Finetuning Pretrained GPT-2 for Dutch TTF Gas Imbalance Prediction: A Mixed Time Series Prediction and Classification Approach

Author:
King Hang Siu
310240

Supervisors:
Prof. Dr. Dr. Lars Schmidt-Thieme
Johannes Burchert
Carsten Hoelsken, RWE Supply & Trading GmbH

13th March 2024

Thesis submited for

Master of Science in Data Analytics

Wirtschaftsinformatik und Maschinelles Lernen
Stiftung Universität Hildesheim
Universitatsplätz 1, 31141 Hildesheim

Statement as to the sole authorship of the thesis:

TITLE OF THE THESIS. I hereby certify that the master's thesis named above was solely written by me and that no assistance was used other than that cited. The passages in this thesis that were taken verbatim or with the same sense as that of other works have been identified in each individual case by the citation of the source or the origin, including the secondary sources used. This also applies for drawings. sketches, illustration as well as internet sources and other collections of electronic texts or data, etc. The submitted thesis has not been previously used for the fulfilment of a degree requirements and has not been published in English or any other language. I am aware of the fact that false declarations will be treated as fraud.

Date, City Signature

Abstract

While Large Language Models (LLM) have achieved significant success in natural language processing (NLP) and computer vision (CV), their application in time series analysis has been limited due to the lack of large-scale training data. This thesis addresses this challenge by finetuning pre-trained models from language or computer vision, trained on billions of tokens, for time series analysis. This thesis evaluates the Frozen Pretrained Transformer (FPT), which leverages the self-attention and feedforward layers of residual blocks from pre-trained models. The paper introduces MTSPC-GPT, a novel approach that focus on fine-tuning the pre-trained GPT2 model for multi-task learning, specifically for mixed time series prediction and classification. This approach enables the model to concurrently perform classification and forecasting tasks. The study also explores various methods to enhance the predictability of gas-related time series, such as the Dutch TTF Gas Balancing Signal, by integrating external features like weather data, power forecast data, temporal features, and technical indicators. The results indicate that these pre-trained models can deliver comparable or even superior performance on public time series classification datasets and the Dutch TTF Gas Imbalance Dataset. The code for this research is publicly accessible at https://github.com/ttterence927/MTSPC-GPT

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Multivariate time-series data is utilized in various sectors such as finance, healthcare, environmental science, and energy markets, playing a significant role in decision-making processes. The details found in these data sets are important for forecasting, particularly in situations where precision are of high importance. Effective analysis of multivariate time-series allows the prediction of future trends and the facilitation of different strategies [Wei, 2019].

In forecasting, performance is often tied to the ability to learn robust representations that can capture historical patterns and predict future events. The rise of foundation models, especially in Natural Language Processing (NLP) and Computer Vision (CV), gives the potential of advanced representational learning. Transformer-based models and convolutional neural networks are examples of these models that have significantly influenced their respective fields by extracting deeply informative representations [Vaswani et al., 2017, He et al., 2016].

Inspired by the pioneering achievements in NLP and CV, foundation models such as GPT-2 show a great effectiveness in few-shot learning paradigms, attributable to their extensive pre-training across large corpora [Radford et al., 2019]. The implications of this are particularly significant in domains where data scarcity limits model performance, signifying a potential frontier for deploying these skilled learners. Given their ability to infer and adapt to new contexts with minimal fine-tuning, models like GPT-2 have demonstrated that knowledge derived from one domain can provide valuable insights when wisely applied to another [Brown et al., 2020].

This thesis explores the potential of using pre-trained models for the classification and prediction of Dutch TTF Gas Imbalance, with the aim of improving trading decisions. The research investigates the possibility of using

existing knowledge in Natural Language Processing (NLP) and Computer Vision (CV) to understand complex patterns in the energy market's time-series data. This study also contributes to the ongoing discussion on the broader applications of foundational models [Dosovitskiy et al., 2021].

Incorporating Time-Series Data into Large Language Models

In order to adapt large language models (LLMs) to process time-series data, some techniques are required for effective data tokenization. [Nie et al., 2023] introduced Patch Time-Series Transformer (PatchTST), which provides strong evidence that treating multivariate time-series data as multiple, channel-independent univariate time series can lead to improved model performance. This channel-independent approach to patching allows the complex structure of multivariate data to be distilled into formats that a unified model such as the Transformer can process effectively.

With these insights, our method integrates temporal information while leveraging the advantages of patching and channel-independence. This is to allow for an interaction between time-series data and the pre-existing architecture of LLMs, hence gaining from the models' high capability for pattern recognition and prediction.

Utilizing Pre-Trained Large Language Models without Compromising Their Core Features

A principal challenge in applying pre-trained LLMs to new domains is to do so without undermining their original capabilities. As models like InstructGPT and ChatGPT, developed with techniques employed by [Ouyang et al., 2022], have demonstrated the effectiveness of aligning models with instruction-oriented data through a process of careful fine-tuning [Ouyang et al., 2022]. In consideration of these models' success, we propose a two-stage fine-tuning protocol for time-series data. Initially, we aim to introduce the LLM to time-series data using supervised fine-tuning. This step is followed by downstream fine-tuning focused specifically on time-series prediction and classification tasks.

Balancing adaptability to new data types and preservation of the LLMs' intrinsic attributes calls for sophisticated fine-tuning strategies. We approach this by incorporating Parameter-Efficient Fine-Tuning (PEFT) techniques,

such as Layer Normalization Tuning [Qi et al., 2022] and LoRA [Hu et al., 2022], which are designed to enhance the model's flexibility without overhauling the original parameters .

MTSPC-GPT: Proposed Approach to Time-Series Prediction and Classification Using GPT-2

The paper proposes a model fine-tuning the pre- trained GPT2 model for multitask learning, specifically for mixed time series prediction and classification, named MTSPC-GPT. MTSPC-GPT stands for Mixed Time Series Prediction and Classification using a Pretrained GPT-2 [Radford et al., 2019], which by uniting the patching technique, and PEFT techniques [Hu et al., 2022] with a pre-trained GPT-2 to effectively interpret complex multivariate time-series data like Dutch TTF Gas Balancing Signal. The intention is to leverage the sophisticated learning mechanisms of GPT-2 for revealing intricate patterns and dynamics within such data, which can play a pivotal role in forecasting and decision-making processes within the energy market and beyond.

Background and Motivation

Introduction to the Dutch Gas Market The Dutch gas market plays an important role in the European energy system, marked by its comprehensive gas network and substantial natural gas reserves. Historically, the market has depended on the Groningen field, but it is currently undergoing a transition due to the depletion of local resources and a growing global focus on sustainability. This changing landscape requires new strategies for gas balancing, which is important for ensuring the reliability and efficiency of the gas supply. [Kema/TPA, 2012].

Gas Balancing and Its Importance Gas balancing is essential to ensure that the supply of gas meets the demand at all times. Imbalances can lead to price volatility and can undermine the security of supply. Accurate forecasting and real-time adjustments are therefore pivotal in the Dutch gas market, particularly in light of fluctuating renewable energy production and variable consumption patterns [Kema/TPA, 2012].

Motivation Behind the Research The objective of this research is to improve the operational efficiency of gas market systems through the applic-

ation of advanced time-series prediction and visualization techniques. These enhancements aim to provide market participants with more comprehensive information, thereby supporting more informed decision-making processes for traders. Improvements in prediction and classification of gas imbalances by leveraging sophisticated machine learning methodologies such as ensemble tree methods, deep learning, and transformers can enhance the performance of predictive models. Ensemble tree methods, including bagging, boosting, and gradient boosting, combine multiple weak learners to create a stronger, more accurate model [Ke et al., 2017]. Deep learning, a subset of machine learning, uses artificial neural networks with multiple abstraction layers to process data and extract patterns for decision-making [Wei, 2019]. Transformers, a model architecture in natural language processing, have shown significant success in tasks requiring understanding of the context in a sequence of input data [Vaswani et al., 2017]. These advanced techniques, when appropriately applied, can significantly improve the accuracy and robustness of machine learning models, resulting in better decision-making, optimizing market performance and contributing to the overall stability of the energy grid [Kema/TPA, 2012].

Research Objectives

Goals and Questions of the Research The primary goal of this research is to evaluate the effectiveness of the MTSPC-GPT approach in predicting gas imbalances within the Dutch TTF Market. Specific research questions include:

- How does the MTSPC-GPT approach compare to traditional time series prediction models in terms of accuracy and efficiency?
- What are the benefits of multitask learning in the context of gas imbalance prediction?
- Can the inclusion of external features like weather data and power forecast data significantly improve the predictability of the Dutch TTF Gas Balancing Signal?
- What are the limitations of the MTSPC-GPT approach, and how can they be addressed in future research?

Contribution of the Thesis

The contributions of the thesis can be summarized as follows:

- It tackles the challenge of scarce large-scale training data for time series analysis by leveraging pre-trained models from the fields of language or computer vision, which have been trained on extensive datasets.
- It assesses the Frozen Pretrained Transformer (FPT), a model that utilizes the self-attention and feedforward layers of residual blocks from pre-trained models.
- It introduces MTSPC-GPT, which fine-tunes the pre-trained GPT2 model for multitask learning, aimed at mixed time series prediction and classification, allowing the model to simultaneously handle classification and forecasting tasks.
- It investigates various feature engineering techniques to improve the predictability of gas-related time series, such as the Dutch TTF Gas Balancing Signal, by incorporating external features like weather data, power forecast data, temporal features, and technical indicators.
- The research presents the integration of Time-Series with LLMs through tokenization techniques, utilizing patching to convert multi-dimensional time-series data into a format that can be processed efficiently by these models.
- It further outlines the universality of the proposed approach by validating it with a GPT2-Medium and GPT2-Small. This extension verifies that the methodology is not restricted to a single architecture but is versatile and can be adapted to other pre-trained models.
- Lastly, the paper provides empirical validation through evaluations on real-world datasets, specifically the Dutch TTF Balancing signal dataset and public dataset. The model's performance, with its accuracy in forecasting gas imbalance for predetermined future intervals.

Outline of the Thesis

The thesis is structured as follows:

- **Chapter 1: Introduction** - This chapter introduces the motivation behind the thesis and provides an in-depth understanding of the Dutch gas market, including its historical context and the current challenges in gas balancing.
- **Chapter 2: Related Work** - This chapter delves into the research related to gas-related features and time series models.
- **Chapter 3: Methodology** - This chapter discusses the empirical methodology, including the evaluation metrics used to assess the effectiveness of the model.
- **Chapter 4: Experiments** - This chapter presents the results of the experiments, including the data acquisition, data pre-processing and parameter optimization, offering insights into the model's performance and contrasting it with traditional classification methods.
- **Chapter 5: Conclusion** - concludes the thesis

1.1 Understanding the Dutch TTF Balancing Signal

Understanding the Dutch gas market, particularly the Dutch TTF Balancing signal dataset, is important for a comprehensive view of the Dutch gas market. This market's structure is influenced not only by economic and industrial demands but also by a variety of factors. The paper aims to examine the impact of the historical data of balancing signals, weather forecasts, changes in power and gas pricing, and variables such as wind power generation.

The following section provides an in-depth analysis of the Dutch TTF Balancing signal, examining its various components and how they combine to provide a perspective of the Dutch gas market. This analysis allows us to identify the market's distinctive characteristics, examine its current status and historical development, and consider the effects of external variables such as weather conditions and renewable energy outputs.

1.2 Dutch Gas Market

The Dutch gas market has a significant role in the European energy landscape, due to its former reliance on the Groningen gas field, known for its

low-calorific gas [Kema/TPA, 2012]. The Dutch high-pressure gas transmission grid connects the Netherlands with Belgium, Germany, and Denmark, and a North Sea pipeline link to the UK was established in 2006. This infrastructure facilitates the transportation of two gas qualities: G+ gas (a mixture of Groningen gas and high-calorific gas) and H+ gas.

Gasunie, a key player in the Dutch gas market, manages and maintains the country's gas infrastructure, ensuring the reliable supply of both G+ and H+ gas qualities to domestic and international consumers [Kema/TPA, 2012]. With a remarkably high natural gas penetration rate of 98% of households being connected, the Dutch gas market is highly concentrated, particularly for low-calorific gas.

One significant challenge in the Dutch gas market is the requirement for maintaining gas balancing. Shippers often lack the necessary information to manage imbalance risks effectively, and the existing imbalance charges do not always reflect the actual system imbalance [Kema/TPA, 2012]. These challenges have led to calls for a market-based balancing regime.

As a result, Gasunie Transport Services B.V. (GTS) allows consumers of G+ gas to transition to H gas. This decision aimed to enhance market flexibility and accommodate the changing dynamics of the Dutch gas market [Kema/TPA, 2012].

1.3 Gas Balancing in the Netherlands

To maintain a stable and reliable gas transport network in the Netherlands, the gas balancing rules and procedures are essential to ensure that the gas grid remains safe and efficient. This section provides an overview of how gas balancing works in the Netherlands and its significance.

Entry-Exit System and Capacity Reservations

In the Netherlands, gas transportation is organized through an entry-exit system. Shippers, which are entities responsible for moving gas within the network, can reserve capacity at entry and exit points using the PRISMA platform or Virtual Interconnection Points (VIP) such as THE(Trading Hub Europe) and TTF(Title Transfer Facility) [Capacity, 2023]. Entry capacity allows gas injection into the grid, while exit capacity permits gas withdrawal. Directly connected parties can book exit capacity provided they hold the necessary licenses. Shippers can organize these entry and exit points into

portfolios, manage them separately, and are ultimately responsible for ensuring the balance of their gas positions within these portfolios. This collective effort leads to a balanced network [Capacity, 2023].

System Balance Signal (SBS)

The SBS is formed by combining the Portfolio Imbalance Signals (POSs) of all participating shippers within the network. The POS represents the cumulative imbalance position of each shipper. When the network is within permitted limits (the dark green zone), no immediate action is required. However, if the network becomes imbalanced, balancing actions come into play.

