Patel. 2022. "Approaches for Finding Optimal Number of Clusters Using K-Means and Agglomerative Hierarchical Clustering Techniques." In *2022 International Conference on Intelligent Controller and Computing for Smart Power (ICICCSP)*, 1–6. https://doi.org/10.1109/ICICCSP53532.2022.9862439.

Rakthanmanon, Thanawin, Bilson Campana, Abdullah Mueen, Gustavo Batista, Brandon Westover, Qiang Zhu, Jesin Zakaria, and Eamonn Keogh. 2013. "Addressing Big Data Time Series: Mining Trillions of Time Series Subsequences Under Dynamic Time Warping." *ACM Trans. Knowl. Discov. Data* 7 (3): 10:1-10:31. https://doi.org/10.1145/2500489.

Rochet, Jean-Charles, and Jean Tirole. 2003. "Platform Competition in Two-Sided Markets." *Journal of the European Economic Association* 1 (4): 990–1029. https://doi.org/10.1162/154247603322493212.

Rousseeuw, Peter J. 1987. "Silhouettes: A Graphical Aid to the Interpretation and Validation of Cluster Analysis." *Journal of Computational and Applied Mathematics* 20 (November):53–65. https://doi.org/10.1016/0377-0427(87)90125-7.

Ruiz, L. G. B., M. C. Pegalajar, R. Arcucci, and M. Molina-Solana. 2020. “A Time-Series Clustering Methodology for Knowledge Extraction in Energy Consumption Data.” *Expert Systems with Applications* 160 (December):113731. https://doi.org/10.1016/j.eswa.2020.113731.

Russell, S., and W. Lodwick. 1999. “Fuzzy Clustering in Data Mining for Telco Database Marketing Campaigns.” In *18th International Conference of the North American Fuzzy Information Processing Society - NAFIPS (Cat. No.99TH8397)*, 720–26. https://doi.org/10.1109/NAFIPS.1999.781788.

Schank, Thomas, and Dorothea Wagner. 2005. “Approximating Clustering Coefficient and Transitivity.” *Journal of Graph Algorithms and Applications* 9 (2): 265–75.

Schiller, Emily, Simon Müller, Kathrin Ebertsch, and Jan-Philipp Steghöfer. 2024. “No Data Left Behind: Exogenous Variables in Long-Term Forecasting of Nursing Staff Capacity.” In *2024 IEEE 11th International Conference on Data Science and Advanced Analytics (DSAA)*, 1–10. https://doi.org/10.1109/DSAA61799.2024.10722806.

Scott, John. 2017. "Social Network Analysis: Research Methods." Bloomsbury Publishing.

“Search Engine Noun - Definition, Pictures, Pronunciation and Usage Notes | Oxford Advanced Learner’s Dictionary at OxfordLearnersDictionaries.Com.” 2024. https://www.oxfordlearnersdictionaries.com/definition/english/search-engine.

Sezer, Omer Berat, Mehmet Ugur Gudelek, and Ahmet Murat Ozbayoglu. 2020. “Financial Time Series Forecasting with Deep Learning : A Systematic Literature Review: 2005–2019.” *Applied Soft Computing* 90 (May):106181. https://doi.org/10.1016/j.asoc.2020.106181.

Shering, Thomas, Eduardo Alonso, and Dimitra Apostolopoulou. 2024. “Investigation of Load, Solar and Wind Generation as Target Variables in LSTM Time Series Forecasting, Using Exogenous Weather Variables.” *Energies* 17 (8): 1827. https://doi.org/10.3390/en17081827.

Sinaga, Kristina P., and Miin-Shen Yang. 2020. “Unsupervised K-Means Clustering Algorithm.” *IEEE Access* 8:80716–27. https://doi.org/10.1109/ACCESS.2020.2988796.

Srinivasan, Dina. 2020. “Why Google Dominates Advertising Markets: Competition Policy Should Lean on the Principles of Financial Market Regulation.” *Stanford University*, 24, Stanford Technology Law Review (1): 55–175.

StatCounter. 2024. “Search Engine Market Share Worldwide.” StatCounter Global Stats. November 2024. https://gs.statcounter.com/search-engine-market-share.

