Master's thesis
Master's Programme in Data Science

Transformers are Zero-Shot / Domain Learners for Technical Analysis

Roope Kolehmainen

September 29, 2025

Supervisor(s): Professor Teemu Roos

Examiner(s): Professor A
Dr. B

University of Helsinki
Faculty of Science

P. O. Box 68 (Pietari Kalmin katu 5)
00014 University of Helsinki

HELSINGIN YLIOPISTO — HELSINGFORS UNIVERSITET — UNIVERSITY OF HELSINKI

Tiedekunta — Fakultet — Faculty Faculty of Science	Koulutusohjelma — Utbildningsprogram — Degree programme Master's Programme in Data Science

Tekijä — Författare — Author

Roope Kolehmainen

Työn nimi — Arbetets titel — Title

Transformers are Zero-Shot / Domain Learners for Technical Analysis

Työn laji — Arbetets art — Level Master's thesis	Aika — Datum — Month and year September 29, 2025	Sivumäärä — Sidantal — Number of pages 27

Tiivistelmä — Referat — Abstract

This paper investigates the application of transformer-based language models to the domain of financial time series forecasting, specifically focusing on price prediction through technical analysis. While large language models (LLMs) such as GPT have demonstrated exceptional performance in natural language processing tasks, their potential in financial modeling - particularly in learning and predicting chart-based trading patterns - remains underexplored.

The study first outlines the foundational principles of technical analysis and its role in modeling price dynamics in financial markets. It then introduces the transformer architecture, highlighting its core components, self-attention, multi-head attention, and autoregressive modeling, explains how these enable the detection of long-range, non-linear patterns within sequential data. The research evaluates whether transformers can internalize the temporal structures and heuristics that underlie technical trading strategies.

This work contributes to a novel intersection of deep learning and financial engineering, testing the hypothesis that models originally developed for language can meaningfully emulate or enhance rule-based financial analysis.

ACM Computing Classification System (CCS):
Computing methodologies → Machine learning → Machine learning applications → Forecasting
Applied computing → Business → Financial engineering
Computing methodologies → Natural language processing → Language models

Avainsanat — Nyckelord — Keywords

layout, summary, list of references

Säilytyspaikka — Förvaringsställe — Where deposited

Muita tietoja — Övriga uppgifter — Additional information

Contents

1. Introduction

Technical analysis posits that, at least in the short term, the pricing of securities does not strictly follow a stochastic or purely random process. Instead, it suggests that price movements are influenced by investor behavior, which can give rise to identifiable patterns and trends in historical price data. The objective of technical trading is to detect and exploit these patterns to inform profitable trading decisions.

In contrast to fundamental analysis, which relies on indicators such as a firm's financial performance, macroeconomic data, or asset-based valuation, technical analysis focuses exclusively on signals derived from past market behavior. This focus on sequential price dependencies aligns conceptually with the autoregressive modeling paradigm found in contemporary natural language processing (NLP), where future tokens are predicted based on sequences of preceding inputs. Transformer-based models, in particular, have demonstrated exceptional capability in capturing structural patterns in sequential data.

This parallel raises a key research question: *Can transformer architectures, which excel at modeling complex syntactic and semantic structures in language, also be applied to financial time series to extract and forecast meaningful patterns in market behavior*?

Technical analysis provides a compelling testbed for this question. It encompasses a broad range of trading strategies, from intraday high-frequency trading to longer-horizon approaches like momentum and swing trading, all of which depend on identifying repeatable patterns in price movements. Forecasting tasks in this domain can be framed as binary or multiclass classification problems (e.g., up / down movements or buy / hold / sell actions), as well as point-estimation tasks such as trend direction or volatility forecasting.

Moreover, financial markets differ significantly across asset classes and timeframes, requiring tailored modeling strategies. These design decisions include choosing between encoder-decoder versus decoder-only transformer architectures, selecting data granularity (e.g., high-frequency vs. daily), and determining feature sets (e.g., OHLC - Open, High, Low, Close - data, technical indicators).

