Predicting the performance of social media recruitment advertising

Luciën Tuijp

Vrije Universiteit Amsterdam
MSc Business Analytics

In collaboration with

Dutchwebshark

University supervisors:
First supervisor: dr. Shujian Yu
Second reader: prof. dr. Guszti Eiben

Company Supervisor:
Twan Houwers

Contents

Abstract

Leveraging a novel feature engineering pipeline—including a Large Language Model (LLM) used with structured function calling to extract over 40 unique creative fields from ad media—these extracted fields are processed and combined with other metadata to form a static feature vector enriched to 270 features. This vector is later expanded via one-hot encoding (OHE) for compatibility with the sequential deep learning architectures. A rigorous 5-fold GroupKFold cross-validation strategy was used to address the ad campaign life cycle across three sub-questions (SQ1–SQ3). **The key findings establish a clear, actionable temporal hierarchy of predictive power:**

1. **Early intervention (SQ1):** Pre-launch creative features are weak predictors of ad success. Incorporating just 1–2 days of cumulative performance data substantially improves the model's ability to identify both classes: overall accuracy rises modestly, while the F1-score for the critical "not chosen" class reaches $\approx \mathbf{0.79}$ by Day 2, validating an early-intervention system to flag and mitigate budget waste.
2. **Tactical forecasting (SQ2):** For next-day performance, sequential deep learning models—the Temporal Convolutional Network (TCN) and Gated Recurrent Unit (GRU) —outperformed traditional models (XGBoost and Seasonal Autoregressive Integrated Moving Average with Exogenous variables (SARIMAX)) by capturing sequential momentum and $m = 7$ seasonality. The TCN with static features achieved the lowest Mean Absolute Error (MAE) for clicks (a $\mathbf{45.5}\%$ reduction over the baseline, $\mathbf{MASE = 0.558}$), while the GRU without static features was superior for the sparse leads target (a $\mathbf{24.5}\%$ reduction, $\mathbf{MASE = 0.421}$). However, no model, including the baseline or any with static features, showed statistically significant improvements, likely due to the limited statistical power of the 5-fold cross-validation.
3. **Strategic analysis (SQ3):** The primary contribution is the **paradox of static feature importance**: while non-predictive for short-term volatility (SQ2), LLM-extracted creative features show statistically significant correlations with long-term efficiency metrics (e.g., funnel conversion rate and cost per lead). Actionable themes—such as salary inclusion, motivational tone, and casual presentation—emerge as strategic levers for maximizing ROI.

The research also confirms a **top-of-funnel disconnect** ($r = 0.34$ between clicks and leads), motivating specialized forecasting models. Overall, this thesis delivers a multi-stage quantitative framework that moves beyond traditional reactive advertising, integrating temporal performance dynamics with statistically validated creative design principles.

1 Introduction

Every day, marketing teams are faced with a crucial decision: which new ad creative is worth the investment? Get it right, and a campaign can deliver exceptional returns. Get it wrong, and significant portions of a budget can be wasted on assets that fail to connect with an audience. While digital advertising platforms provide a wealth of data, many of these critical go/no-go decisions still rely on a blend of creative intuition and reactive A/B testing. This traditional approach means that by the time an underperforming ad is identified, valuable time and resources have already been lost.

This thesis explores a more proactive approach. It investigates whether data science can be used to move from reacting to results to proactively predicting them. The central goal is to build a quantitative framework that can forecast an ad's performance at different stages of its life. This is done, by developing and evaluating machine learning models that learn from two distinct types of information: the static, pre-launch features of an ad (such as salary, call to action and creative style), and the dynamic performance data from its (first) days online. By understanding what these two sources of data can tell us, it can help marketers make data-driven decisions to optimize their ad spend and save valuable time.

To guide this investigation, the research is centered on one overarching question:

RQ: How can a quantitative framework be developed to accurately forecast and evaluate the performance of social media recruitment ads by systematically assessing the predictive power of creative features in comparison to dynamic performance data?

This central question is explored through three sub-questions, aligned with the key stages of a campaign's life cycle—from initial design to live performance management:

SQ1 (Classification): To what extent can the potential success ("Good" vs. "Bad") of a recruitment ad be predicted using only its creative features before launch (Day 0), how does this predictive accuracy evolve as early performance data (Days 1-14) becomes available, and which features are most indicative of this potential?

SQ2 (Forecasting): How accurately do different models (**SARIMAX**, **XGBoost**, **GRU** and **TCN**) forecast next-day ad performance (clicks/leads)? Does the inclusion of creative features significantly improve this accuracy over using historical performance data alone, and which features are the most important predictors in this forecasting context?

SQ3 (Analysis): Which specific creative features have a statistically significant impact on an ad's overall lifetime performance efficiency metrics: **Click-Through Rate (CTR)**, **Conversion Rate (CR)**, **Cost Per Click (CPC)**, **Cost Per Lead (CPL)**, and **Funnel Conversion Rate (FCR)**? Furthermore, what is their relative importance, and do significant interaction effects exist between these top features?

2 Literature review

Click-Through Rate (CTR) prediction is a related and well-researched field [19], but it represents only one part of the job application funnel. A complete funnel is a combination of both the CTR and the Conversion Rate (CR). Even more challenging in this context is the prediction of the final outcome—the Funnel Conversion Rate (FCR) from *reach* to an application (leads)—which is the multiplication of these two rates ($\text{CTR} \times \text{CR}$) and results in target values that are several magnitudes smaller.

The high frequency of advertising data, combined with the extreme sparsity (zero-inflation) of low-funnel events like leads, presents a significant modeling challenge. Research has shown that traditional statistical methods struggle to accurately model data exhibiting this *intermittent demand* characteristic [14]. Furthermore, predicting the success of newly launched ads or creatives—a task known as the *"cold start" problem* [15]—requires leveraging static creative features effectively before sufficient performance history is accrued.

Therefore, this thesis addresses a gap in the current literature by focusing on models that can jointly handle the high volatility of ad performance time series and the zero-inflation inherent in predicting sparse, low-funnel conversion events (leads). To the best of my knowledge, Temporal Convolutional Networks (TCNs) have not yet been applied to marketing recruitment ad performance prediction. Furthermore, the comparative evaluation of TCNs alongside GRU, XGBoost, and SARIMAX on this type of data constitutes a novel contribution of this work.

2.1 The role of creative features in recruitment

According to Alniacik and Alniacik (2025) [2], CR should be higher when job ads are more informative. Their research concludes that more informative ads help candidates better judge if the job is appropriate for them, which makes them more likely to apply. Their recommendation is that HR professionals should craft jobs that have clear, specific information about roles and requirements, especially when targeting less experienced candidates. Focusing on clarity and specificity can significantly increase application rates.

Similarly, Mahjoub and Kruyen (2021) [7] state that efficient recruitment relies largely on the design and content of advertisements. In a comprehensive literature review spanning over four decades of research, they identified the most important ad features: informativeness, clarity, attractiveness, credibility, specificity, organizational image, and inclusiveness. They conclude that effective job ads should be clear, informative, and tailored for their respective target audience.

2.2 Models for performance prediction

CTR prediction models in the literature can be grouped into four categories: (1) multivariate statistical models, (2) factorization machine (FM)-based models, (3) deep learning models, and (4) tree-based models (Yang & Zhai, 2022 [8]). However, many existing CTR predicting models ignore distinct characteristics in multimedia advertising, namely image and video features.

2.2.1 Tree-based models

Tree-based models are commonly used for this type of tabular prediction. For example, Bhulai et al. (2017) [3] examined application rates for a major Dutch company using features such as job title, work location, and required education level. Data sources included the company's Applicant Tracking System, Google Analytics, and Twitter activity. Among the machine learning models compared, a random forest (RF) yielded the most accurate results.

2.2.2 Temporal Convolutional Networks (TCN)

Edizel et al. (2017) [4] introduced the use of temporal convolutions for CTR prediction in commercial search engines. More recently, Guo et al. (2025) [5] were the first to apply a temporal convolutional network (TCN) to lifetime sequence modeling (LSM) on TaoBao and WeChat. Their Context-Aware Interest Network (CAIN) outperformed prior LSM methods across both datasets.

2.2.3 Challenges in conversion rate (CR) prediction

According to Lu et al. (2017) [6], CR prediction may seem similar to the much researched topic of CTR prediction. However, a conversion requires significantly more user engagement and is a much less researched topic. One of the main challenges in predicting CR versus CTR is conversion rarity. Compared with CTR, CR is generally several magnitudes smaller, which makes the prediction inherently more challenging.

3 Methodology

3.1 Research design

The primary aim of this study is to evaluate and forecast advertisement performance through a three-pronged approach, addressing the distinct stages of a campaign's life cycle. As visualized in Figures 1 and 2, this design is broken down into:

1. A **classification** task to identify high-potential ads at or near launch (SQ1).
2. A **time-series forecasting** task to predict daily performance for mature ads (SQ2).
3. An **inferential analysis** task to determine the statistical impact of creative features on overall lifetime performance (SQ3).

The first two components, the predictive pipelines for SQ1 and SQ2, are illustrated in Figure 1.

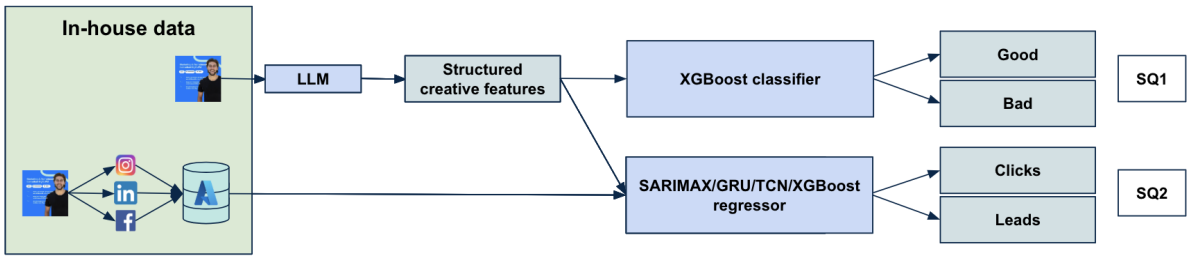


Figure 1: Research design for the predictive pipelines (SQ1 and SQ2), showing the flow from in-house data to classification and forecasting models.

The **first component (SQ1)** is a classification model developed to predict whether an ad will perform well (*good* versus *bad*) based on its static, pre-launch creative features. This model is then sequentially enhanced by incorporating early-life performance data (e.g., from days 1, 2, 5, etc.) to evaluate the added predictive value of initial performance.

The **second component (SQ2)** is a comparative forecasting analysis of four different modeling techniques for predicting daily clicks and leads. The chosen models—eXtreme Gradient Boosting (**XGBoost**) [9], **Temporal Convolutional Network (TCN)** [10], **Gated Recurrent Unit (GRU)** [11], and **Seasonal AutoRegressive Integrated Moving Average with eXogenous variables (SARIMAX)** [12]—facilitate a comparison between a classical statistical model, an advanced tree-based ensemble, and two deep learning architectures.

The **third component (SQ3)** is a dedicated analytical pipeline, shown in Figure 2, designed to move beyond prediction and identify why certain ads perform well. This analysis investigates the statistical relationship between the static creative features and an ad's final lifetime performance metrics, assessing feature significance, relative importance, and interaction effects.

For the **forecasting analysis (SQ2)**, a 14-day starting point was chosen to ensure all models, particularly SARIMAX, had sufficient data to establish statistical stability and capture initial weekly patterns. To ensure a fair comparison, the machine learning models (XGBoost, TCN, and GRU) were trained and validated using the same comprehensive dataset, pre-processing, and Optuna-based optimization steps. However, due to its different statistical nature, the SARIMAX model was treated as a specialized baseline: its parameters (orders) were optimized *per-series* using `pmdarima.auto_arima` (minimizing AIC), and it was fit using only a minimal set of three performance-based regressors and no static creative features.

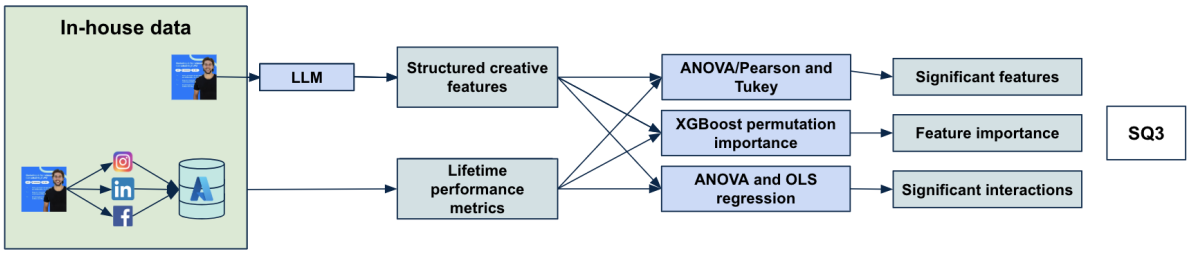


Figure 2: Research design for the analytical pipeline (SQ3), showing the three parallel analyses used to evaluate creative feature impact.

3.2 Data description and cleaning

To investigate the predictability of social media advertisement performance, a dataset was provided by Dutchwebshark, a data-driven recruitment company. The dataset contains historical campaign data from 2022 to the present. These campaigns are characterized by a high volume of short, independent time series. While the mean ad duration is approximately 33 days, the median is only 17 days, indicating the data is heavily skewed by a long tail of outlier campaigns.

The data for this research was gathered from two sources:

Azure SQL Database: Contains the primary dataset: a structured table of daily ad performance logs. The 26 columns can be logically grouped as follows:

- **Performance Metrics:** Daily aggregated counts for key performance indicators (e.g., `Clicks (All)`, `Amount Spent`, `Leads`, `Reach` and `Post Engagement`).
- **Efficiency Metrics:** Calculated ratios based on the performance metrics (e.g., `CPC (All)`, `CTR (All)` and `Frequency`).
- **Hierarchy & Identifiers:** Columns that define the ad's structure and origin (e.g., `Campaign ID`, `Ad Set name`, `Ad ID`) and the client account (e.g., `Account Name` and `Client_ID`).
- **Time Column:** A daily `Date` field, which forms the basis of the time series.
- **Descriptive & Creative Data:** Text fields for identification (`Campaign Name`, `Ad name`) and a `Ad Image URL` linking to the ad's visual asset.

Meta Ads Manager: Manual extraction of creative media files and campaign objective information.

3.2.1 Azure SQL - Performance data

The raw performance data was extracted from an Azure SQL table, structured by `ad_id`, `platform`, and `date`. To ensure data quality and consistency, only data meeting the following criteria was exported:

- **Campaign-level filtering:**
  - Campaigns must have been inactive for at least 30 days to ensure all performance data is final and complete.
  - Campaigns must have run for a minimum of 7 days.
  - Campaigns must have reached a minimum of 1,000 people.

- **Row-level cleaning:**
  - Individual rows containing illogical data were dropped. This included:
    * Clicks $>$ Reach
    * Leads $>$ Clicks
    * Amount Spent < 0

This row-level cleaning resulted in the removal of 373 (0.6%) records, leaving a final dataset of 64,489 daily performance rows. These records are aggregated into 1,948 unique time series (one per ad per platform) originating from 794 distinct ads.

3.2.2 Meta Ads Manager

The Azure SQL database contained expired temporary URLs for the ad media. Automating media retrieval was explored extensively, as several potential avenues were investigated:

- A custom Python scraper was built to crawl the **Meta Ads Library** with two primary objectives:
  1. **Media Matching:** To automatically find and extract the missing media files for the ads already in the performance database. This was found to be unfeasible, as the public identifiers in the Ads Library do not correspond to the internal `ad_id`s from the Ads Manager, making automated matching impossible.
  2. **Data Augmentation:** To extract new, publicly available ads and their information to enlarge the dataset for training. This approach was also abandoned for performance modeling, as the public library only provides `reach` estimates and lacks the essential `spend`, `clicks`, and `leads` data required for the analysis.
- The **Facebook Graph API** was also found to be insufficient, as it only returned low-resolution `64x64` thumbnails for older ads and could not reliably retrieve historical campaign objective data.
- Windsor.ai was unable to retrieve the full-sized, original media files or the historical campaign objectives, such as leads or traffic.

Given that all automated avenues for bulk retrieval were exhausted, the only reliable method remaining was manual retrieval. A total of 643 unique media files were downloaded by manually navigating the Meta Ads Manager interface, saving each file with its corresponding `ad_id` as the filename. During this process, the campaign objective data was also manually extracted for each campaign. This number of media files is lower than the 794 distinct ads in the database because the permission to access some old clients' pages was gone.

3.3 Data visualization and exploration

Exploratory data analysis (EDA) was conducted to identify outliers, understand data properties, and inform modeling decisions.

Initial visualizations of performance data revealed significant outliers in the `leads` metric for certain ads, as seen in Figure 3. Investigation showed these were often caused by improperly configured custom conversions, where link clicks were erroneously registered as leads. The `ad_id`s associated with this erroneous data were removed from the dataset. This process also

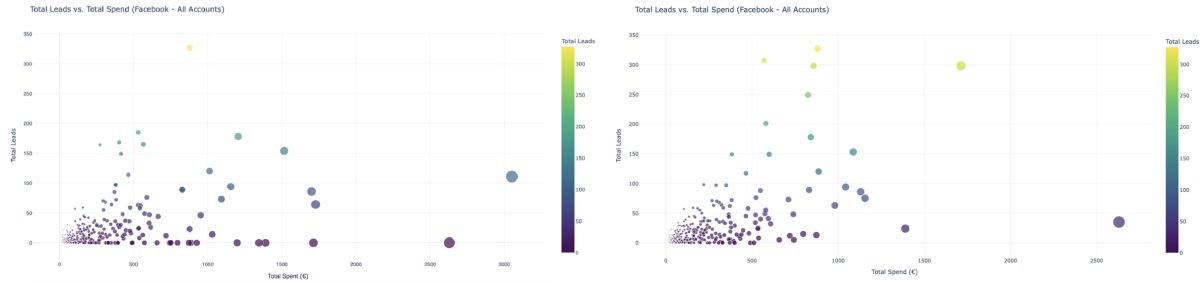


Figure 3: Total leads vs. total spend, illustrating performance outliers on the left, cleaned data on the right.

uncovered missing leads data in the database, which was subsequently backfilled using data from Windsor.ai.

Analysis of efficiency metrics (CR, CPC, CTR, CPL) in Figure 4 demonstrated rapid convergence. The distributions of these metrics stabilize within the first few days of a campaign. This finding suggests that an ad's long-term efficiency is largely determined early in its life cycle, validating the research design's focus on using early performance to predict future outcomes.

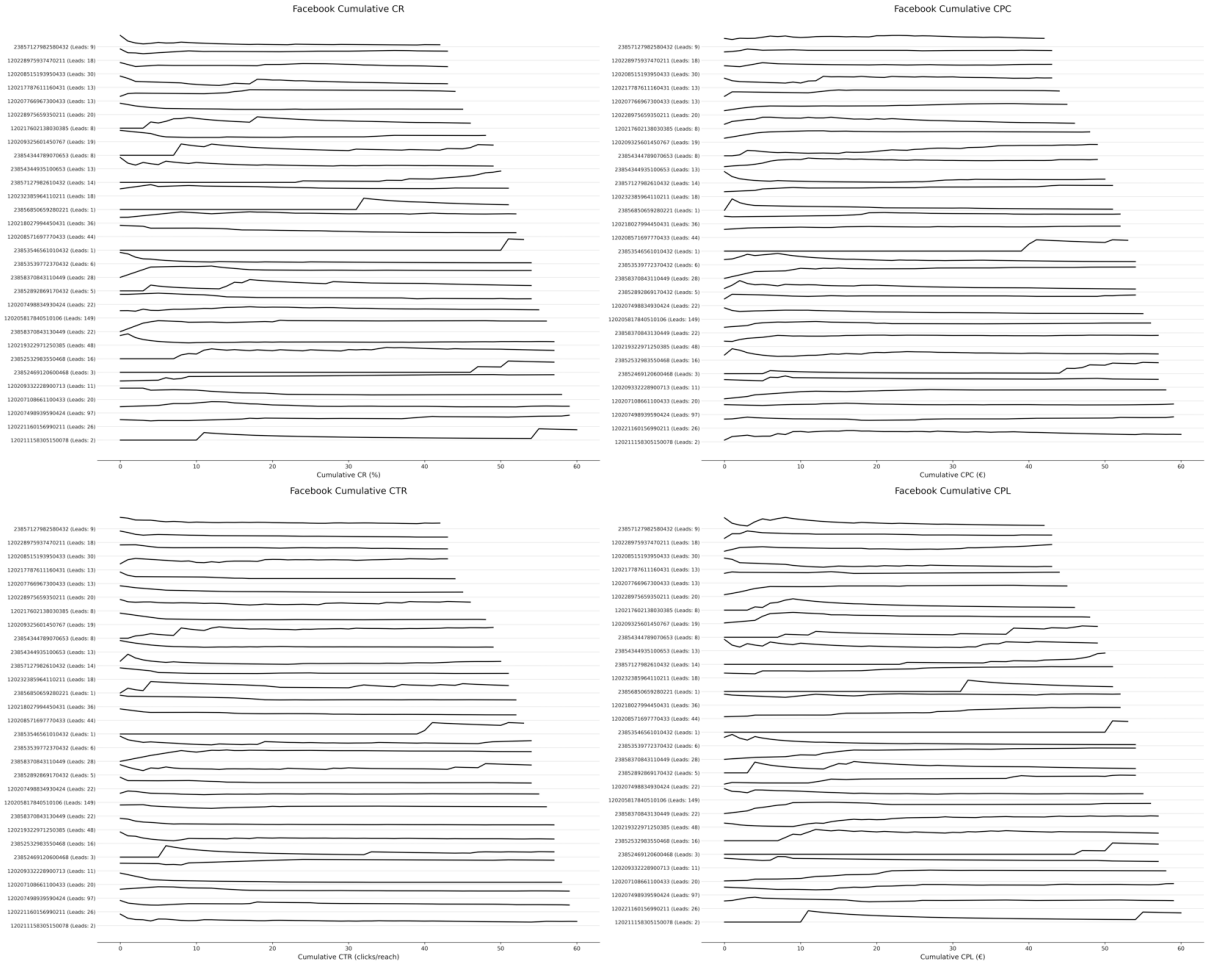


Figure 4: Convergence of cumulative efficiency metrics (CR, CPC, CTR, CPL) over ad lifetime.

A pairplot (Figure 5) was used to visualize the distributions and relationships between key daily metrics. The pairplot (left) highlighted the extreme right-skew of the data, particularly for `clicks_all`, `leads`, and `amount_spent`, indicating that most daily values are low, with rare

high-performance days.

The corresponding correlation heatmap (right) revealed two critical insights that informed the research design. First, it showed a strong positive *volume cluster* among `amount_spent`, `reach`, and `clicks_all` (correlations 0.67–0.71), indicating that increases in spend are generally accompanied by higher reach and clicks. Second, it exposed a significant *top-of-funnel disconnect*: the correlation between `clicks_all` and `leads` was weak (0.34), and the correlation between `ctr_all` and `leads` was even weaker (0.20), suggesting that clicks and CTR are poor proxies for lead generation. These findings validate the decision to build separate forecasting models for `clicks` and `leads`, as their drivers are distinct.

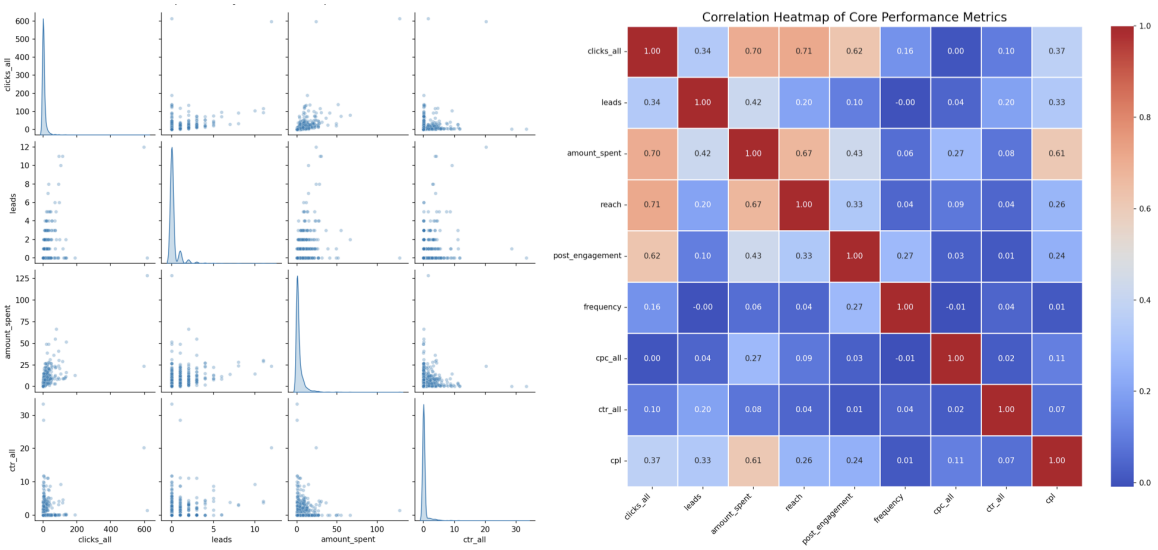


Figure 5: Pairplot and Correlation Heatmap of key daily performance metrics. The pairplot (left) shows distributions and scatter plots. The heatmap (right) shows Pearson correlation coefficients.

3.3.1 Time-series properties

To understand the temporal dependencies in the data, Autocorrelation (ACF) and Partial Autocorrelation (PACF) plots were generated for the primary target variables (Figure 6).

The plots revealed two key properties. First, the ACF plots for both `Clicks` and `Leads` show a very slow, gradual decay from a high initial value. This pattern is a classic indicator of a non-stationary time series with a strong trend and high persistence (e.g., a strong memory). It confirms that a day's performance is highly correlated with its recent past, strongly justifying the use of lagged features and rolling windows for the forecasting models.

Second, the **PACF plots** clearly showed significant spikes at Lag 7. This finding confirms a strong *weekly seasonality* in the data (e.g., performance on a Monday is highly correlated with performance on the previous Monday). These insights directly informed two key modeling decisions:

1. A **14-day lookback period** was consistently used across the **XGBoost, TCN, and GRU** models for generating input sequences and lagged/rolling features. Using a multiple of the primary seasonal period (7 days) ensures that these models receive input containing complete weekly cycles, providing sufficient historical context.

2. For **SARIMAX**, while the specific $(p, d, q)(P, D, Q)$ orders were optimized per-series (see Section 3.9.2), the PACF analysis confirmed the appropriateness of setting the seasonal

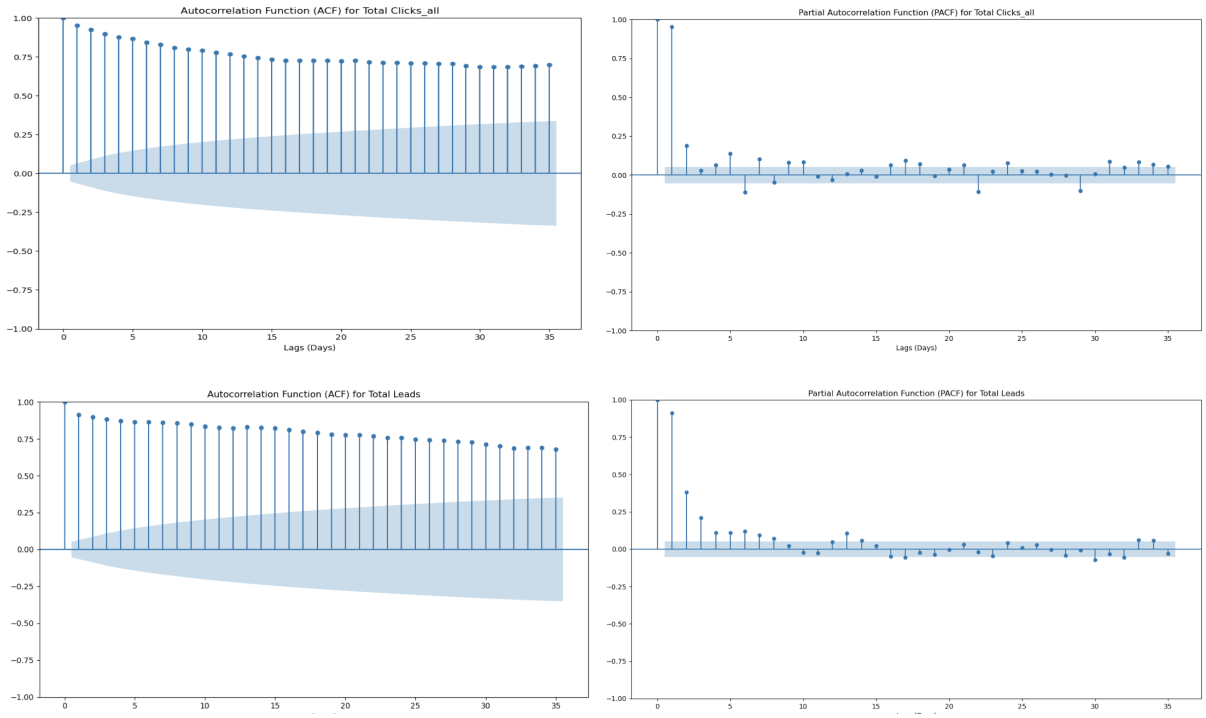


Figure 6: ACF and PACF plots for Total Clicks (Top) and Total Leads (Bottom).

period to $m = 7$ within the `pmdarima.auto_arima` search and the 14-day window was used for the rolling refitting process.

3.4 Feature engineering

To enhance the predictive power of the models, a diverse set of features was engineered from the raw data. The objective was to capture performance trends, temporal patterns, campaign strategy, and the specific content of each advertisement.

3.4.1 Creative feature extraction via LLM

To convert the 643 unstructured media files (images and videos) into structured, usable data, Google's `gemini-2.5-pro-latest` model was employed. The extraction utilized the model's *function calling* capability (sometimes referred to as "tool use"), which provides more reliable and structured output than a simple text prompt.

A comprehensive JSON schema was defined, pre-specifying over 40 desired attributes. These features were grouped into five main categories: `creative_format`, `composition_and_style`, `content_and_offer`, `demographics`, and `brand_and_emotion`.

A batch processing script iterated through the media files, making automated API calls. For each file, the model was prompted to use the `extract_ad_features` function, which forced its response to conform to the predefined JSON schema. The resulting structured JSON output formed the basis for the **Ad Creative and Content features (3.4.4)**. Additional features described in Sections 3.4.2, 3.4.3, and 3.4.5 were engineered from the structured Azure SQL database.

3.4.2 Performance, trend, and temporal features

This category includes features that capture both the *when* (temporality) and *how well* (performance) of marketing activities, providing insights into the timing and effectiveness of campaigns.

- **Temporal Features:**
  - `day_of_week`: A categorical feature (0-6) to capture weekly patterns.
  - `month`: A categorical feature (1-12) to capture broader seasonal trends.
  - `is_weekend`: A binary feature (1 for Saturday/Sunday, 0 otherwise).
- **Ratio and Efficiency Metrics:**
  - `cpc_all` (Cost Per Click): Calculated as amount_spent/clicks_all.
  - `cpl` (Cost Per Lead): Calculated as amount_spent/leads.
  - `ctr_all` (Click-Through Rate): Calculated as clicks_all/reach.
  - `frequency`: Calculated as impressions/reach (available as a base metric).
- **Rolling Window Statistics:** To capture recent trends and volatility, rolling (moving) statistics were calculated using 7-day (short-term) and 14-day (medium-term) windows. Both the **mean** and **standard deviation** were calculated for:
  - `amount_spent`
  - `leads`
  - `reach`
  - `clicks_all`
- **Base Metrics:** The core, non-lagged metrics (e.g., `amount_spent`, `leads`, `post_engagement`, `reach` and `frequency`) were retained to provide the model with the most direct information for each observation.

3.4.3 Textual naming features

In practice, campaign, ad set, and ad names often contain structured metadata about targeting or strategy. To leverage this, features were engineered by parsing these names.

- **Keyword Indicators:** A set of binary features flagging the presence of specific keywords in the `campaign_name`, `ad_set_name`, and `ad_name`. Examples include:
  - Job terms: `..._has_baan`, `..._has_vacature`, `..._has_solliciteer`
  - Sectors: `..._has_finance`, `..._has_it`, `..._has_techniek`
  - Seniority/Type: `..._has_junior`, `..._has_senior`, `..._has_fulltime`
- **Name Length:** The `..._word_count` feature was calculated for each naming level.