In cases of imbalance, the national network operator may buy or sell gas to resolve the issue, with the costs caused by the parties responsible for causing the imbalance. Shippers submit daily forecasts, and some shippers are obligated to apply a damping formula, which will be discussed in later sections, to their forecasts. Near-real-time data assesses imbalances, which are shared through Portfolio Imbalance Signals (POS), leading to the activation of the System Balance Signal (SBS) if it's not at zero. Additionally, Within-Day Balancing Actions (WDBA) can be taken without adhering to a specific balancing period [Gasunie, 2023b].

As illustrated in Figure 1.1 from [Gasunie, 2023b], different degrees of network imbalance are distinguished and ranked from small to large as light-green zone imbalances (Im^L), orange zone imbalances (Im^O), and red zone imbalances (Im^R). These zones are determined based on surplus or shortage of gas measured in MWh over time.

The equations provided in Figure 1.1 indicate how these zones are calculated:

$$Im^L = SBS - GZL \text{ if } LGZL < |SBS| > GZL$$
$$Im^O = SBS - GZL \text{ if } OZL > |SBS| > LGZL$$
$$Im^R = SBS - GZL \text{ if } |SBS| > OZL$$

These equations help in determining whether immediate actions need to be taken for rebalancing or not.

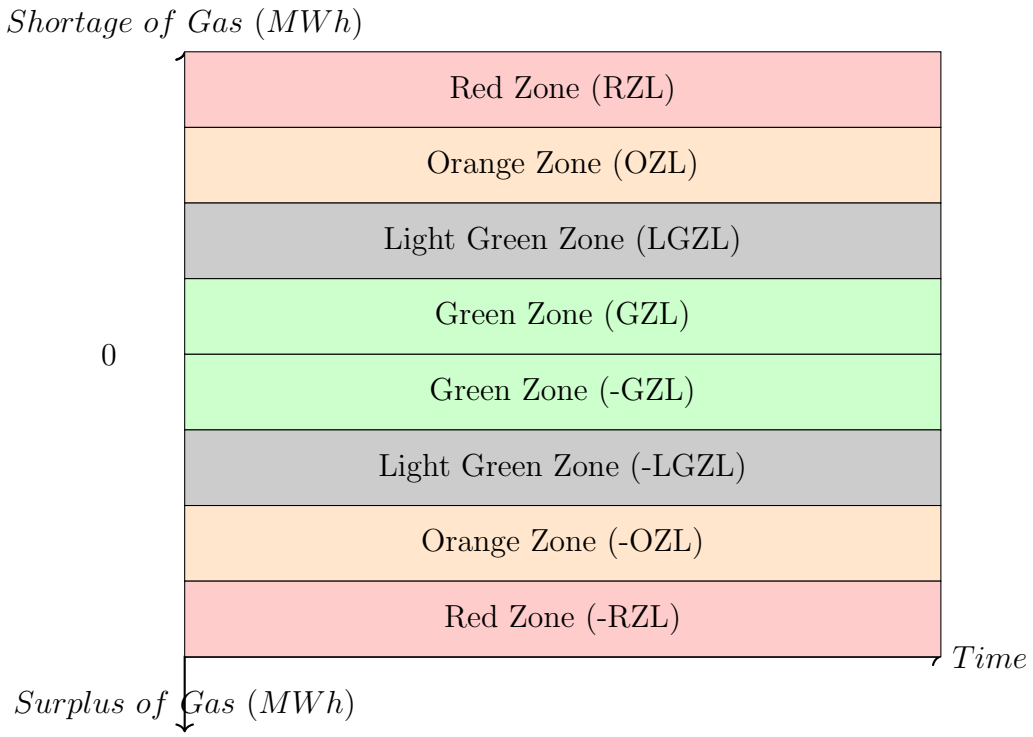

Figure 1.1: Different Degrees of Network Imbalance

Damping for Network Stability

Damping is introduced in gas balancing and designed to ensure the stability and reliability of the network. Without damping, changes in gas flow on the exit side of the network could lead to uncontrolled fluctuations on the entry side. This could potentially result in system instability and disruptions [Gasunie, 2023c].

Damping introduces controlled delays and adjustments in gas flows, helping to maintain a balanced and consistent gas supply. Over time, the sum of all entries and exits aligns, upholding system integrity and operational efficiency [Gasunie, 2023c].

The damping formula is given by:

$$E_d(h) = \alpha \cdot E_{\text{exit}}(h) + (1 - \alpha) \cdot E_d(h - 1)$$

where:
- $E_d(h)$ is the damped exit in the current hour,
- $E_{\text{exit}}(h)$ is the exit in the current hour,
- $E_d(h - 1)$ is the damped exit of the last hour,
- α is a damping factor.

For the first hour of the gas day, it applies that $E_d(h - 1) = E_{\text{exit}}(h)$.

The formula calculates the damped exit for a given hour, which is a weighted sum of the actual exit during that hour and the damped exit from the previous hour. The weight (α) determines how much influence each component has on the damped exit [Gasunie, 2023c].

Calculating Imbalances and SBS

GTS (Gasunie Transport Services B.V.) computes the final near-real-time imbalance for each portfolio per hour. This imbalance is added to the previous hour's Portfolio Imbalance Signal to determine the POS for the current hour. The SBS is then derived by summing up all POS values [Gasunie, 2023d].

Throughout the hour, the projected SBS value is recalculated every 5 minutes using anticipated POS values for that hour's end. This projected SBS aids in deciding whether a balancing action is necessary on the ICE Endex Gas Exchange. It also helps in assessing the current POS and whether any changes to planned nominations are needed [Gasunie, 2023d].

Shippers' Role in Balancing

In the gas balancing system, the aim is to be a "helper" rather than a "causer" of imbalances. When the system is in a "long" state, "long" portfolios are considered the "causers," while "short" portfolios are the "helpers." Conversely, in a "short" system state, "short" portfolios become the "causers," and "long" portfolios become the "helpers." The goal is always to be on the opposite side of the system's imbalance during balancing actions [Gasunie, 2023d].

Balancing Actions

Within-Day TTF Gas

When the gas network is in a state categorized as "light green zone" due to imbalance, Gasunie Transport Services B.V. (GTS) engages in buying or selling Within-Day (WD) TTF gas products on the ICE Endex Gas Exchange [Gasunie, 2023a]. This involves ensuring equal gas intake or delivery until the end of a specified period, typically starting from 4 hours after the given instruction hour (current hour +3).

For instance, if a balancing action is initiated at 13:20 hours, the lead time is 3 hours, and gas will be delivered or received from 17:00 hours to 06:00 hours of the following day. Within-Day TTF gas transactions play a pivotal role in addressing minor imbalances and maintaining network stability.

Next Hour TTF Gas

In cases where the gas network is in the "orange" or "red" zone, indicating a more significant imbalance, GTS engages in buying or selling Next Hour TTF gas during a balancing action [Gasunie, 2023a]. This involves gas intake or delivery for a single hour, commencing from the subsequent hour following the instruction.

For example, if a balancing action is executed at 13:20 hours, with no lead time (0-hour lead time), gas will be delivered or received from 14:00 hours to 15:00 hours. Next Hour TTF gas transactions are essential for rectifying substantial imbalances promptly.

Summary

Gas balancing in the Netherlands maintain a stable and efficient gas transport network. System Balance Signal (SBS) guides actions to keep the network within acceptable limits. Shippers, through responsible management and predictive strategies, contribute significantly to maintaining a stable gas transport network [Gasunie, 2023a].

Chapter 2

Related Work

2.1 Factors Affecting Gas balancing Signals

Gas balancing signals are critical indicators used by transmission system operators (TSOs) to ensure that the supply of natural gas matches the demand in real-time. These signals typically involve requests for adjustments in the output from various energy producing or consuming units. However, these signals are not immune to external influences. One significant factor that affect them are weather shock.

Impact of Weather Shocks on Gas balancing Signals

Weather conditions have a profound impact on both the supply and demand sides of natural gas markets. On the demand side, extreme temperatures can lead to higher consumption due to increased use of heating or cooling systems. On the supply side, weather can affect the production capabilities of renewable energy sources like wind and solar power. For instance, wind turbines produce less energy when wind speeds are low and solar panels are less effective during cloudy or rainy days [Lucidi et al., 2022].

Moreover, weather shocks such as storms or hurricanes can damage infrastructure, leading to a sudden loss of generating capacity. Hydropower generation is also vulnerable to droughts or excessive rainfall, which can disrupt water levels and flows [Tanaka et al., 2022].

Weather predictions, hence, are an integral part of managing balancing signals. Forecast errors can lead to significant imbalances, requiring TSOs to issue balancing signals more frequently and at higher magnitudes to correct for unforeseen weather-related disruptions [Ketterer, 2014].

2.2 Time Series Forecasting Model

Traditional Time Series Forecasting Models

Traditional models for time series forecasting in energy markets include autoregressive (AR), moving average (MA), and autoregressive integrated moving average (ARIMA) models. While AR models use a combination of past values of the variable, MA models use past forecast errors. The ARIMA model combines both approaches and includes differencing to make the time series stationary [Box et al., 2015].

Another traditional approach is the seasonal autoregressive integrated moving-average (SARIMA), which extends ARIMA to account for seasonality, which is a significant aspect in energy consumption and production [Noor et al., 2022].

These models have been widely used due to their simplicity, mathematical tractability, and the interpretability of their forecasting results. However, they often require the time series to be linear and stationary, and may not cope well with complex patterns or sudden changes, limiting their effectiveness in highly volatile energy markets.

Advanced Time Series Forecasting Models

Advancements in computational power and machine learning have led to the development of more sophisticated time series forecasting models. These include the vector autoregressive (VAR) model, which can capture the relationship between multiple interdependent time series, and state space models like the Kalman filter, which are ideal for time series that involve noise and other forms of uncertainty [Zivot and Wang, 2003].

Machine learning techniques such as artificial neural networks (ANNs), support vector machines (SVMs), and random forest (RF) approaches are being applied to forecast energy time series. These models can learn complex non-linear relationships and interactions in the data without the need for the series to be stationary or linear [Weron, 2014].

Deep learning models, particularly recurrent neural networks (RNNs) like long short-term memory (LSTM) networks, have shown great promise in capturing long-term dependencies and intricate patterns in time series data that traditional models may not [Hochreiter and Schmidhuber, 1997].

Hybrid models that combine traditional statistical methods with machine learning algorithms have also been developed to leverage the strengths of both

approaches. For example, an ARIMA model might be used in conjunction with an ANN to improve prediction accuracy for energy market time series [Zhang, 2003].

The continuous evolution of forecasting methods is essential to accommodate the increasing complexity and dynamism of energy markets, driven by factors such as growing renewable energy sources, market liberalization, and more active consumer participation.

2.3 Recent Advances in Time Series Forecasting Models

More Recent Advanced Time Series Forecasting Models

With the continuous evolution of machine learning, several advanced models have been proposed for time series forecasting. Here is a review of some cutting-edge transformer-based models:

N-HiTS and N-BEATS N-HiTS (Neural Hierarchical Interpolation for Time Series) and N-BEATS (Neural Basis Expansion Analysis for Time Series Forecasting) are neural network models designed with a focus on interpretability and scalability. N-BEATS, in particular, stands out for its stack of fully connected layers and backward and forward residual links. N-HiTS builds upon N-BEATS by structuring the model in a hierarchical manner to exploit multi-resolution representations of the input data [Oreshkin et al., 2020].

FEDformer and Autoformer FEDformer (Frequency Enhanced Decomposed Transformer) improves upon the traditional transformer architecture by enhancing the attention mechanism's sensitivity to frequency-domain information. Autoformer is another variant that introduces a novel autocorrelation mechanism, which aims to capture recurrent patterns in time series data effectively [Wu et al., 2022].

DLinear The DLinear model is a simple yet surprisingly effective approach for long-term time series forecasting (LTSF). It challenges the prevailing use of complex Transformer-based models for LTSF by demonstrating that a set of embarrassingly simple one-layer linear models, named LTSF-Linear, can outperform these sophisticated models across various real-life datasets. The

DLinear model is a variant of LTSF-Linear that incorporates a decomposition scheme to enhance performance when clear trends are present in the data. Specifically, DLinear decomposes the input data into trend and seasonal components using a moving average kernel, then applies separate one-layer linear models to each component before summing the outputs to produce the final prediction. This approach allows DLinear to effectively capture and leverage trend information in time series data, leading to improved forecasting accuracy. The paper's findings suggest that the temporal modeling capabilities of Transformers for time series analysis may be exaggerated and that simpler linear models like DLinear can serve as powerful baselines for future research in LTSF.[Zeng et al., 2022].

PatchTST The PatchTST model is a Transformer-based model designed for multivariate time series forecasting and self-supervised representation learning. It operates on two key components: segmentation of time series into subseries-level patches, which are used as input tokens to the Transformer, and channel-independence, where each channel contains a single univariate time series that shares the same embedding and Transformer weights across all the series[Nie et al., 2023]. The patching design has three main benefits. First, it retains local semantic information in the embedding. Second, it reduces the computation and memory usage of the attention maps quadratically given the same look-back window. Third, it allows the model to attend to a longer history. The channel-independent patch time series Transformer (PatchTST) can significantly improve long-term forecasting accuracy compared to state-of-the-art Transformer-based models. It also performs well in self-supervised pretraining tasks, outperforming supervised training on large datasets. Transferring of masked pre-trained representation on one dataset to others also produces state-of-the-art forecasting accuracy. The model works by dividing multivariate time series data into different channels, which share the same Transformer backbone, but the forward processes are independent. Each channel univariate series is passed through an instance normalization operator and segmented into patches. These patches are used as Transformer input tokens. For self-supervised representation learning, patches are randomly selected and set to zero, and the model reconstructs the masked patches. [Nie et al., 2023].