Statista. 2023. “Global Online Travel Market Size 2023.” Statista. 2023. https://www.statista.com/statistics/1179020/online-travel-agent-market-size-worldwide/.

Statista. 2024. “Search Advertising - Worldwide | Statista Market Forecast.” October 2024. https://www-statista-com.eu1.proxy.openathens.net/outlook/amo/advertising/search-advertising/worldwide.

Sundermeyer, Martin, Ralf Schlüter, and Hermann Ney. 2012. “LSTM Neural Networks for Language Modeling.” In *Interspeech 2012*, 194–97. ISCA. https://doi.org/10.21437/Interspeech.2012-65.

Suntwal, Sandeep, Mithun Paul, Rebecca Sharp, and Mihai Surdeanu. 2020. “On the Importance of Delexicalization for Fact Verification.” arXiv. https://doi.org/10.48550/arXiv.1909.09868.

Tavenard, Romain. 2017. “Time Series Clustering — Tslearn 0.6.3 Documentation.” Tslearn. 2017. https://tslearn.readthedocs.io/en/stable/user_guide/clustering.html.

Thorndike, Robert L. 1953. “Who Belongs in the Family?” *Psychometrika* 18 (4): 267–76. https://doi.org/10.1007/BF02289263.

Valurank. 2024. “MiniLM-L6-Keyword-Extraction.” 2024. https://huggingface.co/valurank/MiniLM-L6-Keyword-Extraction.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2023. “Attention Is All You Need.” arXiv. https://doi.org/10.48550/arXiv.1706.03762.

Vinh, Nguyen Xuan, Julien Epps, and James Bailey. 2009. “Information Theoretic Measures for Clusterings Comparison: Is a Correction for Chance Necessary?” In *Proceedings of the 26th Annual International Conference on Machine Learning*, 1073–80. ICML ’09. New York, NY, USA: Association for Computing Machinery. https://doi.org/10.1145/1553374.1553511.

Watts, Duncan J., and Steven H. Strogatz. 1998. “Collective Dynamics of ‘Small-World’ Networks.” *Nature* 393 (6684): 440–42. https://doi.org/10.1038/30918.

West, Mike. 1997. “Time Series Decomposition.” *Biometrika* 84 (2): 489–94. https://doi.org/10.1093/biomet/84.2.489.

Wolfstetter, Elmar. 1996. “AUCTIONS: AN INTRODUCTION - Wolfstetter - 1996 - Journal of Economic Surveys - Wiley Online Library.” 1996. https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-6419.1996.tb00018.x.

Wolpert, D.H., and W.G. Macready. 1997. “No Free Lunch Theorems for Optimization.” *IEEE Transactions on Evolutionary Computation* 1 (1): 67–82. https://doi.org/10.1109/4235.585893.

Wu, Zonghan, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S. Yu. 2021. "A Comprehensive Survey on Graph Neural Networks." *IEEE Transactions on Neural Networks and Learning Systems* 32 (1): 4–24. https://doi.org/10.1109/TNNLS.2020.2978386.

Würfel, Max, Qiwei Han, and Maximilian Kaiser. 2021. "Online Advertising Revenue Forecasting: An Interpretable Deep Learning Approach." arXiv. https://doi.org/10.48550/arXiv.2111.08840.

Yu, Bing, Haoteng Yin, and Zhanxing Zhu. 2018. "Spatio-Temporal Graph Convolutional Networks: A Deep Learning Framework for Traffic Forecasting." In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*, 3634–40. https://doi.org/10.24963/ijcai.2018/505.

Zia, Mohammad, and Ram C. Rao. 2019. "Search Advertising: Budget Allocation Across Search Engines." *Marketing Science* 38 (6): 1023–37. https://doi.org/10.1287/mksc.2019.1186.

Zweig, Katharina Anna, and Michael Kaufmann. 2011. "A Systematic Approach to the One-Mode Projection of Bipartite Graphs." *Social Network Analysis and Mining* 1 (3): 187–218. https://doi.org/10.1007/s13278-011-0021-0.