3.4.4 Ad creative and content features

This extensive set of features, derived from the LLM extraction (3.4.1), quantifies the visual and semantic elements of each ad creative. These were processed from a multi-faceted classification:

- **Brand & Emotion:** Capturing the ad's affective tone and branding (e.g., `brand_name_visible` and `emotional_tone_Positive`).
- **Composition & Style:** Describing the visual construction (e.g., `setting_Outside`, `visual_complexity_Medium` and `time_of_day`).
- **Content & Offer:** Detailing the core message, including:
  - *Offer Elements:* `cta_present`, `advantages_listed`, `logo_visible` and `salary_mentioned`.
  - *Job Details:* `job_details_industry_...` and `job_details_job_function_...`.
  - *Salary Details:* `salary_extraction_currency_EUR` and `salary_extraction_period_Month`.
  - *Visual Content:* `number_of_people` and `person_activity_Office Work`.
- **Creative Format:** Technical descriptors (e.g., `media_type_Video`, `camera_angle_Eye level` and `amount_of_text_Low`).
- **Demographics:** Visual indicators of people shown (e.g., `demographics_gender_Mixed` and `demographics_age_group_Adult`).
- **Creative Meta-Features:** Summary features aggregating content richness (e.g., `num_advantage_cat` (count) and `has_any_advantage_cat` (binary)).
- **Question Analysis:** For ads containing a question in their copy, a keyword analysis was performed (e.g., `question_has_jij` and `question_has_hoe`

3.4.5 Campaign configuration features

This final group captures high-level strategic choices from the Azure SQL data.

- **Platform Placement:** One-hot encoded binary features (`platform_facebook` and `platform_instagram`, etc.).
- **Campaign Objective:** One-hot encoded feature (`extra_veld3_Leads` and `extra_veld3_Traffic`) derived from the manual extraction (3.2.2).

3.5 Feature selection for classifier model

Following feature engineering, a crucial selection step was performed for the ad-level classification model. The model's objective is to identify which creative elements drive performance, *independent of campaign-level settings*. This required isolating the variance between individual ads *within* the same campaign.

Therefore, all features representing campaign-level configurations, ad set settings, or post-launch performance were **removed**. These features (e.g., campaign objective, platform) are often constant for all ads in a campaign and provide no contrast for an ad-level model. This ensures the model focuses exclusively on the ad's intrinsic creative attributes.

The following feature categories were **excluded**:

1. **Performance, Temporal, and Contextual Features:** To prevent data leakage for the "0-day" classification, all features related to performance or external context were removed (e.g., `amount_spent`, `cpl`, all `..._roll_mean_...` features and `day_of_week`).

2. **Campaign and Ad Set Configuration Features:** These were removed as they are campaign-level constants (e.g., `extra_veld3_Leads`, `platform_facebook`, all `campaign_name...` and `ad_set_name...` features).

3. **Shared Job-Specific Attributes:** Attributes related to the job (industry, function, level) were also removed, as these are typically properties of the campaign, not the individual ad creative (e.g., all `..._job_details_industry_...` and `..._job_details_function_...` features).

The **final selected feature set** for the classifier consists exclusively of features describing the ad's creative, content, and copy (e.g., `brand_and_emotion...`, `composition_and_style...`, `creative_format...`, `demographics...` and `num_advantage_cat`, etc.). This refined set enables the model to isolate the creative and textual levers that distinguish a successful ad from an unsuccessful one.

3.6 Data splitting and validation

A robust **5-fold GroupKFold cross-validation** strategy was used for all models. The `campaign_id` was used as the grouping key. This ensures that all ads from a single campaign exist entirely within either the training or the test set for a given fold, preventing any data leakage between campaigns and simulating a real-world scenario of predicting performance for new, unseen campaigns. The split was approximately 80% training and 20% testing for each fold.

Evaluation metrics used for the forecasting models (Section 3.9) were as follows:

- **Mean Absolute Error (MAE):** Measures the average absolute difference between predicted and actual values, providing an interpretable measure of prediction accuracy.
- **Root Mean Squared Error (RMSE):** Emphasizes larger errors through squaring, making it sensitive to outliers and useful for evaluating volatility.
- **F1-score:** Represents the harmonic mean of precision and recall for the classification-based evaluation of up/down or outperform/underperform prediction tasks.
- **Mean Absolute Scaled Error (MASE):** A scale-independent measure of forecast accuracy that allows comparison across series; values below 1.0 indicate forecasts more accurate than a naïve benchmark [22].

For the classification model (Section 3.8), standard binary classification metrics were used: Accuracy, Precision, Recall, F1-score, the Area Under the Precision–Recall Curve (AUC-PR), and the Confusion Matrix.

3.7 Hyperparameter tuning

To ensure each model (except SARIMAX) performed optimally, hyperparameters were tuned automatically using the **Optuna** framework [1]. Optuna employs Bayesian optimization algorithms, such as the default Tree-structured Parzen Estimator (TPE), to efficiently search large hyperparameter spaces.

A **nested cross-validation** approach was used. For each of the 5 outer folds:

1. A separate Optuna study was conducted with 15 trials.
2. The training data of that fold was split 80/20 into an internal training and validation set.

3. For each trial, a temporary model was trained on the internal training set and evaluated on the internal validation set.

4. The **MAE** on the internal validation set was used as the objective for Optuna to minimize. Early stopping was used to prune unpromising trials.

5. After 15 trials, the best hyperparameter set was selected and used to train the final model for that fold on the *full* training set. This final model was then evaluated on the unseen test set.

The hyperparameter optimization strategy differed between model families. For the deep learning (GRU, TCN) and gradient boosting (XGBoost) models, a global search was performed using **Optuna**. This involved testing 15 trial combinations to find the hyperparameters that minimized the Mean Absolute Error (MAE) on a validation set. The specific hyperparameters tuned for each model were:

- **XGBoost:** `n_estimators`, `max_depth`, `learning_rate`, `subsample` and `colsample_bytree`
- **GRU:** `hidden_dim`, `n_rnn_layers` and `dropout`
- **TCN:** `kernel_size`, `num_filters` and `dropout`

In contrast, the **SARIMAX** model parameters were selected on a *per-series* basis. For each individual ad's time series, the optimal non-seasonal order (p, d, q) and seasonal order $(P, D, Q, m = 7)$ were determined using the `pmdarima.auto_arima` function. This process automatically searches for the combination of orders that best minimizes the **AIC (Akaike Information Criterion)**, which balances model fit with parsimony.

3.8 Classifier: Good/bad performance predictor

This classification model was designed to act as a proactive tool, enabling an informed decision on budget allocation *before* significant budget is spent. The goal is to identify and filter out "Bad" performing ads, which waste time and money. As seen in Figure 7, the budget spent on the underperforming (purple) ad could have been saved.

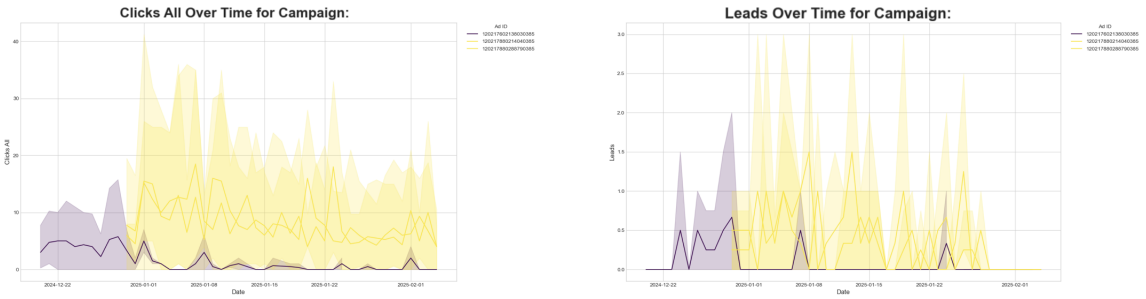


Figure 7: Example campaign timeline (Clicks and Leads) illustrating a "Bad" (purple) vs. "Good" (yellow) ad.

An ad's performance class was defined on a campaign-by-campaign basis. Let:

- S_i = total spend of ad i
- $S^c_{\max}$ = maximum total spend of any ad in campaign c

Then, the label for ad i is defined as:

$$\text{Performance Label}_i = \begin{cases} \text{"Good" (Chosen)}, & \text{if } \frac{S_i}{S^c_{\max}} \geq 0.20 \\ \text{"Bad" (Not Chosen)}, & \text{if } \frac{S_i}{S^c_{\max}} < 0.20 \end{cases}$$

This logic defines "Bad" ads as those likely deactivated either manually by an optimizer or automatically by Meta's algorithm due to poor performance, resulting in limited budget allocation.

An XGBoost classifier (`XGBClassifier` from the `xgboost` library) was used for this task. The model was trained and evaluated sequentially for different amounts of available performance data:

1. **Day 0:** Using only the static creative features identified in Section 3.5, extracted from the first day's record for each ad.
2. **Days N (1, 2, 3, 5, 7, 10, 14):** Sequentially augmenting the static creative features with the cumulative performance features available up to day N (e.g., raw metrics like `clicks_all`, `amount_spent`, and rolling statistics calculated up to that day).

For each day N, a 5-fold group cross-validation (grouping by `campaign_id`, as described in Section 3.6) was performed. Within each fold, a preprocessing pipeline was applied:

- Numerical features were imputed with zero (`SimpleImputer(strategy='constant', fill_value=0)`) and then standardized (`StandardScaler`).
- Categorical features were passed through.
- A `VarianceThreshold` was applied to remove zero-variance features within the training set of that fold.
- The XGBoost classifier was then trained on the preprocessed data.

Hyperparameter Tuning Note: While Optuna was used for hyperparameter tuning in the forecasting models (Section 3.7), initial experiments with Optuna for this classification task did not yield significant performance improvements over the default `XGBClassifier` parameters. Therefore, for simplicity and given the lack of substantial gain, the results reported in Section 4.1 were generated using the default hyperparameters of the `XGBClassifier`.

This sequential analysis allows for measuring the precise predictive value added by early performance data (SQ1). Furthermore, feature importance (based on XGBoost's internal gain metric, averaged across folds and days) was extracted to gain insight into which creative and performance attributes contribute most to a "Good" or "Bad" classification, providing statistical evidence to inform future creative processes (SQ1). Evaluation focused on standard binary classification metrics (Accuracy, F1-Score, Precision, Recall) calculated per class, aggregated confusion matrices, and Precision-Recall curves to account for class imbalance.

3.9 Next-day performance forecasting

This section details the comparison of XGBoost, TCN, GRU, and SARIMAX for forecasting next-day ad performance. All models were used to predict both **clicks** and **leads** as the target variable. For the leads forecast, only data from campaigns with a 'Leads' objective was used. "As the forecasting models are based on regression and do not have an inherent non-negativity constraint, raw predictions could result in small negative values for count-based metrics like 'clicks'. To correct for this, all final forecasts were post-processed by applying a lower bound of zero (i.e., $Forecast = \max(0, \text{Raw\_Forecast})$), ensuring all predicted values are logically possible."

Feature Importance Strategy for Sequential Models For the deep learning models (`GRU`, `TCN`), the final SHAP analysis focused exclusively on interpreting the **dynamic performance features** (lagged and rolling metrics). This was a deliberate methodological choice due to the high dimensionality of the combined input space (approximately 850 features, including

453 static creative features). The objective was to isolate the contribution of **dynamic causality**—momentum, volatility, and trend—in day-to-day forecasting. Consequently, while the static creative features were provided as contextual inputs to the models (as static covariates) to refine sequential processing, their direct contribution was intentionally excluded from the final SHAP interpretation plots.

3.9.1 Data preparation

Prior to training, data was prepared to meet the input requirements of each model:

- **Scaling:** All continuous target and covariate features were scaled to a [0, 1] range using a `MinMaxScaler`.
- **Formatting (TCN/GRU):** Data was formatted into fixed-length sequences. Based on the 7-day seasonality found in 3.3.1, a **14-day lookback period** was used. The models use a sequence of 14 past time steps to predict the next time step (i.e., day 15).
- **Formatting (XGBoost):** The time series was "unrolled" into a tabular format using the 14-day lookback period to create a flat feature vector for each time step.
- **Formatting (SARIMAX):** Data was provided as Pandas Series/DataFrames for endogenous and exogenous variables within the rolling forecast loop. Time series were filtered to include only those with a minimum length of 15 days (`MIN_TRAIN_LENGTH`) to ensure sufficient data for the initial 14-day lookback window used for both order selection and the first prediction step.

3.9.2 Baseline model

The EDA (3.3) showed that efficiency metrics converge quickly. This implies that after a few days, performance (clicks/leads) should be strongly correlated with spend alone. Therefore, a baseline model was engineered using a global efficiency factor calculated from the **training set**: global_efficiency_factor = total_target/total_spend. This factor represents the mean 1/CPC or 1/CPL. The baseline prediction uses only the *previous day's spend*, a sequential one-step-ahead forecast using a lagged predictor.

- **For leads:** $\text{leads}(t) = (\text{total\_train\_leads}/\text{total\_train\_spend}) \times \text{spend}(t-1)$
- **For clicks:** $\text{clicks}(t) = (\text{total\_train\_clicks}/\text{total\_train\_spend}) \times \text{spend}(t-1)$

3.9.3 TCN

The Temporal Convolutional Network (TCN) [10], an architecture using dilated causal convolutions to capture long-range dependencies in sequences, was implemented using the `darts.models.TCNModel`.

- **Model Specification:** The TCN model was configured with an `input_chunk_length` equal to the lookback period (14 days) and an `output_chunk_length` of 1 day, suitable for one-step-ahead forecasting. It processes sequences of past target values and past covariates to predict the next target value.
- **Feature Sets:** Similar to XGBoost, two versions were evaluated:
    1. **Performance Only:** This version used the 14-day history of the target variable (`clicks_all`) and the time-varying performance features (lags, rolling stats, temporal features) as *past covariates*. No static creative features were provided to the model itself.

2. **Performance + Creative:** This version used the same past covariates as "Performance Only" but additionally incorporated the static creative features (e.g., features starting with `creative_format_`, `brand_and_emotion_`) as *static covariates*. These are provided once per series and allow the model to condition its predictions on the ad's specific characteristics.

- **Data Preparation:**
  - **TimeSeries Creation:** As with XGBoost, data within each fold was converted into lists of `darts.TimeSeries` objects, filtered to include only series with at least 15 days (`LOOKBACK_PERIOD + 1`). Value columns included the target and the past (performance) covariates. Crucially, the static creative features (plus a unique tag derived from `ad_platform_id`) were included during TimeSeries creation using the `static_cols` argument.
  - **Static Covariate Encoding:** Since TCNModel can directly handle static covariates but requires them to be numeric, a `darts.dataprocessing.transformers.StaticCovariatesTransformer` was used. This transformer applied one-hot encoding (via `sklearn.preprocessing.OneHotEncoder`) to all categorical static features (including the ad ID tag). It was fit on the combined train and test series within the fold to learn the full vocabulary before transforming the train and test lists separately. The original ad ID tag was effectively encoded but excluded from the list of static features passed to the model during training/prediction to avoid data leakage. Numeric static features (if any, although excluded in the provided script's final feature definition) would bypass encoding.
  - **Scaling:** Both the target variable and the past covariates (performance features) were scaled independently using `darts.dataprocessing.transformers.Scaler` with `global_fit=True`. Static covariates were *not* scaled. The static covariates (now numerically encoded) were re-attached to the scaled target TimeSeries objects before model training and prediction.
- **Hyperparameter Tuning:** Optuna [1] was used within each fold using the nested validation approach (Section 3.7). The objective function (`objective_tcn`) minimized MAE on the internal validation split over 15 trials. Tuned hyperparameters included `kernel_size`, `num_filters`, and `dropout`. PyTorch Lightning's `EarlyStopping` callback (monitoring validation loss with a patience of 5 epochs) was used within Optuna trials and during the final model training to prevent overfitting and determine the optimal number of training epochs (up to a maximum of 100 for Optuna, 200 for the final fit). Training utilized GPU acceleration where available.
- **Training and Forecasting:** The best hyperparameters from Optuna were used to configure the final TCN model for the fold. This model was trained on the full (scaled) training data for that fold, again using early stopping based on a 90/10 split of the training data for validation during the final fit. Similar to XGBoost, predictions for the test set were generated using `historical_forecasts` with `start=LOOKBACK_PERIOD` and `retrain=False`, performing sequential one-step-ahead forecasts using the single trained model. Scaled predictions were inverse-transformed using the fitted target scaler. As the model's final linear output layer is unconstrained and can produce negative values, a post-processing step was applied to clip all final, unscaled predictions at zero (i.e., $\max(0, \hat{y})$) to enforce the non-negativity constraint.

3.9.4 GRU

The Gated Recurrent Unit (GRU) [11], a type of Recurrent Neural Network (RNN) known for its efficiency in capturing temporal dependencies, was implemented using the `darts.models.BlockRNNModel` with the `model='GRU'` parameter.

- **Model Specification:** The `BlockRNNModel` was configured with an `input_chunk_length` of 14 days and an `output_chunk_length` of 1 day. This architecture processes sequences of past target values, past covariates, and static covariates to predict the next target value.
- **Feature Sets:** Like XGBoost and TCN, two versions were evaluated:
  1. **Performance Only:** Utilized the 14-day history of the target variable (`clicks_all`) and the time-varying performance features (lags, rolling stats, temporal features) as *past covariates*. No static creative features were provided to the model.
  2. **Performance + Creative:** Used the same past covariates but also incorporated the static creative features (e.g., features starting with `creative_format_`) as *static covariates*, allowing the GRU layers to condition their hidden state based on the ad's characteristics.
- **Data Preparation:** The data preparation steps were identical to those used for the TCN model (Section 3.9.6):
  - **TimeSeries Creation:** Data was converted to `darts.TimeSeries` objects, filtered for minimum length (15 days), and included past covariates (performance features) in the main value columns and static covariates (creative features + ad ID tag) via the `static_cols` argument.
  - **Static Covariate Encoding:** The `StaticCovariatesTransformer` with `OneHotEncoder` was used to numerically encode all categorical static features, fitting on the combined train/test set vocabulary before transforming. The encoded ad ID tag columns were excluded from the features passed to the model.
  - **Scaling:** Target and past covariates were scaled using `Scaler(global_fit=True)`. Encoded static covariates were not scaled but were re-attached to the scaled target TimeSeries.
- **Hyperparameter Tuning:** Optuna [1] was used with the nested validation approach (Section 3.7). The objective function (`objective_gru`) minimized MAE on the internal validation split over 15 trials. Tuned hyperparameters included `hidden_dim` (size of the GRU hidden state), `n_rnn_layers` (number of stacked GRU layers), and `dropout`. PyTorch Lightning's `EarlyStopping` callback (monitoring validation loss, patience 5) was used during Optuna trials and final training. GPU acceleration was utilized.
- **Training and Forecasting:** The best hyperparameters were used to configure the final `BlockRNNModel(model='GRU')`. It was trained on the full fold training data using early stopping based on a 90/10 validation split. Predictions were generated using `historical_forecasts` (`start=LOOKBACK_PERIOD`, `retrain=False`), performing sequential one-step-ahead forecasts. Predictions were inverse-transformed using the target scaler. As the model's final linear output layer is unconstrained and can produce negative values, a post-processing step was applied to clip all final, unscaled predictions at zero (i.e., $\max(0, \hat{y})$) to enforce the non-negativity constraint.

3.9.5 XGBoost

XGBoost (eXtreme Gradient Boosting) [9] , a highly efficient gradient boosting library, was implemented using the `darts.models.SKLearnModel` wrapper. This approach frames the time series forecasting problem as a supervised regression task.

- **Model Specification:** An `XGBRegressor` model was used. The `SKLearnModel` wrapper automatically creates lagged features from the target variable and past covariates. Based on the seasonality analysis and consistency with other models, a lookback window of 14 days was used (`lags=14`, `lags_past_covariates=14`). This means the model used the previous 14 days of the target variable and all selected features to predict the next day's value.
- **Feature Sets:** Two distinct versions of the XGBoost model were trained and evaluated for each fold:
    1. **Performance Only:** This version used only the dynamic performance features. These included lagged values (up to 14 days) of core metrics (`clicks_all` and `amount_spent`, etc.), rolling window statistics (7 and 14-day mean/std deviation), and temporal features (`day_of_week`, `month`, `is_weekend`). Crucially, current-day performance metrics were excluded from the feature set to prevent data leakage, ensuring only past information was used for prediction.
    2. **Performance + Creative:** This version augmented the "Performance Only" set by adding all static creative features (derived from the LLM analysis and name parsing, e.g., features starting with `creative_format_`, `brand_and_emotion_`, `_has_` and `_word_count`, etc.). These static features provide the model with context about the specific ad being predicted, constant across all its time steps.
- **Data Preparation:** Input data was sourced from the master feature set CSV. Within each fold, the training and testing dataframes were converted into lists of `darts.TimeSeries` objects, grouped by `ad_platform_id`. Time series shorter than 15 days (`LOOKBACK_PERIOD + 1`) were filtered out. Both the target variable (`clicks_all`) and the feature sets were scaled independently using `darts.dataprocessing.transformers.Scaler` before being fed into the model.
- **Hyperparameter Tuning:** Optuna [1] was used for hyperparameter optimization within each fold, following a nested validation procedure similar to that described in Section 3.7. The training data of the fold was split 80/20 for internal training and validation. The Optuna objective function (`objective_xgboost`) minimized the MAE on the internal validation set over 15 trials (`N_TRIALS_OPTUNA`). Tuned hyperparameters included `max_depth`, `learning_rate`, `subsample`, and `colsample_bytree`. To leverage XGBoost's built-in early stopping mechanism during Optuna trials (preventing overfitting and speeding up trials), the lagged validation data (`eval_set`) was manually constructed using a helper function (`create_lagged_data`) and passed to the underlying `XGBRegressor.fit()` method via the Darts wrapper's `fit_kwargs`. A fixed high number of estimators (2000) was set, allowing early stopping (with a patience of 15 rounds) to determine the optimal number of trees.
- **Training and Forecasting:** After identifying the best hyperparameters via Optuna for a given fold, a final XGBoost model instance was created with these parameters. This final model was trained once on the *entire* training dataset for that fold (again, potentially using early stopping with a manually constructed validation set from the last 10% of the training data). Predictions for all test series in the fold were then generated using the `historical_forecasts` method with `start=LOOKBACK_PERIOD` and `retrain=False`.

This means the single, optimized model trained on the fold's training data was used to make sequential one-step-ahead predictions across the entire forecast horizon for each test series without being refit at each step. Predictions were inverse-transformed using the fitted target scaler before evaluation. As the regressor's output is unconstrained and could theoretically produce negative values, a post-processing step was applied to clip all final, unscaled predictions at zero (i.e., $\max(0, \hat{y})$) to enforce the non-negativity constraint of the target variable.

- **Feature Importance:** The feature importances (based on gain) were extracted from the underlying `XGBRegressor` model after training on the full data for each fold and for each version (Perf. Only vs. Perf. + Creative). The script included logic to attempt matching importance scores with the correct lagged feature names generated by the Darts wrapper. These fold-level importances were later aggregated to provide an overall view.

3.9.6 SARIMAX

The SARIMAX (Seasonal AutoRegressive Integrated Moving Average with Exogenous Variables) [12] model from the `statsmodels` library was implemented as a classical time-series benchmark. Given its univariate nature, a separate SARIMAX model was fit for each individual ad time-series within each cross-validation fold.

- Model Specification: The model was specified to handle the data's specific characteristics:
  - **Exogenous Regressors (X)**: A limited set of regressors was chosen for parsimony and direct relevance to short-term dynamics: `amount_spent_lag_1` (recent budget), `clicks_all_lag_1` or `leads_all_lag_1` (recent momentum), and `is_weekend` (weekly pattern).
  - **Target Transformation:** Both the `clicks_all` and `leads_all` targets represent non-negative count data ($y \geq 0$). Such data often produce non-normally distributed residuals, violating a key assumption of the standard SARIMAX model. Moreover, the Gaussian likelihood used in SARIMAX can yield negative forecast values, which are not meaningful for count outcomes. To resolve this mismatch, stabilize the vari-

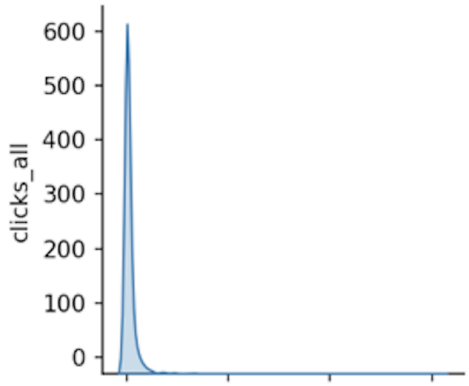


Figure 8: Distribution of raw `clicks_all` data (from the EDA pair plot in subsection 3.3). The extreme right-skew and non-negativity justifies the use of a stabilizing square root transformation.

ance, and make the data's distribution more symmetric, a square root transformation ($\sqrt{y}$) was applied. The model was therefore specified to predict the square root of clicks, not the raw count.

- Order Selection: To account for the unique characteristics of each ad, the optimal non-seasonal order (p, d, q) and seasonal order $(P, D, Q, m = 7)$ were determined *once per series.* This was achieved using the `pmdarima.auto_arima` function on the transformed $(\sqrt{y})$ initial 14 days of data (`LOOKBACK_PERIOD`) for that specific ad. The function automatically searched for the best orders minimizing the AIC (Akaike Information Criterion) within a constrained search space (e.g., $p, q \leq 2$; $P, Q \leq 1$). A fallback order of $(1, 0, 1)(1, 0, 0, 7)$ was used if `auto_arima` failed.

- Forecasting Method: A rolling forecast approach was employed. Starting from day 15 (`MIN_TRAIN_LENGTH`), the model was refit for each subsequent day using the square root of the target data $(\sqrt{y})$ from the preceding 14 days (`LOOKBACK_PERIOD`). The predetermined optimal order and the three exogenous regressors were used in each refitting step. The refitted model produced a one-step-ahead forecast in the transformed (square root) space. This forecast was then **squared ($\hat{y}^2_{\text{transformed}}$) and corrected for re-transformation bias by adding the model's residual variance ($\hat{\sigma}^2$)** to return an **unbiased prediction** in the original scale, mathematically guaranteeing a non-negative forecast.

3.10 Creative feature impact analysis (SQ3)

To address SQ3—identifying which creative features significantly impact lifetime performance and their relative importance—a dedicated analysis pipeline was executed. This analysis focused exclusively on the relationship between the static, pre-launch creative features and the overall lifetime efficiency metrics calculated for each ad.

3.10.1 Data preparation for lifetime analysis

The analysis began by loading the pre-generated master feature set, which contained daily data including both performance and creative features. This daily data was then aggregated to create a dataset with one row per unique ad (`ad_platform_id`). During aggregation:

- Core performance metrics (`clicks_all`, `leads`, `reach` and `amount_spent`) were summed over the entire lifetime of each ad to get totals.

- Static creative features (derived from the LLM extraction and other sources).

- Five key lifetime efficiency metrics were calculated based on these totals, corresponding to the metrics mentioned in SQ3:

  - Click-Through Rate (CTR): $\frac{\text{total\_clicks}}{\text{total\_reach}} \times 100$

  - Conversion Rate (CR): $\frac{\text{total\_leads}}{\text{total\_clicks}} \times 100$

  - Funnel Conversion Rate (FCR): $\frac{\text{total\_leads}}{\text{total\_reach}} \times 100$

  - Cost Per Click (CPC): $\frac{\text{total\_spend}}{\text{total\_clicks}}$

  - Cost Per Lead (CPL): $\frac{\text{total\_spend}}{\text{total\_leads}}$

Crucially, to isolate the impact of creative choices, all dynamic performance features (e.g., daily metrics, rolling statistics, date-based features like `day_of_week`, `month`) were explicitly identified and **dropped** from the aggregated dataset before the main analysis. This ensures that the subsequent steps evaluate only the predictive power of the creative elements themselves on the final lifetime outcomes.

3.10.2 Statistical significance testing

For each of the five lifetime efficiency metrics (used as the target variable), the relationship with each individual creative feature was assessed for statistical significance using a threshold of $p < 0.05$:

- **Categorical creative features** (including binary flags): One-way Analysis of Variance (ANOVA) was used to test if the mean efficiency metric differed significantly across the different categories or levels of the feature.
- **Post-hoc analysis:** For categorical features found significant by ANOVA, the Tukey Honestly Significant Difference (HSD) test was performed to identify which specific pairs of categories had statistically significant differences in their mean efficiency metric.
- **Continuous creative features** (e.g., `content_and_offer_number_of_people`, counts like `num_advantage_cat`): Pearson correlation coefficient was calculated to measure the linear relationship between the feature and the efficiency metric, along with its associated p-value.

These tests identify creative features that have a statistically verifiable association with the overall performance metrics.

3.10.3 Relative importance ranking

To determine the relative importance of the creative features in predicting the lifetime efficiency metrics, Permutation Importance was calculated. For each efficiency metric:

- An XGBoost regression model was trained using **only** the identified creative features as predictors and the lifetime efficiency metric as the target.
- The permutation importance of each creative feature was computed by measuring the decrease in model performance (R-squared or similar metric) when the values of that single feature were randomly shuffled.
- Features were ranked based on their mean importance score across multiple shuffling repeats.

This method provides a model-based assessment of which creative features are most influential in predicting the final outcome, complementing the individual statistical tests.

3.10.4 Interaction effects analysis

To investigate potential interaction effects between the most important creative features, Ordinary Least Squares (OLS) regression models were employed.

- The top N (e.g., $N = 10$) most important *categorical* creative features (identified via permutation importance) were selected. This focus on categorical-categorical interactions was a deliberate choice to prioritize the discovery of the most interpretable and directly actionable insights, as opposed to the more complex analysis of continuous-variable interactions.
- For pairs of these top features, an OLS model was fitted including main effects for both features and their interaction term (e.g., Target $\sim C(\text{Feature1}) + C(\text{Feature2}) + C(\text{Feature1}) : C(\text{Feature2})$).
- An ANOVA was performed on the fitted OLS model to obtain the p-value specifically for the interaction term.

- Interaction plots were generated for visual inspection, annotated with the interaction p-value.

This step helps uncover synergistic or antagonistic effects where the impact of one creative feature depends on the level of another.

3.10.5 Subset analysis

All analyses (statistical significance, importance ranking, interaction effects) were performed on all the ads of which the corresponding extracted creative features were present.

This comprehensive analysis pipeline directly addresses SQ3 by systematically evaluating the significance, importance, and interactions of creative features concerning key lifetime performance indicators, after controlling for dynamic performance influences.

4 Results

This section presents the results of the classification and forecasting models developed to address the research questions.

4.1 Classifier Performance (SQ1)

The first sub-question (SQ1) investigated the extent to which the potential success ("Good" vs. "Bad", defined in Section 3.8) of a recruitment ad could be predicted using only its creative features (Day 0) and how this prediction evolved with the inclusion of early performance data (Days 1-14). An XGBoost classifier was trained and evaluated using 5-fold group cross-validation for each time point.

4.1.1 Overall Performance Evolution

Figure 9 illustrates the mean performance metrics (F1-Score, Accuracy, Recall and Precision) across the 5 folds as more days of cumulative training data were included. The table below (summarized from log output) shows the aggregated performance for the "Total" dataset split by the target class.

Table 1: Mean Classifier Performance Metrics (5-Fold CV - Total Ads)

Class	Metric	Day 0	Day 2	Day 7	Day 14
Not Chosen	F1-Score	0.729	0.793	0.796	0.819
	Recall	0.770	0.830	0.822	0.821
	Precision	0.693	0.760	0.771	0.819
Chosen	F1-Score	0.226	0.446	0.579	0.681
	Recall	0.197	0.399	0.543	0.679
	Precision	0.267	0.509	0.620	0.690
Overall	Accuracy	0.599	0.699	0.725	0.770

Note: "Chosen" refers to the 'Good' class, "Not Chosen" to the 'Bad' class.

Several key observations emerge:

- **Day 0 Performance:** Using only creative features, the model achieved a mean accuracy of approximately 59.9%. However, performance was significantly imbalanced. The F1-score for the majority class ("Not Chosen") was reasonable (0.73), but very low for the minority class ("Chosen", 0.23), indicating difficulty in identifying potentially successful ads based solely on pre-launch characteristics. Recall for "Chosen" ads was particularly low (around 20%).
- **Impact of Early Data:** Incorporating just the first few days of performance data greatly improved the model's ability to identify "Chosen" ads. The F1-score for this minority class jumped from 0.23 to 0.45 by Day 2 and continued to rise steadily, reaching 0.68 by Day 14. Crucially, the model's already-strong ability to identify the "Not Chosen" class was also solidified; its F1-score rose from 0.73 to over 0.80 by Day 2. This shows that the main value of early data is in learning to separate the rare "Good" ads from the "Bad" majority. Overall accuracy also increased significantly, reaching approximately 71% by Day 2 and 77% by Day 14.