TimeNet Time series analysis is crucial across various domains, including weather forecasting, anomaly detection, and action recognition. Traditional

methods struggle with the complex temporal patterns in 1D time series data. The TimesNet model, introduced by Wu et al.[Wu et al., 2023], addresses these challenges by leveraging the observation of multi-periodicity in time series. It transforms 1D time series into 2D tensors to model intra- and inter-period variations using 2D kernels, overcoming the limitations of 1D representation. The core of TimesNet, the TimesBlock, adaptively discovers multi-periodicity and extracts temporal variations from the transformed 2D tensors through a parameter-efficient inception block. This innovative approach allows TimesNet to achieve state-of-the-art performance in five mainstream time series analysis tasks, including forecasting, imputation, classification, and anomaly detection. The model's architecture and functionality demonstrate its effectiveness in capturing both short-term and long-term dependencies, showcasing its potential as a foundational model for time series analysis.

GPT4TS The GPT4TS model represents a significant advancement in leveraging pre-trained models for time series analysis across a variety of tasks. This model, based on the concept of Frozen Pretrained Transformer (FPT), utilizes the architecture and parameters of pre-trained language or computer vision models, without altering the self-attention and feedforward layers of the residual blocks. The primary innovation lies in the application of these pre-trained models, originally designed for NLP or CV tasks, to time series analysis by fine-tuning them on specific time series tasks. The paper demonstrates that this approach can achieve state-of-the-art or comparable performance across all major types of time series analysis tasks, including classification, anomaly detection, forecasting, and few-shot learning. The effectiveness of the GPT4TS model is attributed to the self-attention mechanism's ability to behave similarly to principal component analysis (PCA), which facilitates the bridging of domain gaps and underscores the universality of pre-trained transformers. Extensive experiments validate the model's performance, showing significant improvements over traditional methods and other deep learning approaches in tasks such as forecasting, imputation, and anomaly detection. The GPT4TS model's success in leveraging cross-modality transferred knowledge for time series analysis without substantial modifications to the pre-trained models underscores its potential as a versatile and powerful tool for time series analysis. [Zhou et al., 2023].

2.4 Transfer Learning in Time-Series Analysis

In-Modality Transfer Learning via Pre-Trained Models

Recent advancements have underscored the efficacy of pre-trained models across various domains, from natural language processing (NLP) and computer vision (CV) to vision-and-language (VL) tasks. In NLP, the focus has been on developing contextual word embeddings that enhance performance on downstream tasks. The advent of deep transformer models, powered by increased computational capabilities, has significantly improved representation capabilities across a myriad of language tasks. Notably, BERT [Devlin et al., 2019] utilizes transformer encoders and a masked language modeling task to predict randomly masked tokens within a text, demonstrating the model's robust representation ability. Similarly, OpenAI's GPT [Radford and Narasimhan, 2018] and its successor, GPT2 [Radford et al., 2019], leverage transformer decoders trained on extensive language corpora, showing remarkable transferability to various downstream tasks. This concept of pre-training has also been successfully applied to CV tasks, with models like DEiT[Touvron et al., 2021] employing a teacher-student strategy using convolutional neural networks (CNNs) as the teacher, and BEiT[Bao et al., 2022] adapting the BERT model for visual tasks by converting images into visual tokens. Despite these successes, the application of pre-trained models to general time series analysis remains limited, primarily due to challenges in training sample availability and the need for domain-specific models for tasks such as classification, anomaly detection, and forecasting.

Evidence from Recent Studies

The Frozen Pretrained Transformer (FPT) has showcased it can be effectively adapted for time series analysis, achieving comparable or superior performance across major time series tasks without altering the model's core layers [Zhou et al., 2023]. The integration of pre-trained models into time series analysis, although in its nascent stages, shows promising potential. By leveraging the foundational principles established in NLP and CV, these models can significantly enhance the efficiency and effectiveness of time series analysis across a broad spectrum of tasks, marking a pivotal step towards the universal applicability of pre-trained models [Zhou et al., 2023].

Cross-Modality Knowledge Transfer

Transformers have demonstrated their capability to manage tasks across different modalities by converting input data into embeddings. This raises the intriguing question of whether transformers possess a universal representation capability that could facilitate knowledge transfer across various domains. The VLMo model, introduced by Bao et al.[Bao et al., 2021], employs a novel stagewise pre-training approach that leverages attention blocks, pre-trained solely with image data, to enhance language processing capabilities. A significant study by Lu et al[Lu et al., 2022]. explores the efficacy of utilizing a frozen, pre-trained language model for knowledge transfer across different domains, showcasing its superior performance compared to an end-to-end transformer model trained on data from those domains. Another paper, Voice2series[Yang et al., 2021], applies a pre-trained speech processing model to time series classification, achieving remarkable results. Despite these advancements, research on cross-modality knowledge transfer, especially for time series forecasting and general time series analysis, remains scarce.

Parameter-Efficient Fine-Tuning

Parameter-efficient fine-tuning (PEFT) techniques have been proposed in both Natural Language Processing (NLP) and Computer Vision (CV) to fine-tune fewer parameters in various downstream tasks. The primary objective of these techniques is to reduce computational costs by fine-tuning a smaller set of parameters, while still achieving or even surpassing the performance of full fine-tuning. Notable examples of these techniques include the adapter method[Houlsby et al., 2019], which introduces small modules between transformer layers, and prefix tuning, which adds tunable prefixes to the keys and values of the multi-head attention at every layer[Li and Liang, 2021]. Another technique, Low-Rank Adaptation (LoRA)[Hu et al., 2022], injects trainable low-rank matrices into transformer layers to approximate the weight updates. A unified view of these PEFT methods is provided in the literature [He et al., 2022].

2.5 Time Series Classification

Time Series Classification (TSC) remains a pivotal task across numerous domains such as finance, healthcare, and engineering, where the goal is to categorize time-series data into distinct classes based on their temporal features. Researchers have developed various models to address this problem, each with their unique architectures tailored for capturing temporal dependencies within data. This paper provides an overview of the current state of research in TSC, examining several prominent models including Classical Recurrent Neural Networks (RNNs), Temporal Convolutional Networks (TCNs), Transformers, ROCKET, LSTNet, AutoFormer, Non-stationary Transformers, FEDFormer, ETSFormer, FlowFormer, DLinear, TimesNet and GPT4TS alongside discussing their application domains, performance, strengths, and weaknesses.

Classical Recurrent Neural Networks (RNNs)

RNNs have traditionally been the cornerstone model for TSC. These networks, capable of capturing sequential information through their recurrent connections, perform exceptionally in tasks where past information is necessary for current prediction [Lipton et al., 2015]. A known weakness, however, is their difficulty managing the vanishing and exploding gradient problems, hindering the learning of long-term dependencies [Pascanu et al., 2013].

Tree models

Tree models serve as a useful baseline in time series classification due to their interpretability and ability to handle tabular data effectively. When time series data is compressed into technical indicators, it essentially transforms into a tabular format with historical features from $t-N$ to $t-1$ incorporated into the t timeframe. Tree models can handle this type of data well, as they can easily manage different types of variables and deal with missing values.

Moreover, tree models are known for their interpretability, which is a crucial aspect when dealing with time series data. They provide clear decision rules and allow for an understanding of which features are important for classification. This is particularly useful in time series classification where understanding the impact of past events on future predictions is important.

According to [Grinsztajn et al., 2022] tree models often outperform deep learning models on tabular data. This is because deep learning models, while

powerful, often require large amounts of data and extensive tuning to perform well. On the other hand, tree models are simpler and more efficient, making them a good choice for a baseline model.

In terms of accuracy, while deep learning models have shown significant improvements in time series classification[Zheng et al., 2014], tree models still hold their ground as a reliable and interpretable method for time series classification. Therefore, using a tree model as a baseline provides a solid starting point for time series classification tasks.

LightGBM LightGBM is a gradient boosting decision tree algorithm. It works by training a series of weak decision tree models in a sequential manner, where each subsequent model attempts to correct the errors made by the previous one. The model uses two novel techniques to improve efficiency and scalability: Gradient-based One-Side Sampling (GOSS) and Exclusive Feature Bundling (EFB). GOSS excludes a significant proportion of data instances with small gradients, using the remaining instances to estimate the information gain. This is based on the observation that data instances with larger gradients play a more important role in the computation of information gain. EFB, on the other hand, bundles mutually exclusive features (those that rarely take nonzero values simultaneously) to reduce the number of features. This is done using a greedy algorithm, as finding the optimal bundling of exclusive features is NP-hard. These techniques allow LightGBM to speed up the training process of conventional gradient boosting decision tree models by up to over 20 times while achieving almost the same accuracy. [Ke et al., 2017]

CatBoost CatBoost addresses the shortcomings of existing gradient boosting algorithms, particularly in handling categorical features and preventing prediction shift due to target leakage. The model works by constructing an ensemble of decision trees in a sequential manner, where each tree is fitted to the residual errors made by the previous trees. CatBoost introduces two key innovations: ordered boosting and an algorithm for processing categorical features. Ordered boosting is a permutation-driven approach that avoids the target leakage problem by ensuring that the model for each training example is built without using that example's target value. The algorithm for categorical features uses target statistics, which are calculated without including the target value of the example being processed, thus preventing leakage and ensuring unbiased predictions. These techniques allow CatBoost

to outperform other gradient boosting implementations on various datasets by effectively handling categorical data and avoiding prediction shift, leading to more accurate and reliable models. [Prokhorenkova et al., 2019]

Temporal Convolutional Networks (TCN)

TCNs offer a compelling alternative to RNNs, employing stacked dilated convolutions to enable the network to view larger receptive fields, thus better learning long-range temporal correlations [Bai et al., 2018]. TCNs are known for their parallelism and stable gradients, standing as a more efficient and effective model for long sequences.

Random Convolutional Kernel Transform (ROCKET)

ROCKET stands out by using random convolutional kernels to transform time series data into feature representations that are then utilized by linear classifiers [Dempster et al., 2020]. Its strengths lie in its simplicity and execution speed, achieving comparable or superior performances to deep learning methods with lesser computational overhead.

Long- and Short-term Time-series network (LSTNet)

LSTNet combines convolutional neural networks (CNNs) and RNNs to simultaneously capture short-term and long-term patterns in time-series data [Lai et al., 2018]. The model has been shown to be effective in multivariate time series forecasting, although its complex architecture can lead to longer training times.

Autoformer

Autoformer: Decomposition Transformers with Auto-Correlation for Long-Term Series Forecasting, addresses the challenge of long-term forecasting in time series data. Traditional Transformer-based models, while effective in capturing long-range dependencies through self-attention mechanisms, struggle with the intricate temporal patterns present in long-term forecasting scenarios. These patterns can obscure reliable dependencies, and the computational cost of Transformers scales quadratically with the sequence length, leading to efficiency bottlenecks. *Autoformer* innovates by incorporating a novel decomposition architecture that integrates series decomposi-

tion directly into the model, moving beyond the conventional pre-processing approach. This design allows the model to progressively decompose and refine complex time series data, enhancing its ability to handle intricate temporal patterns. Additionally, the *Autoformer* introduces an Auto-Correlation mechanism inspired by stochastic process theory. This mechanism focuses on the periodicity of the series, enabling the model to discover dependencies and aggregate representations at the sub-series level more efficiently and accurately than traditional self-attention mechanisms. The Auto-Correlation mechanism operates with a complexity of $O(L \log L)$ for a series of length L, significantly improving both computational efficiency and information utilization. The *Autoformer* model demonstrates state-of-the-art accuracy in long-term forecasting, achieving a 38% relative improvement on six benchmarks across various practical applications, including energy, traffic, economics, weather, and disease forecasting. In summary, the *Autoformer* model addresses the limitations of previous Transformer-based models in long-term time series forecasting by introducing a decomposition architecture and an Auto-Correlation mechanism. These innovations allow the model to efficiently and accurately capture long-term dependencies and intricate temporal patterns, leading to significant improvements in forecasting performance across a range of applications[Wu et al., 2022].

FEDformer

The *FEDformer*, or *Frequency Enhanced Decomposed Transformer*, is a novel method for long-term series forecasting that combines Transformer with a seasonal-trend decomposition method. This approach addresses the computational expense of Transformer-based methods and their inability to capture the global view of time series. The decomposition method captures the global profile of time series while Transformers capture more detailed structures. The *FEDformer* also exploits the fact that most time series have a sparse representation in well-known bases such as Fourier transform, developing a frequency enhanced Transformer. This method is more efficient than the standard Transformer, with a linear complexity to the sequence length. Empirical studies with six benchmark datasets show that *FEDformer* can reduce prediction error by 14.8% and 22.6% for multivariate and univariate time series, respectively[Zhou et al., 2022].

Non-stationary Transformers

Non-stationary Transformers is a novel framework for time series forecasting that addresses the issue of over-stationarization in Transformer-based models. Over-stationarization occurs when stationarization methods, used to make time series data more predictable, remove too much of the inherent non-stationarity of the data, leading to less instructive forecasting of real-world events. The proposed framework consists of two interdependent modules: Series Stationarization and De-stationary Attention. Series Stationarization normalizes each input series and restores the output with original statistics for better predictability. De-stationary Attention, on the other hand, recovers the intrinsic non-stationary information into temporal dependencies by approximating distinguishable attentions learned from raw series. The Non-stationary Transformers framework significantly improves the performance of mainstream Transformers, reducing Mean Squared Error (MSE) by 49.43% on Transformer, 47.34% on Informer, and 46.89% on Reformer, making them the state-of-the-art in time series forecasting [Liu et al., 2023].

ETSformer

The ETSformer is a novel time-series Transformer architecture designed for time-series forecasting. It leverages the principle of exponential smoothing methods to improve the performance of Transformers for time-series data. The ETSformer introduces a level-growth-seasonality decomposed Transformer architecture, which leads to more interpretable and disentangled decomposed forecasts. It also proposes two novel attention mechanisms - the exponential smoothing attention and frequency attention, which are designed to overcome the limitations of the traditional attention mechanism for time-series data. The model works by first extracting global periodic patterns as seasonality, and subsequently extracting growth as the change in level in an exponentially smoothed manner. The final forecast is a composition of level, growth, and seasonal components, making it human interpretable. The model has been validated through extensive experiments on the long sequence time-series forecasting benchmark [Woo et al., 2022].