To explore the applicability of transformer models in this context, this study adopts a dual approach. First, it investigates zero-shot inference using general-purpose large language models (LLMs) such as LLaMA, which have not been fine-tuned on financial data. These models are prompted to analyze market behavior based solely on pre-trained world knowledge and reasoning capabilities. Second, the study develops a domain-specific transformer model trained directly on historical price sequences to learn market-specific dynamics for forecasting tasks.

By comparing these two paradigms - zero-shot generalization versus supervised, task-specific learning - and benchmarking them against traditional baselines (e.g., ARIMA or LSTM), this study seeks to evaluate the strengths, limitations, and practical utility of transformer-based architectures in modeling and predicting patterns in financial time series.

RESULT PLACEHOLDER: Results from this study indicate that [brief description of result; e.g., domain-specific transformer models outperform baseline approaches in certain classification tasks]. Notably, the [model name or approach] achieved [X] % accuracy / F1-score / Sharpe ratio on [task], surpassing [baseline model] by a margin of [Y]%. Conversely, the zero-shot inference using general-purpose large language models showed [brief insight e.g., limited quantitative performance but some qualitative reasoning ability]. These findings suggest that [core insight e.g., transformer models can capture meaningful temporal dependencies in financial time series when trained in-domain], but also highlight [a limitation or nuance].

2. Background and Related Work

Despite the explosive growth in both financial machine learning (ML) and natural language processing (NLP), the direct application of transformer-based large language models (LLMs) to technical analysis for stock prediction remains largely unexplored. Historically, financial forecasting models have relied heavily on statistical and deep learning techniques such as ARIMA, LSTM, GRU, and CNN architectures to model price series. These methods emphasize pattern recognition and temporal dependencies in numerical data, forming the core of traditional technical analysis approaches.

Conversely, transformer-based LLMs like GPT, BERT, and their financial-domain variants (e.g., FinBERT, BloombergGPT) have primarily been utilized for text-based analysis, including sentiment extraction, earnings call interpretation, and market news classification. While some hybrid applications integrate text sentiment with price data, these typically do not utilize LLMs to process raw time series in a manner akin to classical technical charting.

To our knowledge, no prior work has directly leveraged transformer-based LLMs to perform technical-style price movement forecasting - that is, using LLMs to identify and act upon patterns such as breakouts, reversals, or momentum shifts directly from price sequences, candlestick patterns, or volume indicators. This research therefore occupies a novel intersection between the interpretability of traditional technical analysis and the pattern abstraction capacity of LLMs.

Although a few early-stage studies (e.g., StockTime, ElliottAgents) suggest that transformers can capture market structure when fine-tuned on historical charts, these remain exceptions and do not generalize to LLM-based prediction in a traditional technical analysis framework. Our study aims to bridge this gap by exploring whether general-purpose LLMs can emulate or enhance rule-based trading logic, offering a new lens through which to view market dynamics.

2.1 Technical Analysis in Financial Markets

Technical analysis (TA) refers to the forecasting of future price movements in financial markets through the examination of historical market data, primarily price and volume.

In contrast to fundamental analysis, which seeks to determine a security's intrinsic value using financial statements or macroeconomic indicators, TA is rooted in the belief that prices already reflect all publicly available information (consistent with the *semi-strong form* of the Efficient Market Hypothesis, EMH). It further assumes that investor psychology and collective market behavior produce identifiable, repetitive price patterns over time [11].

TA is implemented through a wide variety of trading strategies suited to different investment horizons:

- **Momentum trading** identifies assets with strong upward or downward trends and follows the trend direction, supported by evidence from [8] showing persistence in short-term return patterns.
- **Swing trading** targets price movements over several days or weeks, using chart patterns and oscillators to identify temporary fluctuations within broader trends.
- **Mean reversion strategies** are based on the premise that prices tend to revert to a historical mean, and are widely used in statistical arbitrage and high-frequency trading (HFT) contexts.
- **Intraday and high-frequency trading (HFT)** applies TA principles at ultra-short intervals - minutes or even milliseconds - aiming to exploit micro-inefficiencies and liquidity gaps in real-time markets.