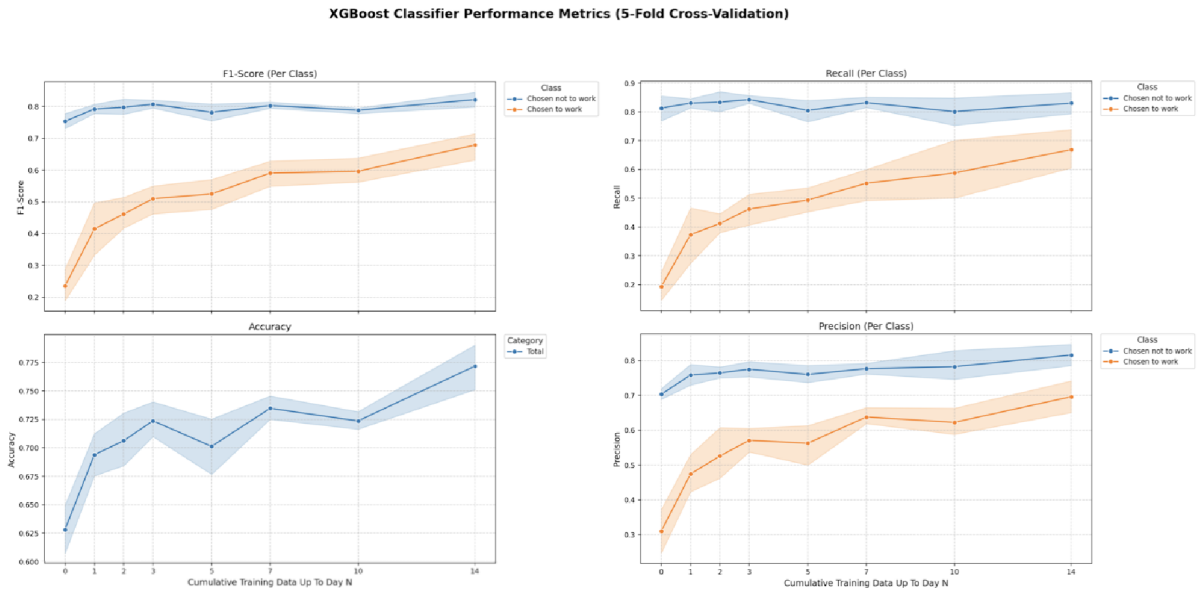


Figure 9: Model Performance Metrics Evolution (5-Fold Cross-Validation). Shaded areas represent +/- one standard deviation across folds.

- **Metric Convergence:** Most performance gains occurred within the first 2-5 days. While metrics continued to improve slightly up to Day 14, the marginal benefit of additional data decreased over time. Precision and Recall for both classes generally improved, suggesting the model became better at both identifying the correct class and avoiding misclassifications as more data became available.
- **Class Imbalance Effect:** The performance difference between the "Chosen" (minority) and "Not Chosen" (majority) classes remained noticeable throughout, although the gap narrowed significantly with the addition of performance data. This highlights the inherent challenge of predicting the rarer "Good" outcome.

4.1.2 Confusion Matrices and Class Distinction

Figure 10 shows the aggregated confusion matrices across the 5 folds for Days 0, 2, 7, and 14.

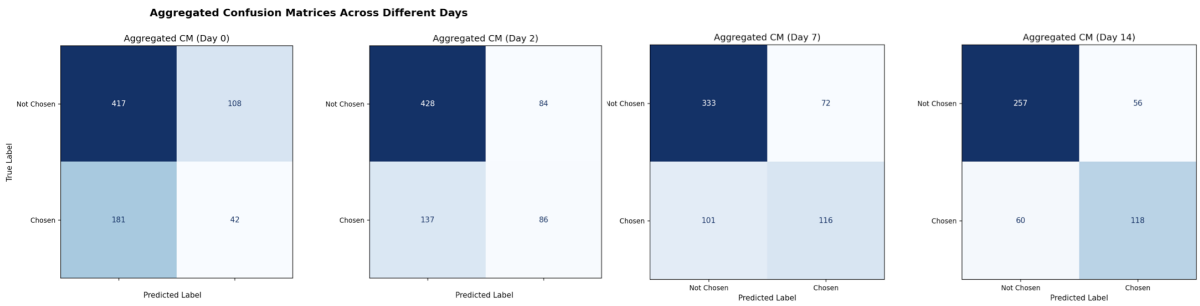


Figure 10: Aggregated Confusion Matrices Across Folds for Different Training Data Durations.

The matrices visually confirm the trend observed in the metrics:

- At Day 0, using only static features, the model correctly identified **42** "Chosen" ads (True Positives, TP) while misclassifying **181** as "Not Chosen" (False Negatives, FN). It performed well on the "Not Chosen" class, achieving **417** correct predictions (True Negatives, TN) against **108** False Positives (FP).

– By Day 2, the number of correctly identified "Chosen" ads (TP) was **86**, with False Negatives (FN) decreasing to **137**. The model's performance on the "Not Chosen" class saw True Negatives (TN) rise slightly to **428** and False Positives (FP) drop to **84**, validating the use of early performance data.

– At Day 7 and Day 14, the model continued to refine its distinction. True Positives (TP) were **116** (Day 7) and **118** (Day 14). False Negatives (FN) continued to decrease to **101** (Day 7) and **60** (Day 14). Concurrently, True Negatives (TN) were **333** and **257**, respectively, demonstrating a clear and consistent improvement in the model's ability to identify the "Chosen" class.

Overall, the confusion matrices demonstrate the model's increasing ability to distinguish between the two performance classes as it gains access to early performance signals.

4.1.3 Precision-Recall Analysis

To better evaluate the model's performance on the imbalanced classes, Precision-Recall (PR) curves were generated for key time points, focusing on the "Not Chosen" (majority) class as the positive target for consistency with the plots generated. Figure 11 shows the PR curves for Day 0, 2, 7, and 14 combined.

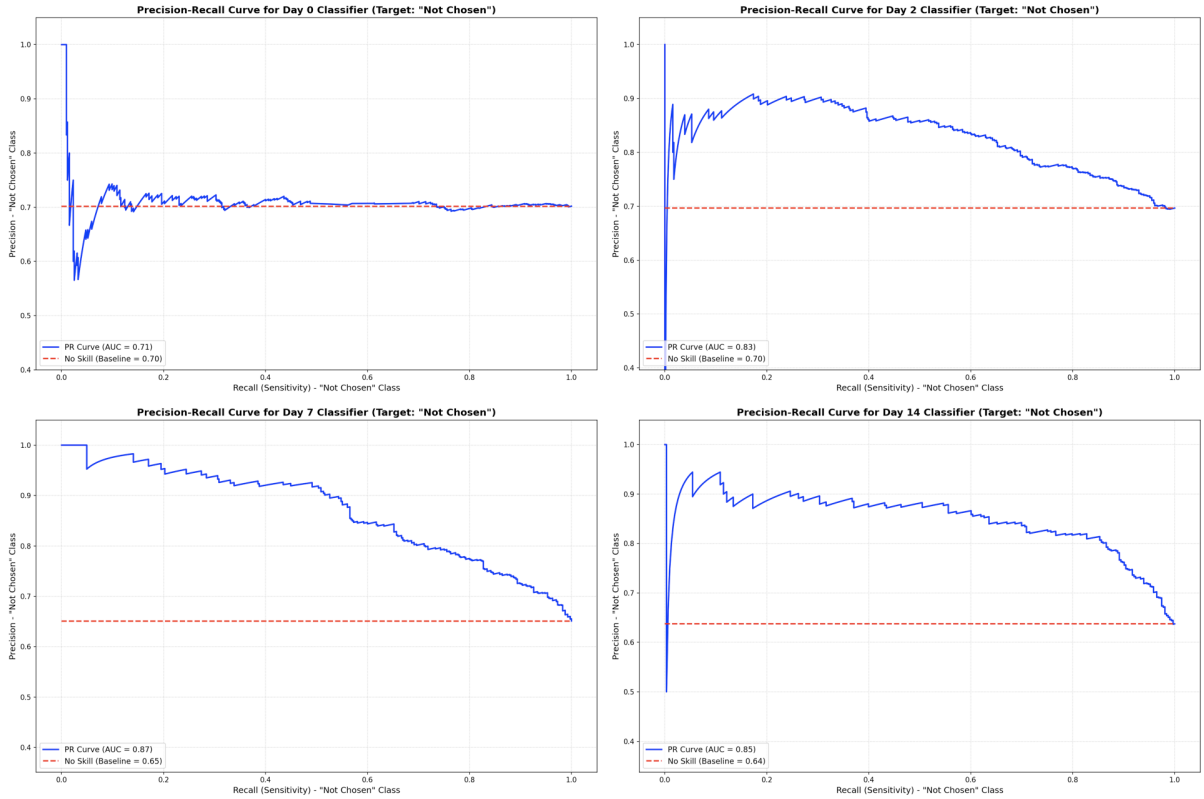


Figure 11: Precision-Recall Curves for the "Not Chosen" Class at Different Training Data Durations (Aggregated Across Folds). The red dashed line indicates the no-skill baseline.

The Area Under the PR Curve (AUC-PR) provides a summary measure of the model's ability to achieve high precision and high recall simultaneously.

- At Day 0, the AUC-PR was 0.71, only slightly above the no-skill baseline of 0.70. This confirms the limited predictive power of creative features alone for this class definition and dataset.

- By Day 2, the AUC-PR jumped significantly to 0.83, indicating a much better trade-off between precision and recall.
- The AUC-PR continued to improve, reaching 0.87 by Day 7 and 0.85 by Day 14 (slight variations between Day 7 and 14 might be due to fold aggregation). The curves consistently stay well above the no-skill line, demonstrating the model's value once performance data is included.

These curves reinforce that while creative features offer minimal predictive signal at launch (Day 0), incorporating just a few days of performance data allows the model to differentiate between "Good" and "Bad" ads with significantly higher confidence.

4.1.4 Feature Importance

To understand which features contributed most to the predictions, feature importance was calculated based on the XGBoost model's internal gain metric, averaged across all folds and days where the feature was used. Figure 12 shows the top 20 creative features, while Figure 13 shows the top 20 performance features (relevant for Day 1 onwards).

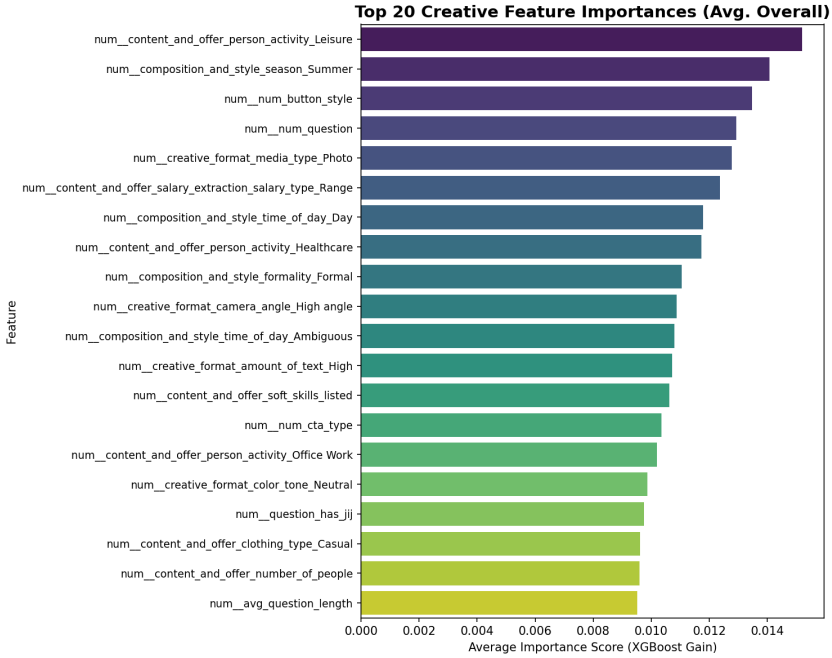


Figure 12: Top 20 Creative Feature Importances (Averaged Across Folds and Days).

Key insights from feature importance include:

- **Top creative features:** The most influential creative attributes identified by the model are depicting **leisure activities** (`num_content_and_offer_person_activity_Leisure`), the **season being summer** (`num_composition_and_style_season_Summer`), and the **number of button styles** (`num_num_button_style`). Other highly important factors include the **number of questions** (`num_num_question`), using a **photo** (`num_creative_format_media_type_Photo`), and mentioning a **salary range** (`num_content_and_offer_salary_extraction_salary_type_Range`). Activities like **office work** and **healthcare**, as well as creative formatting choices such as **formal style** and **high text amount**, also ranked highly.
- **Top performance features:** Once available, recent performance metrics dominate importance. The top three are **raw amount spent** (`num_amount_spent`), **raw reach**

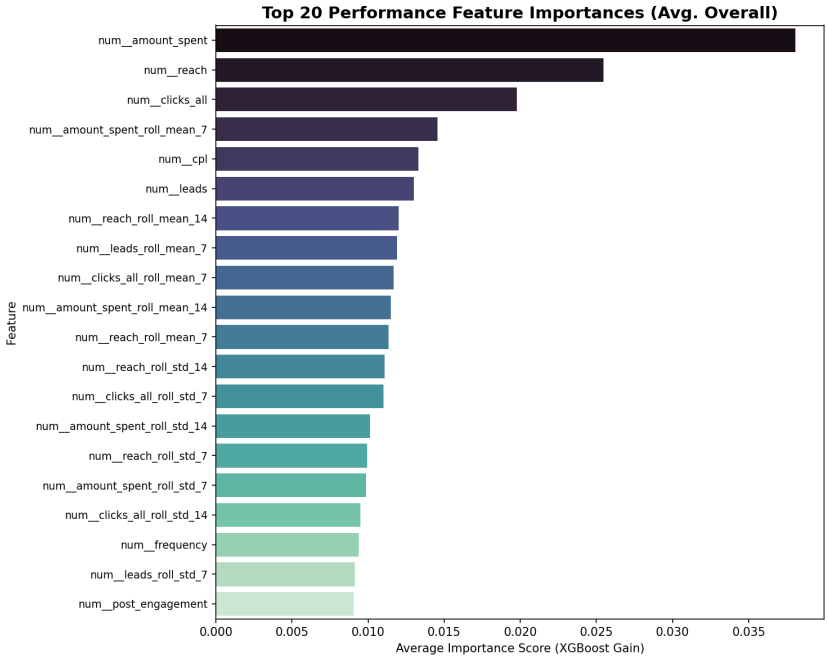


Figure 13: Top 20 Performance Feature Importances (Averaged Across Folds and Days 1-14).

(`num_reach`), and **raw clicks** (`num_clicks_all`). Short-term rolling means (e.g., `num_amount_spent_roll_mean_7`) were more important than longer-term averages or standard deviations initially. Efficiency metrics like **cost per lead** (`num_cpl`) also quickly gained importance. Notably, text-derived features (word counts, average question length) were absent from the top 20, indicating the model relies primarily on **direct performance momentum and volatility** once early data is available.

These importance rankings help answer the final part of SQ1, highlighting specific creative elements that associate with the "Good"/"Bad" label at Day 0, and confirming the dominance of direct performance metrics once they become available.

4.2 Forecasting Performance (SQ2)

The second sub-question (SQ2) evaluated the accuracy of TCN, XGBoost, GRU, and SARIMAX models against a baseline for forecasting next-day `clicks_all` and `leads`. This section summarizes the comparative performance of all models and provides a detailed analysis based on the 5-fold cross-validation experiments.

4.2.1 Overall Model Comparison (Clicks)

Table 2 presents the aggregated mean performance metrics for all models predicting `clicks_all`. The models are sorted by Mean Absolute Error (MAE) from lowest (best) to highest (worst).

The `clicks` table is already using the correct, consistent baseline values.

Table 2: Overall Mean Forecasting Performance (5-Fold CV - Target: `clicks`)

Model	MAE	RMSE	F1-Score	MASE	Runtime (hrs)	ΔMAE vs Baseline
TCN (Perf. + Creative)	**1.597**	5.180	0.782	**0.558**	27.01	+45.5%
TCN (Perf. Only)	1.970	5.006	0.622	0.692	27.01	+32.8%
GRU (Perf. Only)	2.223	7.122	0.598	0.775	34.59	+24.1%
Realistic Baseline	**2.929**	9.513	0.777	0.907	–	Reference
SARIMAX (Simple)	3.248	13.994	0.810	0.953	1.42	-10.9%
XGBoost (Perf. + Creative)	3.341	12.002	0.660	1.036	4.23	-14.0%
XGBoost (Perf. Only)	3.346	12.000	0.647	1.036	4.23	-14.2%
GRU (Perf. + Creative)	4.489	9.689	0.439	1.567	34.59	-53.2%

The results for predicting clicks show a clear hierarchy:

- **TCN models** were the top performers, with `TCN (Perf. + Creative)` achieving the best MAE (1.597), a **45.5% improvement** over the baseline.
- **GRU (Perf. Only)** also performed well, improving MAE by **24.1%** over the baseline.
- **SARIMAX** and **XGBoost models** underperformed, failing to surpass the `Realistic Baseline`.
- Adding **creative features** improved TCN performance but worsened GRU performance (MAE 4.489), highlighting sensitivity to high-dimensional inputs.

4.2.2 Overall Model Comparison (Leads)

Table 3 summarizes results for the sparser `leads` target, using the averaged baseline metrics from all experiments.

Table 3: Overall Mean Forecasting Performance (5-Fold CV - Target: `leads`)

Model	MAE	RMSE	F1-Score	MASE	Runtime (hrs)	ΔMAE vs Baseline
GRU (Perf. Only)	**0.165**	0.464	0.482	**0.421**	30.56	+24.5%
TCN (Perf. Only)	0.192	0.558	0.522	0.513	20.51	+12.1%
TCN (Perf. + Creative)	0.200	0.520	0.525	0.535	20.51	+8.5%
GRU (Perf. + Creative)	0.206	0.534	0.401	0.525	30.56	+5.7%
XGBoost (Perf. Only)	0.216	0.621	0.025	0.548	3.13	+1.1%
XGBoost (Perf. + Creative)	0.216	0.623	0.033	0.549	3.13	+1.1%
Realistic Baseline	**0.219**	**0.597**	**0.432**	**0.546**	–	Reference
SARIMAX (Simple)	0.306	1.693	0.518	0.717	1.24	-40.0%

The results confirm the Deep Learning models' superiority for sparse targets and SARIMAX's weakness:

- **GRU (Perf. Only)** achieved the best MAE (0.165), a **24.5% improvement** over the baseline.
- **TCN models** performed moderately well, with improvements of **12.1%** (Perf. Only) and **8.5%** (Perf. + Creative).
- **XGBoost** achieved only a marginal improvement (**1.1%**).
- **SARIMAX** failed on this task, performing **40.0% worse** than baseline.
- Adding **creative features** to GRU worsened performance (MAE 0.206), suggesting these features added noise for sparse targets.

The predictive skill for the sparse `leads` target demonstrated a clear performance hierarchy, substantially exceeding the capability of the engineered baseline. The inclusion of the Mean Absolute Scaled Error (MASE) provides definitive validation of model superiority, as all models achieved MASE < 1.0, indicating performance statistically better than a zero-skill benchmark. The **GRU (Perf. Only)** model achieved the lowest Mean Absolute Error (MAE = 0.165) and the highest skill, with MASE = **0.421**. This confirms that the GRU's forecast error is only 42.1% of that produced by the Naïve benchmark, making it the most robust architecture for this low-volume, high-volatility task. The **TCN** models followed closely (MASE $\approx$ 0.51–0.53), demonstrating comparable structural competence. Notably, even the **Realistic Baseline** achieved a respectable MASE = **0.546**, confirming that both GRU and TCN delivered significant gains over an already skillful engineered reference.

Runtime and Computational Trade-Offs While the TCN and GRU models delivered superior predictive performance, their 5-fold cross-validation required 20–35 hours of sequential training on a `4gdn.xlarge` Linux PyTorch instance. By contrast, XGBoost and SARIMAX required only 1–4 hours. This highlights the trade-off between predictive power and computational cost, especially when handling high-dimensional and sparse targets.

4.2.3 Model-Specific Analysis: TCN (Clicks)

This analysis is based on the TCN clicks prediction run. The TCN models were trained using a 14-day lookback, with **27 time-varying performance features** (excluding the target). The static creative input was derived from a base of **270 categorical and binary features** that were then transformed using One-Hot Encoding (OHE) to create the final **453 static features** required by the TCN/GRU models.

Table 4: TCN performance and hyperparameters (**clicks**).

Variant	MAE	RMSE	F1-Score	MASE	Runtime (hrs)	Typical Hyperparameters (Range)
TCN (Perf. + Creative)	1.597	5.180	0.782	0.558	27.01	kernel_size=4–5, num_filters=32–48, dropout=0.10–0.34
TCN (Perf. Only)	1.970	5.006	0.622	0.692	27.01	kernel_size=3–4, num_filters=16–48, dropout=0.14–0.40
Realistic Baseline	**2.929**	**9.513**	**0.777**	**0.907**	–	–

Visual Analysis Figure 14 visualizes the distribution of Mean Absolute Error (MAE) across the 5 cross-validation folds for each model. It clearly shows that `TCN (Perf. + Creative)` not only has the lowest median MAE but also a tighter distribution than the baseline, indicating more consistent performance across the different test sets.

The scatter plots in Figure 15 compare the predicted values (y-axis) against the true values (x-axis) for all test predictions aggregated from the 5 folds. The `Realistic Baseline` (left) shows significant under-prediction, with most points falling below the red 'Perfect Prediction'

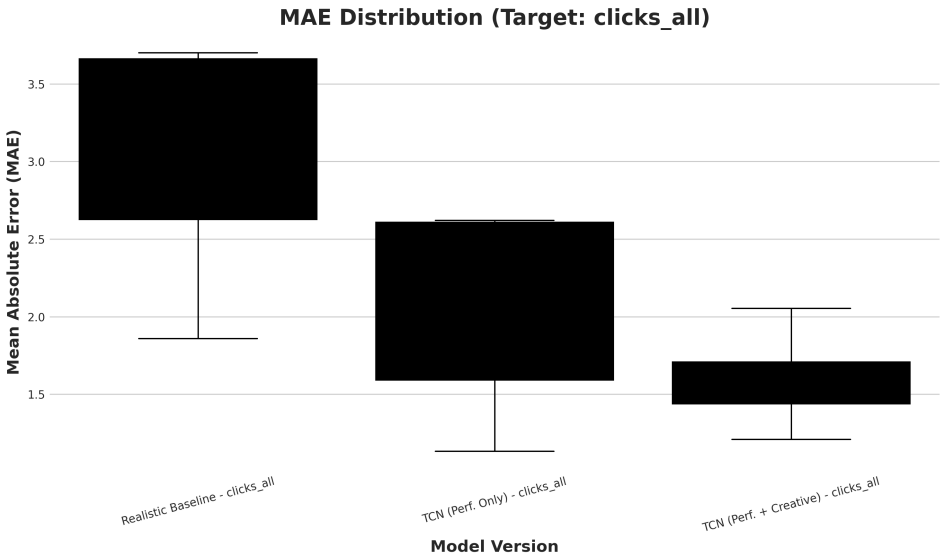


Figure 14: MAE Distribution by Model (Target: `clicks_all`). Results from 5-fold cross-validation.

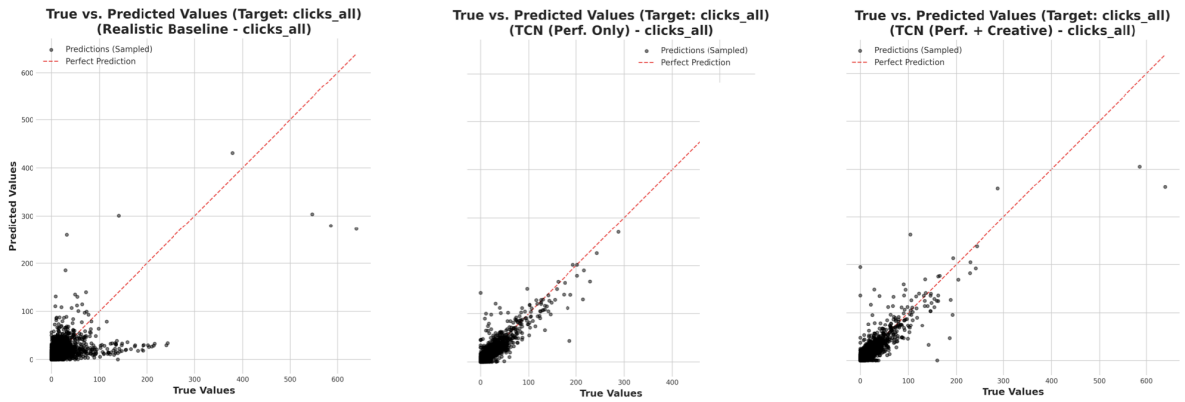
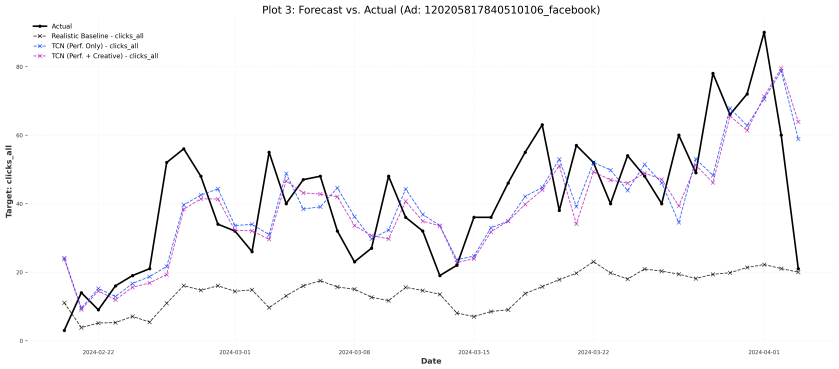


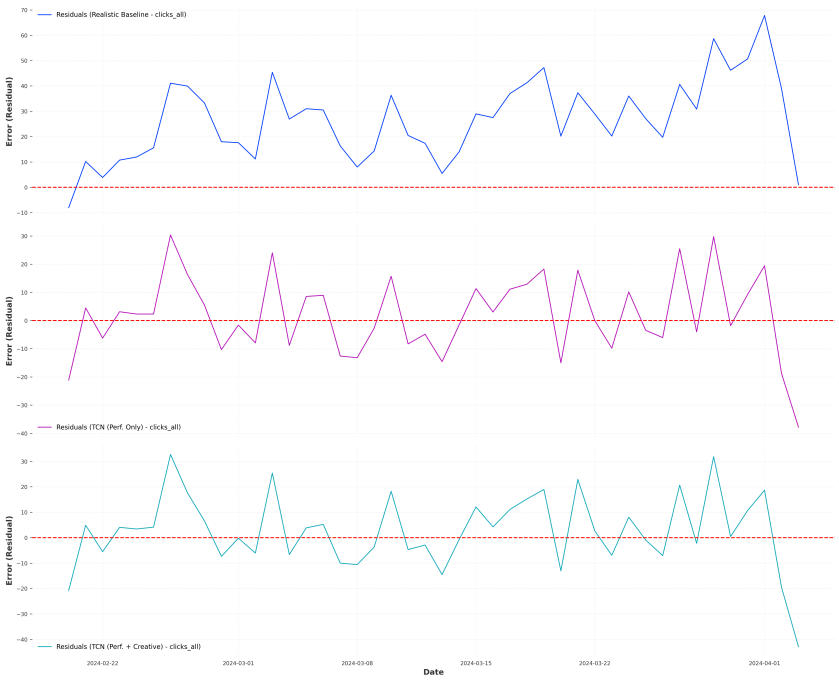
Figure 15: True vs. Predicted Values for `clicks_all` (Aggregated from all 5 folds).

line. The TCN models (two on the right) are much more tightly clustered around the line, demonstrating their superior accuracy.

Figures 16a and 16b provide a detailed forecast for a single representative ad. In Figure 16a, the TCN models (blue and purple) track the volatile actual data (black) far more closely than the baseline (black-dashed). Figure 16b confirms this: the baseline's residuals (blue, top) show a large, consistent positive error (under-prediction), while the TCN models' residuals (purple and teal) are centered much closer to zero.



(a) Example Forecast vs. Actual for a single ad.



(b) Corresponding Residuals (Error) over time for the same ad.

Figure 16: Example TCN Time Series Forecast and Residual Plots for `clicks_all`.

Feature Importance Analysis (TCN Clicks) Key insights from the SHAP analysis (Kernel Explainer on dynamic features) comparing the two TCN models for forecasting `clicks_all` include:

- **Dominance of Target Volatility (Perf. Only):** For the baseline `TCN (Perf. Only)` model (Figure 17), the top predictive features are dominated by the volatility of the target variable, `clicks_all`. Specifically, `clicks_all_roll_std_14_lag_14` and `clicks_all_roll_std_14_lag_12` demonstrate the highest importance, indicating the model prioritizes autocorrelation and historical volatility patterns of the target series in the absence of creative context.
- **Shift to Upper-Funnel Reach (Perf. + Creative):** When the static creative features are added (`TCN (Perf. + Creative)`, Figure 18), the model shifts its focus almost entirely to `reach` metrics. Features such as `reach_roll_std_14_lag_7` and `reach_roll_mean_7_lag_7` become the most important predictors. This highlights that the creative context enables the model to leverage the direct causal relationship between recent reach momentum and clicks.
- **Silent Static Features and Methodological Rationale:** Consistent with the `leads` analysis, none of the 453 static creative features appear in the SHAP summary plots. This outcome is due to the methodological constraint:

    1. **Computational Tractability:** The combined input space ($\approx$ 850 features) is too large for efficient and interpretable Kernel Explainer analysis.
    2. **Causality Focus:** The SHAP analysis is intentionally restricted to the dynamic inputs to interpret how the model uses temporal momentum and short-term trends (the lagged performance history) in its day-to-day forecasting logic. The static features serve as a constant contextual baseline that refines the TCN's sequential processing capability.