FlowFormer

FlowFormer is a transformer-based neural network architecture designed for learning optical flow, which is the pattern of apparent motion of objects, sur-

faces, and edges in a visual scene caused by the relative motion between an observer and the scene. FlowFormer works by tokenizing the 4D cost volume built from an image pair, encoding these cost tokens into a cost memory with alternate-group transformer (AGT) layers in a novel latent space, and then decoding the cost memory via a recurrent transformer decoder with dynamic positional cost queries. The model has demonstrated strong performance on the Sintel benchmark, achieving significant error reduction and strong generalization performance. The model's architecture includes a cost volume encoder that embeds the 4D cost volume into a latent cost space and fully encodes the cost information in such a space, and a recurrent cost decoder that estimates flows from the encoded latent cost features. The model is designed to effectively process cost volumes, which are compact yet rich representations widely explored in optical flow estimation communities, for estimating accurate optical flows [Huang et al., 2022].

Chapter 3

Methodology

Model Structure

The architecture for time series classification is designed to leverage the strengths of pre-trained models from other domains, such as natural language processing (NLP), and adapt them for time series data. The model structure incorporates elements from well-established transformer models, including GPT2 [Radford et al., 2019], BERT [Devlin et al., 2019], and BEiT [Bao et al., 2022], to explore the effectiveness of cross-domain knowledge transfer.

The proposed model's architecture is depicted in 3.1, focusing on classifying and forecasting the inputted time series. This series undergoes instance normalization and patching before being processed by the TS input embedding module to generate primary tokens for transformers. Our model incorporates a Frozen Pretrained Block, preserving the positional embedding layers and self-attention blocks from pre-trained models. We freeze the self-attention blocks during fine-tuning but fine-tune the positional embeddings and layer normalization layers to enhance downstream tasks with minimal effort, following standard practices by [Lu et al., 2022] and [Houlsby et al., 2019]. Additionally, we redesign and train the input embedding layer to project the time-series data to the dimensions required by the specific pre-trained model, employing linear probing to reduce the training parameters. This approach allows us to leverage the learned knowledge from pre-trained language models efficiently, by concatenating learnable prompts from a frequency adapter with input tokens and retraining certain components during fine-tuning.

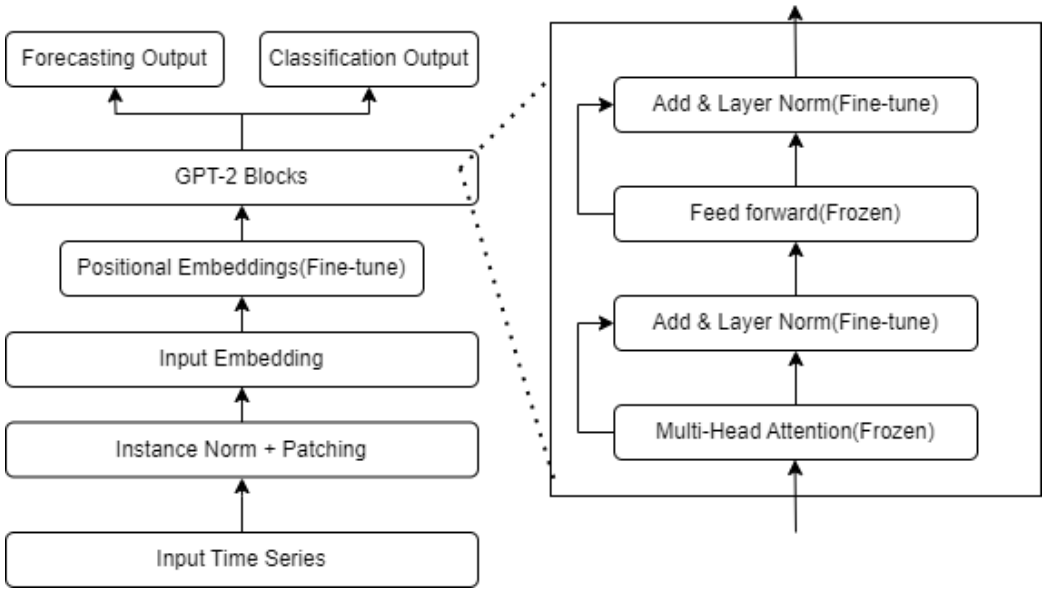

Figure 3.1: Model architecture. The self-attention and feedforward layers within the transformer blocks are kept frozen, while the embedding layer, normalization layers, and output layer are the only components that undergo training.

3.1 Instance Normalization

Instance Normalization has recently been proposed to help mitigate the distribution shift effect between the training and testing data [Ulyanov et al., 2017, Kim et al., 2022]. This process is essential for the performance of pre-trained models. We use a data normalization block, known as non-affine reverse instance norm, to enhance knowledge transfer by normalizing input time series based on their mean and variance, and then reintroducing these parameters in the output. The normalization for a univariate time series X with mean Exp and standard deviation Var is given by

$$\hat{X} = \frac{X - \text{Exp}}{\sqrt{\text{Var} + \epsilon}}.$$

3.2 Patching

Forward Process

The paper denotes the i-th univariate series of length L, starting at time index 1, as $x_{1:L}^{(i)} = (x_1^{(i)}, \ldots, x_L^{(i)})$, where $i = 1, \ldots, M$. The input $(x_1, \ldots, x_L)$ is then split into M univariate series $x^{(i)} \in \mathbb{R}^{1 \times L}$. Each of these univariate series $x^{(i)}$ is independently fed into the Transformer backbone according to the channel-independence setting. The Transformer backbone produces prediction results $x^{\wedge(i)} = (\hat{x}_{L+1}^{(i)}, \ldots, \hat{x}_{L+T}^{(i)}) \in \mathbb{R}^{1 \times T}$ accordingly.

Patching

Each input univariate time series $x^{(i)}$ is first divided into patches, which can be either overlapped or non-overlapped. Denoting the patch length as P and the stride - the non-overlapping region between two consecutive patches as S, the patching process generates a sequence of patches $x_p^{(i)} \in \mathbb{R}^{P \times N}$ where N is the number of patches, $N = \left\lfloor \frac{(L-P)}{S} \right\rfloor + 2$. Here, The paper pads S repeated numbers of the last value $x_L^{(i)} \in \mathbb{R}$ to the end of the original sequence before patching.

With the use of patches, the number of input tokens can be reduced from L to approximately L/S. This implies the memory usage and computational complexity of the attention map are quadratically decreased by a factor of S. Thus, constrained on the training time and GPU memory, the patch design allows the model to see the longer historical sequence, which can significantly improve the forecasting performance[Nie et al., 2023].

3.3 Frozen Pre-trained Block

The concept of a "Frozen Pre-trained Block" refers to a scenario where certain layers of a pre-trained model are kept unchanged during the fine-tuning process on a new task. This approach is predicated on the hypothesis that the pre-trained layers already encapsulate valuable knowledge that can be leveraged for the new task, thereby remove the need for additional training. Experiments with BERT-frozen and BEiT-frozen models carried out in [Zhou et al., 2023], which were initially pre-trained on images, were conducted to assess this hypothesis by examining the models' proficiency in transferring their pre-acquired knowledge to disparate domains. The favorable out-

comes of these experiments indicate that the capacity for knowledge transfer is not an exclusive attribute of GPT2-based language models but rather a pervasive characteristic of various pre-trained models [Zhou et al., 2023]. This discovery is important because it suggests that pre-trained models can be easily adapted for novel tasks with relative ease. In the fine-tuning phase for downstream tasks, only the layer normalization layers are fine-tuned, which is a widely accepted practice [Lu et al., 2022]. These layers are instrumental in stabilizing the learning process and can be swiftly adjusted to accommodate new data, thereby ameliorating the model's performance on the target task with minimal supplementary training.

3.4 Input embedding

To adapt a pre-trained NLP model to handle time-series data, we use the input embedding layer. This layer is tasked with transforming the time-series input into a format that aligns with the model's expected input structure. To achieve this, the embedding layer must project the data into the appropriate dimensional space required by the pre-trained model [Houlsby et al., 2019]. One effective strategy for this is linear probing, which not only facilitates the necessary dimensionality transformation but also streamlines the model by reducing the overall number of trainable parameters. This approach is particularly beneficial when adapting models to new tasks or data modalities, as it allows for leveraging the knowledge encapsulated in pre-trained models while minimizing the computational burden associated with training large numbers of parameters from scratch [Houlsby et al., 2019].

3.5 Loss Function

The paper introduces a dual-target training approach, which caters to both regression and classification tasks within a single model framework. This strategy leverages the strengths of two distinct loss functions: Mean Squared Error (MSE) for regression and Cross-Entropy (CE) for classification. The MSE loss quantifies the discrepancy between the predicted values and the actual ground truth for continuous outcomes, making it ideal for regression tasks. On the other hand, the CE loss is employed for classification problems, where it measures the performance of the model in correctly predicting the class of each input by comparing the predicted class probabilities with the

actual class labels.

The integration of these two loss functions into a unified model is achieved through a weighted combination, allowing the model to simultaneously learn from both regression and classification targets. This approach is formalized in the overall loss function as follows:

$$\text{Loss} = (1 - w) \cdot \text{Loss}_{\text{MSE}} + w \cdot \text{Loss}_{\text{CE}} \tag{3.1}$$

where w is a weighting factor that balances the contribution of each loss component to the overall training objective. The MSE component of the loss is defined as:

$$\text{Loss}_{\text{MSE}} = \frac{1}{M} \sum_{i=1}^{M} \frac{1}{T} \sum_{t=L+1}^{L+T} \|\hat{x}_t^{(i)} - x_t^{(i)}\|_2^2 \tag{3.2}$$

and the CE component is given by:

$$\text{Loss}_{\text{CE}} = -\sum_{c=1}^{C} y_{o,c} \log(p_{o,c}) \tag{3.3}$$

In this formulation, M represents the number of time series or samples, T is the length of each time series, $\hat{x}_t^{(i)}$ denotes the predicted value, $x_t^{(i)}$ is the true value, C is the number of classes, $y_{o,c}$ is a binary indicator (0 or 1) if class label c is the correct classification for observation o, and $p_{o,c}$ is the predicted probability that observation o belongs to class c.

This multi-target training approach, by employing both MSE and CE losses, enables the model to effectively handle both tasks with sharing knowledge [Luo et al., 2023].

Chapter 4

Experiments

Our proposed methods were evaluated using two distinct datasets. The first is the Gas Balancing Signal Dataset. Due to company policy, the source code about this dataset cannot be disclosed. However, it provided a robust platform for testing our methods in a real-world, industry-specific context. The second dataset used for evaluation is publicly available, allowing for transparency and reproducibility in our testing procedures.

4.1 Gas Imbalance Dataset Description

Understanding Gas Imbalance Prediction

As shown in Figure 4.1 illustrates the time series of the gas balancing signal SBS, which is a critical factor in managing gas distribution systems. The graph plots the gas surplus against time (in hours), with the blue line representing the SBS time series. The objective of the paper is to develop a predictive model that can forecast gas imbalance events at least 3 hours in advance.

Defining Gas Imbalance

Gas imbalance occurs when the absolute value of the SBS exceeds a predefined threshold, denoted as LGZ (Light Green Zone)(as shown in Figure 4.1). This threshold is represented by the dashed horizontal lines labeled as `+LGZ` and `-LGZ` on the graph. The points where the SBS time series intersects these thresholds are of particular interest, as they indicate an imbalance event. These points are highlighted with red dots on the graph.

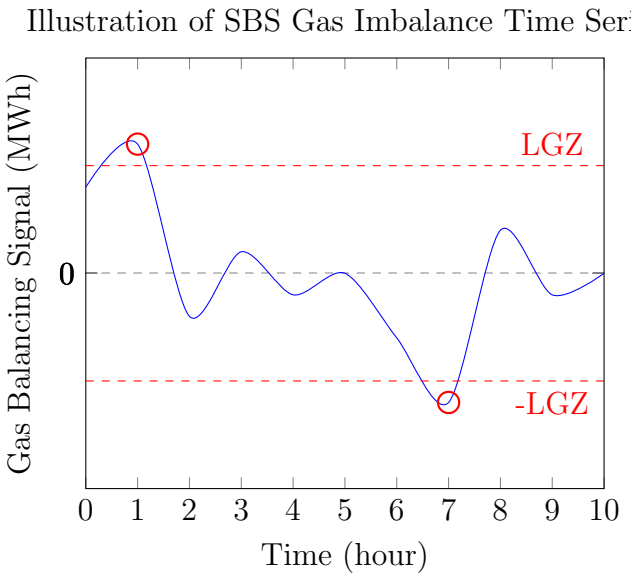

Figure 4.1: Simple illustration of the time series of the gas balancing signal SBS, with the threshold levels indicated by the dashed red lines representing ±LGZ. The instances where the SBS signal crosses these thresholds, indicating a gas imbalance event, are highlighted with red circles.

Defining Objective

The study aims to forecast the gas imbalance events with a lead time of at least three hours. This would mitigate the effects of these imbalances on the gas distribution system. To achieve this, we will utilize a variety of time series forecasting methodologies and machine learning algorithms, including LightGBM classification, to analyze historical data and identify patterns that predict imbalance events.

Switching Objective to Classification

Initially, the focus was on a prediction task, but it was observed that the model heavily relied on autocorrelation for its predictions. This implies that the historical values of the time series itself were the most significant features, leading to a lack of forecasting power for outlier events. However, further analysis revealed that other features, such as seasonality and weather forecast data, also play significant roles. This is particularly important when the goal is to predict outliers, such as spikes in Gas Balancing Signal. The variability in short-term gas supply and demand, often triggered by factors such as

weather forecast inaccuracies or unforeseen outages, can lead to outliers in data analysis. To address this challenge, we have shifted our approach from forecasting to classification. Specifically, we aim to predict whether there will be a gas imbalance within the next 5 to 12 hours. For this purpose, we classify the outcome as '1' if an imbalance is anticipated, and '0' if not.