Technical analysis employs a variety of analytical tools designed to identify patterns and predict market movements. These tools generally fall into three main methodological classes, each reflecting a distinct approach to interpreting price and volume data:

- **Chart patterns**, such as head-and-shoulders, triangles, and double tops/bottoms, offer visual cues that practitioners interpret as signals of trend continuation or reversal.
- **Indicators and oscillators**, including moving averages, the Moving Average Convergence Divergence (MACD), the Relative Strength Index (RSI), and Bollinger Bands, provide smoothed, quantitative insights into price dynamics, trend strength, and market conditions [**?**, 1].
- **Statistical models**, such as the Autoregressive Integrated Moving Average (ARIMA) model [5] and the Generalized Autoregressive Conditional Heteroskedasticity (GARCH) model [2], aim to model autocorrelation, volatility clustering, and non-stationary behavior in asset returns.

Despite its widespread use among practitioners, TA has faced sustained criticism in academic finance, particularly from proponents of the EMH. According to EMH, particularly in its strong and semi-strong forms [6], all relevant information is already reflected in asset prices, making it impossible to systematically outperform the market through pattern recognition or historical price analysis. From this perspective, TA is no more effective than random guessing, and any perceived patterns are likely artifacts of data mining or overfitting.

Critics also argue that many technical patterns lack formal theoretical grounding, are not rigorously defined, and are subject to confirmation bias in interpretation. Empirical studies that fail to adjust for transaction costs, slippage, and overfitting in backtesting further challenge the validity of many TA-based strategies.

However, TA has also been defended on both empirical and behavioral grounds. For example, [3] demonstrated that simple moving average and trading range breakout rules generated significant excess returns in historical simulations. Behavioral finance adds further nuance by suggesting that markets are not always rational; cognitive biases, herd behavior, and overreactions may indeed cause patterns that TA can exploit at least temporarily.

Recent work by [12] shows that TA continues to play a practical role in energy and financial markets, especially for short-term speculation and risk management. Moreover, the rise of algorithmic and quantitative trading has enabled more rigorous and systematic implementations of TA, increasingly enhanced by data-driven methods such as machine learning [19].

2.2 Machine Learning in Financial Forecasting

The increasing availability of financial data and advances in computational power have accelerated the adoption of machine learning (ML) techniques in financial forecasting. While traditional TA relies on predefined rules and visual patterns to guide trading decisions [1], machine learning offers a data-driven alternative capable of uncovering complex, non-linear relationships that may be difficult to formalize explicitly. In recent years, this has led to the development of hybrid systems that combine technical indicators with statistical learning models to improve predictive accuracy and robustness.

Nti et al. [13] provide a comprehensive systematic review of stock market prediction techniques, highlighting the transition from rule-based and econometric models to modern ML approaches. Their findings show that supervised learning, especially classification models for directional movement prediction (e.g., up or down), dominates the literature, with support vector machines (SVMs), decision trees, and ensemble methods such as random forests being widely adopted. At the same time, feature engineering remains closely tied to TA: most studies extract input variables from traditional indicators like RSI, MACD, and moving averages, showing that even in ML contexts, technical analysis continues to shape model design.

Deep learning has further expanded the boundaries of financial modeling, offering tools for both feature extraction and temporal dependency modeling. Olorunnimbe and Viktor [14] survey deep learning applications in stock market prediction, noting the growing use of CNNs, LSTMs, and attention-based architectures. Their review emphasizes the importance of rigorous backtesting, hyperparameter optimization, and overfitting controlâareas where deep learning has both introduced new capabilities and new challenges. Notably, they observe that while model complexity has increased, predictive performance gains remain context-specific, and there is no universally superior architecture.