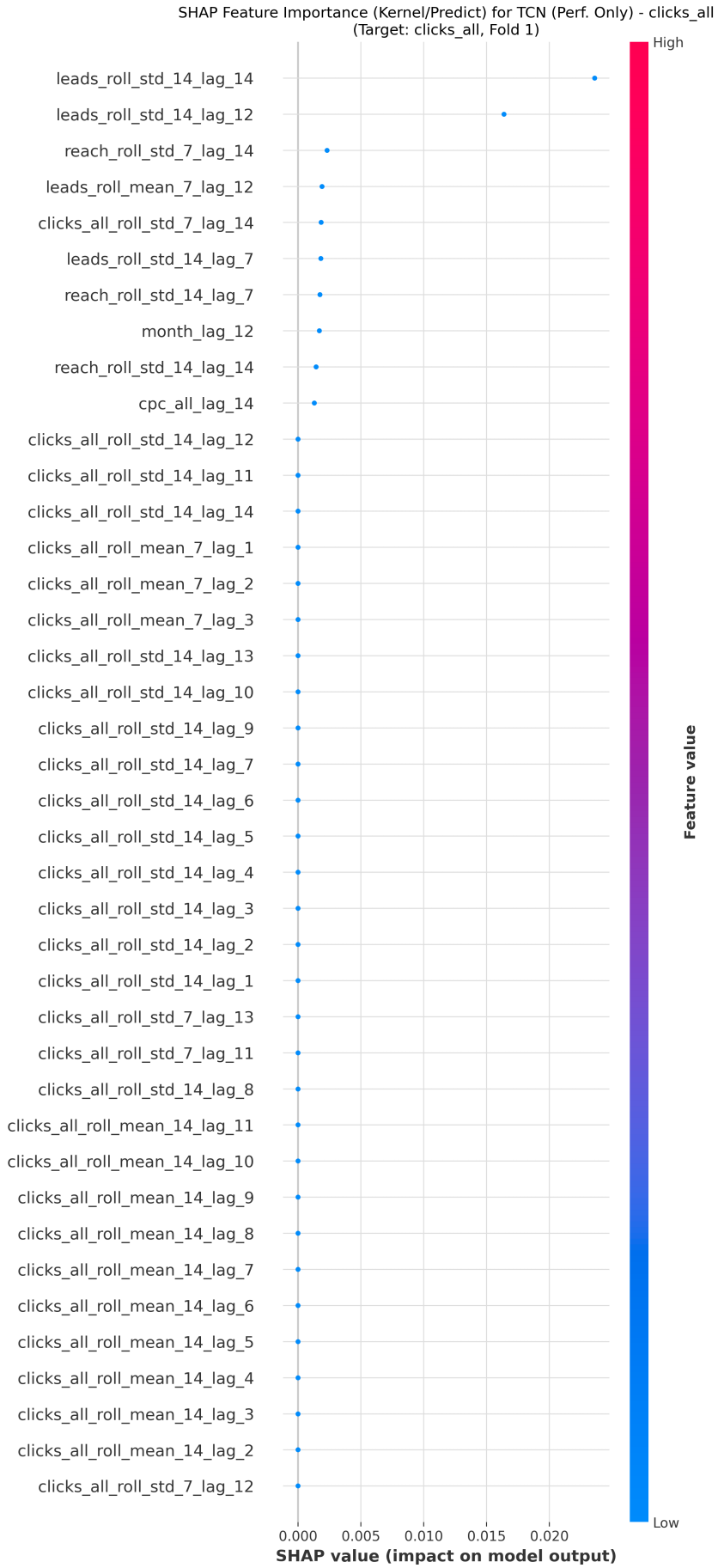


Figure 17: SHAP Feature Importance for TCN (Perf. Only) - `clicks_all` (Fold 1)

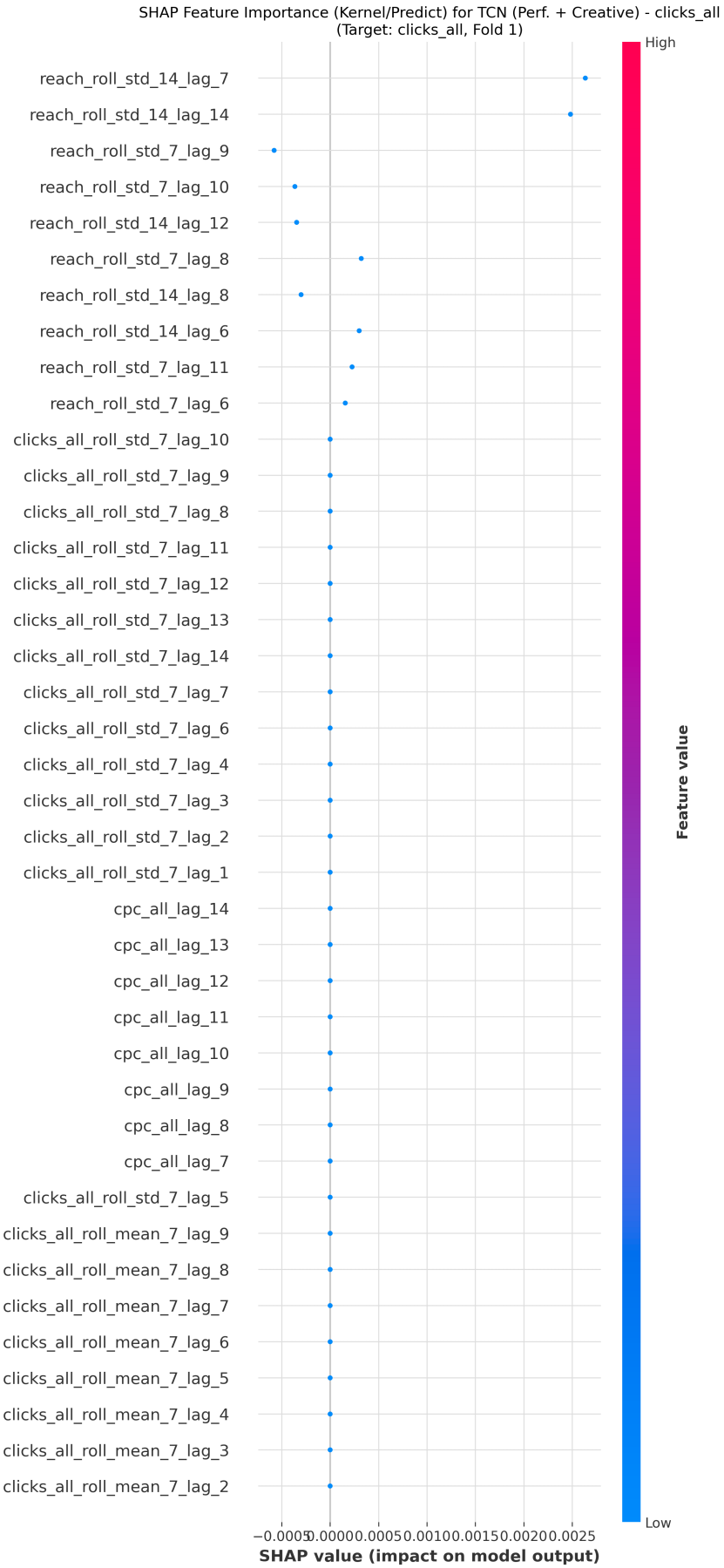


Figure 18: SHAP Feature Importance for TCN (Perf. + Creative) - `clicks_all` (Fold 1)

4.2.4 Model-Specific Analysis: TCN (Leads)

1. Model Configuration and Overall Performance This analysis is based on the TCN leads prediction run.

Key Model Parameters The TCN models were trained using the following configuration:

- **Lookback Period (Input Chunk Length):** 14 days
- **Forecast Horizon (Output Chunk Length):** 1 day
- **Static Feature Count (OHE):** 453 (derived from 270 base features)
- **Dynamic Covariate Count:** 27 time-varying performance features

Quantitative Performance The aggregated metrics highlight the TCN's strong, consistent performance on the sparse leads target.

Table 5: TCN performance and hyperparameters (**leads**).

Variant	MAE	RMSE	F1-Score	MASE	Runtime (hrs)	Typical Hyperparameters (Range)
TCN (Perf. Only)	0.192	0.558	0.522	0.513	20.51	kernel_size=4–5, num_filters=32–48, dropout=0.23–0.33
TCN (Perf. + Creative)	0.200	0.520	0.525	0.535	20.51	kernel_size=3–5, num_filters=16–48, dropout=0.13–0.35
Realistic Baseline	**0.223**	**0.604**	**0.437**	**0.563**	–	–

Visual Analysis Figure 19 shows the MAE distribution across the 5 folds. It visually confirms the findings from Table 3: the two TCN models perform similarly, with both being significantly more accurate (lower MAE) and more consistent than the `Realistic Baseline`.

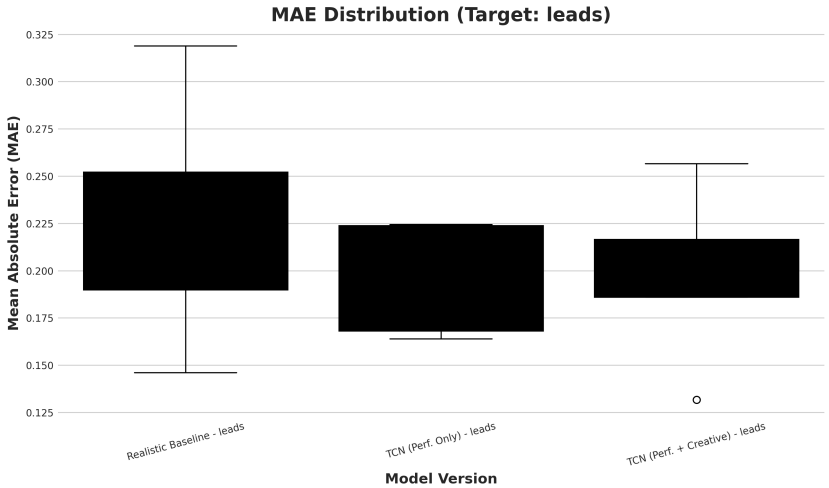


Figure 19: MAE Distribution by Model (Target: `leads`).

The scatter plots in Figure 20 reinforce this. The baseline model (left) shows a wide, scattered pattern. The two TCN models (two on the right) are much more tightly clustered around the perfect prediction line, especially in the critical 0-5 leads range where most data points lie.

The example forecast for a representative ad (Figure 21a) and its residuals (Figure 21b) illustrate the TCN models' effectiveness. Both TCN models (blue and purple) track the

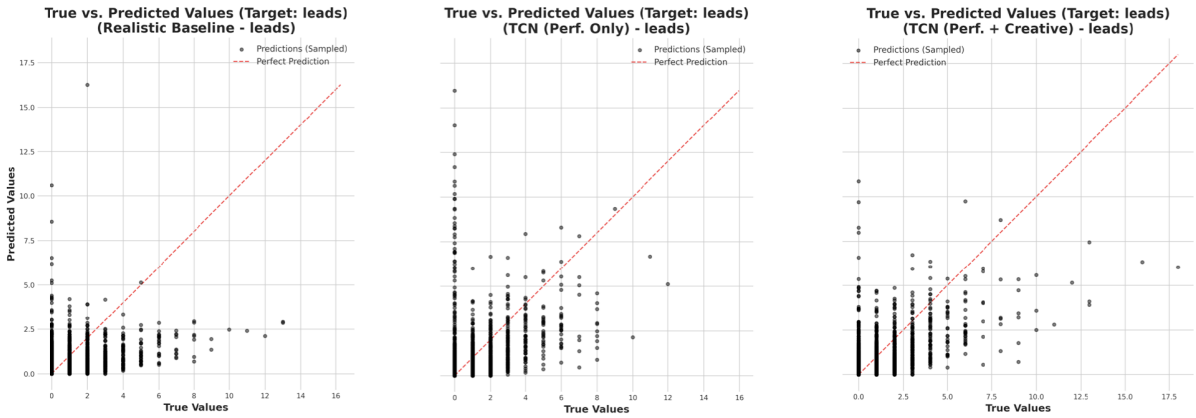
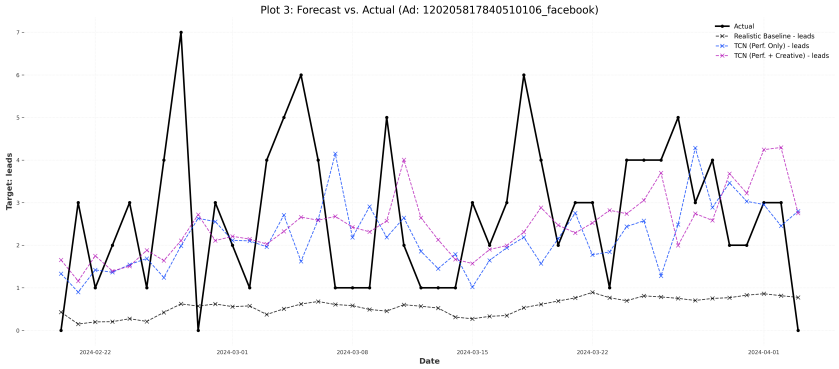
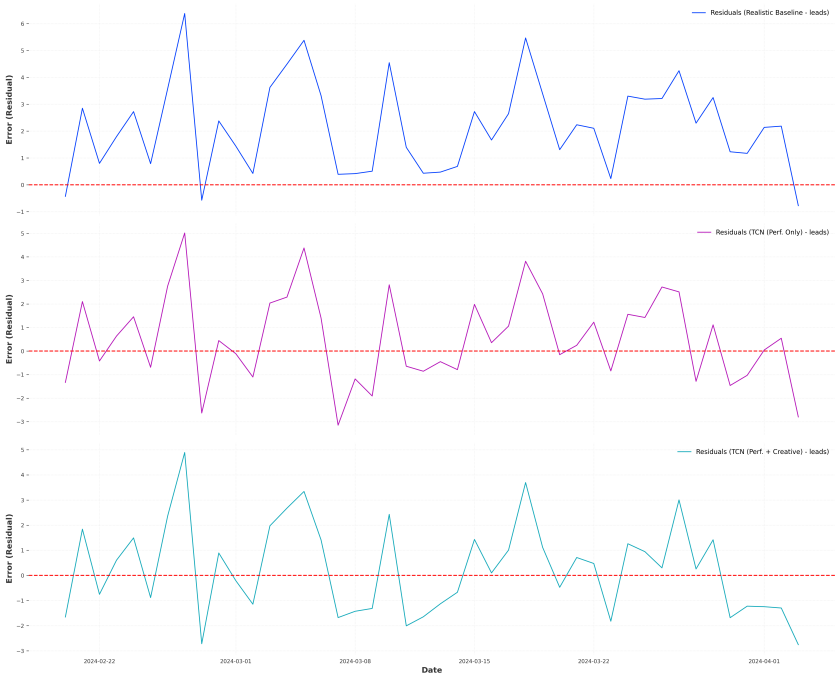


Figure 20: True vs. Predicted Values for `leads` (Aggregated from all 5 folds).

sparse, spiky actuals (black) far better than the flat-lining baseline (black-dashed). The residuals for the TCN models (**middle and bottom panels**) are clustered near zero, oscillating symmetrically around the zero-error line, which indicates **low bias**. In contrast, the baseline's residuals (**top panel**, blue) are almost entirely positive and consistently far above zero, confirming that the baseline model significantly and persistently **under-predicts** the actual lead volume.



(a) Example Forecast vs. Actual for a single ad.



(b) Corresponding Residuals (Error) over time for the same ad.

Figure 21: Example TCN Time Series Forecast and Residual Plots for `leads`.

Feature Importance Analysis (TCN Leads) Key insights from the SHAP analysis (Kernel Explainer on dynamic features) comparing the two TCN models for forecasting `leads` include:

- **Dominance of Direct Performance History (Perf. Only):** For the baseline `TCN (Perf. Only)` model (Figure 22), the top predictive features are overwhelmingly related to the historical target variable, `leads`. Specifically, recent rolling statistics, such as `leads_roll_std_14_lag_13` and `leads_roll_mean_7_lag_1`, are the primary drivers of the forecast. This indicates the model relies heavily on the volatility and recent trend of the leads series itself.
- **Shift in Focus with Creative Features:** When the static creative features are added (`TCN (Perf. + Creative)`, Figure 23), the top features shift their focus to dynamic metrics from the upper funnel. The most important predictors become the historical rolling statistics of `reach` and `cpc_all` (e.g., `reach_roll_std_7_lag_12` and `cpc_all_lag_14`). This suggests that the inclusion of the static creative context enables the model to better utilize the momentum and trend of related performance features.

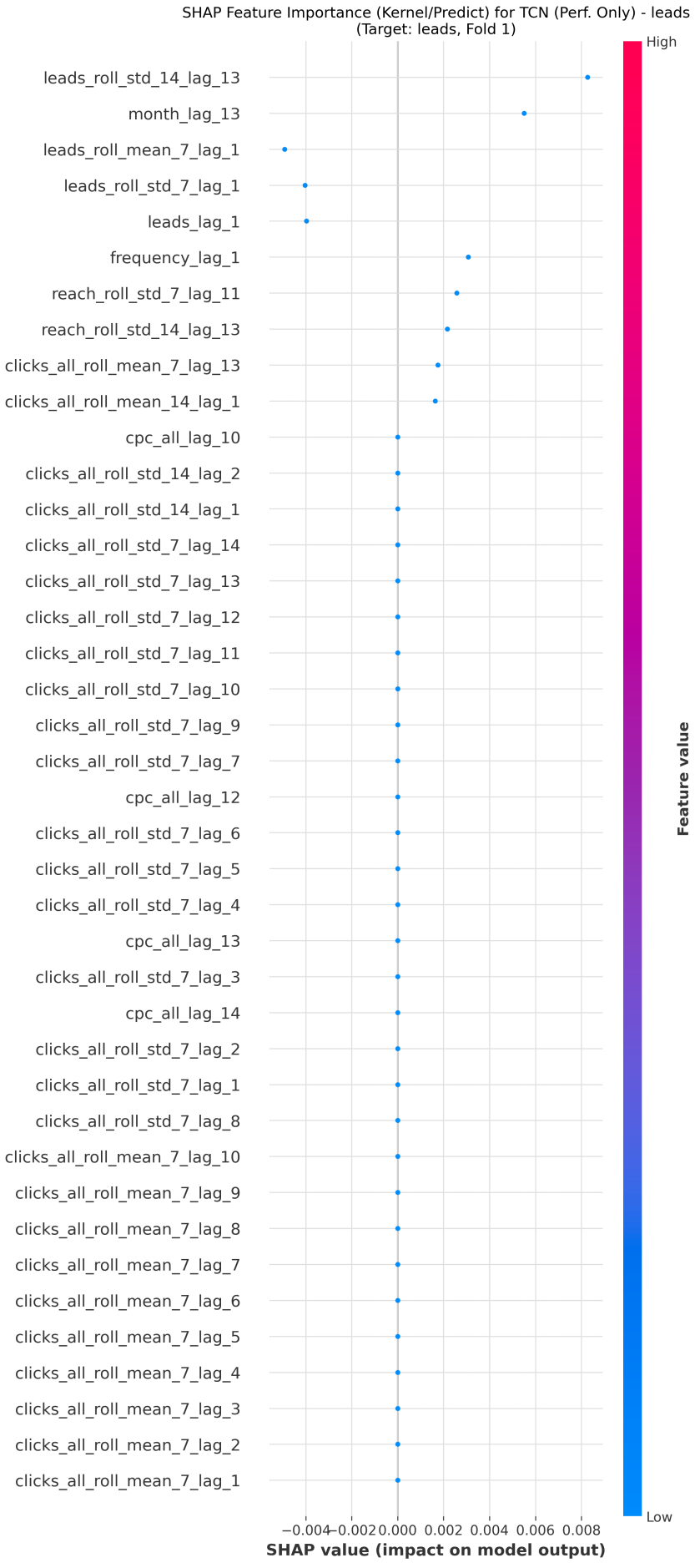


Figure 22: SHAP Feature Importance for TCN (Perf. Only) - `leads` (Fold 1)

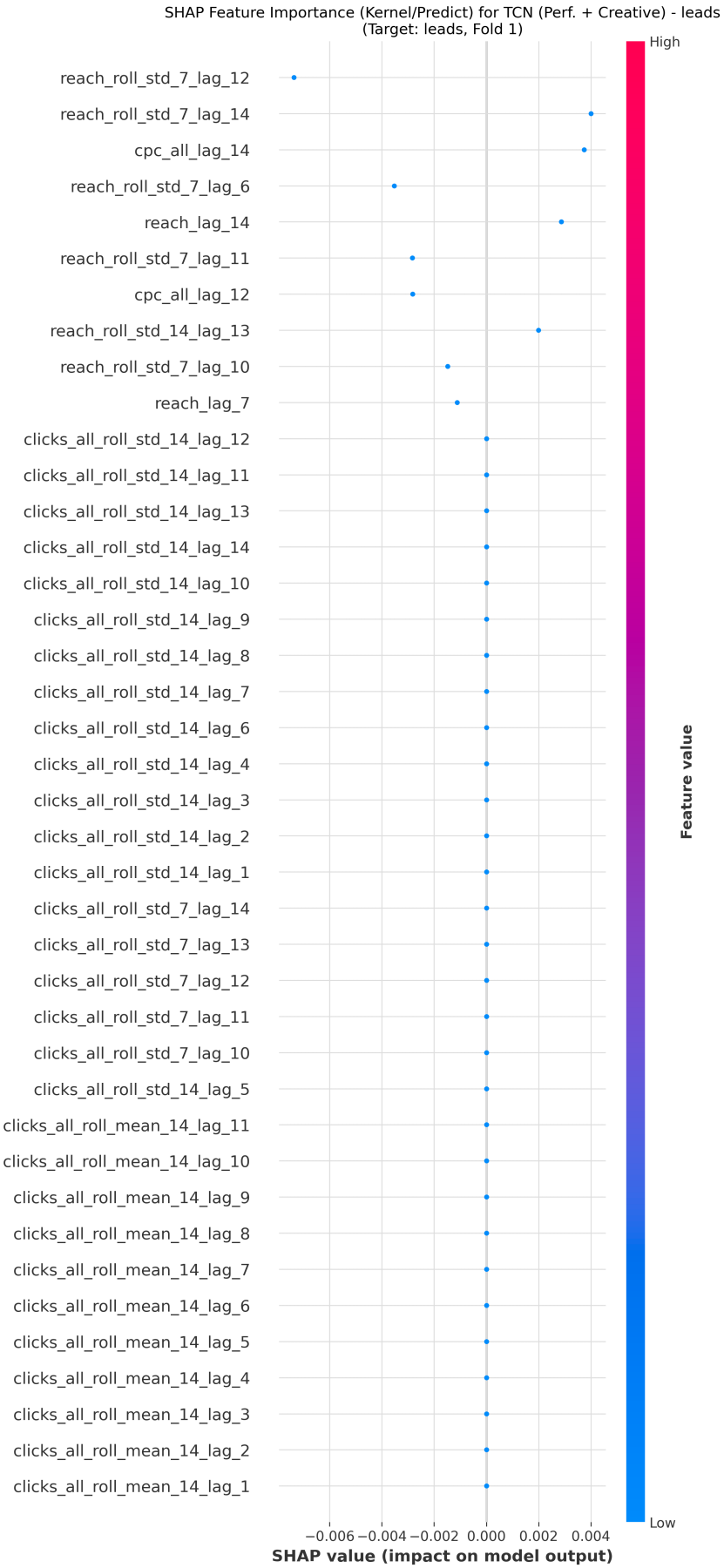


Figure 23: SHAP Feature Importance for TCN (Perf. + Creative) - `leads` (Fold 1)

4.2.5 Model-Specific Analysis: GRU (Clicks)

1. Model Configuration and Overall Performance This analysis is based on the Gated Recurrent Unit (GRU) model for the `clicks_all` target.

Key Model Parameters The final GRU model configuration used the following parameters:

* **Lookback Period (Input Chunk Length):** 14 days
* **Forecast Horizon (Output Chunk Length):** 1 day
* **Static Feature Count (OHE):** 453 (for Perf. + Creative model)
* **Dynamic Covariate Count:** 27 performance features + 1 target

Quantitative Performance The aggregated metrics for the GRU models confirm that the Performance Only model was significantly more stable and accurate than the model including creative features, which showed an increased error (MAE 4.489).

Table 6: GRU Overall Mean Forecasting Performance (5-Fold CV - Target: `clicks`)

Variant	MAE	RMSE	F1-Score	MASE	Runtime (hrs)	Typical Hyperparameters (Range)
GRU (Perf. Only)	**2.223**	7.122	0.598	0.775	34.59	hidden_dim=16–48, n_rnn_layers=2–3, dropout=0.20–0.40
Realistic Baseline	**2.929**	**9.513**	**0.777**	**0.907**	–	–
GRU (Perf. + Creative)	4.489	9.689	0.439	1.567	34.59	hidden_dim=32–64, n_rnn_layers=2–3, dropout=0.10–0.30

2. Visual Analysis Figure 24 visualizes the distribution of Mean Absolute Error (MAE) across the 5 cross-validation folds. It clearly shows the `GRU (Perf. Only)` model achieving a tight, low error distribution, while the creative-inclusive model demonstrates a much higher median MAE and wider variance.

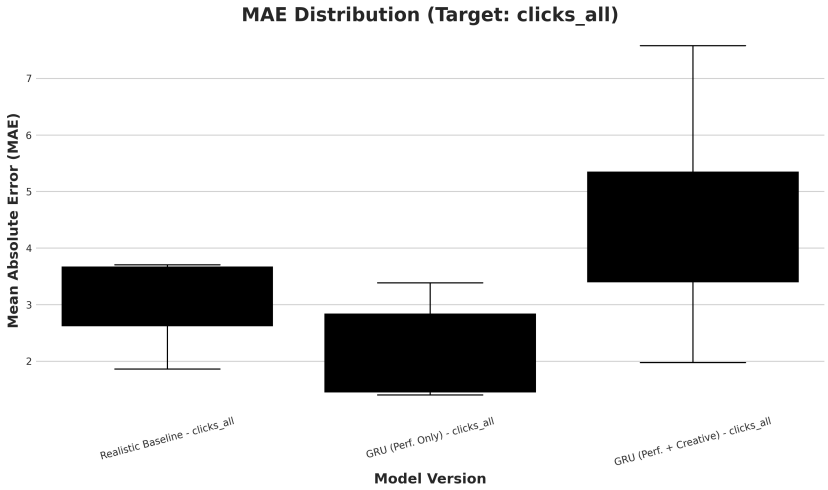


Figure 24: MAE Distribution by Model (Target: `clicks_all`). Results from 5-fold cross-validation.

The scatter plot in Figure 25 shows the GRU predictions (aggregated from all 5 folds) clustered tightly around the diagonal line, confirming strong correlation and low bias when compared to baseline models.

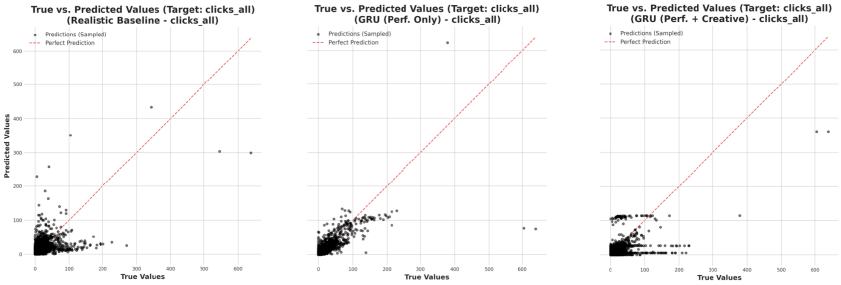
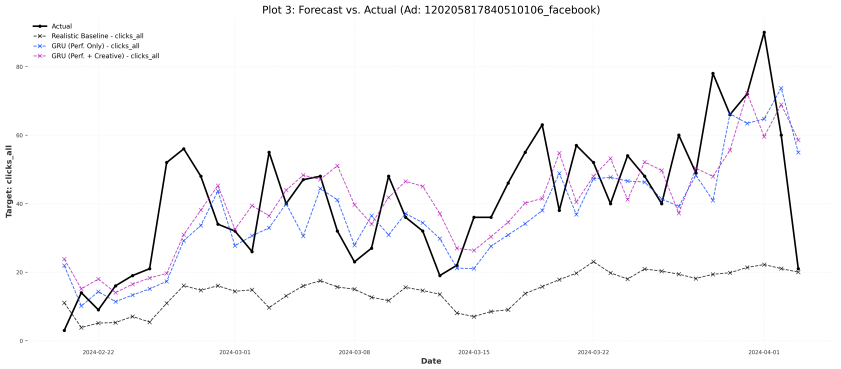
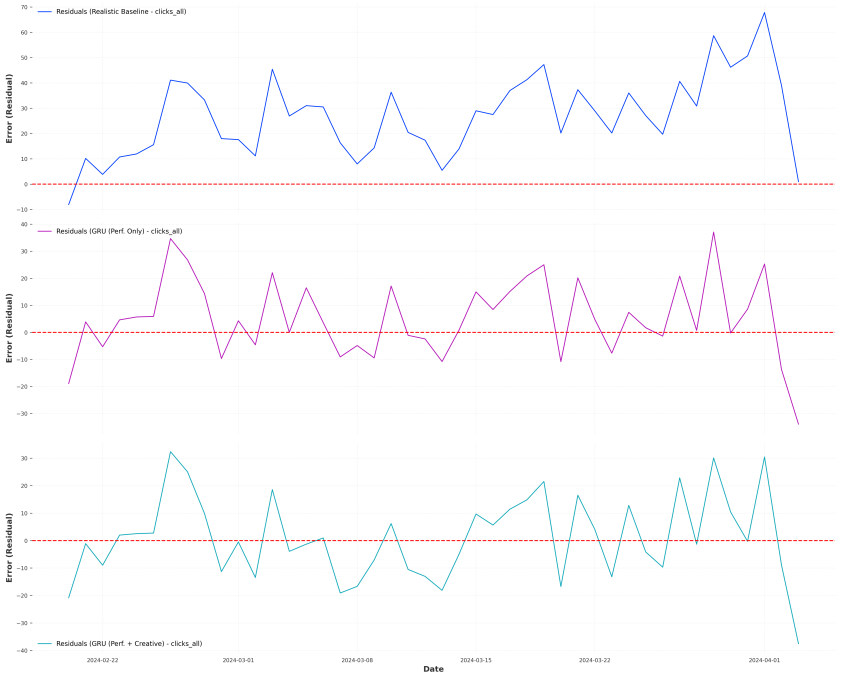


Figure 25: True vs. Predicted Values for `clicks_all` (Aggregated from all 5 folds).

Figures 26a and 26b illustrate the forecast for a single representative ad. The residuals plot (Figure 26b) confirms that the `GRU (Perf. Only)` errors are tightly centered around the zero-error line, indicating low overall bias.



(a) Example Forecast vs. Actual for a single ad.



(b) Corresponding Residuals (Error) over time for the same ad.

Figure 26: Example GRU Time Series Forecast and Residual Plots for `clicks_all`.

3. Feature Importance Analysis (GRU Clicks)

Key Findings

* **Target Volatility Dominance:** In the **Performance Only** model (Figure 27), the top predictors are dominated by rolling standard deviation metrics of the target variable (`clicks_all_roll_std`). This signifies the model's reliance on understanding the historical volatility profile.
* **Reach Reinforcement:** In the **Performance + Creative** model (Figure 28), the importance of `reach` metrics increases, suggesting the static creative context guides the recurrent unit to better interpret upper-funnel momentum.

4.2.6 Model-Specific Analysis: GRU (Leads)

1. Model Configuration and Overall Performance This analysis is based on the Gated Recurrent Unit (GRU) model for the `leads` target, which represents the most sparse and challenging prediction task.

Key Model Parameters The GRU model configuration was consistent with the following parameters:

* **Lookback Period (Input Chunk Length):** 14 days
* **Forecast Horizon (Output Chunk Length):** 1 day
* **Static Feature Count (OHE):** 453 (for Perf. + Creative model)
* **Dynamic Covariate Count:** 27 performance features + 1 target

Quantitative Performance The aggregated metrics highlight the model's efficiency on sparse data and confirm the detrimental effect of adding static complexity to this specific task.

Table 7: GRU Mean Forecasting Performance (5-Fold CV - Target: `leads`)

Variant	MAE	RMSE	F1-Score	MASE	Runtime (hrs)	Typical Hyperparameters (Range)
GRU (Perf. Only)	**0.165**	0.464	0.482	0.421	30.56	hidden_dim=16–64, n_rnn_layers=3, dropout=0.10–0.40
GRU (Perf. + Creative)	0.206	0.534	0.401	0.525	30.56	hidden_dim=32–64, n_rnn_layers=2–3, dropout=0.12–0.34
Realistic Baseline	**0.215**	**0.595**	**0.419**	**0.520**	–	–

2. Visual Analysis Figure 29 visually confirms the table, showing the `GRU (Perf. Only)` model achieving the lowest median MAE with a very narrow error distribution, indicating high consistency.

The scatter plot in Figure 30 shows that while the majority of predictions cluster near zero due to data sparsity, the GRU models successfully capture the rare, non-zero lead spikes far better than the baseline.

Figures 31a and 31b demonstrate the GRU's effectiveness on the single example series. The residuals plot (Figure 31b) shows that the `GRU (Perf. Only)` errors (purple line) are balanced and centered around zero, indicating low bias, while the baseline's error (top, blue line) is consistently high.