Multi-task learning

The proposed model aims to accomplish both classification and forecasting tasks, embodying a multi-task prediction approach. This approach is supported by various studies that demonstrate improved performance when employing multi-task learning. For instance, a study on atmospheric particulate matter prediction utilized attentive multi-task learning to predict air quality in cities. The study found that the multi-task learning model outperformed single-task models and other state-of-the-art methods in terms of accuracy performance [Song et al., 2022]. Similarly, a study on Alzheimer's disease progression prediction employed a high-order multi-task learning model to explore temporal correlations in imaging and cognitive data. The model's sparsity allowed for the selection of a small number of imaging measures while maintaining high prediction accuracy[Wang et al., 2012]. Another study in the intensive care unit setting used a flexible Transformer-based model to accurately predict seven clinical outcomes related to readmission and patient mortality over multiple future time horizons[Shickel et al., 2021]. These studies collectively show the potential of multi-task learning in enhancing the performance of predictive models.

Data Preprocessing

The dataset under consideration comprises the Gas Balancing Signal, which is recorded at 5-minute intervals. This data starts from August 2016, to January 2024. We have transformed the raw time series into an Open-High-Low-Close (OHLC) format, as shown in Figure4.2, and resampled the time series into 1-hour intervals. The OHLC structure is also known as a candlestick, or when referring to multiple such structures, a candlestick pattern. This terminology is derived from the visual resemblance of the data structure to a candle, where the body of the candlestick is defined by the range between the opening and closing prices, and the shadows represent the high and low prices in relation to the body's top or bottom. A black-bodied candlestick indicates a price decrease, while a white-bodied candlestick signifies a price

increase. This resampling technique is widely adopted in financial time series analysis and is particularly beneficial for the following reasons:

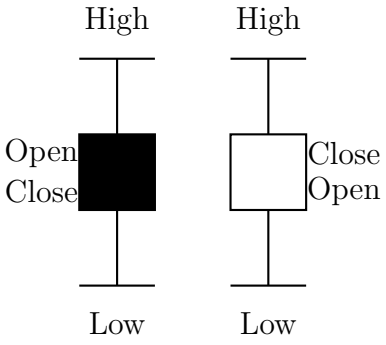

Figure 4.2: OHLC data structures

- **Dimensionality Reduction:** The OHLC format effectively reduces the dimensionality of the dataset by summarizing the data into four points per specified time frame, which in our case is daily. This reduction simplifies the dataset while retaining critical information about the price movements within each interval.
- **Feature Enrichment:** By converting the time series into OHLC data, we can leverage a variety of technical indicators that are commonly used in the analysis of financial markets. These indicators serve as powerful features for predictive models as they encapsulate key aspects of the data's trend, momentum, volatility, and volume.

Technical indicators

Technical indicators are particularly useful in this context as they are designed to signal potential turning points in the data or to confirm the strength of a particular trend, e.g. Moving Averages (MA), Relative Strength Index (RSI), and Bollinger Bands (BBAND), which can be calculated from the OHLC data. These indicators have been shown to be effective in various domains, including energy markets, for both prediction and classification tasks [Demir et al., 2020]. Here are the general overview of some technical indicators we used, including their descriptions and formulas:

Moving Average (MA)

The Moving Average smooths the price data and help identify the direction of the trend. The formula for a Simple Moving Average (SMA) is:

$$\text{SMA} = \frac{\sum_{i=1}^{n} \text{Price}_i}{n}$$

where n is the number of periods.

Moving Average Convergence Divergence (MACD)

MACD is a trend-following momentum indicator that shows the relationship between two moving averages. It is calculated by subtracting the 26-period Exponential Moving Average (EMA) from the 12-period EMA.

$$\text{MACD} = \text{EMA}_{12} - \text{EMA}_{26}$$

Relative Strength Index (RSI)

The RSI measures the speed and change of price movements. It oscillates between 0 and 100. Traditionally, and according to [J. W. Wilder, 1978], RSI is considered overbought when above 70 and oversold when below 30.

$$\text{RSI} = 100 - \frac{100}{1 + \text{RS}}$$

where RS is the average gain of up periods during the specified time frame divided by the average loss of down periods.

Bollinger Bands (BB)

Bollinger Bands are consists of a middle band being an N-period simple moving average (SMA), an upper band at K times an N-period standard deviation above the middle band, and a lower band at K times an N-period standard deviation below the middle band.

$$\text{Middle Band} = \text{SMA}(N)$$

$$\text{Upper Band} = \text{SMA}(N) + K \times \text{SD}(N)$$

$$\text{Lower Band} = \text{SMA}(N) - K \times \text{SD}(N)$$

where SD is the standard deviation over the last N periods, N is the period of the SMA, and K is the number of standard deviations from the SMA.

Rate of Change (ROC)

The Rate of Change indicator measures the percentage change in price between the current price and the price a certain number of periods ago.

$$\text{ROC} = \left(\frac{\text{Current Price} - \text{Price}_n \text{ periods ago}}{\text{Price}_n \text{ periods ago}} \right) \times 100$$

These indicators are widely used in technical analysis to predict future market movements based on historical prices and volumes. Each has its own strengths and can be used in conjunction with others to develop trading strategies.

Benefits of using technical indicators:

- **Stationary Features:** Technical indicators such as RSI can transform raw financial data into features that are more stationary. Stationarity is a desirable property in time series modeling, including ML applications, because it implies that the statistical properties of the series (mean, variance) do not change over time. This can make models more stable and predictions more reliable[Yıldırım et al., 2021].

- **Historical Context:** Indicators like RSI provide a way to encapsulate historical price movements and market sentiment into a single, quantifiable measure. This historical context can be crucial for ML models to identify patterns or trends in the data that are predictive of future price movements[Ntakaris et al., 2020].

- **Data Compression:** By summarizing past price actions into a single value, technical indicators effectively compress data. This can reduce the dimensionality of the problem space, making ML models more computationally efficient and potentially reducing the risk of overfitting. It allows models to focus on key features that encapsulate significant market information[Ntakaris et al., 2020].

- **Pattern Recognition and Prediction Enhancement:** Incorporating technical indicators as features in ML models can enhance the

model's ability to recognize patterns and improve prediction accuracy. For instance, RSI can help identify overbought or oversold conditions, aiding in predicting market movements and optimizing trading strategies[Yıldırım et al., 2021].

- **Versatility in Market Analysis:** Technical indicators are versatile tools that can be adapted for use in various market conditions. They can signal potential trend reversals, momentum, and volatility, which are critical aspects for successful trading strategies when combined with ML models[Yıldırım et al., 2021].

Weather Shocks Information

In addition to the Gas Balancing Signal, the paper incorporates a comprehensive set of exogenous variables that are likely to influence the gas imbalance. These variables include weather shocks information and energy-related data, which are crucial for creating a robust predictive model.

Weather conditions have a significant impact on short term gas demand and supply[Aslam et al., 2020], and therefore, they are essential predictors for gas imbalance. We have included weather-related features such as temperature, moisture, wind, and cloud cover for key locations that influence the gas market dynamics, namely London, Berlin, Groningen, and Amsterdam. These features are expected to capture the variability in gas usage due to weather fluctuations.

Energy Market Data

The integration of renewable energy sources into the power grid introduces additional variability that can affect the gas imbalance. To account for this, we have included the wind power generation forecast change for Germany and the Netherlands. These forecasts provide insights into the expected availability of wind energy, which can substitute or supplement gas-fired power generation.

Power Price

The power price is a economic indicator that reflects the balance between supply and demand in the energy market. By including power price inform-

ation in our model, we aim to capture the economic factors that can signal changes in the gas imbalance.

The combination of the gas balancing signal with weather and energy market data is expected to enhance the predictive capabilities of our model. By capturing a wide range of factors that influence the gas imbalance, we aim to provide accurate forecasts that can be used to make informed decisions in the management of gas distribution systems.

Temporal Features

Temporal features is also considered predicting gas imbalances, as they capture the cyclical patterns and trends that are inherent in energy demand and supply dynamics. To incorporate time as a feature in our predictive models, we calculate various temporal attributes from the date information in our dataset. Specifically, we use the following transformations in our DataFrame 'df':

- `df['dt_day']` represents the day of the week, calculated by converting the `date` column to a datetime object and then using the `.dt.dayofweek` attribute, to which we add 2 to adjust the day index as needed.
- `df['dt_wk']` captures the week of the year, obtained by using the `.dt.isocalendar().week` method on the datetime object.
- `df['dt_month']` denotes the month of the year, derived from the `.dt.month` attribute of the datetime object.
- To account for the cyclical nature of these temporal features, we also create sine-transformed versions of these attributes, such as `df['dt_day_sin']`, `df['dt_wk_sin']`, and `df['dt_month_sin']`, using the sine function and a scaling factor of

  $$\pi \times 2$$

  divided by the period of the cycle (7 for days, 52 for weeks, and 12 for months). Specifically, the sine-transformed week of the year is calculated as:

  $$df['dt\_wk\_sin'] = \sin\left(\frac{2\pi \cdot df['dt\_wk']}{52}\right)$$

  and the sine-transformed month of the year is calculated as:

  $$df['dt\_month\_sin'] = \sin\left(\frac{2\pi \cdot df['dt\_month']}{12}\right)$$

These transformations help the model to understand and leverage the periodicity in the data. In fact, according to Figure 4.3, a feature importance ranking of a model trained using LightGBM [Ke et al., 2017] , a gradient boosting framework, 'dt_wk_sin' and 'dt_day_sin' are among the top 10 most important features for classifying whether a gas imbalance will occur in the next 5-12 hours. This underscores the significance of temporal features in enhancing the predictive accuracy of models dealing with gas imbalances.

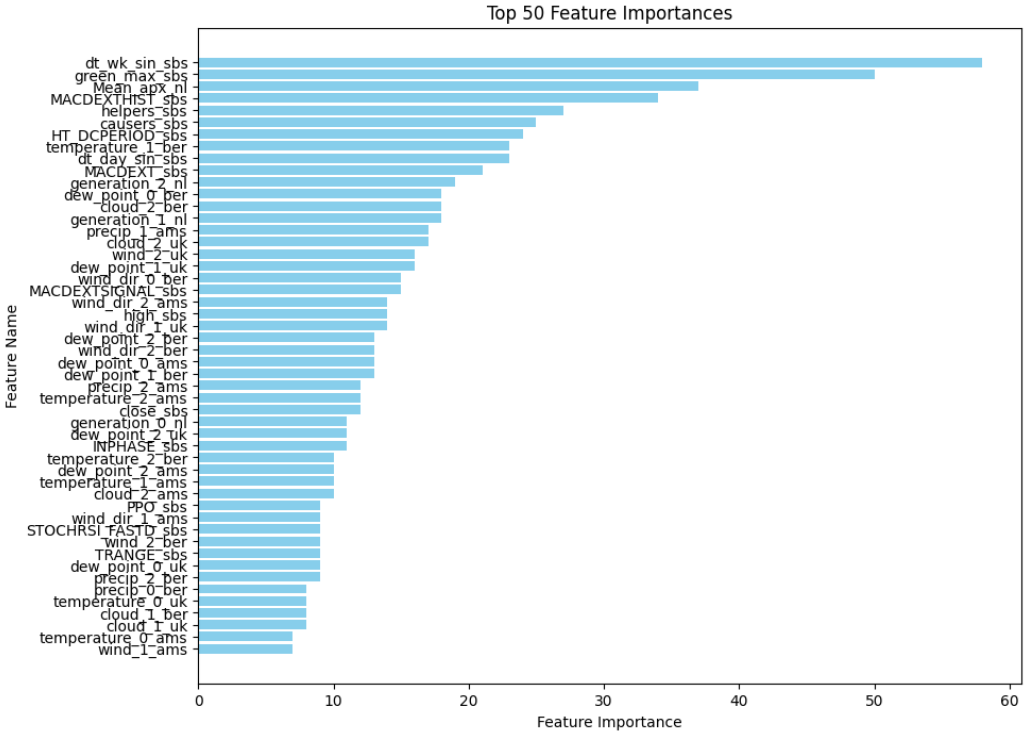

Figure 4.3: Top 50 Important Feature of a classification model to predict whether a gas imbalance will occur in the next 5-12 hours trained using LightGBM

4.2 Feature Engineering

The dataset, after feature engineering and enrichment, contains 108 features, resulting in a dimension of 63744 rows x 108 columns. Here, 63744 represents the number of hours, and 108 represents the number of features. The objective of our machine learning model is to classify whether a gas imbalance will occur in the next 5 to 12 hours and to predict how the gas balancing signal behave in next 1 to 5 hours. Table 4.1 shows Top 50 features with short explanation.