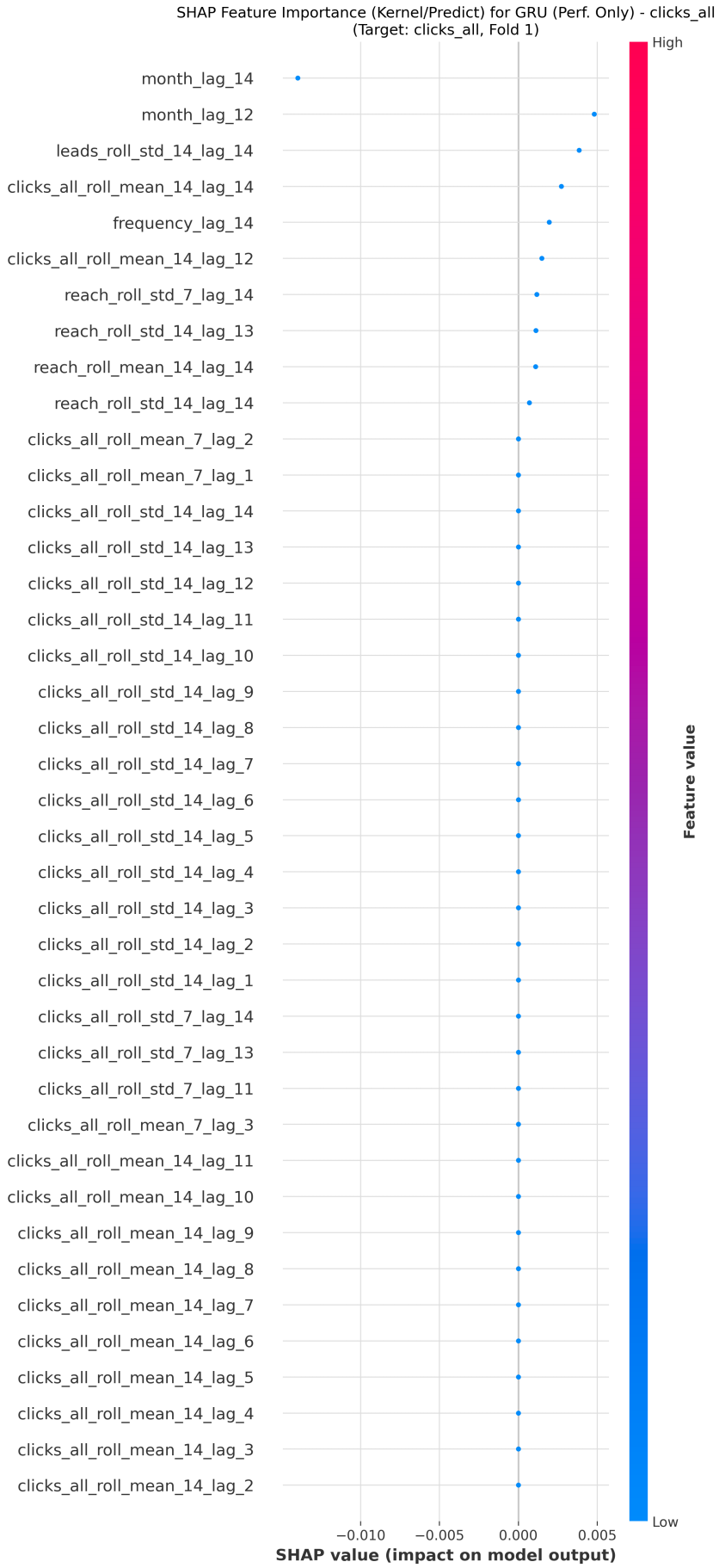


Figure 27: SHAP Feature Importance for GRU (Perf. Only) - `clicks` (Fold 1)

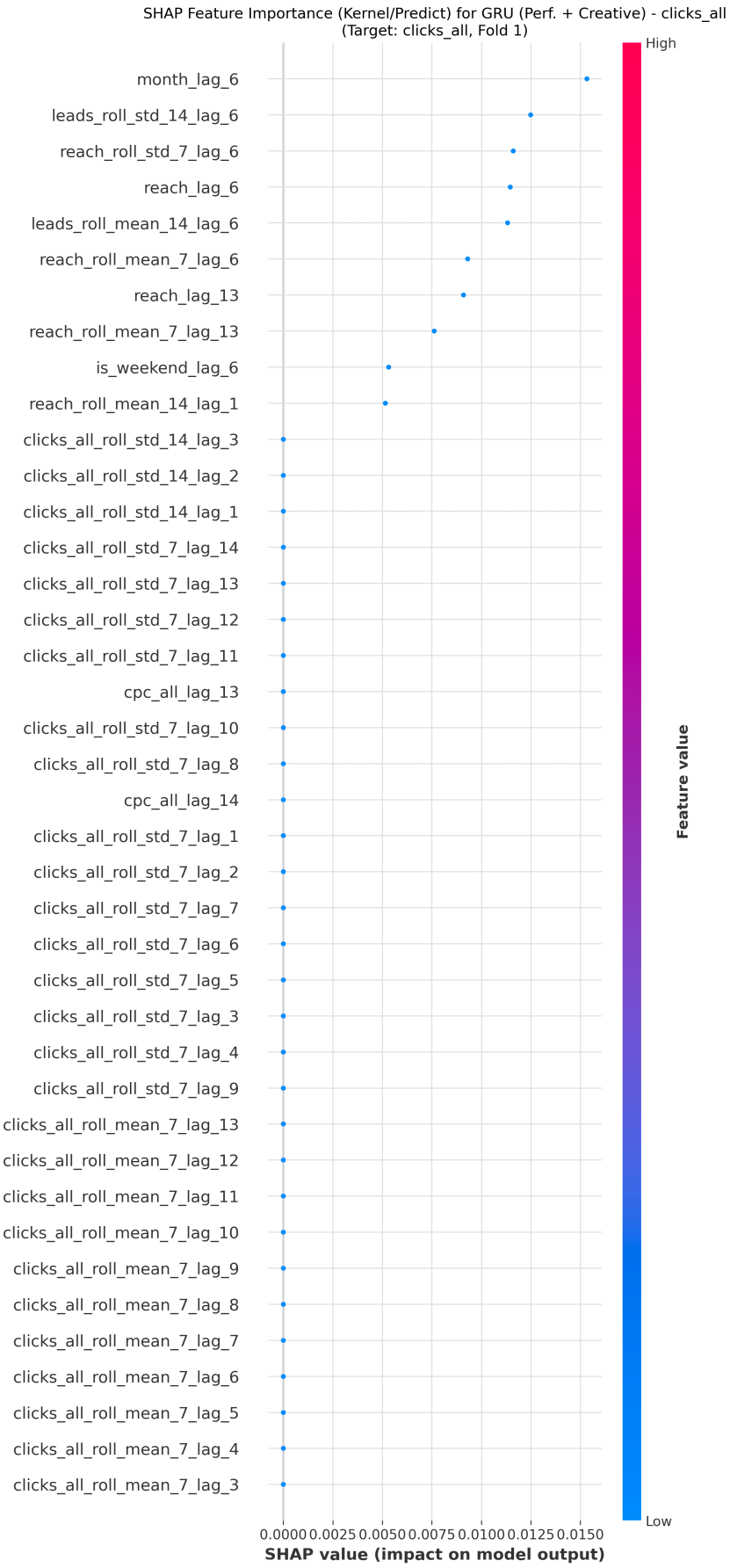


Figure 28: SHAP Feature Importance for GRU (Perf. + Creative) - `clicks` (Fold 1)

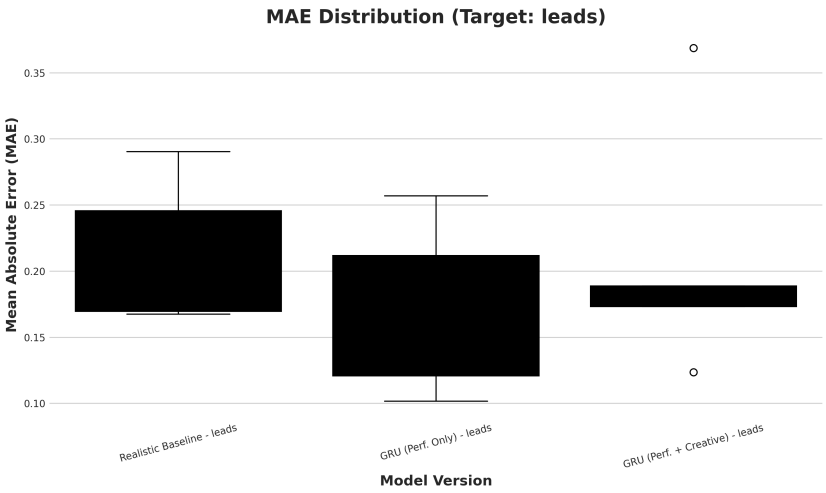


Figure 29: MAE Distribution by Model (Target: `leads`). Results from 5-fold cross-validation.

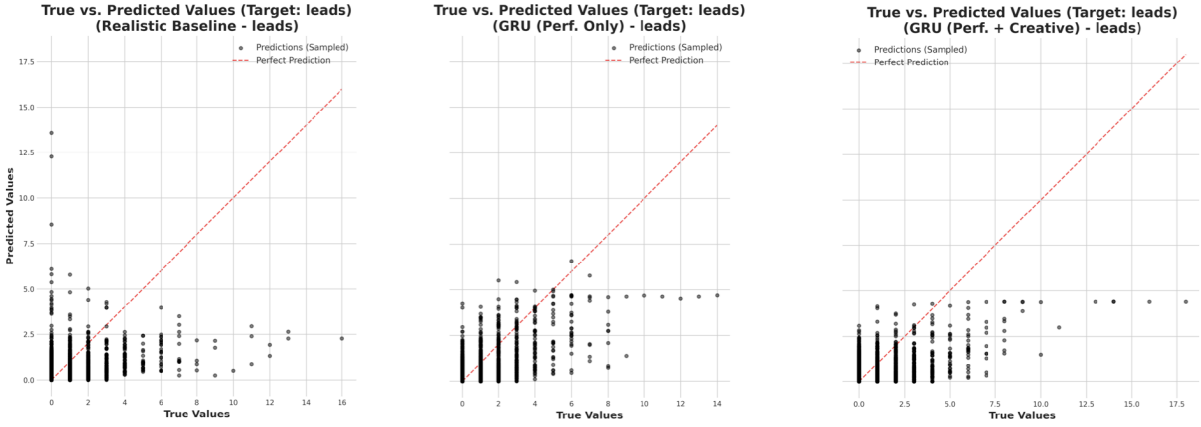
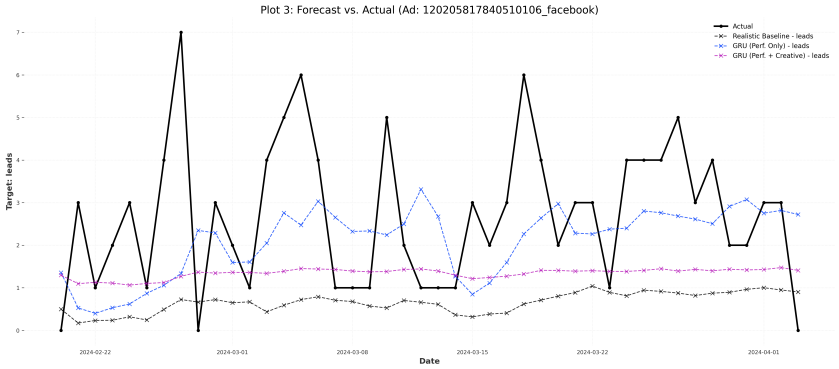
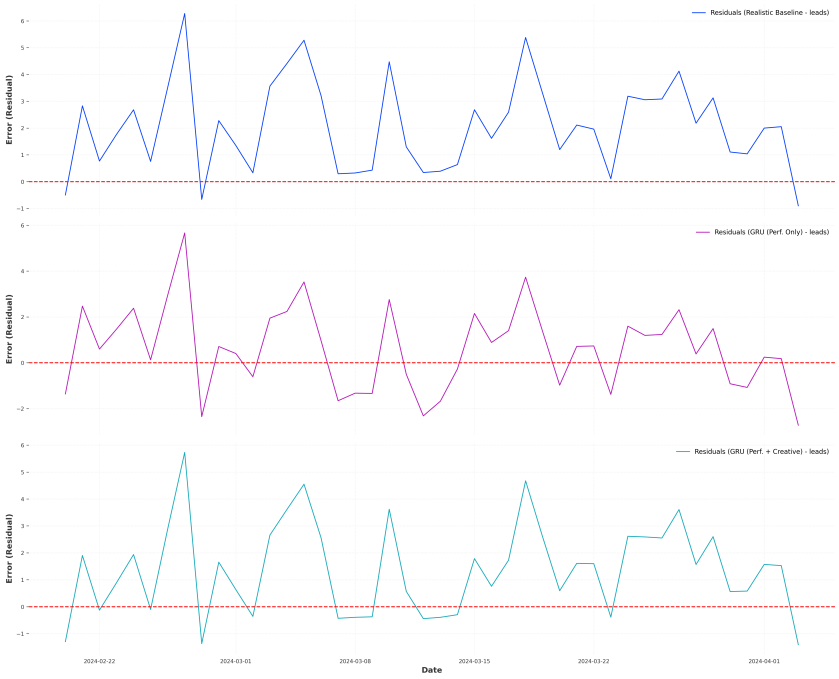


Figure 30: True vs. Predicted Values for `leads` (Aggregated from all 5 folds).



(a) Example Forecast vs. Actual for a single ad.



(b) Corresponding Residuals (Error) over time for the same ad.

Figure 31: Example GRU Time Series Forecast and Residual Plots for `leads`.

3. Feature Importance Analysis (GRU Leads)

Key Findings

* **Shift in Feature Focus:** The SHAP plots reveal a critical shift in the model's logic. The **Performance Only** model (which performed best) relies on a mix of **seasonality** and **auto-regression**. Its most important feature is `month_lag_13`, followed by historical target metrics like `leads_roll_std_14_lag_13` and `leads_roll_mean_7_lag_1`.
* **Creative Features as Noise:** When the 453 static creative features are added, the **Performance + Creative** model's focus shifts entirely *away* from this mix. It abandons the target's history and becomes dominated by upper-funnel `reach` metrics (e.g., `reach_roll_mean_7_lag_12`).
* This shift, combined with the significant drop in model performance (MAE 0.165 $\rightarrow$ 0.206), strongly suggests the static features introduced noise, confusing the model and causing it to abandon the primary auto-regressive and seasonal signals. The static features themselves are not plotted, as the SHAP analysis was methodologically constrained to focus only on the dynamic inputs .

4.2.7 Model-Specific Analysis: XGBoost (Clicks)

This analysis is based on the XGBoost Clicks prediction run. The XGBoost model, being a tree-based ensemble method, uses the **27 time-varying performance features** and the **270 static creative features** directly, without the need for One-Hot Encoding (OHE) on the categorical creative data.

Table 8: XGBoost performance and hyperparameters (**clicks**).

Variant	MAE	RMSE	F1-Score	MASE	Runtime (hrs)	Key Optuna Parameters (Range)
Realistic Baseline	**2.929**	**9.513**	**0.777**	**0.907**	–	–
XGBoost (Perf. + Creative)	3.341	12.002	0.660	1.036	4.23	max_depth=3–4, learning_rate=0.034–0.29, subsample=0.71–0.96, colsample_bytree=0.60–0.91
XGBoost (Perf. Only)	3.346	12.000	0.647	1.036	4.23	max_depth=3–4, learning_rate=0.012–0.26, subsample=0.72–0.91, colsample_bytree=0.61–0.76

Visual Analysis The MAE distribution plot in Figure 34 visually confirms the poor performance reported in Table 2. The error distributions for both XGBoost models are centered higher than the baseline, and the `XGBoost (Perf. Only)` model shows a very wide error variance.

The scatter plots in Figure 35 further illustrate the models' struggles. Compared to the baseline (left), the XGBoost models (two on the right) show significantly more variance and a weaker correlation with the true values, resulting in high RMSE values.

The example time series forecast in Figure 36 (top) and its corresponding residuals (bottom) show that the XGBoost models (purple and teal) are **consistently failing to capture the underlying patterns, mostly predicting zero for every timestep**. This indicates a significant inability to learn the dynamic relationship within the data for this particular ad, resulting in forecasts that do not reflect the actual fluctuations in clicks.

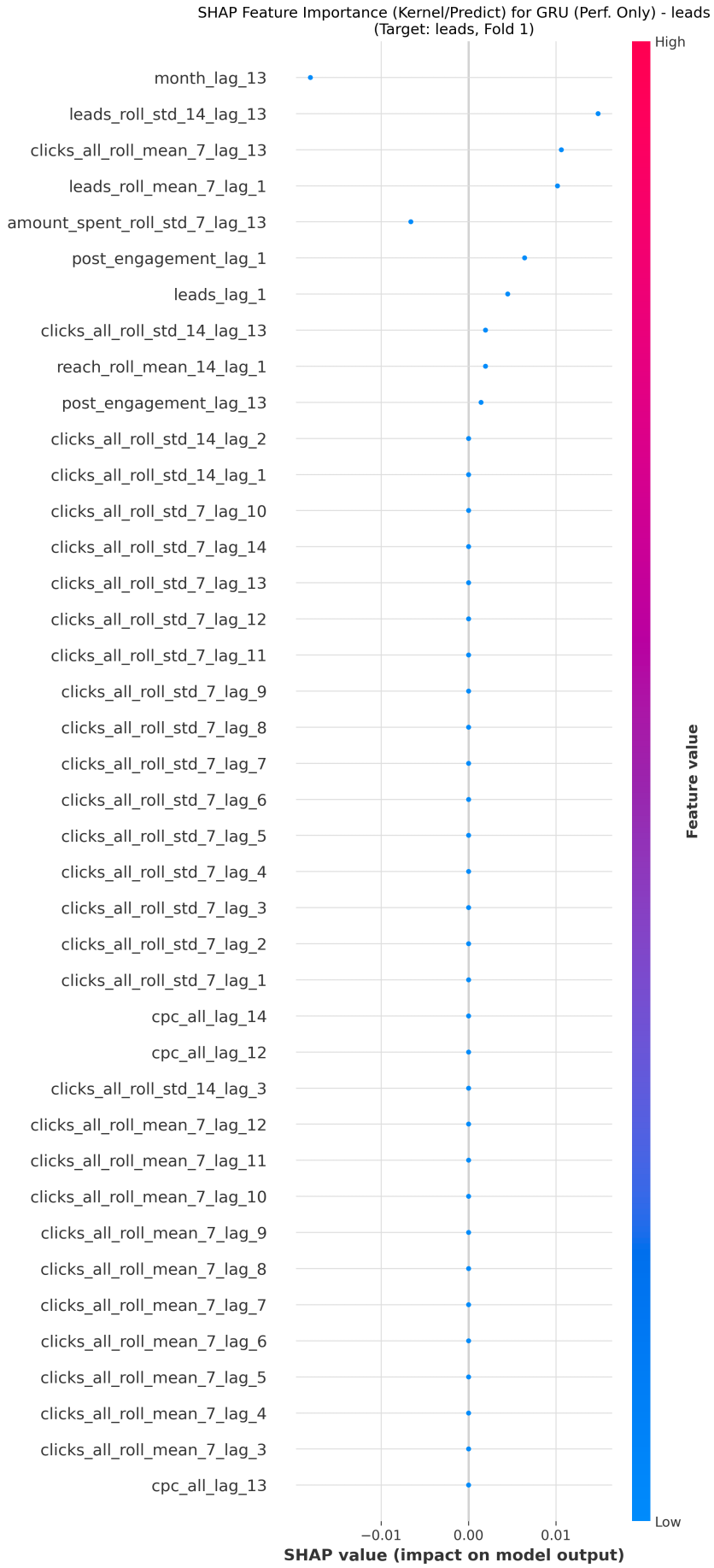


Figure 32: SHAP Feature Importance for GRU (Perf. Only) - `leads` (Fold 1)

Figure 33: SHAP Feature Importance for GRU (Perf. + Creative) - `leads` (Fold 1)

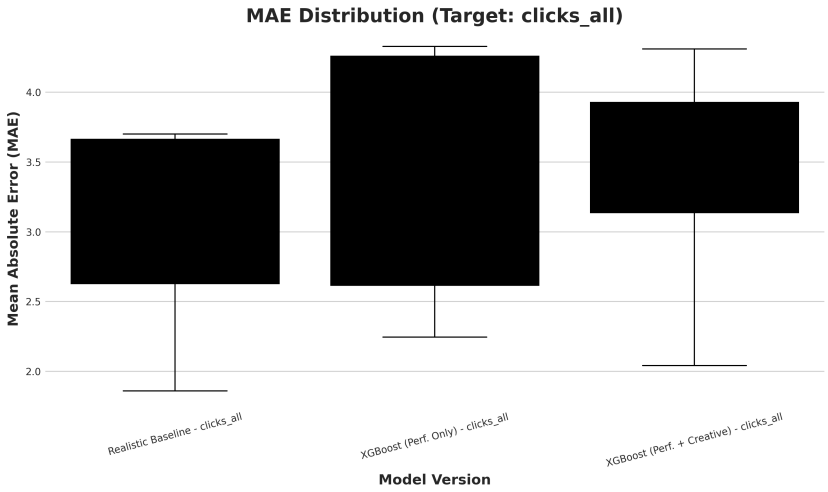


Figure 34: MAE Distribution by Model (Target: `clicks_all`).

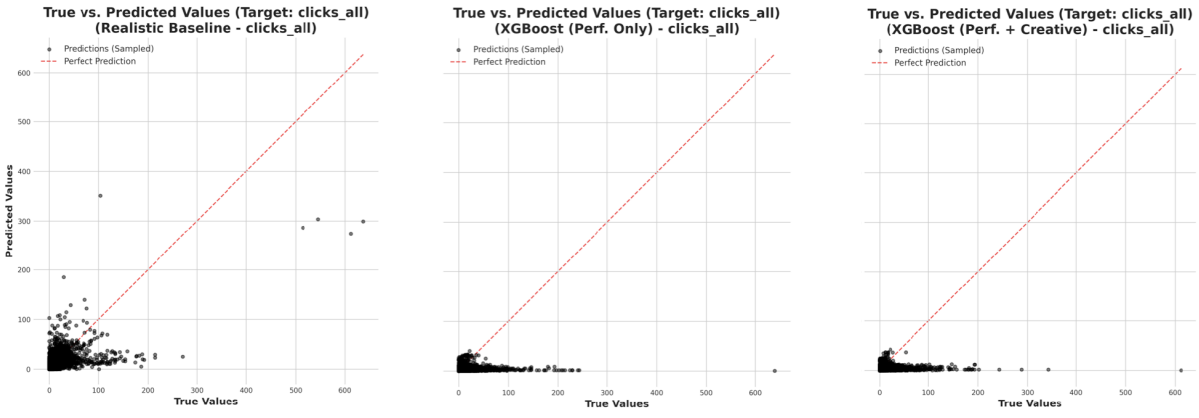
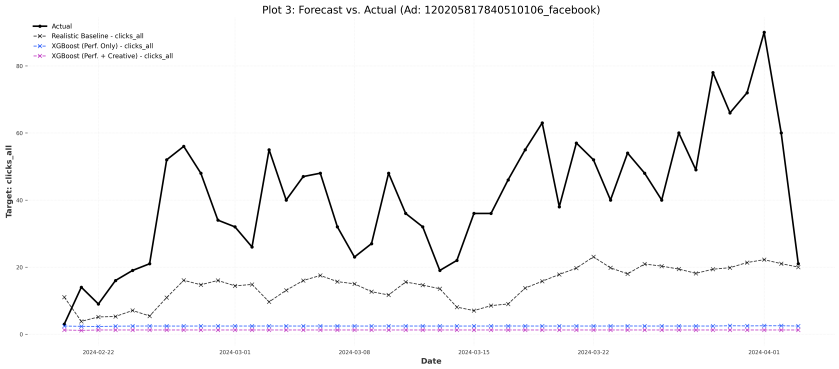
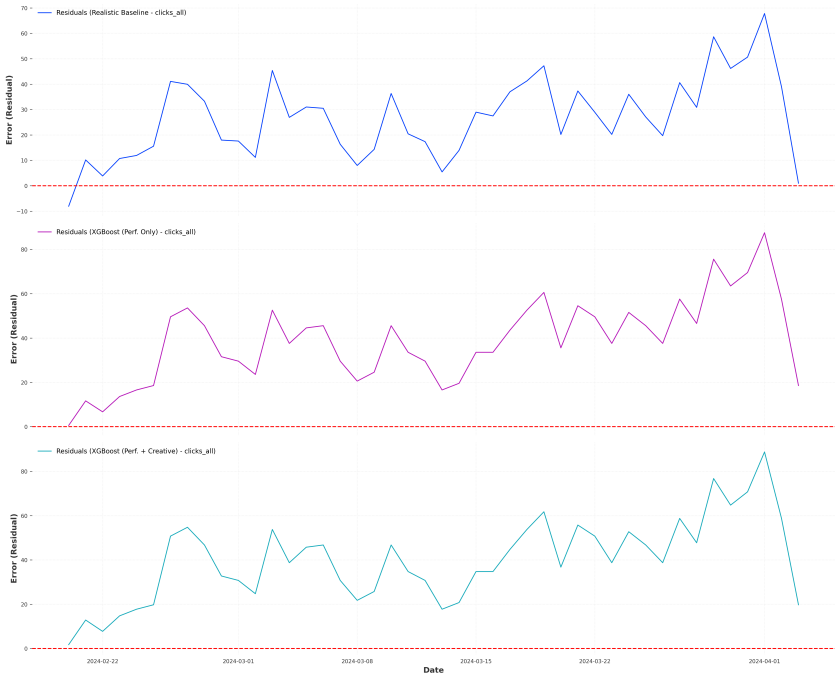


Figure 35: True vs. Predicted Values for `clicks_all` (Aggregated from all 5 folds).



(a) Example Forecast vs. Actual for a single ad.



(b) Corresponding Residuals (Error) over time for the same ad.

Figure 36: Example XGBoost Time Series Forecast and Residual Plots for `clicks_all`.

Feature Importance Analysis (XGBoost) The feature importance plots for XGBoost (Figures 37 and 38) reveal a key difference from the TCN models.

* **Priority of Lag-1 Features:** The XGBoost model's predictions are overwhelmingly dominated by the most recent raw values: `clicks_all_lag1` and `amount_spent_lag1` are the top two features in both models.
* **Creative Feature Signal:** When creative features are added (Figure 38), `num_advantage_cat`—the number of advantages listed in the ad—emerges as a top-tier feature. This confirms that creative features do provide a predictive signal, with XGBoost identifying this measure of "informativeness" as the most important for predicting clicks.

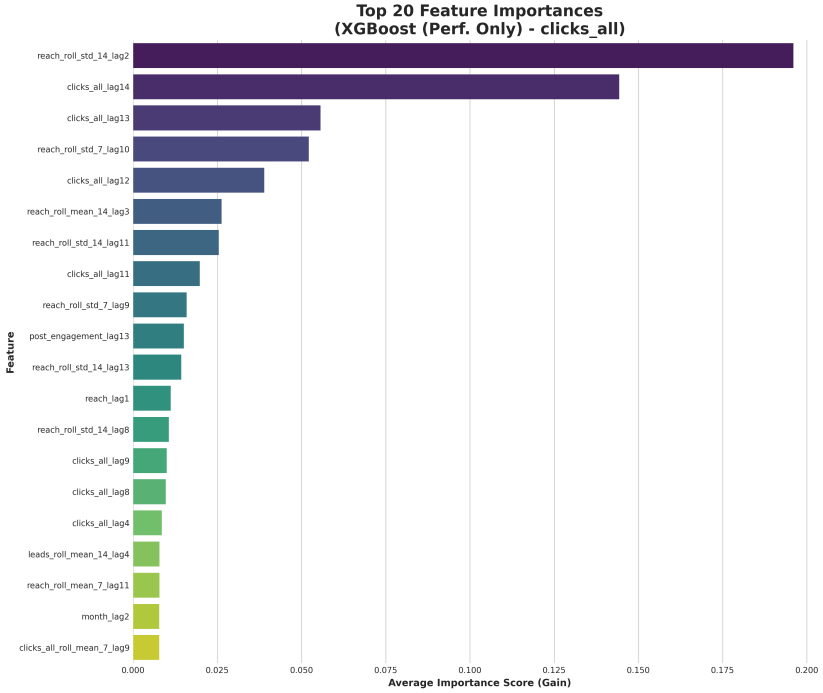


Figure 37: Feature Importance for XGBoost (Perf. Only) - `clicks_all` (Aggregated)

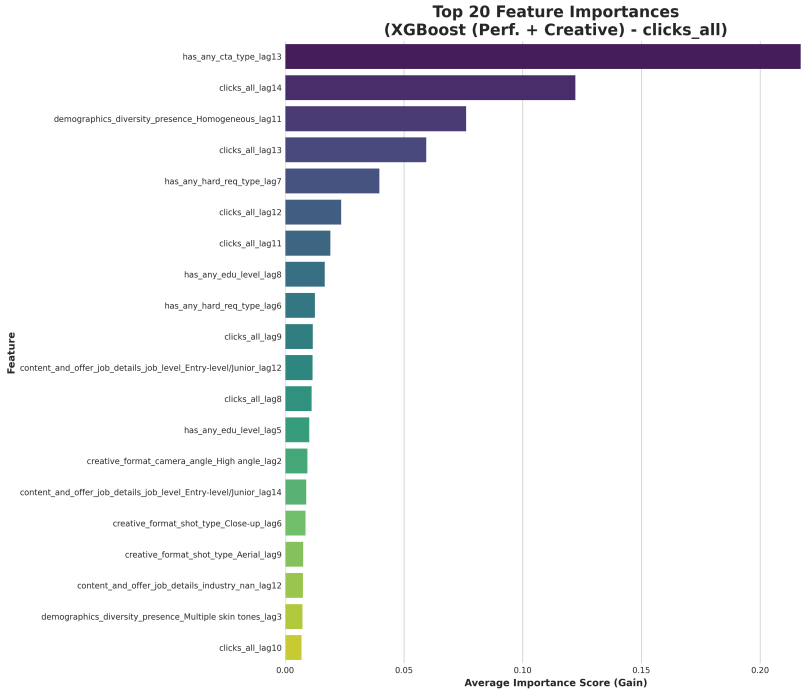


Figure 38: Feature Importance for XGBoost (Perf. + Creative) - `clicks_all` (Aggregated)

4.2.8 Model-Specific Analysis: XGBoost (Leads)

This analysis is based on the XGBoost Leads prediction run. The XGBoost model, being a tree-based ensemble method, uses the **27 time-varying performance features** and the **270 static creative features** directly, without the need for One-Hot Encoding (OHE) on the categorical creative data.

1. Model Configuration and Overall Performance

Key Model Parameters The optimal hyperparameters for the final XGBoost models were determined via Optuna hyperparameter tuning within each fold, leading to the following range and most frequent parameters:

Table 9: XGBoost performance and hyperparameters (**leads**).

Variant	MAE	RMSE	F1-Score	MASE	Runtime (hrs)	Key Optuna Parameters (Range)
XGBoost (Perf. Only)	0.216	0.621	0.025	0.548	3.13	max_depth=3–5, learning_rate=0.037–0.30, subsample=0.61–0.91, colsample_bytree=0.74–0.88
XGBoost (Perf. + Creative)	0.216	0.623	0.033	0.549	3.13	max_depth=3–7, learning_rate=0.14–0.30, subsample=0.72–0.99, colsample_bytree=0.77–1.00
Realistic Baseline	**0.223**	**0.604**	**0.437**	**0.563**	–	–

Quantitative Performance The aggregated metrics show that the XGBoost models achieved only a marginal **3.3**% improvement over the baseline, confirming their struggle with this sparse and intermittent target.

2. Visual Analysis Figure 39 visualizes the Mean Absolute Error (MAE) distribution across the 5 cross-validation folds. It shows that both XGBoost models have error distributions clustered slightly above the baseline, confirming their failure to significantly outperform the simplest prediction method for this sparse, volatile target.

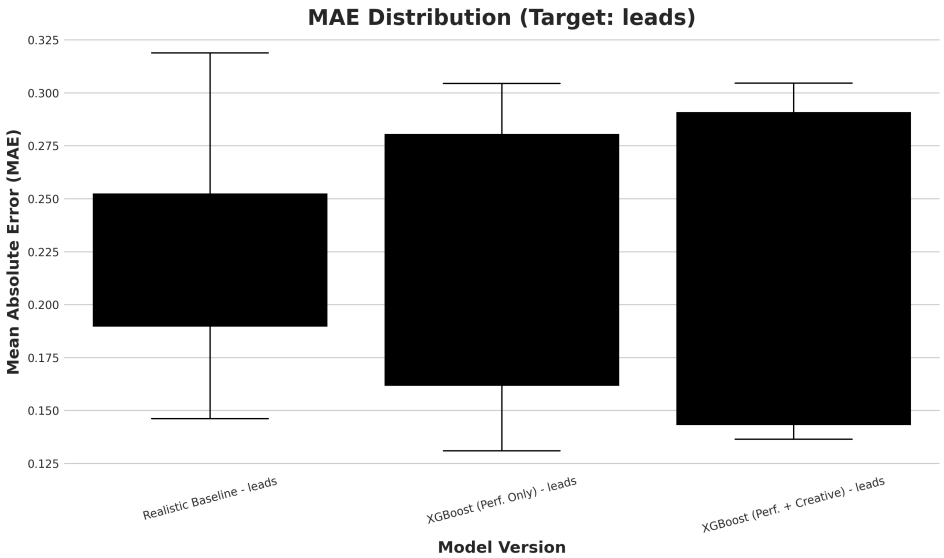


Figure 39: MAE Distribution by Model (Target: `leads`). Results from 5-fold cross-validation.