Feature	Explanation
dt_wk_sin_sbs	Sine transformation of weekdays for SBS data
MACDEXTHIST_sbs	MACD histogram for SBS data
Mean_apx_nl	Power Price of the Netherlands
HT_DCPERIOD_sbs	Hilbert Transform Dominant Cycle Period for SBS data
helpers_sbs	Suppliers Signal
green_max_sbs	Absoulte value of imbalance (light green zone) threshold
MACDEXT_sbs	MACD line value for SBS data
MACDEXTSIGNAL_sbs	MACD signal line value for SBS data
generation_2_nl	Wind power generation forecast (48h) in the Netherlands
causers_sbs	Customers Signal
dt_day_sin_sbs	Sine transformation of days for SBS data
cloud_2_ber	Cloud cover forecast (48h) in Berlin
wind_dir_2_ams	Wind direction forecast (48h) in Amsterdam
cloud_2_ams	Cloud cover forecast (48h) in Amsterdam
generation_1_nl	Wind power generation forecast (24h) in the Netherlands
generation_2_de	Wind power generation forecast (48h) in Germany
precip_1_ams	Precipitation forecast (24h) in Amsterdam
dew_point_2_ber	Dew point forecast (48h) in Berlin
dew_point_0_ams	Dew point forecast (12h) in Amsterdam
temperature_1_ber	Temperature forecast (24h) in Berlin
cloud_1_ber	Cloud cover forecast (24h) in Berlin
PPO_sbs	Percentage Price Oscillator for SBS data
precip_2_ber	Precipitation forecast (48h) in Berlin
dew_point_1_ams	Dew point forecast (24h) in Amsterdam
wind_1_ams	Wind speed forecast (24h) in Amsterdam
wind_dir_1_ber	Wind direction forecast (24h) in Berlin
wind_dir_1_uk	Wind direction forecast (24h) in the UK
dew_point_1_uk	Dew point forecast (24h) in the UK
wind_dir_2_ber	Wind direction forecast (48h) in Berlin
dew_point_2_uk	Dew point forecast (48h) in the UK
wind_dir_2_uk	Wind direction forecast (48h) in the UK
wind_2_uk	Wind speed forecast (48h) in the UK
dew_point_0_uk	Dew point forecast (12h) in the UK
cloud_2_uk	Cloud cover forecast (48h) in the UK
cloud_1_uk	Cloud cover forecast (24h) in the UK
dew_point_0_ber	Dew point forecast (12h) in Berlin
generation_0_nl	Wind power generation forecast (12h) in the Netherlands
wind_dir_1_ams	Wind direction forecast (24h) in Amsterdam
dew_point_1_ber	Dew point forecast (24h) in Berlin
wind_0_ber	Wind speed forecast (12h) in Berlin
wind_1_uk	Wind speed forecast (24h) in the UK
wind_2_ber	Wind speed forecast (48h) in Berlin
high_sbs	Last High price for SBS data
wind_dir_0_ber	Wind direction forecast (12h) in Berlin
dew_point_2_ams	Dew point forecast (48h) in Amsterdam
precip_2_ams	Precipitation forecast (48h) in Amsterdam
precip_0_ams	Precipitation forecast (12h) in Amsterdam
precip_0_ber	Precipitation forecast (12h) in Berlin
cloud_0_ber	Cloud cover forecast (12h) in Berlin
close_sbs	Last Closing price for SBS data

Table 4.1: Top 50 Features with Short Explanations, SBS means the Gas Balancing Signal

4.3 Public Dataset Description

Classification (UEA)	Dim Series	Length	Dataset Size	Information (Frequency)
EthanolConcentration	3	1751	(261, 0, 263)	Alcohol Industry
FaceDetection	144	62	(5890, 0, 3524)	Face (250Hz)
Handwriting	3	152	(150, 0, 850)	Handwriting
Heartbeat	61	405	(204, 0, 205)	Heart Beat
JapaneseVowels	12	29	(270, 0, 370)	Voice
PEMS-SF	963	144	(267, 0, 173)	Transportation (Daily)
SelfRegulationSCP1	6	896	(268, 0, 293)	Health (256Hz)
SelfRegulationSCP2	7	1152	(200, 0, 180)	Health (256Hz)
SpokenArabicDigits	13	93	(6599, 0, 2199)	Voice (11025Hz)
UWaveGestureLibrary	3	315	(120, 0, 320)	Gesture

Table 4.2: **Dataset descriptions**. The dataset size is organized in (Train, Validation, Test).

UEA Classification Datasets

The UEA multivariate time series classification archive, introduced in 2018 [Bagnall et al., 2018], represents a significant advancement in the field of machine learning, specifically targeting the classification of multivariate time series data. This archive, developed through a collaborative effort between the University of East Anglia and the University of California, Riverside, includes 30 datasets that cover a wide array of applications, such as gesture recognition, medical diagnosis, and audio recognition. These datasets are characterized by their diversity in case types, dimensions, and series lengths, providing a comprehensive resource for evaluating the performance of machine learning algorithms in handling complex, real-world data. The creation of this archive aims to stimulate research in multivariate time series classification by offering a standardized benchmark, similar to the impact of the UCR archive on univariate time series classification research. The paper choose 10 UEA classification datasets, as shown in Table 4.2

4.4 Baseline

For the paper, we have chosen a set of baseline models that are well-represented in the literature and have been evaluated in comprehensive empirical studies of time series, as reported in [Wu et al., 2023]. These baselines encompass a variety of model architectures, including:

- Tree models such as LightGBM [Ke et al., 2017], CatBoost [Prokhorenkova et al., 2019]
- CNN-based models such as TimesNet [Wu et al., 2023],
- MLP-based models like DLinear [Zeng et al., 2022],
- Transformer-based models, for instance, Autoformer [Wu et al., 2022], FEDformer [Zhou et al., 2022], ETSformer [Woo et al., 2022], Non-stationary Transformer [Liu et al., 2023], Flowformer [Huang et al., 2022], PatchTST [Nie et al., 2023] and GPT4TS [Zhou et al., 2023]

4.5 Results for the Time Series classification Public Datasets

As shown in Table 4.3, *Ours* achieves an average accuracy of 74.08%, surpassing all baselines including *GPT2(6) FPT* (73.75%) and *TimesNet* (73.46%). From the table 4.4, *Ours* achieves the best rank also, demonstrating the effectiveness of multi-task learning and leveraging prior NLP knowledge for time series representation.

Methods	LightGBM	CatBoost	Rocket	LSTNet	TCN	Auto.	Station.	FED.	Flow.	PatchTST	DLinear	TimesNet	GPT4TS	MTSPC-GPT
EthanolConcentration	43.3	43.1	**45.0**	39.7	28.7	31.4	32.5	31.0	33.6	33.8	32.4	35.5	34.7	34.8
FaceDetection	53.3	53.2	64.3	65.3	52.5	68.0	67.6	65.6	67.2	68.3	67.6	68.2	68.8	**69.2**
Handwriting	16.8	16.5	**58.4**	25.6	53.0	36.5	31.3	36.5	33.5	36.8	26.8	31.8	32.4	32.5
Heartbeat	72.2	72.4	75.2	76.7	75.2	73.3	72.9	73.3	77.2	76.3	74.7	77.6	77.8	**78.4**
JapaneseVowels	57.1	57.2	95.8	97.7	98.5	95.8	98.8	98.0	98.5	97.8	95.8	98.0	98.2	**99.1**
PEMS-SF	76.3	76.1	74.7	86.3	85.7	87.0	80.6	85.7	83.5	86.4	74.7	**89.2**	87.5	87.6
SelfRegulationSCP1	86.3	86.4	90.4	83.6	84.2	89.0	88.3	88.8	92.1	89.7	87.0	91.5	**92.9**	**92.9**
SelfRegulationSCP2	53.3	53.0	53.0	52.5	50.3	57.0	54.2	55.2	50.3	58.2	57.0	**59.2**	58.4	58.6
SpokenArabicDigits	82.7	82.5	70.9	**100.0**	95.3	**100.0**	**100.0**	**100.0**	98.5	**100.0**	81.1	98.7	98.6	98.8
UWaveGestureLibrary	64.4	64.3	**94.0**	87.4	88.0	85.8	87.1	84.9	86.2	87.1	81.7	84.9	88.2	88.9
Average	60.57	60.47	72.17	71.48	71.14	72.38	71.33	71.9	72.06	73.44	67.88	73.46	73.75	**74.08**

Table 4.3: **Full results for the classification task:** A higher accuracy indicates better performance. **Bold: best**

Methods	LightGBM	CatBoost	Rocket	LSTNet	TCN	Auto.	Station.	FED.	Flow.	PatchTST	DLinear	TimesNet	GPT4TS	MTSPC-GPT
EthanolConcentration	2.0	3.0	**1.0**	4.0	14.0	12.0	10.0	13.0	9.0	8.0	11.0	5.0	7.0	6.0
FaceDetection	12.0	13.0	11.0	10.0	14.0	5.0	6.5	9.0	8.0	3.0	6.5	4.0	2.0	**1.0**
Handwriting	13.0	14.0	**1.0**	12.0	2.0	4.5	10.0	4.5	6.0	3.0	11.0	9.0	8.0	7.0
Heartbeat	14.0	13.0	7.5	5.0	7.5	10.5	12.0	10.5	4.0	6.0	9.0	3.0	2.0	**1.0**
JapaneseVowels	14.0	13.0	11.0	9.0	3.5	11.0	2.0	6.5	3.5	8.0	11.0	6.5	5.0	**1.0**
PEMS-SF	11.0	12.0	13.5	6.0	7.5	4.0	10.0	7.5	9.0	5.0	13.5	**1.0**	3.0	2.0
SelfRegulationSCP1	12.0	11.0	5.0	14.0	13.0	7.0	9.0	8.0	3.0	6.0	10.0	4.0	**1.5**	**1.5**
SelfRegulationSCP2	9.0	10.5	10.5	12.0	13.5	5.5	8.0	7.0	13.5	4.0	5.5	**1.0**	3.0	2.0
SpokenArabicDigits	11.0	12.0	14.0	3.0	10.0	3.0	3.0	3.0	9.0	3.0	13.0	7.0	8.0	6.0
UWaveGestureLibrary	13.0	14.0	**1.0**	5.0	4.0	9.0	6.5	10.5	8.0	6.5	12.0	10.5	3.0	2.0
Average	13.0	14.0	6.0	9.0	11.0	5.0	10.0	8.0	7.0	4.0	12.0	3.0	2.0	**1.0**

Table 4.4: **Rank results for the classification task:** A lower rank indicates better performance. **Bold: best**

4.6 Results for the Gas Imbalance Dataset

Evaluation Metric

When evaluating scenarios where the objective is to predict specific events such as gas imbalances, it's crucial to focus on metrics that effectively capture the performance of the model in terms of both its precision and its ability to recall relevant instances. Precision and recall are critical evaluation metrics in classification, especially in contexts where the consequences of false positives and false negatives are significant. Precision measures the accuracy of the positive predictions made by a model, which is the proportion of true positive predictions out of all positive predictions. It is calculated as:

$$\text{Precision} = \frac{TP}{TP + FP}$$

where TP is the number of true positives and FP is the number of false positives.

Recall, on the other hand, measures the model's ability to identify all actual positive cases. It is the proportion of true positive predictions out of all actual positives, including those not predicted correctly by the model. Recall is calculated as:

$$\text{Recall} = \frac{TP}{TP + FN}$$

where FN is the number of false negatives.

In the context of gas imbalance, recall is important because it reflects the model's ability to predict as much gas imbalance as possible, which is crucial for making informed decisions. Precision is equally important because it reflects the accuracy of those predictions. A model with high precision but low recall might miss many actual positive cases, while a model with high recall but low precision might generate too many false alarms.

To balance the measurement of the model's precision and recall, the F1 score becomes a critical metric. It is particularly useful in imbalanced datasets where the cost of false positives and false negatives is high. The F1 score is calculated using the formula:

$$\text{F1} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

A high F1 score indicates a well-balanced performance between precision and recall, which is desirable in many real-world applications. Therefore,

focusing on the **positive F1 score, positive recall, and positive precision** as evaluation metrics can provide a comprehensive understanding of a model's performance in critical decision-making scenarios.

Customized Loss function

To further refine the model's focus on predicting the positive class (e.g., the occurrence of a gas imbalance), an adjustment to the loss function can be made. This involves using a weighted version of the binary cross-entropy loss function, with a higher weight assigned to the positive class (class 1). This adjustment aims to penalize misclassifications of the positive class more than those of the negative class, thereby encouraging the model to improve its detection of positive instances even at the risk of increasing false positives. The rationale behind this approach is that in many practical scenarios, such as predicting gas imbalances, the cost of missing a positive event may be much higher than the cost of a false alarm.

The weighted binary cross-entropy loss function can be expressed as:

$$\text{Loss} = -\frac{1}{N}\sum_{i=1}^{N}[w \cdot y_i \cdot \log(\hat{y}_i) + (1 - y_i) \cdot \log(1 - \hat{y}_i)]$$

where: - N is the number of observations, - y_i is the actual label of observation i, - $\hat{y}_i$ is the predicted probability of observation i being in the positive class, - w is the weight assigned to the positive class, typically greater than 1 to emphasize its importance.

By focusing on maximizing the F1 score for the positive class and adjusting the loss function to prioritize the detection of positive instances, models can be better tuned to the specific requirements of tasks like predicting gas imbalances in time series data.

Experiment Result

Experiment Setup

The gas imbalance dataset for the experiment was divided into two distinct sets: training and testing. The training data starts from August 10, 2016 18:00:00 to December 11, 2020 11:00:00. This dataset has 36,920 rows and 108 columns. The test data, used for evaluating the model's performance, starts from December 11, 2020 12:00:00 to January 8, 2024 10:00:00, having 24,614

rows and 108 columns. Both datasets were recorded at an hourly interval, ensuring a consistent temporal resolution across the data. The proposed model was trained on an AWS g5.xlarge instance, equipped with 24 GB of VRAM. Special thanks to RWE Supply & Trading GmbH for providing the computational resources.

Experiment Result

As demonstrated in our results (see Table 4.5), our proposed model outperforms the current state-of-the-art model, TimesNet, and achieves better results than the GPT4TS model.

Results Discussion

In our comparative analysis of predictive models, including LightGBM, CatBoost, TimesNet, GPT4TS, and our proposed model, we evaluated performance across three scenarios:

1. Using the full dataset with all 108 time series channels(see Table 4.5).
2. Applying only Technical Indicators and Temporal Features of the Gas Balancing Signal (see Table 4.6).
3. Only the OHLC (Open, High, Low, Close) features of the Gas Balancing Signal (see Table 4.7).