The scatter plots in Figure 40 compare the predicted values (y-axis) against the true values (x-axis) for all test predictions aggregated from the 5 folds. The models

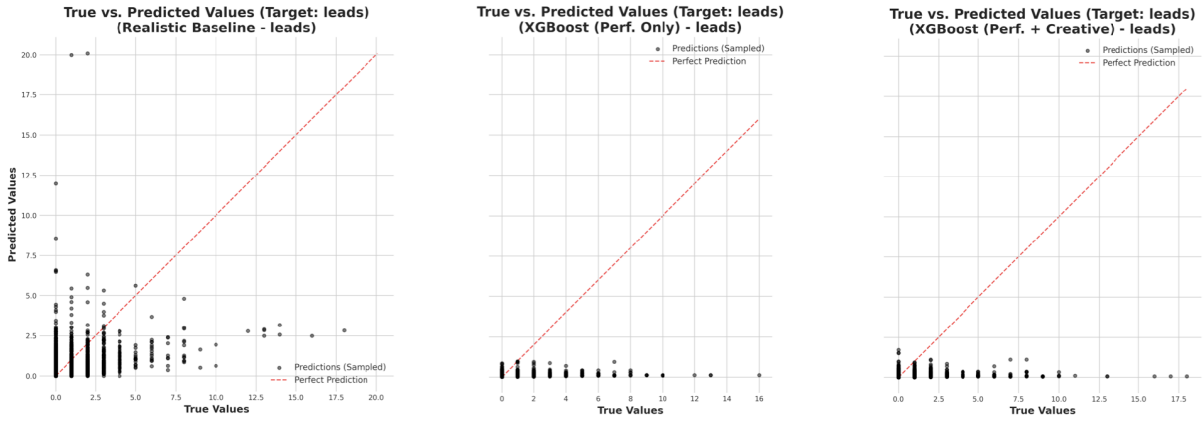
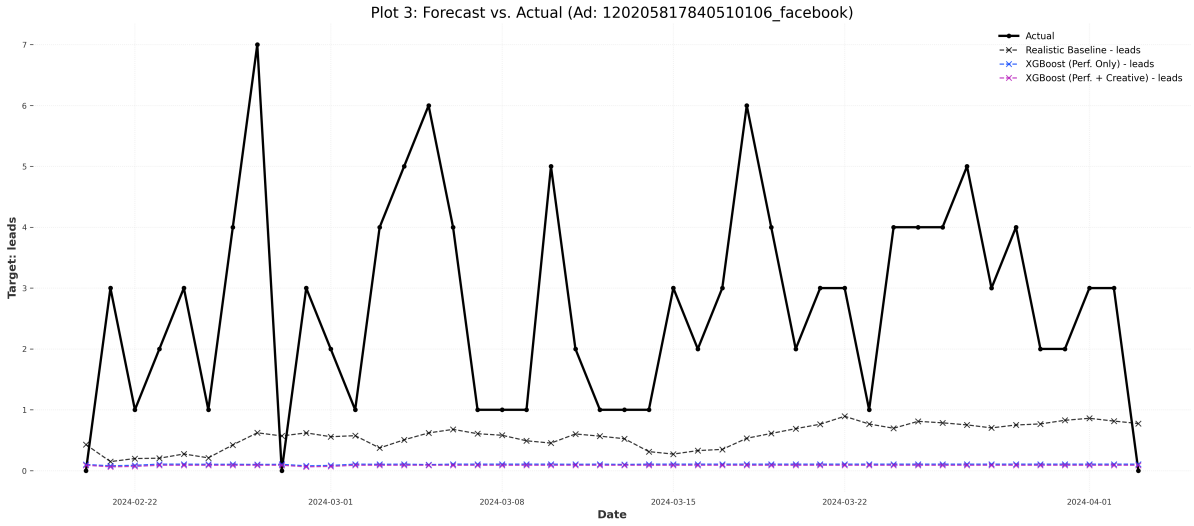


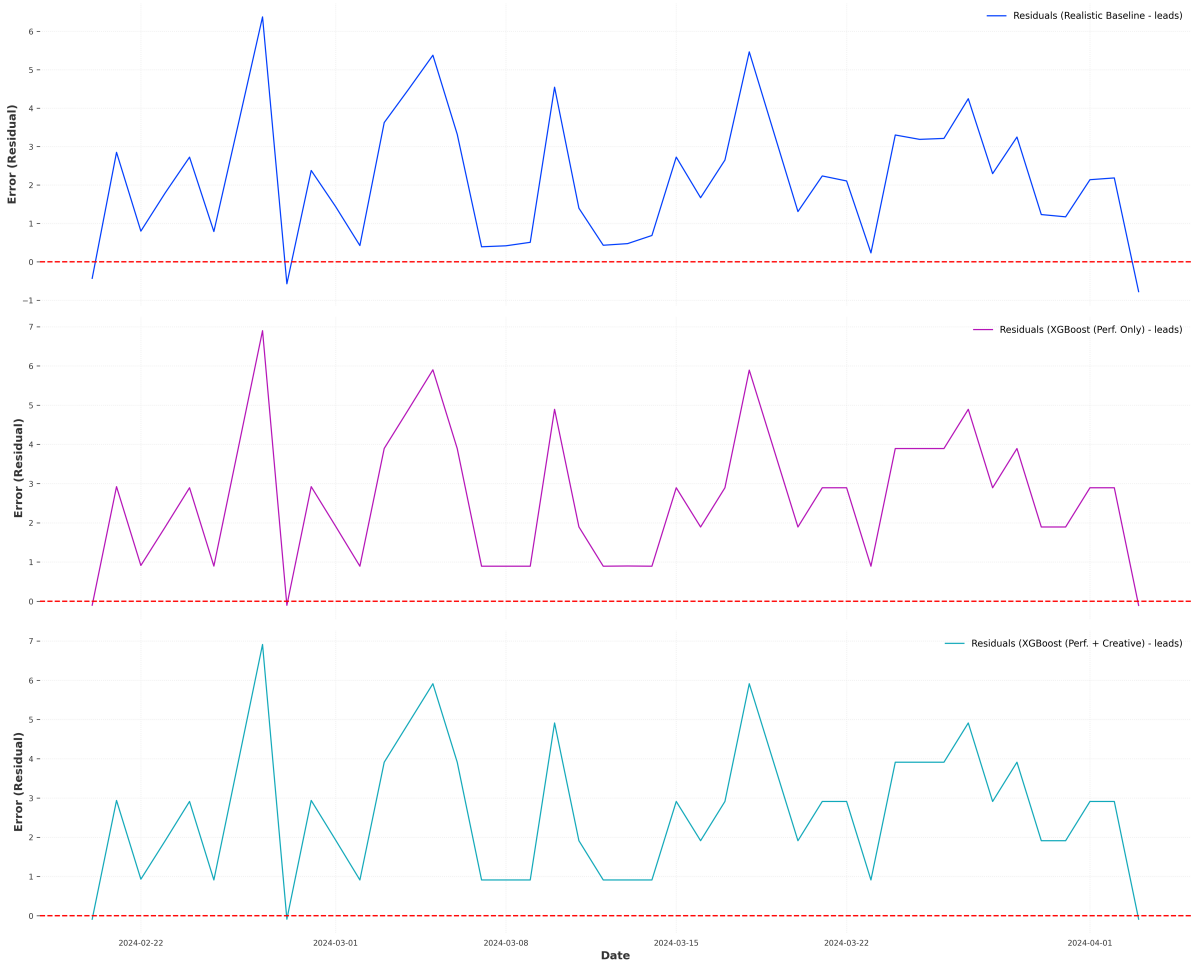
Figure 40: True vs. Predicted Values for `leads` (Aggregated from all 5 folds).

show a very high concentration of predictions clustered around zero, indicating that XGBoost struggled to detect and forecast the infrequent, non-zero lead spikes.

Figures 41a and 41b illustrate the forecast for a single representative ad. The XGBoost forecasts (Figure 41a) appear as flat lines near zero for most timesteps, leading to high residuals (Figure 41b) that closely track the actual lead spikes. This behavior is characteristic of models that fail to capture the underlying causal dynamics of a time series, instead treating the target as near-zero noise.



(a) Example Forecast vs. Actual for a single ad.



(b) Corresponding Residuals (Error) over time for the same ad.

Figure 41: Example XGBoost Time Series Forecast and Residual Plots for `leads`.

3. Feature Importance Analysis

Key Findings The feature importance analysis confirms that the XGBoost model for leads prioritizes the most recent values of the target and related high-level metrics.

* **Target Dominance:** In the **Performance Only** model (Figure 42), the predictions are dominated by `leads_lag_1` (the leads metric from the previous day), followed by volatility statistics like `leads_roll_std_7`. The model primarily extrapolates based on the most immediate preceding value.
* **Creative Signal vs. Dynamic Feature:** The **Leads** target does not show a dominant creative feature breaking into the top tier. The model relies almost exclusively on the short-term performance history, suggesting that creative features provided minimal signal gain for this highly sparse, low-funnel target.

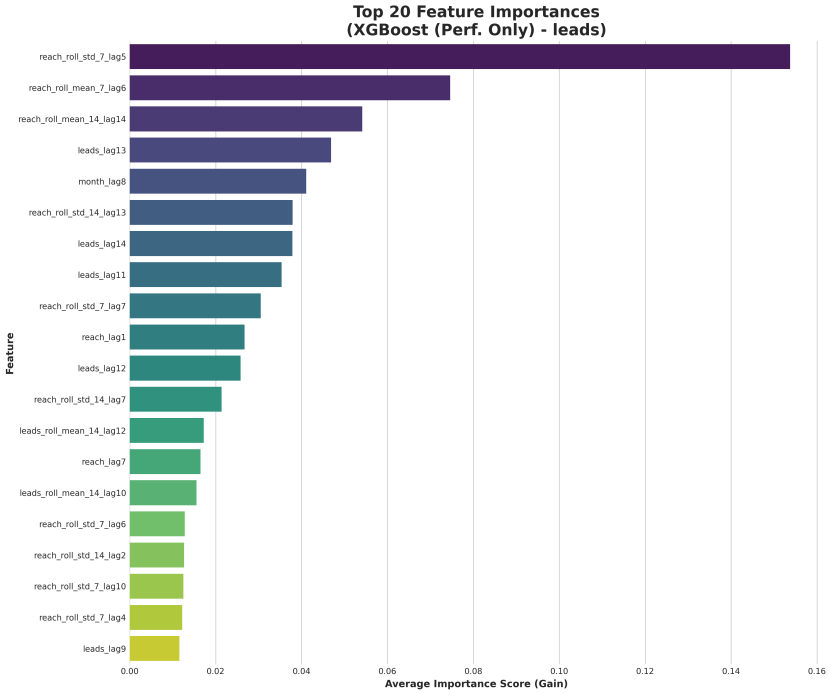


Figure 42: Feature Importance for XGBoost (Perf. Only) - `leads` (Aggregated)

4.2.9 Model-Specific Analysis: SARIMAX (Clicks)

This analysis is based on the SARIMAX clicks prediction run, where a separate model was fit for each time series using exogenous features (X).

1. Model Configuration and Overall Performance

Key Model Parameters The SARIMAX model utilized the following setup for its auto-fitting process across the cross-validation folds:

* **Transformation:** Square Root (Applied to stabilize variance and handle zero values)

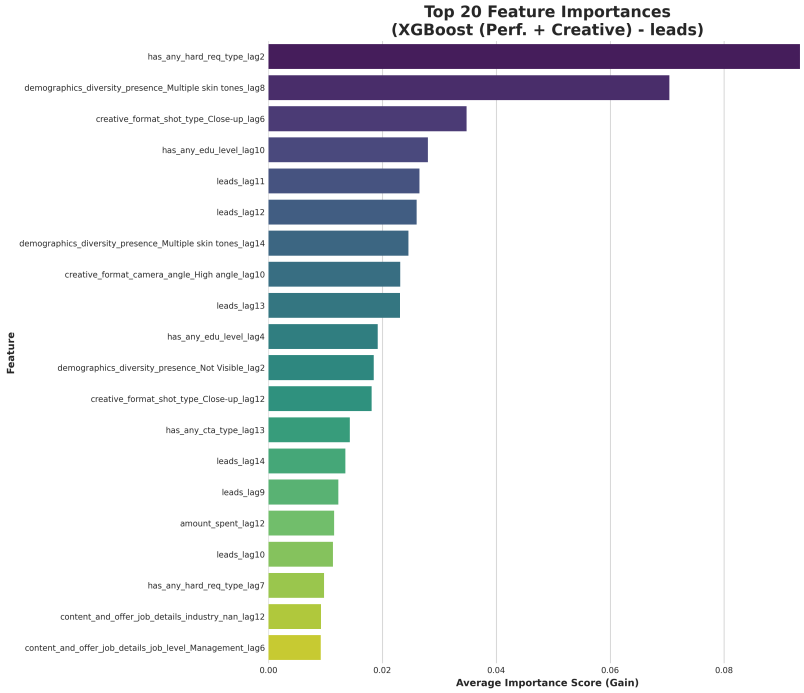


Figure 43: Feature Importance for XGBoost (Perf. + Creative) - `leads` (Aggregated)

* **Exogenous Features (X):** 3 ('amount_spent_lag_1', 'clicks_lag_1, 'is_weekend')
* **Dominant Orders:** $(\mathbf{0}, \mathbf{0}, \mathbf{0})$ was the most frequent order, found in $\sim$ **66**% of the series.

Table 10: SARIMAX performance and parameters (**clicks**).

Variant	MAE	RMSE	F1-Score	MASE	Runtime (hrs)	Dominant (p,d,q) Orders
Realistic Baseline	**2.929**	**9.513**	**0.777**	**0.907**	–	–
SARIMAX (Simple)	3.248	13.994	0.810	0.953	1.42	(0,0,0) ∼66%, (1,0,0) ∼14%, (0,0,1) ∼9%

Quantitative Performance The aggregated metrics for SARIMAX place it as a middle-performer, but it **failed** to beat the `Realistic Baseline` (MAE 2.929), performing **10.9**% worse (MAE 3.248).

2. Visual Analysis The MAE distribution plot in Figure 44 shows that the SARIMAX model's error distribution (right) is more volatile (wider range) than the `Realistic Baseline` (left).

The scatter plots in Figure 45 compare the predicted values (y-axis) against the true values (x-axis). Both the `Realistic Baseline` (left) and the `SARIMAX` model (right) show a heavy cluster of predictions near zero, struggling to capture the high-variance, high-click days. The SARIMAX plot shows a slightly wider spread but still fails to align closely with the perfect prediction line, confirming its mediocre performance.

The example forecast (Figure 46a) and its residuals (Figure 46b) illustrate that while the SARIMAX forecast (blue) tracks the actual data more dynamically than the flat-lining baseline, its residuals show more frequent over- and under-corrections.

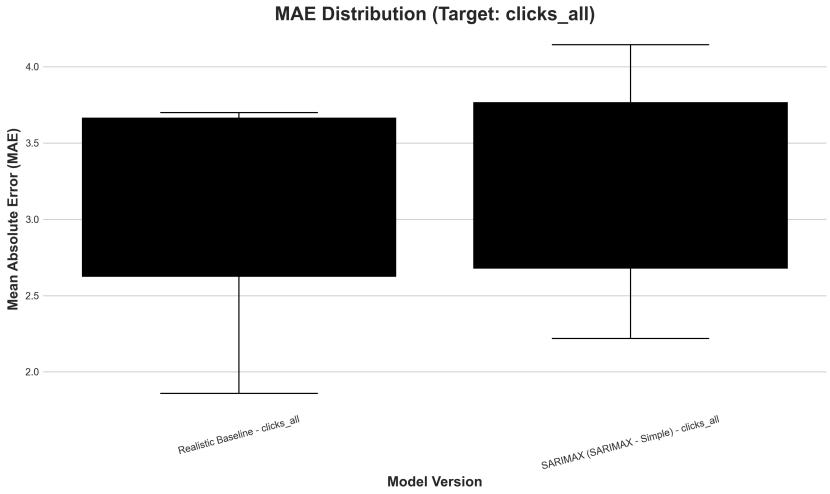


Figure 44: MAE Distribution by Model (Target: `clicks_all`). Baseline (left), SARIMAX (right).

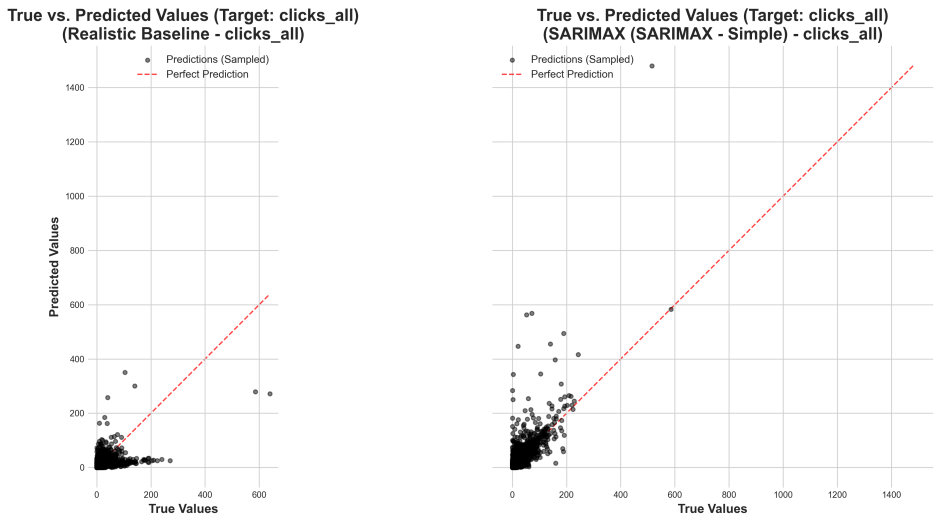
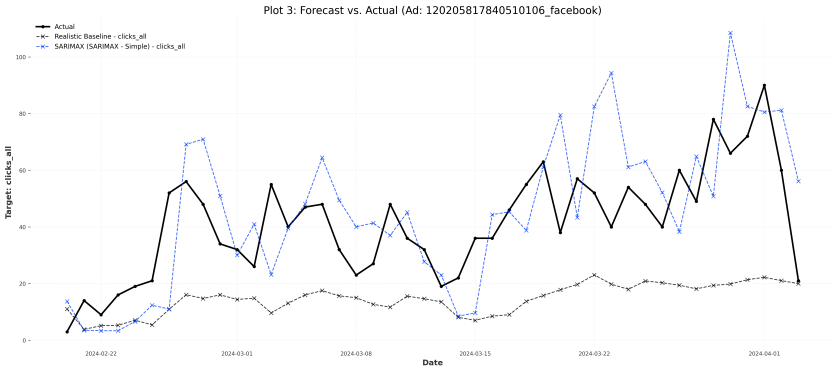
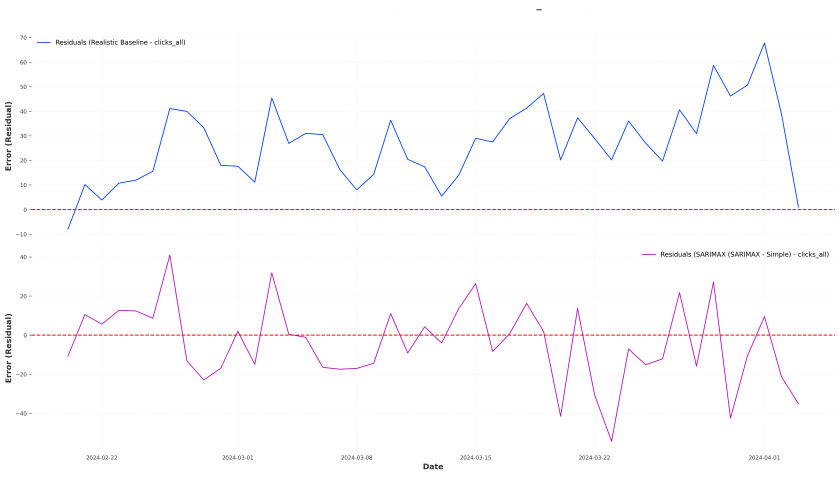


Figure 45: True vs. Predicted Values for `clicks_all` (Aggregated from all 5 folds). Baseline (left), SARIMAX (right).



(a) Example Forecast vs. Actual for a single ad.



(b) Corresponding Residuals (Error) over time for the same ad.

Figure 46: Example SARIMAX Time Series Forecast and Residual Plots for `clicks_all`.

4.2.10 Model-Specific Analysis: SARIMAX (Leads)

This analysis is based on the SARIMAX leads prediction run, which was explicitly filtered for leads objective campaigns. As noted in the overall comparison, this model **failed to outperform its baseline** on this sparse target.

1. Model Configuration and Overall Performance

Key Model Parameters The Auto-SARIMAX process optimized model orders for each time series, resulting in the following structure:

* **Transformation:** Square Root (Applied to stabilize variance)
* **Exogenous Features (X):** 3 ('amount_spent_lag_1', 'leads_lag_1', 'is_weekend')
* **Most Frequent Order:** $(\mathbf{0}, \mathbf{0}, \mathbf{0})$ (Found in **89%** of series), confirming that for the vast majority of sparse lead series, no meaningful auto-regressive (AR) or moving-average (MA) signal could be found.

Table 11: SARIMAX performance and parameters (**leads**).

Variant	MAE	RMSE	F1-Score	MASE	Runtime (hrs)	Dominant (p,d,q) Orders
Realistic Baseline	**0.213**	**0.585**	**0.435**	**0.536**	–	–
SARIMAX (Simple)	0.306	1.693	0.518	0.717	1.24	(0,0,0) ∼89%, (0,0,1) ∼4%, (1,0,0) ∼4%

Quantitative Performance The aggregated metrics confirm the quantitative failure of SARIMAX on this task, as its MAE was worse than the Realistic Baseline by **43.4**%.

2. Visual Analysis The visual analysis confirms the quantitative failure. The MAE distribution plot (Figure 47) shows the SARIMAX model's error (right) is noticeably higher and more volatile than the baseline's (left).

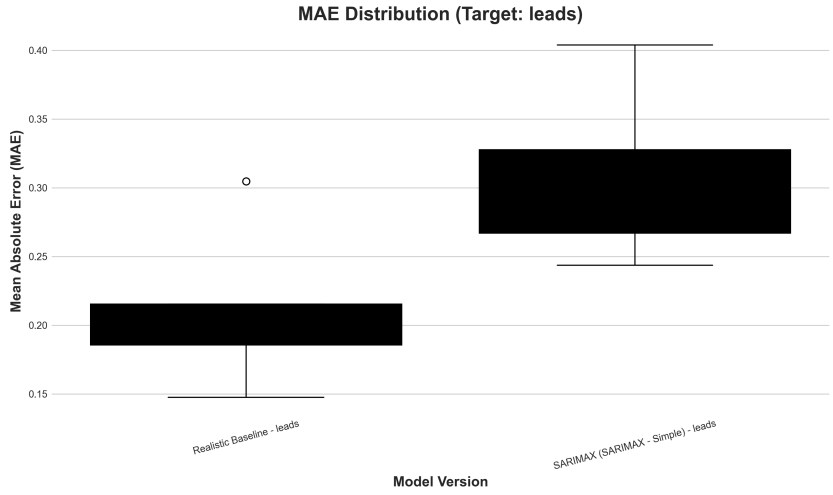


Figure 47: MAE Distribution by Model (Target: `leads`). Baseline (left), SARIMAX (right).

The scatter plot in Figure 48 illustrates the extreme difficulty of this task. For both the baseline and the SARIMAX model, the vast majority of true and predicted values are clustered at zero.

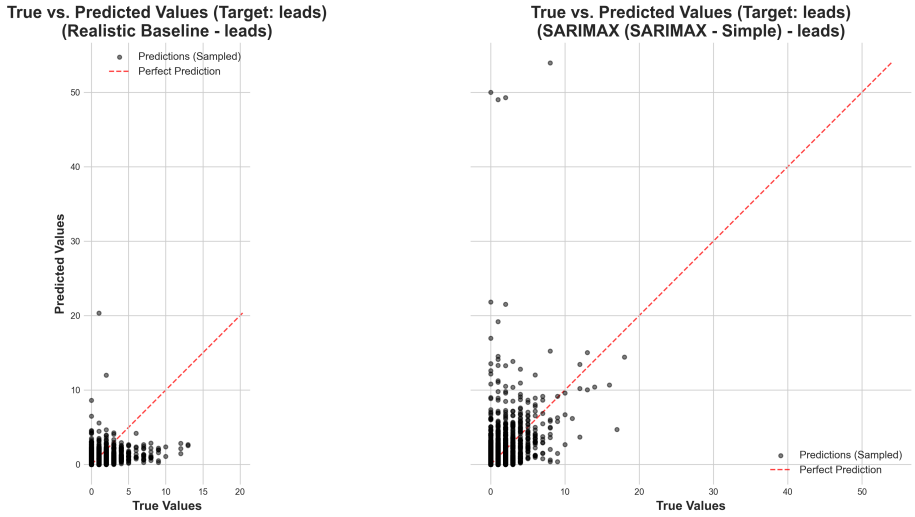


Figure 48: True vs. Predicted Values for `leads` (Aggregated from all 5 folds). Baseline (left), SARIMAX (right).

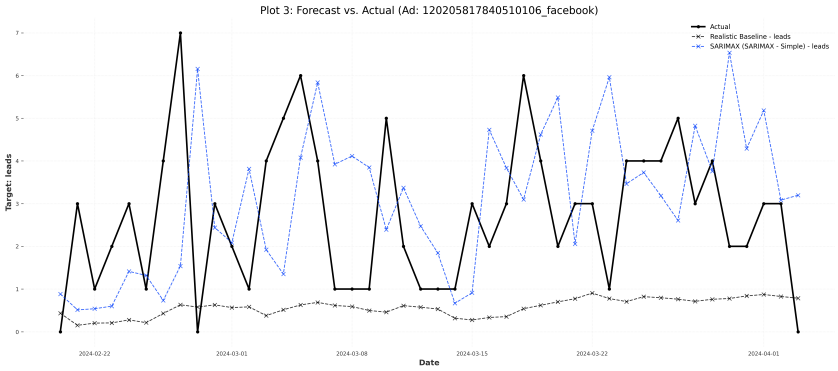
The example forecast in Figure 49a (top) shows both the SARIMAX (blue) and baseline (black) forecasts as a flat line near zero, completely failing to capture any of the ad's actual lead spikes. The corresponding residuals plot (Figure 49b, bottom) shows that the error for both models is nearly identical to the actual data.

4.3 Statistical Analysis of Model Performance

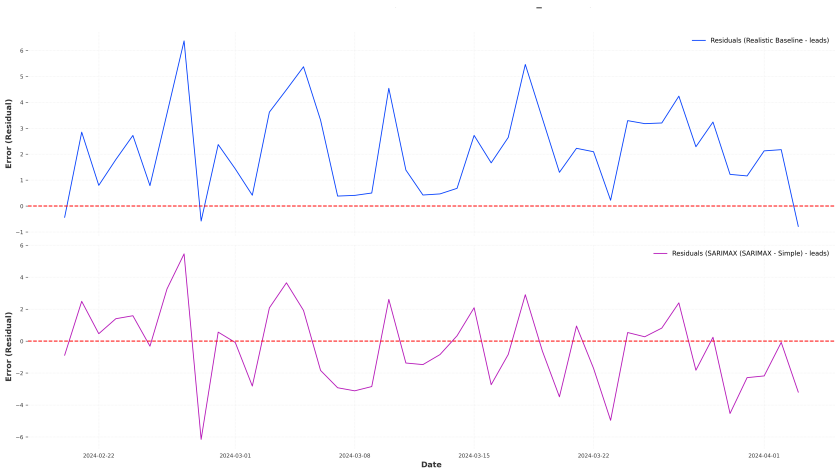
To assess whether differences in predictive performance between models were statistically significant, a *Wilcoxon signed-rank test* [20] was conducted using a significance level of $\alpha = 0.05$. The test was applied to the mean absolute error (MAE) obtained per fold during cross-validation.

4.3.1 Summary of Findings

Table 12 summarizes the results for each target variable, comparing all models against the realistic baseline, evaluating the impact of including creative features ('Perf. Only' vs. 'Perf. + Creative'), and comparing models against the best-performing variant.



(a) Example Forecast vs. Actual for a single ad.



(b) Corresponding Residuals (Error) over time for the same ad.

Figure 49: Example SARIMAX Time Series Forecast and Residual Plots for `leads`.

Table 12: Wilcoxon signed-rank test results (p-values) for model comparisons. NS indicates *not significant* at $\alpha = 0.05$.

Comparison	**Target: clicks_all**	**Target: leads**	**Significance**
A. Model vs. Realistic Baseline			
GRU (Perf. Only) vs. Baseline	0.0625	0.1875	NS
GRU (Perf. + Creative) vs. Baseline	0.1875	0.6250	NS
TCN (Perf. Only) vs. Baseline	0.0625	0.3125	NS
TCN (Perf. + Creative) vs. Baseline	0.0625	0.4375	NS
XGBoost (Perf. Only) vs. Baseline	0.4375	0.8125	NS
XGBoost (Perf. + Creative) vs. Baseline	0.3125	0.8125	NS
SARIMAX vs. Baseline	0.6250	0.0625	NS
B. Perf. Only vs. Perf. + Creative (Creative Impact)			
XGBoost (P+C) vs. (P)	1.0000	0.6250	NS
TCN (P+C) vs. (P)	0.1250	0.6250	NS
GRU (P+C) vs. (P)	0.0625	0.8125	NS
C. Model vs. Best Model			
GRU (P) vs. Best	0.0625	0.8125	NS
GRU (P+C) vs. Best	0.0625	0.8125	NS
TCN (P) vs. Best	0.1250	0.4375	NS
TCN (P+C) vs. Best	0.0625	0.1875	NS
XGBoost (P) vs. Best	0.0625	0.1875	NS
XGBoost (P+C) vs. Best	0.0625	0.3125	NS
SARIMAX vs. Best	0.0625	0.0625	NS

4.4 Interpretation of Model Performance

The results indicate that, across all model comparisons at the stringent $\alpha = 0.05$ level, no statistically significant differences were found. This uniform outcome is strongly indicative of a lack of statistical power due to the small sample size ($N = 5$ folds) used for the Wilcoxon test.

While the differences were not statistically verifiable, the models consistently demonstrated strong **practical significance** over the baseline (Table 2 and 3). For instance, the TCN (Perf. + Creative) model reduced the mean MAE for `clicks_all` by **45.5%** (1.597 vs. 2.929), and the GRU (Perf. Only) model reduced the mean MAE for `leads` by **23.3%** (0.165 vs. 0.223).

Despite the high practical gains, the statistical tests confirm:

* The median difference in MAE between the best-performing models and the Realistic Baseline is not significant, preventing a formal declaration of superiority.
* Adding creative features (`Perf. + Creative`) did not lead to a statistically significant change in performance over `Perf. Only` models for any architecture (SQ2).

Implications: The strong practical trends suggest that GRU and TCN models offer substantial performance benefits. However, to confirm these findings and formally reject the null hypothesis, future work must increase the number of folds or use repeated cross-validation to augment the statistical power.

4.5 Creative Feature Impact (SQ3)

To address the third sub-question (SQ3), a dedicated analysis was performed to identify which static creative features significantly impact overall lifetime efficiency metrics, as detailed in Section 3.10. This analysis combined permutation importance to rank features, ANOVA to test for statistical significance, and OLS regression to identify significant interaction effects.

4.5.1 Relative Feature Importance

The analysis began by identifying the most influential features for predicting the lifetime Funnel Conversion Rate (FCR), using permutation importance on an XGBoost model. Figure 50 shows the top 20 most important features for the "Total Ads" dataset.

The results are striking. Three features are overwhelmingly more important than all others:

1. `platform_audience_network`: The strategic choice of ad placement is the single most important predictor.
2. `num_advantage_cat`: The number of advantages listed in the ad (a measure of informativeness).
3. `brand_and_emotion_emotional_tone_Motivational`: The creative's emotional tone.

After these top three, there is a large drop-off in importance. This suggests that for converting users from reach-to-lead, *where* the ad is placed, *what* it promises, and its *emotional tone* are the dominant predictive factors.

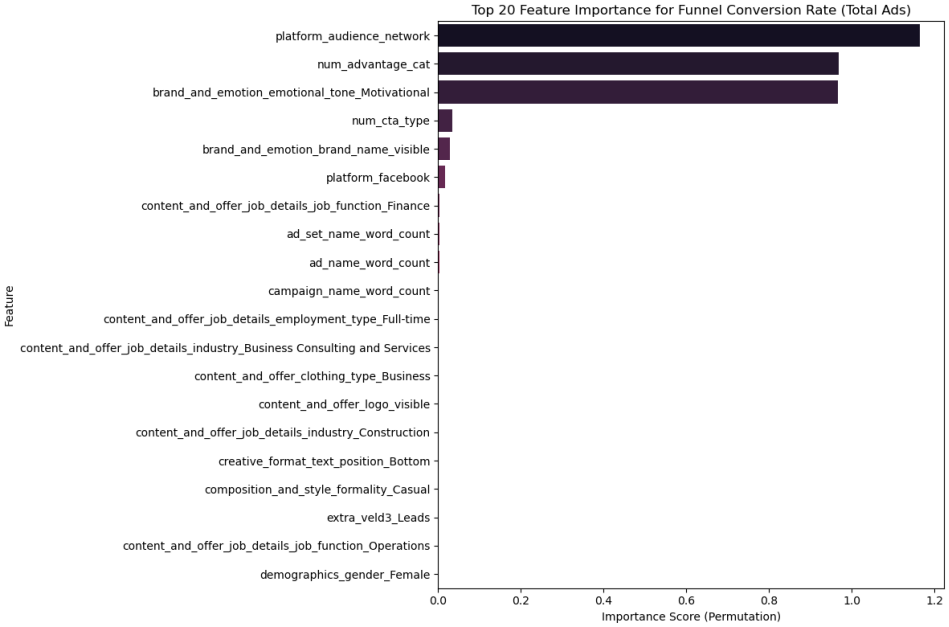


Figure 50: Top 20 Feature Importance for Funnel Conversion Rate (Total Ads) using Permutation Importance.

4.5.2 Interaction Effects

The OLS regression analysis revealed that while the vast majority of feature pairs did not have a significant interaction, a powerful cluster of highly significant interactions exists. These findings suggest that exceptional performance is often not driven by single features but by the synergistic combination of specific strategic, content, and creative choices.

Synergistic Interactions A clear pattern of synergy was found between strategic choices and specific content. As shown in Figure 51, a strong synergistic interaction (p=0.0015) was found between ads for 'Full-Time' employment and the 'Construction' industry. While 'Full-Time' ads for other industries show a low FCR, their performance is amplified largely when targeting 'Construction'.

Antagonistic Interactions Conversely, the analysis also uncovered significant *antagonistic* interactions, where a combination of features leads to worse performance. As shown in Figure 52, ads with a 'Motivational' tone (p=0.0038) performed exceptionally well *off* Facebook. However, when placed *on* the `platform_facebook`, their performance dropped to near-zero. This strongly suggests that this default placement is actively detrimental to this specific, high-potential strategy.

4.5.3 Analysis of Key Feature Themes

The following sections provide a "deep dive" into the most prominent and actionable themes identified from the statistical analysis, highlighting their consistent impact

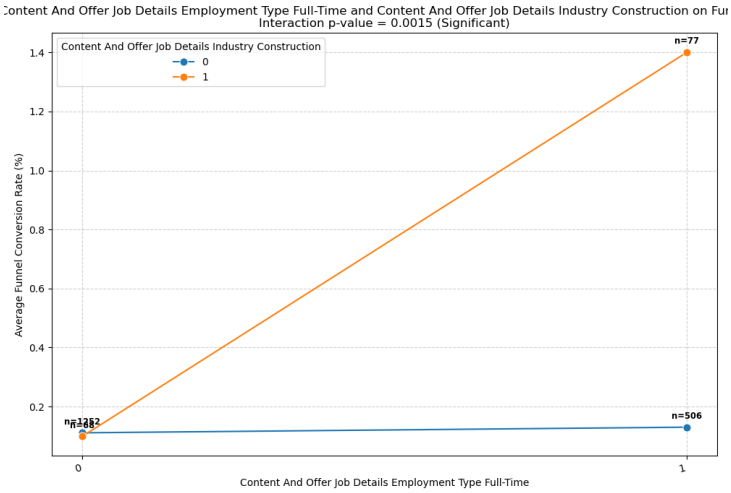


Figure 51: Example of a Significant Synergistic Interaction: Employment Type + Industry (p=0.0015).

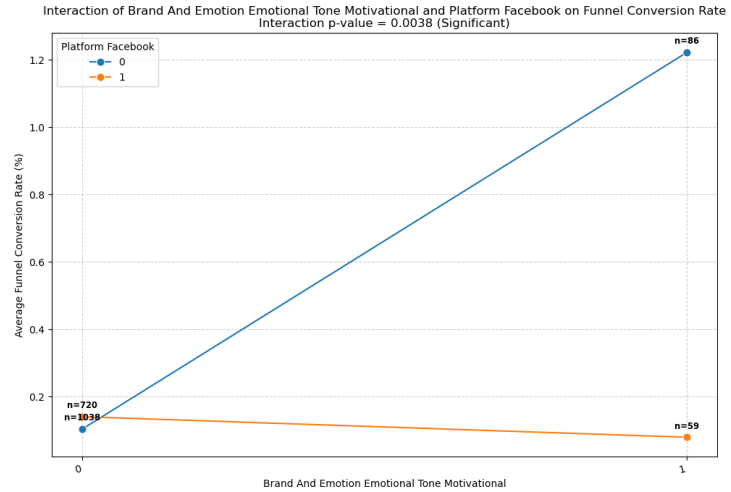


Figure 52: Significant Antagonistic Interaction with 'Facebook' Platform: vs. 'Motivational' Tone (p=0.0038).

across multiple lifetime performance metrics.

The Impact of Salary Information Across all five lifetime efficiency metrics, one of the most consistent findings relates to the inclusion of salary information. This aligns strongly with the literature review (e.g., Alniacik and Alniacik (2025) [2]), which identified informativeness as a critical driver of application intention. The analysis confirmed that mentioning salary was associated with a significantly higher CTR and CR. This powerful combination results in a much better overall FCR, as shown in Figure 53. Furthermore, these efficiency gains translated directly into cost savings, as mentioning salary was also associated with a significantly *lower* CPL.

The Impact of Trainee & Student Targeting A second key strategic finding is the high efficiency of ads targeting trainees, interns, or students. Across multiple

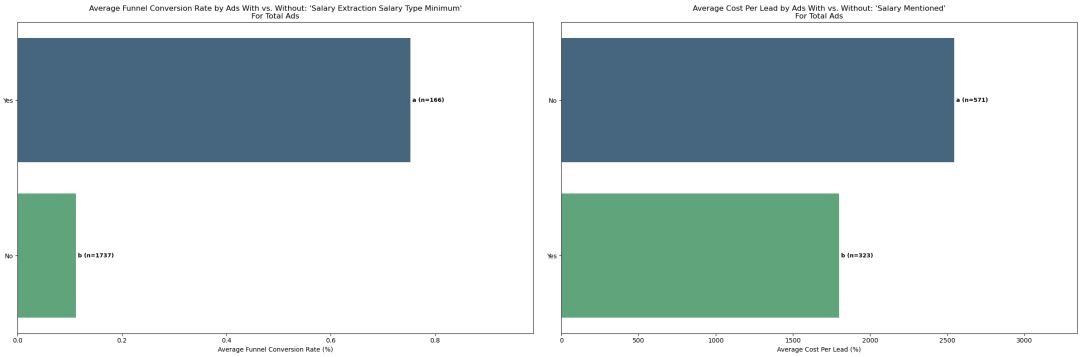


Figure 53: Impact of Salary Information on Funnel Conversion Rate (FCR). Ads mentioning a minimum salary ("Yes") have a significantly higher average FCR ($p < 0.05$) and lower CPL.

feature categories, this audience was consistently associated with superior performance. Ads for trainee-level positions demonstrated a significantly higher FCR. As seen in Figure 54, campaigns with "Traineeship" in their name had a much higher FCR than all other campaigns. This high conversion efficiency directly translated into significant cost savings, as the analysis also confirmed this same feature was associated with a significantly *lower* CPL.

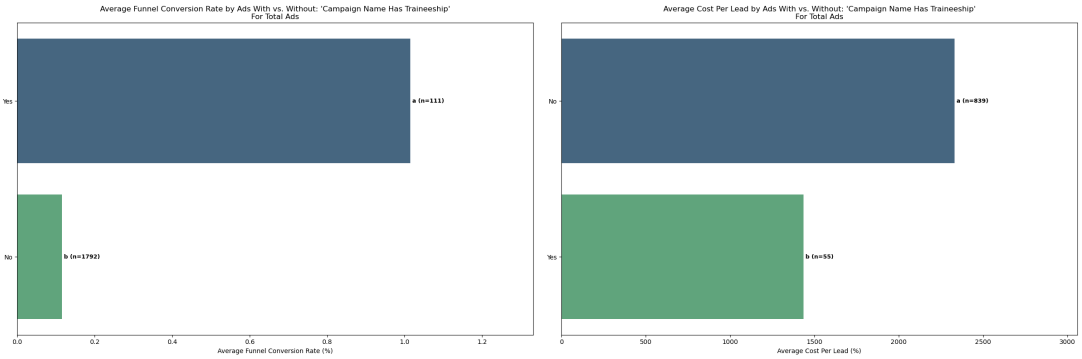


Figure 54: Impact of Traineeship Targeting on Funnel Conversion Rate (FCR). Campaigns for traineeships ("Yes") have a significantly higher average FCR ($p < 0.05$) and lower CPL.

The Impact of Depicting 'Office Work' The analysis of the LLM-extracted visual features revealed that visually depicting 'Office Work' is a highly effective creative strategy for relevant roles. This single feature showed a consistent, positive impact across the performance funnel. The strategy proved effective at the top, achieving a significantly higher average CTR. This initial interest was then converted efficiently, as these ads also had a significantly higher FCR, shown in Figure 55. This combination of higher clicks and better overall conversion makes 'Person Activity: Office Work' a clearly beneficial creative element.

The Impact of a 'Motivational' Tone Another key finding from the LLM-extracted features was the powerful effect of an ad's emotional tone. Specifically, ads identified as having a 'Motivational' tone were linked to significantly better performance. As seen in Figure 56, these ads had a significantly higher average FCR compared to ads that did not. This suggests that a positive, encouraging, and aspirational message is highly effective in converting users. This finding was

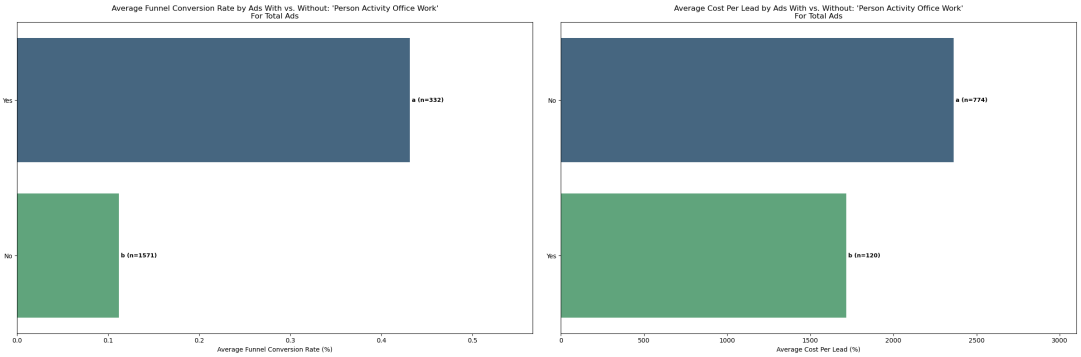


Figure 55: Impact of Depicting 'Office Work' on Funnel Conversion Rate (FCR). Ads showing office work ("Yes") have a significantly higher average FCR ($p < 0.05$) and lower CPL.

also linked to the significant interaction effects, where a 'Motivational' tone showed strong synergistic effects with specific platforms and job functions.

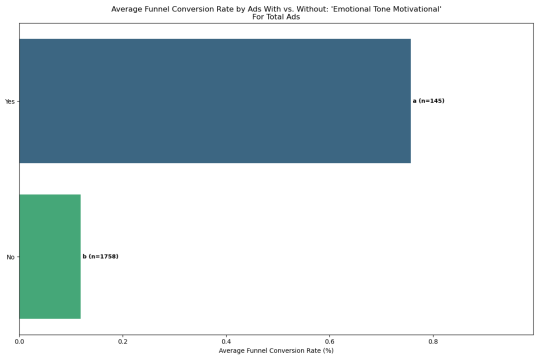


Figure 56: Impact of 'Motivational' Tone on Funnel Conversion Rate (FCR). Ads with this tone ("Yes") have a significantly higher average FCR ($p < 0.05$).

Creative Format (Photo vs. Mixed) The analysis of creative formats revealed a clear distinction in performance. Simple 'Photo' ads emerged as a highly efficient format, while 'Mixed' media ads presented a costly trade-off.

'Photo' ads demonstrated strong, positive performance, showing a significantly higher average CR and, significantly *lower* CPL. This combination makes 'Photo' a clear winner in terms of pure efficiency.

In contrast, 'Mixed' media (such as carousels) showed a more complex performance profile. As seen in Figure 57, 'Mixed' media ads achieved a significantly higher Funnel Conversion Rate (FCR) (Left), indicating a high volume of leads from reach. However, this success came at a significant financial cost, as these same ads also had a significantly *higher* Cost Per Lead (CPL) (Right).

This finding highlights a key strategic trade-off: while 'Photo' ads are a reliable and cost-effective choice, 'Mixed' media can be a powerful tool for achieving high conversion volume but comes at a premium price per lead.

The Efficacy of 'Construction' Industry Ads The analysis of job-specific features identified the 'Construction' industry as a standout performer. This niche demonstrated high efficiency at both the top of the funnel and in overall lead generation. Ads targeting 'Construction' roles achieved a significantly higher average

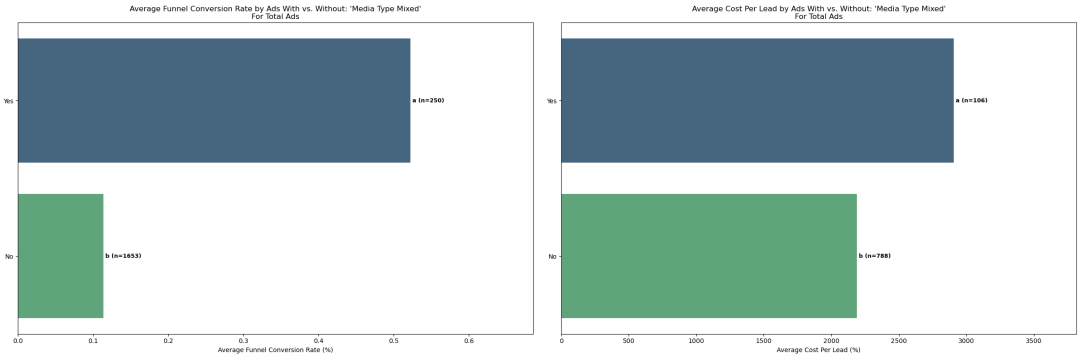


Figure 57: Trade-Off for 'Mixed' Media: FCR (Left) vs. CPL (Right). 'Mixed' media ads ("Yes") show a significantly higher FCR but also a significantly higher CPL ($p < 0.05$).

CTR, suggesting strong initial engagement. This translated to superior overall performance, as these ads also had a significantly higher FCR, as shown in Figure 58.

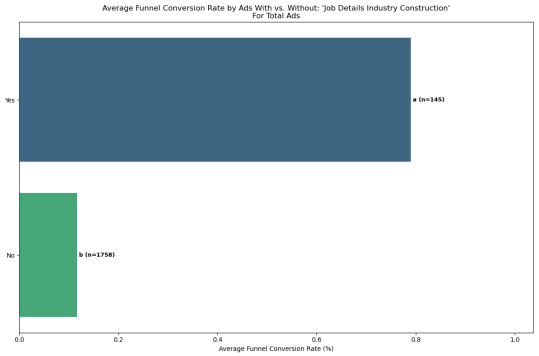


Figure 58: Impact of 'Construction' Industry on Funnel Conversion Rate (FCR). Ads for this industry ("Yes") show a significantly higher average FCR ($p < 0.05$) and lower CPL.

The Value of Casual Presentation The analysis of ad formality and attire revealed another consistent theme: a casual presentation style is highly effective. Features extracted by the LLM, such as `Formality_Casual` and `Clothing_Type_Casual`, were both linked to significantly better performance. This visual style was associated with a significantly higher average CR and a significantly *lower* average CPL. The overall impact is best summarized by the FCR, shown in Figure 59, where 'Casual' ads significantly outperformed their non-casual counterparts.

'Business Development' as a High-Performing Niche A final noteworthy finding is the exceptional performance of ads targeting 'Business Development' roles. This specific job function was not only a strong performer on its own but was a key component in the most powerful synergistic interactions. On its own, 'Business Development' was associated with a significantly higher CTR and, as shown in Figure 60, a significantly higher FCR. This inherent strength was then largely amplified when combined with other features, marking it as a key strategic niche.

The 'Audience Network' Trade-Off A particularly nuanced finding emerged from analyzing the 'Audience Network' platform. This placement was associated with a significantly *higher* CTR and a significantly *higher* overall FCR, suggesting it

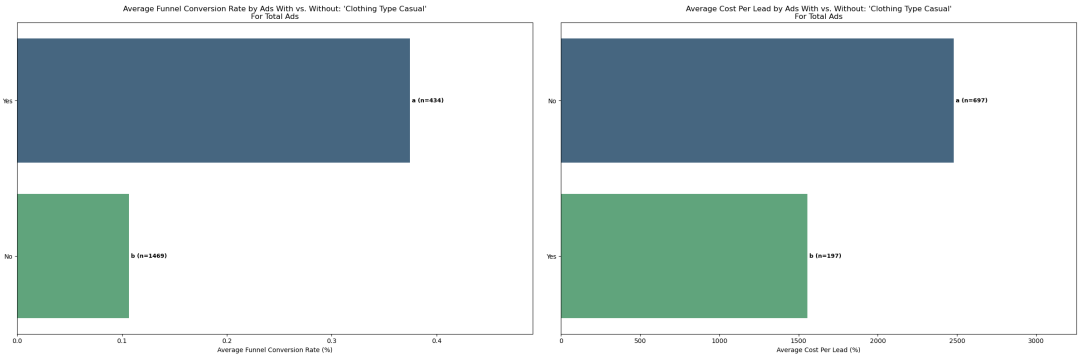


Figure 59: Impact of 'Casual Formality' on Funnel Conversion Rate (FCR). Ads with this attribute ("Yes") have a significantly higher average FCR ($p < 0.05$).

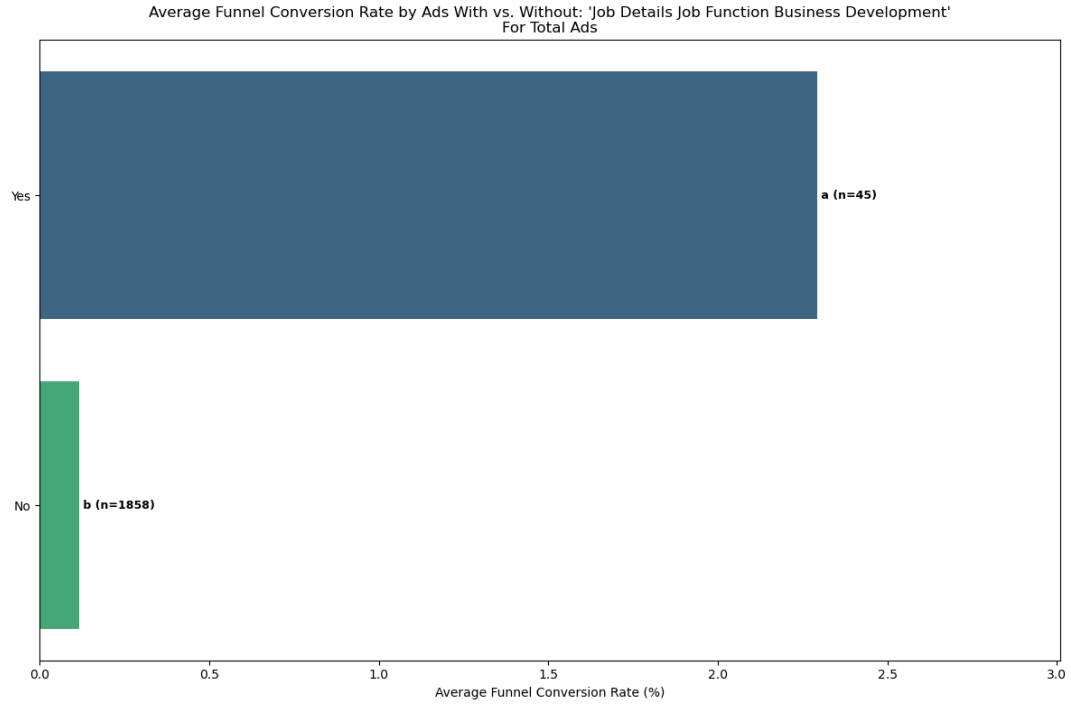


Figure 60: Impact of 'Business Development' Job Function on Funnel Conversion Rate (FCR). Ads for this role ("Yes") have a significantly higher average FCR ($p < 0.05$).

is exceptionally good at capturing attention and converting impressions into leads. However, as shown in Figure 61, it was also linked to a significantly *lower* CR, indicating that the clicks themselves are of lower quality. This low-quality click-through traffic did not negatively impact the final cost; the analysis also confirmed a significantly *lower* CPL. This highlights a key strategic insight: the 'Audience Network' is a powerful tool for driving a high volume of cheap leads, even if the users who click are less qualified.

4.5.4 Analysis for Click-Through Rate (CTR)

The analysis was repeated for the lifetime Click-Through Rate (CTR), which measures the effectiveness of an ad at capturing initial attention (clicks relative to reach).

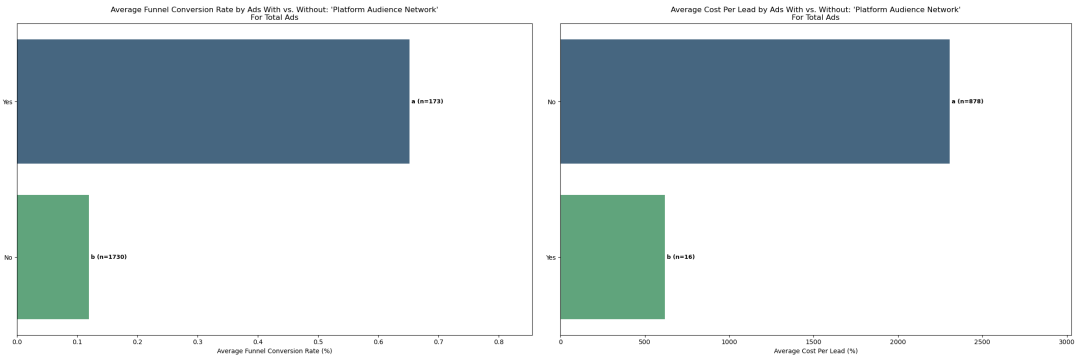


Figure 61: Impact of 'Audience Network' on Conversion Rate (CR). Placement on this platform ("Yes") is associated with a significantly lower average CR ($p < 0.05$) and lower CPL.

Relative Feature Importance for CTR The permutation importance for predicting CTR across all ads is shown in Figure 62. The results clearly indicate that **platform choice is the dominant factor** in predicting CTR.

* `platform_audience_network` is the most important feature by a significant margin.
* This is followed by `platform_unknown` (likely a data artifact, but a strong predictor) and `platform_facebook`.
* Similar to the FCR analysis, `num_advantage_cat` (a measure of informativeness) is also a top-tier predictor.

Naming conventions, such as `ad_name_word_count`, also rank as important, while most specific creative and job-level features show comparatively little predictive power for CTR.

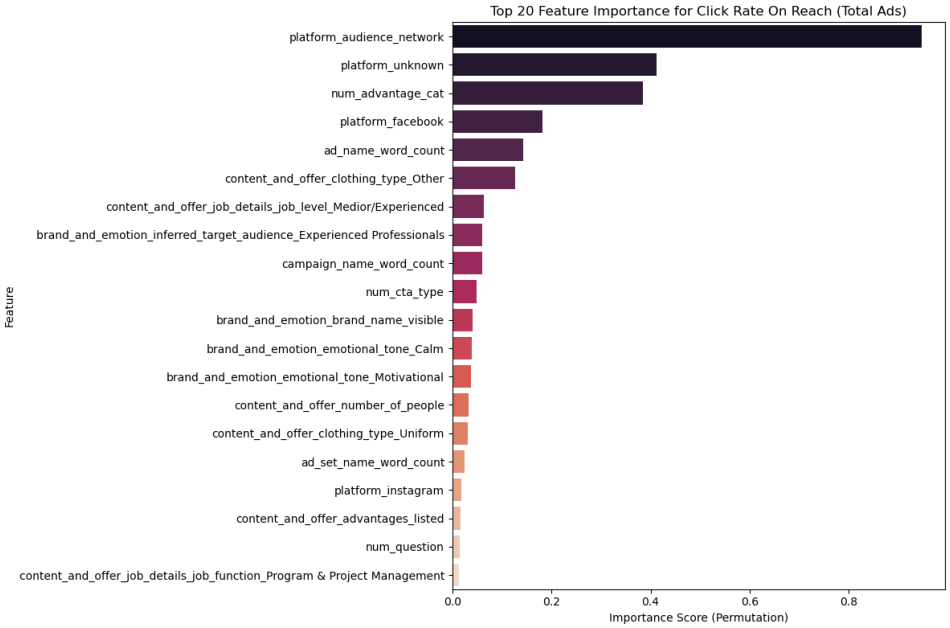


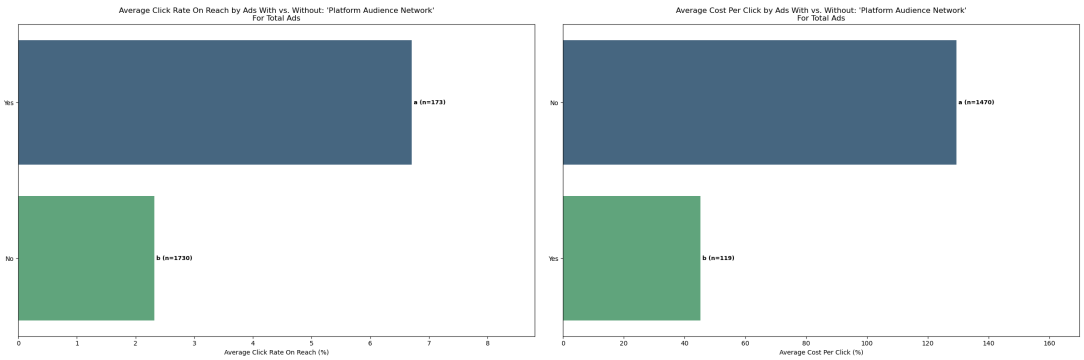
Figure 62: Top 20 Feature Importance for Click-Through Rate (Total Ads) using Permutation Importance.

Statistically Significant Features for CTR The statistical analysis (ANOVA) confirmed that the top-of-funnel performance is highly influenced by platform choice, informativeness, and specific creative elements, often presenting clear trade-offs between engagement (CTR) and cost (CPC).

Platform Trade-Offs

* **Audience Network** delivered a significantly higher CTR (around 6.5%) while simultaneously having a significantly lower average CPC than all other placements, marking it as the most efficient placement for acquiring cheap attention.
* **Facebook** achieved a significantly higher CTR (around 3.3%), with a corresponding decrease in CPC.
* **Instagram** showed the poorest efficiency, achieving a significantly lower CTR while simultaneously having the highest average CPC.

(a)



(b)

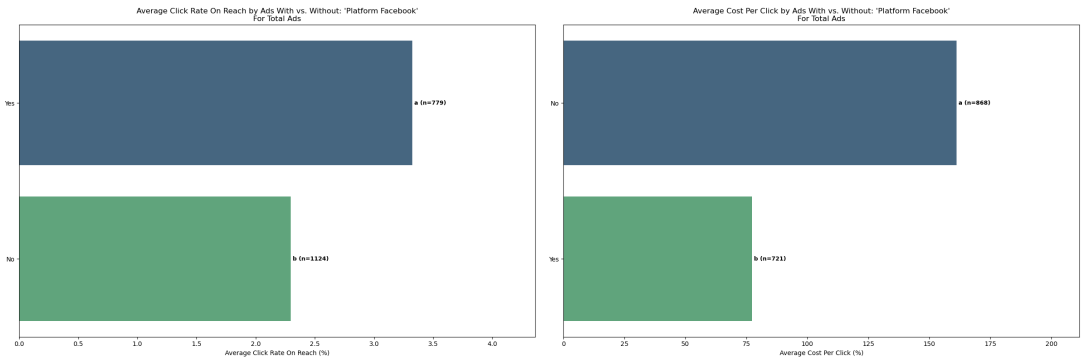
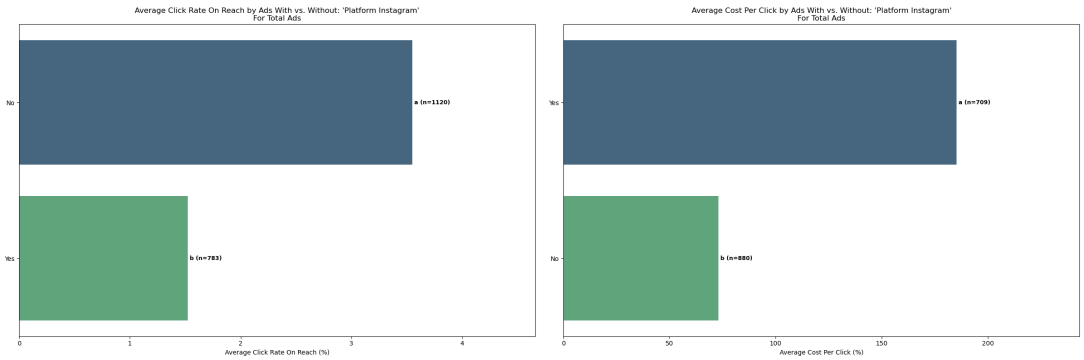


Figure 63: Platform Trade-Offs (1/2): (a) 'Audience Network' (high CTR, low CPC) and (b) 'Facebook' (high CTR, low CPC).

Informativeness and Cost Providing salary details proved to be a dual-benefit strategy for engagement:

* Ads that mentioned salary, specified the currency as EUR, used the period Month, or provided a Salary Range achieved a significantly higher CTR.

(a)



(b)

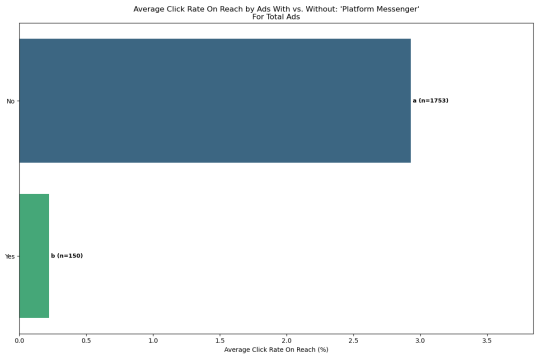


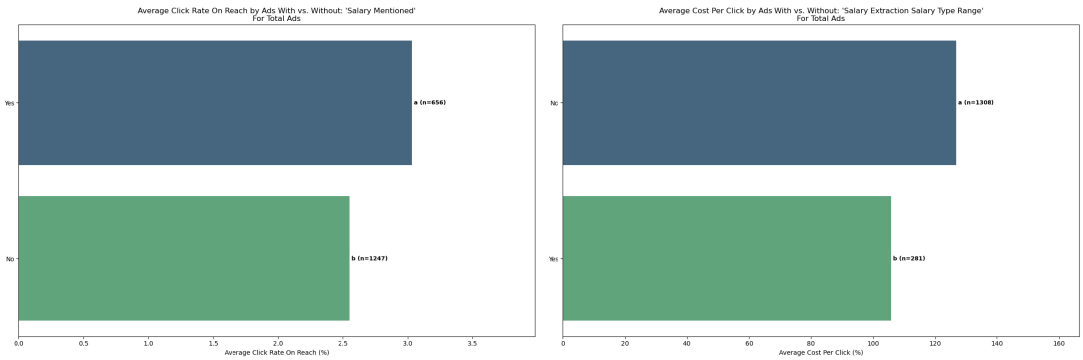
Figure 64: Platform Trade-Offs (2/2): (a) 'Instagram' (low CTR, high CPC) and (b) 'Messenger' (low CTR).

* Crucially, providing a Salary Range or using a Warm color tone also led to a significantly lower Cost Per Click (CPC). Including a question in the ad copy also increased the CTR and reduced the CPC significantly.

Engagement Drivers and Creative Costs

* The most fundamental element, having a Call-to-Action (CTA) present, led to a significantly higher CTR.
* Ads depicting visual diversity (multiple skin tones) achieved a significantly higher CTR and a significantly lower CPC, marking a clear win-win creative strategy.
* A question containing the personal pronoun 'Jij' achieved a significantly higher CTR, and its use made the click less expensive, leading to a significantly lower CPC for ads with any question present.
* Specific job niches like Healthcare ('Zorg') and 'Construction' were linked to a significantly higher CTR and lower CPC.

(a)



(b)

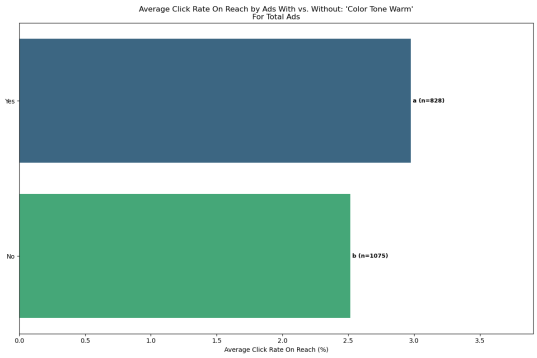


Figure 65: Informativeness Drivers: (a) Mentioning salary or EUR increases CTR and a salary range specified lowers the CPC. (b) A 'Warm' color tone also increases CTR ($p < 0.05$).

Interaction Effects for CTR The OLS regression analysis for CTR revealed several statistically significant interactions, confirming that the effectiveness of a creative feature heavily depends on the platform or the job level being targeted. This is crucial because CTR is the key top-of-funnel engagement metric.