Table 4.5: Experiment result on Gas Balancing Signal Dataset (Full 108 features): A higher value indicates better performance. **Bold: best**

Model	**Precision**	**Recall**	**F1 Score**
Baseline(Random)	0.26	0.49	0.34
LightGBM	**0.37**	**0.71**	**0.49**
CatBoost	0.37	0.70	0.48
TimesNet	0.35	0.65	0.45
GPT4TS	0.36	0.68	0.47
MTSPC-GPT(6)	0.36	0.68	0.47
MTSPC-GPT(GPT-2 Medium)	0.37	0.68	0.48
MTSPC-GPT(3)	0.37	0.69	0.48

Table 4.6: Experiment result on Gas Balancing Signal Dataset (Only Technical Indicators features + Temporal Features): A higher value indicates better performance. **Bold: best**

Model	Precision	Recall	F1 Score
Baseline(Random)	0.26	0.49	0.34
LightGBM	0.35	0.67	0.46
CatBoost	0.34	0.66	0.45
TimesNet	0.34	0.65	0.45
GPT4TS	0.35	0.66	0.46
MTSPC-GPT(6)	0.35	0.66	0.46
MTSPC-GPT(GPT-2 Medium)	0.36	0.66	0.47
MTSPC-GPT(3)	**0.36**	**0.67**	**0.47**

Table 4.7: Experiment result on Gas Balancing Signal Dataset (Only OHLC features): A higher value indicates better performance. **Bold: best**

Model	Precision	Recall	F1 Score
Baseline(Random)	0.26	0.49	0.34
LightGBM	0.34	0.60	0.43
CatBoost	0.33	0.59	0.42
TimesNet	0.34	0.62	0.44
GPT4TS	0.35	0.62	0.45
MTSPC-GPT(6)	0.36	0.63	0.46
MTSPC-GPT(GPT-2 Medium)	0.36	0.64	0.46
MTSPC-GPT(3)	**0.36**	**0.66**	**0.47**

Our propsed model demonstrated better results in comparison to the other models in all scenarios except when trained on the complete dataset. This suggests that the large number of features may have introduced challenges. The observed performance differences between transformer and tree-based models, particularly when comparing the full feature set to a reduced one, can be attributed to the complexity and dimensionality of the data, as well as the inherent strengths of each model type for specific tasks.

Data Complexity and Dimensionality

Transformers, originally designed for natural language processing, are proficient at capturing long-range dependencies through self-attention mechanisms. However, their performance can be hindered by increased input

dimensionality, which amplifies the complexity of these dependencies and may necessitate more data and computational resources for effective training [Islam et al., 2023].

In contrast, tree-based models, such as decision trees and their ensembles like Random Forests and Gradient Boosted Trees, are better equipped to handle high-dimensional data. They excel at identifying important features at each decision node, which can be particularly beneficial when dealing with a large number of features [Raja and Babu, 2019].

Suitability for the Task

For tasks where understanding global dependencies is crucial, transformers can be highly effective, even with a reduced number of channels. Their self-attention mechanism allows them to integrate information across the entire input space, which is advantageous for tasks that require a comprehensive context, like image classification or sequence modeling [Hatamizadeh et al., 2023].

Tree-based models, however, may be more suitable for tasks that involve high-dimensional data with intricate, non-linear relationships where long-range dependencies are less critical. Their hierarchical nature enables them to focus on the most informative features and interactions [Grinsztajn et al., 2022].

Tree-based model v.s. Transformers model

Tree-based models have shown exceptional performance relative to transformer-based models on the Gas Balancing Signal Dataset with full feature engineering. This can also be attributed to the dataset's low signal-to-noise ratio and the presence of meticulously engineered features, which encompass over 100 features, including weather data and electricity pricing data. Conversely, in the context of simpler, publicly available datasets, transformers model tend to outperform tree-based approaches. This finding is consistent with the insights presented in the literature [Grinsztajn et al., 2022].

Generality of Pre-trained Models for Cross-Domain Knowledge Transferring

In this section, we present experiments that demonstrate the adaptability of our proposed model when switching between different backbone Large Language Models (LLMs). Specifically, we focus on the performance of the model when using GPT-2 Small and GPT-2 Medium as the backbone LLMs.

The primary differences between these two models are in their size, which includes the number of parameters, layers, and the size of their embeddings.

- **Number of Parameters:** GPT-2 Medium has 355 million parameters, which is significantly more than the 117 million parameters of GPT-2 Small. The higher number of parameters in GPT-2 Medium allows it to have a greater capacity to learn and model complex patterns in data [Radford et al., 2019].
- **Embedding Size:** The embedding size refers to the size of the vectors used to represent words or tokens. GPT-2 Medium has larger embedding sizes, which is 1024, compared to GPT-2 Small, which have an embedding size of 768. Larger embeddings can capture more nuanced semantic information [Radford et al., 2019].

The results, as shown in Tables Table 4.5, Table 4.6, and Table 4.7, indicate that there is no significant improvement when switching from the smaller LLM (GPT-2 Small) to the larger LLM (GPT-2 Medium) in the Gas Imbalance Dataset. This lack of improvement is likely due to the low signal-to-noise ratio in the dataset.

Hyperparameter Optimization

Optimizing number of GPT2 Layers

Studies such as those by [Zhou et al., 2022] and [Nie et al., 2023] have generally found that including no more than 3 encoder layers in transformer-based methods yields effective results for time-series forecasting. This finding suggests a preference for relatively shallow architectures in this specific application domain, contrasting with the deeper architectures common in other tasks such as natural language processing. A challenge with applying pre-trained models, which typically feature at least 12 layers, to time-series forecasting is the risk of overfitting. Overfitting occurs when a model learns the details and noise in the training data to the extent that it negatively impacts the model's performance on new data. This issue underscores the importance of selecting an appropriate number of layers to balance model complexity with generalization capabilities.

[Zhou et al., 2023] identifies 6 layers as the optimal configuration for the number of GPT2 layers. This finding indicates that a moderate number of layers can potentially offer a balance between capturing complex patterns in

the data and avoiding overfitting. However, experimental results, as shown in Table 4.5, 4.6 and 4.7, suggest that a pre-trained model achieves better performance with as few as 3 layers. The result may be due to the high dimensionality of the data. This outcome led to the selection of a 3-layer GPT2 architecture as the default for the authors' experiments.

LightGBM

In the Full 108 features training, among the various models assessed, LightGBM emerged as the top performer, with its optimal parameters being identified through the Optuna hyperparameter optimization framework. The optimal parameters were found by a Hyperparameter Optimization Framework named Optuna [Akiba et al., 2019]:

- `learning_rate`: 0.06885629101782148,
- `num_leaves`: 154,
- `min_child_samples`: 53,
- `max_depth`: 7,
- `reg_alpha`: 0.050785503299925774,
- `reg_lambda`: 0.0030219634281817786.

These results underscore the effectiveness of tree-based models and the importance of feature engineering in achieving better performance.

4.7 Implications for Gas Trading

From the results, the best performance seems to be achieved by training with all 108 features using the LightGBM model. It achieves a Precision of 0.37, Recall of 0.71, and F1 score of 0.49. This implies that 71% of the Gas imbalance can be identified, with a 37% chance that the prediction is correct, which is more than 40% better than a random guess.

For practical reasons, it is also advantageous to use a Tree-based model because it requires less computational resources and offers better interpretability. This is due to the ability to access the feature importance, which can be used to further improve the features, refering to Figure 4.3.

Although the proposed model shows promise and its results are comparable to other state-of-the-art models, we have decided to deploy the LightGBM as the classification model and DeepTCN as the Probabilistic Forecasting model due to considerations of computational resource requirements, complexity, and performance. The Probabilistic Forecasting model is specifically trained to predict the Gas Balancing Signal for the upcoming 1 to 5 hours. The visualization includes the gas balancing signal, upper interval (95%) prediction, median prediction, lower interval (5%) prediction, classification signal, and the threshold, refering to Figure 4.4 and 4.5. From these signals, we can identify 4 trading scenarios:

1. If the classification signal is higher than the threshold, it indicates that the risk of imbalance is quite high. In such a case, it is suggested to balance some of the position if we are in the same direction as the Gas balancing signal.

2. If the classification signal is low but the upper interval (or lower interval) prediction indicates that a Gas Imbalance is likely to occur within the next 5 hours, it is suggested to follow the classification signal first and continue monitoring the prediction signal.

3. If both the classification signal and prediction signal are low, it suggests that the risk is low, or the risk cannot be predicted by our model.

4. If both the classification signal and prediction signal indicate that the risk of imbalance is high, it is suggested to balance most of the position if we are in the same direction as the Gas balancing signal.

We believe that these information allow traders for more informed decisions without the constant need to monitor for unusual weather data or abrupt changes in the imbalance signal.

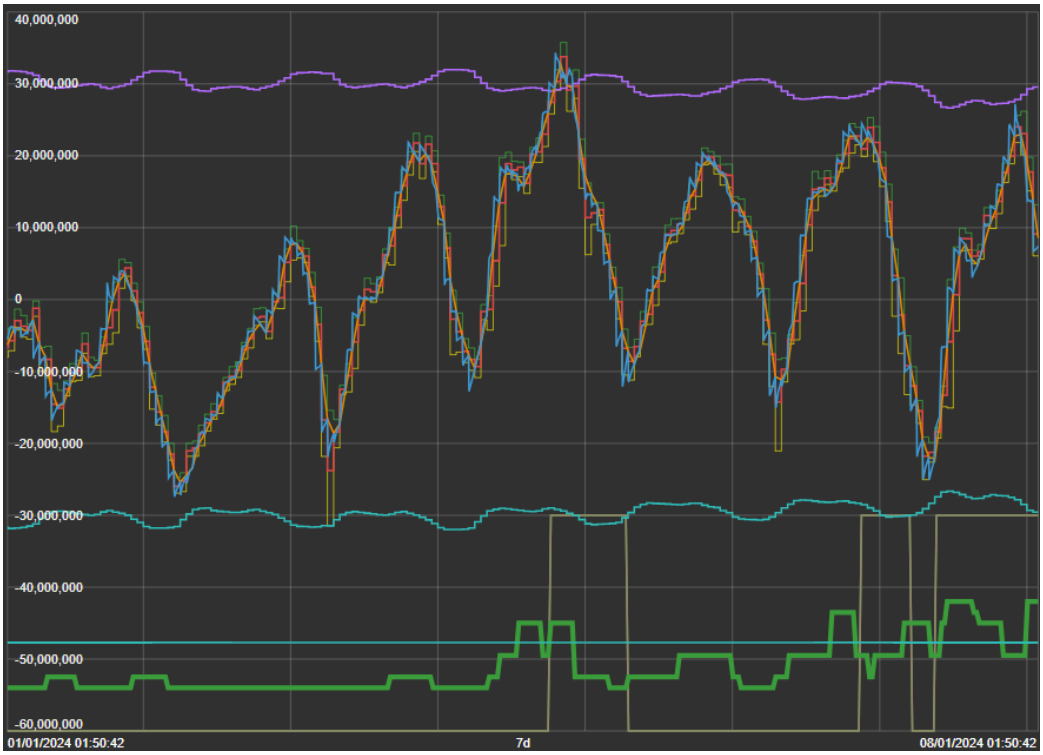

Figure 4.4: Visualization of Predictive Indicators: Blue represents the Gas Balancing Signal; Purple and Light Blue denote the thresholds for Imbalance; Green indicates the Probability of Imbalance for the next 5-12 hours; Dark Yellow signifies the Prediction of Imbalance within the Hour.

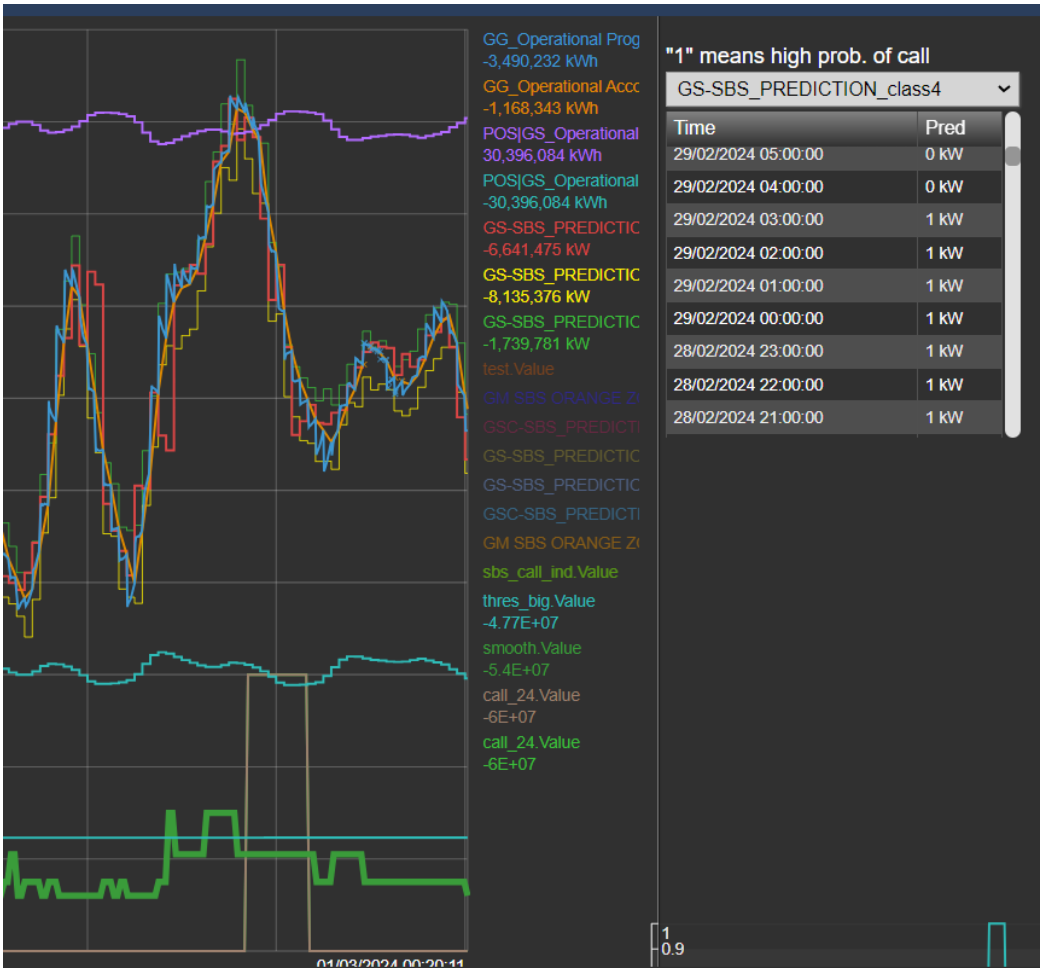

Figure 4.5: Visualization of Imbalance hour prediction table: "1" means high probability of imbalance in that hour

Chapter 5

Conclusion

In conclusion, the thesis presents a novel application of the GPT-2 model, fine-tuned for multitasking in time series prediction and classification. The research demonstrates that by incorporating external features such as weather and power forecast data, the model's predictive capabilities on complex datasets like the Dutch TTF Gas Balancing Signal are enhanced. While tree-based models have shown to be particularly effective for this dataset, likely due to its low signal-to-noise ratio and the inclusion of over 100 engineered features, pre-trained models like GPT-2 perform better on less dimension, publicly available datasets. These findings highlight the importance of feature engineering in time series analysis. Despite these findings, the potential for leveraging Large Language Models (LLMs) in time series tasks remains significant. The growing number of research on time series foundation models and fine-tuning techniques further underscores this potential. Future research could explore the use of LLMs for feature engineering, including the development of methods for automatic feature generation and selection. This could potentially lead to more effective and efficient time series models, further underscoring the potential of LLMs in this domain.

Acknowledgement

I would like to express my sincere gratitude to Carsten Hoelsken, Johannes Burcher, Julian Dorsch, Ilya Noginskiy and Masih Hakimi Nik for their valuable suggestions and proofreading assistance throughout the development of this paper. Their detailed and perceptive feedback significantly enhanced the paper's quality and clarity.

Bibliography

[Akiba et al., 2019] Akiba, T., Sano, S., Yanase, T., Ohta, T., and Koyama, M. (2019). Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery Data Mining*.

[Aslam et al., 2020] Aslam, Z., Javaid, N., Adil, M. N., Ijaz, M. T., ur Rahman, A., and Ahmed, M. (2020). An enhanced convolutional neural network model based on weather parameters for short-term electricity supply and demand. In *International Conference on Advanced Information Networking and Applications*.

[Bagnall et al., 2018] Bagnall, A., Dau, H. A., Lines, J., Flynn, M., Large, J., Bostrom, A., Southam, P., and Keogh, E. (2018). The uea multivariate time series classification archive, 2018.

[Bai et al., 2018] Bai, S., Kolter, J. Z., and Koltun, V. (2018). An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271*.

[Bao et al., 2022] Bao, H., Li, L. H., Wang, J., Zhang, T., Liu, Z., Gao, J., Han, W., Liu, J., Zhang, T., Liu, J., et al. (2022). Beit: Bert pre-training of image transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11568–11577.

[Bao et al., 2021] Bao, H., Wang, W., Dong, L., Liu, Q., Mohammed, O. K., Aggarwal, K., Som, S., and Wei, F. (2021). Vlmo: Unified vision-language pre-training with mixture-of-modality-experts.

[Box et al., 2015] Box, G., Jenkins, G., Reinsel, G., and Ljung, G. (2015). *Time series analysis: Forecasting and control*. John Wiley Sons.

[Brown et al., 2020] Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. (2020). Language models are few-shot learners. *Advances in Neural Information Processing Systems*.

[Capacity, 2023] Capacity (2023). Capacity booking: Entry and exit capacity. `https://www.gasunietransportservices.nl/en/shipper-trader/capacity-booking/entry-and-exit-capacity`.

[Demir et al., 2020] Demir, S., Mincev, K., Kok, K., and Paterakis, N. G. (2020). Introducing technical indicators to electricity price forecasting: A feature engineering study for linear, ensemble, and deep machine learning models. *Applied Sciences*, 10(1).

[Dempster et al., 2020] Dempster, A., Petitjean, F., and Webb, G. I. (2020). Rocket: Exceptionally fast and accurate time series classification using random convolutional kernels. *Data Mining and Knowledge Discovery*, 34(5):1454–1495.

[Devlin et al., 2019] Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 4171–4186, Minneapolis, MN, USA.

[Dosovitskiy et al., 2021] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. (2021). An image is worth 16x16 words: Transformers for image recognition at scale. *International Conference on Learning Representations*.

[Gasunie, 2023a] Gasunie (2023a). Balancing actions. `https://www.gasunietransportservices.nl/en/shipper-trader/balancing-regime/balancing-actions-wdba`.

[Gasunie, 2023b] Gasunie (2023b). Balancing regime. `https://www.gasunietransportservices.nl/en/shipper-trader/balancing-regime`.

[Gasunie, 2023c] Gasunie (2023c). Balancing regime: Damping. https://www.gasunietransportservices.nl/en/shipper-trader/balancing-regime/damping.

[Gasunie, 2023d] Gasunie (2023d). Balancing regime: Sbs and pos. https://www.gasunietransportservices.nl/en/shipper-trader/balancing-regime/sbs-and-pos.

[Grinsztajn et al., 2022] Grinsztajn, L., Oyallon, E., and Varoquaux, G. (2022). Why do tree-based models still outperform deep learning on tabular data?

[Hatamizadeh et al., 2023] Hatamizadeh, A., Yin, H., Heinrich, G., Kautz, J., and Molchanov, P. (2023). Global context vision transformers. In Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S., and Scarlett, J., editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 12633–12646. PMLR.

[He et al., 2022] He, J., Zhou, C., Ma, X., Berg-Kirkpatrick, T., and Neubig, G. (2022). Towards a unified view of parameter-efficient transfer learning. In *International Conference on Learning Representations*.

[He et al., 2016] He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

[Hochreiter and Schmidhuber, 1997] Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.

[Houlsby et al., 2019] Houlsby, N., Giurgiu, A., Jastrzebski, S., Morrone, B., De Laroussilhe, Q., Gesmundo, A., Attariyan, M., and Gelly, S. (2019). Parameter-efficient transfer learning for nlp. In Chaudhuri, K. and Salakhutdinov, R., editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2790–2799, Long Beach, California, USA. PMLR.

[Hu et al., 2022] Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. (2022). Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

[Huang et al., 2022] Huang, Z., Shi, X., Zhang, C., Wang, Q., Cheung, K. C., Qin, H., Dai, J., and Li, H. (2022). Flowformer: A transformer architecture for optical flow.

[Islam et al., 2023] Islam, S., Elmekki, H., Elsebai, A., Bentahar, J., Drawel, N., Rjoub, G., and Pedrycz, W. (2023). A comprehensive survey on applications of transformers for deep learning tasks.

[J. W. Wilder, 1978] J. W. Wilder, J. (1978). *New Concepts in Technical Trading Systems*. Trend Research.

[Ke et al., 2017] Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., and Liu, T.-Y. (2017). Lightgbm: A highly efficient gradient boosting decision tree. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

[Kema/TPA, 2012] Kema/TPA (2012). Investigation into the new dutch gas balancing regime and market model wholesale gas.

[Ketterer, 2014] Ketterer, J. (2014). The impact of wind power generation on the electricity price in germany. *Energy Economics*, 44:270–280.

[Kim et al., 2022] Kim, T., Kim, J., Tae, Y., Park, C., Choi, J.-H., and Choo, J. (2022). Reversible instance normalization for accurate time-series forecasting against distribution shift. In *International Conference on Learning Representations*.

[Lai et al., 2018] Lai, G., Chang, W.-C., Yang, Y., and Liu, H. (2018). Modeling long- and short-term temporal patterns with deep neural networks. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 95–104. ACM.

[Li and Liang, 2021] Li, X. L. and Liang, P. (2021). Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4582–4597. Association for Computational Linguistics.

[Lipton et al., 2015] Lipton, Z. C., Berkowitz, J., and Elkan, C. (2015). A critical review of recurrent neural networks for sequence learning. *arXiv preprint arXiv:1506.00019*.

[Liu et al., 2023] Liu, Y., Wu, H., Wang, J., and Long, M. (2023). Non-stationary transformers: Exploring the stationarity in time series forecasting.

[Lu et al., 2022] Lu, K., Grover, A., Abbeel, P., and Mordatch, I. (2022). Frozen pretrained transformers as universal computation engines. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(7):7628–7636.

[Lucidi et al., 2022] Lucidi, F. S., Pisa, M. M., and Tancioni, M. (2022). The effects of temperature shocks on energy prices and inflation in the euro area.

[Luo et al., 2023] Luo, J., Liu, S., Cai, Z., Xiong, C., and Tu, G. (2023). A multi-task learning model for non-intrusive load monitoring based on discrete wavelet transform. *The Journal of Supercomputing*, 79(8):9021–9046.

[Nie et al., 2023] Nie, Y., Nguyen, N. H., Sinthong, P., and Kalagnanam, J. (2023). A time series is worth 64 words: Long-term forecasting with transformers.

[Noor et al., 2022] Noor, T. H., Almars, A. M., Alwateer, M., Almaliki, M., Gad, I., and Atlam, E.-S. (2022). Sarima: A seasonal autoregressive integrated moving average model for crime analysis in saudi arabia. *Electronics*, 11(23).

[Ntakaris et al., 2020] Ntakaris, A., Kanniainen, J., Gabbouj, M., and Iosifidis, A. (2020). Mid-price prediction based on machine learning methods with technical and quantitative indicators. *PLoS ONE*, 15(6):e0234107.

[Oreshkin et al., 2020] Oreshkin, B. N., Carpov, D., Chapados, N., and Bengio, Y. (2020). N-beats: Neural basis expansion analysis for interpretable time series forecasting.

[Ouyang et al., 2022] Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al. (2022). Training language models to follow instructions with human feedback. In *International Conference on Machine Learning*.

[Pascanu et al., 2013] Pascanu, R., Mikolov, T., and Bengio, Y. (2013). On the difficulty of training recurrent neural networks. In Dasgupta, S. and McAllester, D., editors, *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pages 1310–1318, Atlanta, Georgia, USA. PMLR.

[Prokhorenkova et al., 2019] Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A. V., and Gulin, A. (2019). Catboost: unbiased boosting with categorical features.

[Qi et al., 2022] Qi, W., Ruan, Y.-P., Zuo, Y., and Li, T. (2022). Parameter-efficient tuning on layer normalization for pre-trained language models.

[Radford and Narasimhan, 2018] Radford, A. and Narasimhan, K. (2018). Improving language understanding by generative pre-training.

[Radford et al., 2019] Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. (2019). Language models are unsupervised multitask learners.

[Raja and Babu, 2019] Raja, B. and Babu, T. (2019). A novel feature selection based ensemble decision tree classification model for predicting severity level of copd disease. *Biomedical and Pharmacology Journal*, 12:875–886.

[Shickel et al., 2021] Shickel, B., Tighe, P. J., Bihorac, A., and Rashidi, P. (2021). Multi-task prediction of clinical outcomes in the intensive care unit using flexible multimodal transformers.

[Song et al., 2022] Song, S., Bang, S. Y., Cho, S., Han, H., and Lee, S. (2022). Attentive multi-task prediction of atmospheric particulate matter: Effect of the covid-19 pandemic. *IEEE Access*, PP:1–1.

[Tanaka et al., 2022] Tanaka, K., Matsumoto, K., Keeley, A. R., and Managi, S. (2022). The impact of weather changes on the supply and demand of electric power and wholesale prices of electricity in germany. *Sustainability Science*, 17(5):1813–1825.

[Touvron et al., 2021] Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., and Jégou, H. (2021). Training data-efficient image transformers & distillation through attention. In *Proceedings of the 38th International Conference on Machine Learning*, pages 10347–10357. PMLR.

[Ulyanov et al., 2017] Ulyanov, D., Vedaldi, A., and Lempitsky, V. (2017). Instance normalization: The missing ingredient for fast stylization.

[Vaswani et al., 2017] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. *arXiv preprint arXiv:1706.03762*.

[Wang et al., 2012] Wang, H., Nie, F., Huang, H., Yan, J., Kim, S., Risacher, S. L., Saykin, A. J., and Shen, L. (2012). High-order multi-task feature learning to identify longitudinal phenotypic markers for alzheimer's disease progression prediction. In *Neural Information Processing Systems*.

[Wei, 2019] Wei, W. W. S. (2019). Multivariate time series analysis and applications. *Wiley Interdisciplinary Reviews: Computational Statistics*.

[Weron, 2014] Weron, R. (2014). Electricity price forecasting: A review of the state-of-the-art with a look into the future. *International Journal of Forecasting*, 30(4):1030–1081.

[Woo et al., 2022] Woo, G., Liu, C., Sahoo, D., Kumar, A., and Hoi, S. (2022). Etsformer: Exponential smoothing transformers for time-series forecasting.

[Wu et al., 2023] Wu, H., Hu, T., Liu, Y., Zhou, H., Wang, J., and Long, M. (2023). Timesnet: Temporal 2d-variation modeling for general time series analysis.

[Wu et al., 2022] Wu, H., Xu, J., Wang, J., and Long, M. (2022). Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting.

[Yang et al., 2021] Yang, C.-H. H., Tsai, Y.-Y., and Chen, P.-Y. (2021). Voice2series: Reprogramming acoustic models for time series classification. In *International Conference on Machine Learning*, pages 11808–11819.

[Yıldırım et al., 2021] Yıldırım, D. C., Toroslu, I. H., and Fiore, U. (2021). Forecasting directional movement of forex data using lstm with technical and macroeconomic indicators. *Financial Innovation*, 7:1–36.

[Zeng et al., 2022] Zeng, A., Chen, M., Zhang, L., and Xu, Q. (2022). Are transformers effective for time series forecasting?

[Zhang, 2003] Zhang, G. (2003). Time series forecasting using a hybrid arima and neural network model. *Neurocomputing*, 50:159–175.

[Zheng et al., 2014] Zheng, Y., Liu, Q., Chen, E., Ge, Y., and Zhao, J. (2014). Time series classification using multi-channels deep convolutional neural networks. In *Interational Conference on Web-Age Information Management*.

[Zhou et al., 2022] Zhou, S., Shi, J., Yang, X., Zhang, H., Zhang, C., Lu, Y., and Huang, J. (2022). Fedformer: Fourier enhanced decomposition for time series forecasting. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2876–2886.

[Zhou et al., 2023] Zhou, T., Niu, P., Wang, X., Sun, L., and Jin, R. (2023). One fits all:power general time series analysis by pretrained lm.

[Zivot and Wang, 2003] Zivot, E. and Wang, J. (2003). *Vector Autoregressive Models for Multivariate Time Series*, pages 369–413. Springer New York, New York, NY.