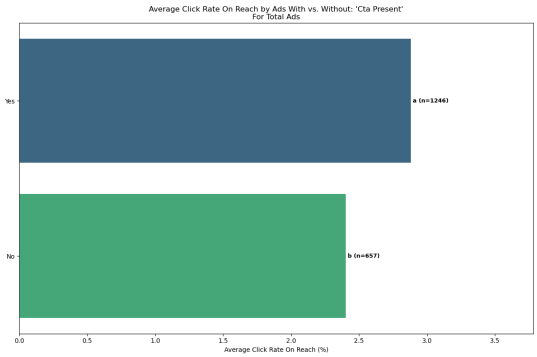
Antagonistic Interaction: Platform vs. Tone (p=0.0179) (see Figure 68, Left)

The Motivational tone and the Facebook platform exhibit a significant antagonistic relationship for engagement. Ads with a Motivational Tone achieve an exceptionally high CTR **off Facebook** (3.65%), but when placed **on Facebook**, their CTR plummets (2.8%). This indicates that the Facebook audience engages **less** with this specific emotional style compared to other platforms.

Antagonistic Interaction: Job Level vs. Uniform (p=0.0347) (see Figure 68, Right)

A significant interaction was found between the job level being advertised and the use of a Uniform in the creative. For Medior/Experienced positions, using a uniform causes the CTR to drop significantly (from 3.4% down to 2.15%). This suggests that **experienced professionals actively disengage** when an ad for an experienced role shows a uniform, potentially associating it with lower-level work or a lack of professional freedom.

(a)



(b)

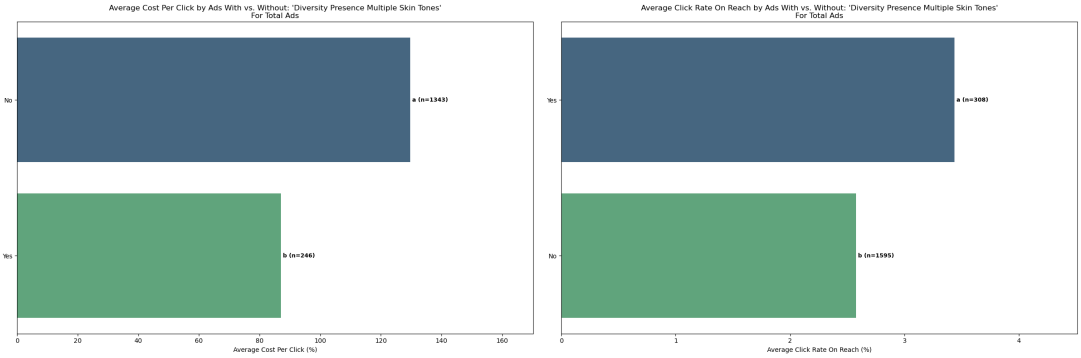


Figure 66: Engagement Drivers (1/2): (a) 'CTA Present' (higher CTR) and (b) 'Visual Diversity' (higher CTR, lower CPC).

5 Discussion

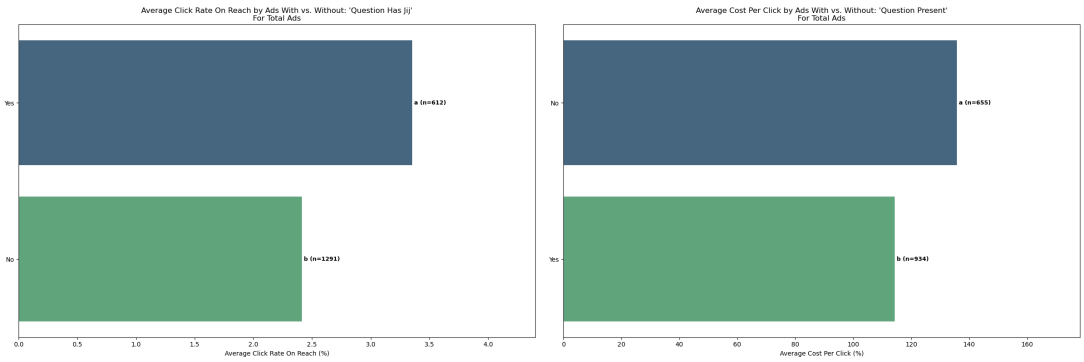
The results of this thesis provide a multifaceted view of ad performance, answering the three sub-questions at different stages of an ad's life cycle. The key findings suggest that: (1) pre-launch creative features are weak predictors of success, but just 1-2 days of live performance data provides a strong signal; (2) for next-day forecasting, sequential deep learning models (GRU, TCN) outperform statistical (SARIMAX) and tree-based (XGBoost) methods, and (3) static creative features have no significant impact on next-day forecasts but show clear, statistically significant correlations with *lifetime* efficiency metrics.

This section reflects on these findings, places them in the context of the broader literature, addresses the methodological choices and limitations of the study, and proposes directions for future research.

5.1 Reflection on Key Findings

The "Top-of-Funnel Disconnect" (Clicks vs. Leads) A foundational insight, first identified in the EDA heatmap (Figure 5), was the weak correlation between top-of-funnel engagement and bottom-of-funnel conversions. The analysis showed that `clicks_all` was a poor proxy for `leads` (correlation: 0.34) and `ctr_all` was even weaker (0.20). This finding validated the methodological choice to treat `clicks` and `leads` as two distinct forecasting targets rather than using clicks to predict leads. The

(a)



(b)

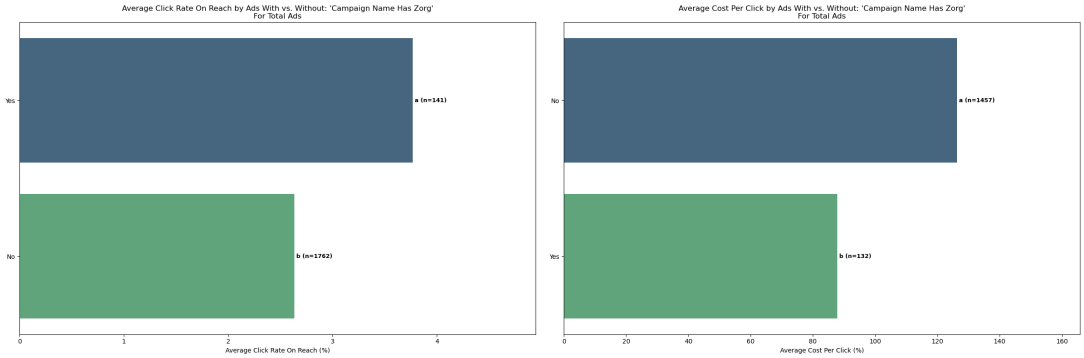


Figure 67: Engagement Drivers (2/2): (a) Using 'Jij'/'Question' (higher CTR lower CPC) and (b) 'Zorg' Campaigns (higher CTR lower CPC).

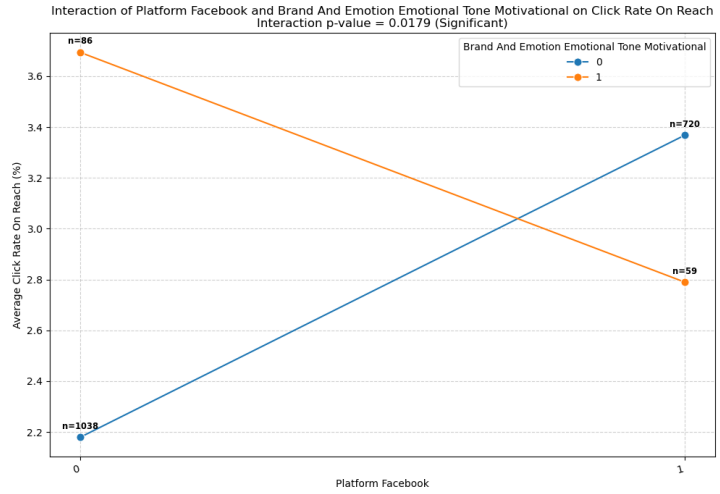
divergent model performance and feature importance for these two targets strongly supports this "top-of-funnel disconnect": the creative and contextual factors that attract a user's *attention* (a click) are not the same as those that drive a user's *intent* (a lead).

The Paradox of Static Feature Importance (SQ2 vs. SQ3) Perhaps the most significant finding of this thesis is the apparent paradox of static feature importance. The results for SQ2 showed that adding static creative features provided no significant improvement for *next-day* forecasting. Conversely, the analysis for SQ3 demonstrated that these same creative features (e.g., emotional tone and salary information) have clear, statistically significant correlations with an ad's total *lifetime* efficiency.

This suggests a clear distinction between tactical and strategic variables.

* **Next-day performance (tactical)** is dominated by short-term, high-frequency patterns. The ACF plots (Figure 6) showed strong persistence and seasonality. The deep learning models (GRU, TCN) likely outperformed XGBoost by being superior at learning these complex sequential "memory" and momentum patterns from the 14-day lookback. In this context, a static feature like "emotional_tone=Motivational" is just noise; it doesn't help predict tomorrow's value, which is almost entirely dependent on today's value and the day of the week.

(a)



(b)

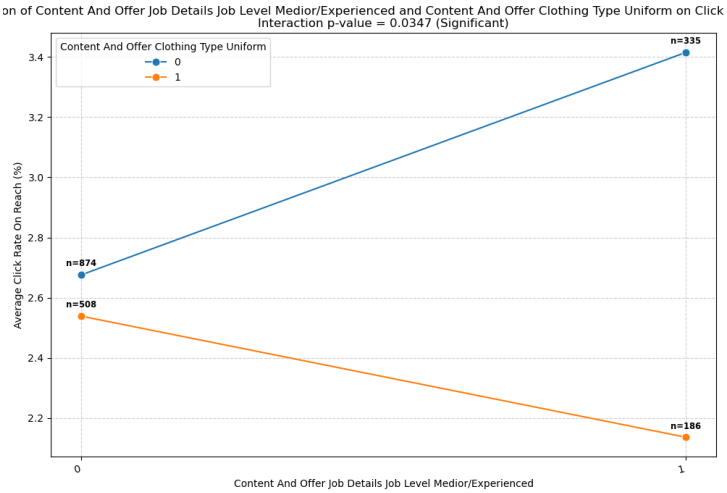


Figure 68: Significant CTR Interaction Effects: (a) Platform vs. Tone (p=0.0179) and (b) Job Level vs. Uniform (p=0.0347).

* **Lifetime performance (strategic)** is an aggregate measure of an ad's fundamental *quality* and its *fit with the audience*. Here, the short-term momentum patterns are averaged out, and the ad's core message—the "why" a user converts—becomes the dominant predictor. This is precisely what static creative features capture.

Interpreting Forecasting Model Performance (SQ2) The superiority of the GRU and TCN models aligns with the time-series properties discovered in the EDA. The strong trend (slow-decaying ACF) and seasonality (PACF lag 7) present a complex sequential problem. The deep learning models, which are purpose-built for sequence-to-sequence tasks, are inherently better at capturing these long-range dependencies than XGBoost, which only sees time through manually engineered lag/rolling features. The SARIMAX model, while explicitly handling seasonality ($m = 7$), was ultimately too rigid and failed on the noisy, volatile, and sparse data

that is characteristic of real-world ad performance.

5.2 Comparison with Related Work

Placing these findings within the context of existing literature reveals both consistencies and novel contributions.

Forecasting Model Hierarchy (SQ2) The superior performance of the sequential deep learning models (GRU, TCN) over the tree-based (XGBoost) and statistical (SARIMAX) methods for next-day forecasting aligns with recent trends in time-series analysis [13]. Studies have similarly demonstrated that deep learning architectures, which learn feature representations automatically from sequences, often outperform models like XGBoost that rely on manually-engineered lag features, especially with noisy or non-stationary data. The poor performance of SARIMAX is also consistent with literature highlighting its limitations in handling the high volatility and zero-inflation common in digital marketing data, a problem more formally known as intermittent demand [14].

The "Cold Start" and "Early Signal" (SQ1) The results from SQ1, which highlight the poor predictive power of pre-launch static features alone, are consistent with the "cold start" problem discussed in online advertising and recommender system literature [15]. More importantly, the finding that just one to two days of live performance data greatly increases predictive accuracy supports the "early-life signal" theory, which has been observed in related fields such as online content virality modeling [16, 17]. This study empirically confirms that this principle holds true for creative ad performance.

Temporal Distinction in Feature Importance (SQ2 vs. SQ3) The "paradox of static feature importance" appears to be a more novel contribution of this thesis. Most related work tends to focus on *either* using creative features to predict *overall* ad success [18] *or* on high-frequency time-series forecasting without static features [19]. The finding that static features had no significant advantage for tactical, next-day momentum-based forecasting (SQ2) but are critical for strategic, lifetime-value prediction (SQ3) provides a clear, actionable distinction not widely discussed in the current literature. This suggests that future research on feature importance should be more precise about the *temporal horizon* being predicted.

5.3 Limitations and Methodological Considerations

While this research provides a comprehensive framework, several methodological choices and limitations must be discussed to contextualize the results.

The "Good" vs. "Bad" Label (SQ1) The binary classification in SQ1 relied on a proxy for success: whether an ad received more than 20% of the maximum spend within its campaign. This 20% threshold is, by definition, arbitrary. More importantly, the resulting "Good" and "Bad" labels are not objective measures of ad quality. Instead, they are largely a reflection of the *specific, subjective decisions* made by the marketing team of Dutchwebshark.

Another company employs different human optimizers, who will have different heuristics, risk tolerances, and optimization strategies. Faced with the exact same set of ads, a different team would make different choices about which ads to keep live, resulting in completely different spending patterns. This means the "Good"/"Bad" labels themselves are an artifact of a specific team's behavior. Therefore, the classification model from SQ1 is highly specific to this one team and cannot be generalized to another company. It is less a model of "ad success" and more a model of "which ads does this specific team choose to keep?" The core finding, however—that early performance data is the most critical predictor for *any* such decision—remains robust.

Forecasting Model Comparison (SQ2) The comparison of forecasting models, while rigorous, has several caveats.

* **SARIMAX Architectural & Methodological Limitations:** The SARIMAX model's comparison to the ML models is complex, as it suffers from both fundamental architectural and specific methodological limitations for this problem.

  Architecturally, as a *univariate* model, it must be trained on each ad's time series individually, which has two critical consequences:

  1. It cannot learn *global, cross-series patterns* from the 500+ other ads in the dataset.
  2. Static creative features (e.g., 'emotional tone') are rendered statistically useless. For any single ad's time series, this feature is a constant value, providing no information about temporal changes, and its entire effect is simply absorbed into the model's intercept.

  Methodologically, the model was further handicapped by the implementation:

  3. The **14-day rolling lookback window** is severely restrictive. This short window is insufficient for `auto_arima` to reliably detect seasonal patterns ($m = 7$, requiring only two cycles) and gives the model "amnesia," preventing it from learning any trends or behaviors that occur over a longer timeframe.

  In contrast, the ML models were trained as *global, multi-series* models (likely with a much longer lookback), allowing them to learn from all ads simultaneously. Therefore, the SARIMAX model was not just "handicapped" by its architecture; it was fundamentally blind to cross-sectional data and hamstrung by a lack of historical context.

– **Metric Calculation and Model Failures:** The reported performance metrics (MAE, RMSE) are calculated *only* on the time series for which a given model successfully produced a forecast. The baseline, by contrast, produced a forecast in 100% of cases. The more complex models, particularly 'auto_arima' within SARIMAX, could fail on short or highly volatile series. This creates a selection bias: the models are being evaluated only on the (likely easier) series they could handle, which skews the performance metrics in their favor relative to the baseline. A more robust approach, especially given the sparse nature of 'leads' data, might be a hurdle model (a two-part model that first predicts the probability of a non-zero event, then predicts the magnitude of that event).

Scalability and Architectural Trade-Off The deep learning (DL) models (`TCN` and `GRU`) were trained sequentially on a Linux PyTorch AMI `4gdn.xlarge` instance, requiring 20 to 35 hours for the complete 5-fold cross-validation. In comparison, the `XGBoost` ($\approx$ 3.68 hours) and `SARIMAX` ($\approx$ 1.33 hours) models trained much faster for this dataset size, highlighting the trade-off between predictive power and computational cost.

However, for a real-world production environment, the architectural design dictates scalability:

1. **Computational Bottleneck of SARIMAX:** The `SARIMAX` model operates on a "model-per-series" architecture, requiring a separate, CPU-bound model fit for every single time series. A system built to handle the full 5,000+ creative inventory would find the maintenance, deployment, and cumulative CPU cost of thousands of individual SARIMAX models to be a significant and limiting bottleneck.
2. **Architectural Scalability of Global Models (ML/DL):** Both the `XGBoost` and the DL (`TCN`, `GRU`) models are trained as *global models*. They rely on one single model capable of forecasting all series, allowing for efficient batch processing. This architecture is the only one that demonstrates a clear path to high-throughput scalability. The DL models, being GPU-native, can use GPU parallelism for low-latency inference, while the `XGBoost` model provides a highly-scalable, CPU-native alternative with significantly faster training times.

Therefore, while statistical "model-per-series" approaches like `SARIMAX` are inefficient and unviable at scale, the **global ML/DL models** are the clear solution for high-volume, production-grade forecasting.

Handling Non-Negative Targets (Clipping vs. Transformation) A key methodological challenge was handling the non-negative nature of the `clicks` and `leads` targets, which are forms of count data ($y \geq 0$). The regression models used (SARIMAX, XGBoost, GRU, TCN) are not inherently constrained and can produce negative forecasts.

Two different strategies were employed to address this:

– For **SARIMAX**, a square root transformation ($\sqrt{y}$) was used. This is a standard statistical approach to stabilize variance and make the data's distribution more symmetric. While this addresses the model-data mismatch, the back-transformation method—squaring the final forecast ($\hat{y} = \hat{y}^2_{transformed}$)—introduces a significant limitation. This "naive" back-transformation, while mathematically guaranteeing a non-negative result, creates a **downward statistical bias**. Due to Jensen's Inequality, the square of the model's mean prediction $(E[\sqrt{y}])^2$ is an estimate of the distribution's *median*, not its *mean* ($E[y]$), and will therefore systematically under-predict the true average.

For **XGBoost, TCN, and GRU**, small estimation errors from the scaled target (e.g., $[0, 1]$) could be magnified into negative values (e.g., -5) by the scaler's inverse-transform.

This issue could be formally addressed in several ways:

1. Using a **target transformation** (e.g., $\log(y + 1)$), which would guarantee non-negative predictions after the inverse-transform.
2. Changing the **loss function** to one designed for non-negative data (e.g., `reg:gamma` in XGBoost or RMSLE).

3. Modifying the **neural network architecture** for TCN and GRU to include a ReLU activation function on the final output layer, constraining its output to be non-negative.

While these approaches are statistically more robust, a simpler *post-processing* step was chosen. To correct this *symptom*, all final, unscaled predictions were clipped at zero ($\max(0, \hat{y})$). This is a common and pragmatic solution for ML/DL models [21].

Validation Strategy The validation strategy was built around a 5-fold `GroupKFold`, using `campaign_id` as the grouping key. This was a critical methodological strength, as it successfully prevents the most common form of data leakage by ensuring that ads from the same campaign never appear in both the training and testing sets.

However, this 'GroupKFold' strategy is not a strict temporal validation. Within the folds, it is possible for a model to be trained on data from January and March and tested on data from February. This potential for temporal leakage could lead to an optimistic estimation of model performance. A more robust, albeit more complex, validation strategy would involve a rolling-forecast origin or a 'TimeSeriesSplit' nested within the 'GroupKFold' structure to ensure that all test data is, in all cases, "in the future" relative to its training data.

LLM Feature Extraction The use of `gemini-2.5-pro-latest` to extract creative features (Section 3.4.1) was a powerful method for converting unstructured media into structured data. However, this method the following limitation. While guided by a strict JSON schema, the model's classifications (e.g., `emotional_tone`, `visual_complexity`) are still a form of "black box" subjective judgment.

5.4 Future Research

The findings and limitations of this study suggest several promising avenues for future research.

Exploring Model Parameters The forecasting models were built with a fixed 14-day lookback and a 1-day forecast horizon. This 14-day window was chosen based on the 7-day seasonality, but it may not be optimal. Future work should experiment with different lookback periods (e.g., 7 days, 21 days) to determine the optimal history length for ML models. Furthermore, a more practical application for the business would be multi-step forecasting (e.g., predicting the next 3 or 7 days). The models, particularly the TCN and GRU, should be re-evaluated on this more challenging task.

Hyperparameter and Feature Optimization The hyperparameter search using Optuna was limited to 15 trials per fold to manage computational cost. A more extensive search (e.g., 50-100 trials) could yield more performant model configurations.

Additionally, the SQ2 models were trained on all available features. The analysis in SQ3 demonstrated that many creative features have a low (or even antagonistic) correlation with performance. A valuable next step would be to re-train the SQ1 and SQ2 models using only a subset of the top 20 or 30 features identified by the permutation importance analysis.

The finding that the addition of creative features did not lead to a statistically significant improvement across any architecture suggests these features may be too noisy or redundant for forecasting.

- **Pruning and Simplification:** Non-informative or low-ranking creative features should be systematically pruned to create streamlined Perf. + Creative models. This process ensures that the model complexity only increases due to features that demonstrably reduce forecasting error.
- **Feature Interaction Analysis:** Before retraining, a dedicated analysis using techniques like SHAP (SHapley Additive exPlanations) values should be performed to quantify the true contribution of individual creative features within the top-performing GRU and TCN models.
- **Alternative Encodings:** Explore different encoding strategies for categorical or text-based creative features, such as advanced embeddings, to capture non-linear relationships that the current approach may have missed.

Enhancing Statistical and Methodological Rigor The statistical analysis established that the current experimental framework, using $N = 5$ cross-validation folds, lacks the power required to decisively confirm the practical superiority of the deep learning models. Future work must focus on increasing statistical robustness.

- **Increase Statistical Power through Validation:** To validate the observed practical improvements and potentially achieve statistically significant results, the number of cross-validation observations must be increased. This can be achieved by expanding the simple $K = 5$ folds to a minimum of $K = 10$ folds, or, preferably, by employing *Repeated K-Fold Cross-Validation (RKCV)*, which repeats the entire K-fold procedure R times. RKCV provides a more stable estimate of the model's true generalization error.
- **Alternative Statistical Testing:** While the Wilcoxon test is appropriate for comparing MAE across folds, the comparison of time series forecast accuracy is often enhanced by utilizing tests that leverage the full error history. If the full series of prediction errors (not just the fold-aggregated MAE) is saved, the *Diebold-Mariano (DM) test* should be applied. The DM test is specifically designed for evaluating predictive accuracy in time series and provides a more nuanced measure of significance.

Improving the LLM Pipeline The LLM feature extraction pipeline could be significantly improved. Future work should explore fine-tuning a smaller, open-source vision model (e.g., LLaVA) on the 643 JSON outputs generated by Gemini. This could create a faster, cheaper, and more reliable model specialized for this specific ad-extraction task. Furthermore, the model's inputs could be enriched by providing it with the ad's *text copy* in addition to the media file, allowing it to correlate the ad's textual promises with its visual style.

Closing the Loop: From Analysis to Action This thesis provides several clear, actionable insights. Future research should focus on operationalizing them.

- **A/B Test Generation:** The findings from SQ3 (e.g., "Motivational tone improves FCR") should be used not as conclusions, but as hypotheses for a new round of structured A/B tests to confirm a causal link.
- **Automated Intervention:** The SQ1 classifier (good/bad) could be used to automatically pause and flag ads for human review, saving budget.
- **Hurdle Model for Leads:** As noted in the limitations, a hurdle model (a classifier for $P(\text{lead} > 0)$ combined with a regressor for $E[\text{leads}|\text{leads} > 0]$) should be tested. This is almost certainly a more accurate approach for sparse, zero-inflated targets like daily leads.

Exploring Data Granularity (Hourly vs. Daily) This study was conducted on data aggregated at a daily level. A promising area for future research would be to analyze the raw hourly performance data. This higher-granularity data could reveal intra-day patterns (e.g., performance dips in the early morning, peaks during evening commutes) that are currently invisible. Deep learning models like TCN and GRU, which excel at finding patterns in long sequences, could potentially leverage this 24-hour cycle to improve next-day forecasting accuracy. However, this approach would present a significant challenge in computational cost, as the dataset's sequence length would increase by a factor of 24. It would also introduce new, complex seasonalities (e.g., time of day), which would need to be carefully modeled alongside existing patterns such as day-of-week effects.

Expanding the Dataset The current dataset, while high-quality, is sourced from a single company. This limits the generalizability of the findings, particularly for the SQ1 classifier, which may be modeling the specific optimization habits of one team (as discussed in 5.3). A significant step forward would be to enlarge the dataset, either by acquiring more historical data from the same company or, ideally, by obtaining data from multiple companies across different industries. A larger, more diverse dataset would allow for the training of more robust and generalizable models, helping to distinguish between universal principles of ad performance and company-specific artifacts.

6 Conclusion

This thesis set out to answer the research question: *How can a quantitative framework be developed to accurately forecast and evaluate the performance of social media recruitment ads by systematically assessing the predictive power of creative features in comparison to dynamic performance data?*

The resulting framework successfully developed, evaluated, and analyzed multiple predictive pipelines, leading to a definitive synthesis of the predictive roles of static creative features versus dynamic performance data across the ad campaign life cycle.

6.1 Synthesis of Findings

The key findings, structured by the sub-questions, demonstrate a clear hierarchy of predictive power and model suitability:

Classification (SQ1): The Power of Early Intervention The XGBoost classifier demonstrated that pre-launch creative features alone are **weak predictors of success** (Day 0 F1-score for "Not Chosen" $\approx$ **0.73**, with low certainty in the minority class). However, the inclusion of just 1–2 days of cumulative performance data proved effective, causing the model's performance on the critical "Not Chosen" class (ads to be filtered) to **improve**. The F1-score for this class reached **0.79** by Day 2, validating the feasibility and value of an **early-intervention system** designed to automatically flag or pause budget allocation before significant spend is wasted.

Forecasting (SQ2): Sequential Models are Superior For the next-day forecasting task, the deep learning architectures outperformed both the tree-based (XGBoost) and statistical (SARIMAX) models, aligning with existing literature on sequential data. These results confirm that deep learning models are better equipped to capture the data's persistence and $m = 7$ seasonality.

- **Clicks):** The **TCN (Perf. + Creative)** model was the strongest, achieving a 45.5% reduction in Mean Absolute Error (MAE) compared to the Realistic Baseline, with MASE = **0.558**.
- **Leads (Sparse Target):** The **GRU (Perf. Only)** model was the most accurate, achieving a **24.5**% MAE reduction, with MASE = **0.421**. This means the GRU's forecast error was only 42.1% of the Naïve benchmark's error, validating its structural skill for the sparse leads target.

Analysis (SQ3): The Paradox of Feature Importance The most significant contribution of this work lies in revealing the **Paradox of Static Feature Importance**. While the deep learning models in SQ2 confirmed that adding creative features provides no statistically significant improvement for **tactical, next-day forecasting**, the dedicated analysis for SQ3 showed that these same features are **critical for strategic, long-term performance evaluation**. Creative attributes such as **Salary Inclusion**, **Casual Presentation**, and **Motivational Tone** showed clear, statistically significant correlations with superior lifetime efficiency metrics (e.g., FCR, CPL, CTR), providing actionable creative hypotheses.

The Top-of-Funnel Disconnect A foundational insight validated across the entire framework was the **top-of-funnel disconnect**: clicks are a poor proxy for leads (correlation $r = 0.34$). The factors that drive attention (CTR) are not the same as those that drive conversion (CR/FCR). This disconnect justifies the dual-target forecasting approach and highlights that creative strategy must be designed to optimize the entire funnel, not just initial engagement.

6.2 Overall Conclusion and Outlook

The quantitative framework developed herein provides a comprehensive, multi-stage strategy for optimizing recruitment ad spend. It moves beyond traditional human intuition and simple linear forecasting by establishing that effective campaign management requires a hybrid approach:

1. **Early Prediction:** Using performance signals from the first two days to quickly flag and pause poor ads (SQ1).

2. **Tactical Forecasting:** Employing **sequential deep learning models (TCN/GRU)** for accurate, short-term budget allocation (SQ2).

3. **Strategic Design:** Leveraging the **statistically significant insights** from creative feature analysis to inform creative design that maximizes lifetime value and cost efficiency (SQ3).

Despite strong practical results, this thesis acknowledges the methodological limitation of **low statistical power** (Section 4.3). While the magnitude of the error reduction strongly supports the superiority of the TCN and GRU models, future work should employ **Repeated K-Fold Cross-Validation** and the **Diebold–Mariano test** to elevate these promising findings to statistically verifiable conclusions.

Overall, this thesis provides a clear and robust roadmap for future research, cementing the integration of advanced deep learning and structured creative analysis as a cornerstone of data-driven recruitment marketing.

References

[1] Akiba, T., Sano, S., Yanase, T., Ohta, T., & Koyama, M. (2019). Optuna: A next-generation hyperparameter optimization framework. In *Proc. 25th ACM SIGKDD Int. Conf. on Knowledge Discovery & Data Mining (KDD '19)* (pp. 2623–2631).

[2] Alniacik, E., & Alniacik, U. (2025). How recruitment ad informativeness influence application intentions: mediating role of perceived fit and moderating role of employment experience. *Human Resources Management and Services*. https://doi.org/10.18282/hrms3576.

[3] Bhulai, S., Folman, P. F. F. G., & van der Heijden, J. F. M. G. (2017). Predicting Applicant Attrition in a Large-Scale Recruitment Process. In *Proceedings of the 2017 IEEE International Conference on Big Data (Big Data)* (pp. 4626-4632). https://doi.org/10.1109/BigData.2017.8258514.

[4] Edizel, B., Mantrach, A., & Bai, X. (2017). Deep Character-Level Click-Through Rate Prediction for Sponsored Search. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '17)* (pp. 305–314). https://doi.org/10.1145/3077136.3080811.

[5] Guo, T., Yang, Z., Zeng, Q., Chen, M. (2025). *Context-Aware Lifelong Sequential Modeling for Online Click-Through Rate Prediction*. arXiv preprint arXiv:2502.12634. https://arxiv.org/abs/2502.12634.

[6] Lu, Q., Pan, S., Wang, L., Pan, J., Wan, F., Yang, H. (2017). A Practical Framework of Conversion Rate Prediction for Online Display Advertising. In *Proceedings of the ADKDD'17*. https://doi.org/10.1145/3124749.3124750.

[7] Mahjoub, A., Kruyen, P. (2021). Efficient recruitment with effective job advertisement: an exploratory literature review and research agenda. *International Journal of Organization Theory and Behavior*. https://doi.org/10.1108/ijotb-04-2020-0052.

[8] Yang, Y., Zhai, P. (2022). Click-through rate prediction in online advertising: A literature review. *Information Processing & Management*, 59(2), 102853. https://doi.org/10.1016/j.ipm.2021.102853.

[9] Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In *Proc. 22nd ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining (KDD '16)* (pp. 785–794). https://doi.org/10.1145/2939672.2939785

[10] Bai, S., Kolter, J. Z., & Koltun, V. (2018). An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271*. https://arxiv.org/abs/1803.01271

[11] Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *Proc. 2014 Conf. on Empirical Methods in Natural Language Processing (EMNLP)* (pp. 1724–1734). https://doi.org/10.3115/v1/d14-1179

[12] Box, G. E. P., Jenkins, G. M., Reinsel, G. C., & Ljung, G. M. (2015). *Time series analysis: Forecasting and control*. John Wiley & Sons.

[13] Lim, B., & Zohren, S. (2021). Deep learning for time series forecasting: A survey. *arXiv preprint arXiv:2101.12072*.

[14] Syntetos, A. A., & Boylan, J. E. (2005). The accuracy of intermittent demand forecasting methods. *International journal of forecasting*, 21(2), 303–314. Elsevier.

[15] Schein, A. I., Popescul, A., Ungar, L. H., & Pennock, D. M. (2002). Methods and metrics for cold-start recommendations. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 253–260).

[16] Szabó, G., & Huberman, B. A. (2010). Predicting the early popularity of online content. *Communications of the ACM*, 53(8), 80–88. ACM.

[17] Cheng, J., Adamic, L. A., Dow, P. A., Kleinberg, J. M., & Leskovec, J. (2014). Can cascades be predicted? In *Proceedings of the 23rd international conference on World wide web* (pp. 925–936).

[18] Teixeira, T. S., Wedel, M., & Pieters, R. (2012). Emotion-induced engagement in Internet video advertisements. *Journal of Marketing Research*, 49(2), 144–159. SAGE Publications.

[19] McMahan, H. B., Holt, G., Sculley, D., Kucukelbir, A., Yilmaz, E., Mediratta, R., Caven, D., Mladenov, M., Wang, Z., O'Malley, S., & others. (2013). Ad click prediction: a view from the trenches. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 1222–1230).

[20] Wilcoxon, F. (1945). Individual comparisons by ranking methods. *Biometrics Bulletin*, 1(6), 80–83.

[21] Géron, A. (2022). *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow* (3rd ed.). O'Reilly Media.

[22] Hyndman, R. J., & Koehler, A. B. (2006). *Another look at measures of forecast accuracy*. International Journal of Forecasting, 22(4), 679–688.