Both reviews agree that a key difficulty in applying ML to financial time series lies in their inherent non-stationarity and high noise levels. Moreover, success is often sensitive to the choice of input features, target definition, and evaluation metrics. For instance, models trained to minimize mean squared error (MSE) may not align with trading objectives like risk-adjusted returns or drawdown minimization.

To bridge the gap between traditional TA and machine learning, some researchers have explored the use of technical indicators as feature encodings for deep models, while others have experimented with direct time-series representations such as candlestick charts or OHLCV sequences. Achelis [1] provides a practical taxonomy of indicators and pattern recognition strategies, many of which serve as the foundation for modern input pipelines in ML-based trading systems.

Taken together, these developments reflect a shift in financial forecasting from manually engineered heuristics toward automated pattern discovery and learning. However, challenges such as data snooping, model interpretability, and real-world trading constraints remain open. These concerns have motivated the exploration of more advanced sequential models, most notably transformer-based architectures, which are introduced in the following section.

2.3 Transformer Architectures

Transformer architectures, introduced by Vaswani et al. (2017) in the seminal paper *Attention Is All You Need*, have fundamentally reshaped sequential modeling, particularly in the field of natural language processing (NLP). Unlike recurrent neural networks (RNNs) and their variants such as long short-term memory (LSTM) models, which process sequences step-by-step, transformers operate on entire sequences in parallel. This parallelism enables significant gains in training efficiency and scalability.

More importantly, transformers effectively model long-range dependencies through the mechanism of self-attention, overcoming the memory and gradient limitations of earlier recurrent architectures [17, 15].

The core innovation of the transformer lies in its **attention mechanism**, particularly *self-attention*, which allows the model to dynamically weigh the relevance of all elements in a sequence when processing any individual element. This mechanism underpins the model's ability to capture context-dependent and non-local relationships across sequences.

- **Self-Attention**: Enables each token (or time step) to attend to every other position in the input sequence. This facilitates the modeling of complex dependencies regardless of sequence distance, which is essential in domains such as language understanding and financial forecasting.
- **Multi-Head Attention**: Extends self-attention by computing attention in parallel across multiple subspaces. Each "head" can focus on different types of relationships, allowing the model to simultaneously capture various levels of structure and interaction within the data.
- **Positional Encoding**: Since the transformer architecture does not impose any sequential ordering inherently, positional encodings are introduced to inject information about the relative or absolute positions of tokens. These encodings allow the model to retain a notion of temporal structure, which is critical in both text and time series data [17].

Transformers have achieved state-of-the-art performance across a wide range of NLP tasks, including machine translation, question answering, summarization, and generative text modeling [4]. Their success is attributed to the capacity to model deep contextual relationships, capture latent structure in unstructured data, and scale effectively with large datasets and model sizes.

These capabilities are not unique to language. Financial time series, much like natural language, exhibit complex, non-linear dependencies, recurring motifs, and variable-length temporal structures. The non-local computation enabled by self-attention is particularly well-suited for identifying delayed or non-contiguous relationships - such as multi-day momentum, periodic reversals, or volatility regimes - that are often targeted by technical analysts [19].

Transformers can be trained in an **autoregressive** fashion, where the model predicts the next element in a sequence conditioned on all past elements:

$$P(x_t \mid x_{<t}) = P(x_t \mid x_{t-1}, x_{t-2}, \ldots, x_1)$$

This formulation underlies many large language models (LLMs) such as GPT, which generate coherent sequences token-by-token [16]. In the context of financial modeling, this same autoregressive structure can be leveraged to predict future price values or directional movements based on prior historical data. It aligns conceptually with the logic of technical analysis, where traders infer future behavior from historical patterns.

Importantly, the transformers' ability to condition on arbitrarily long contexts, unlike traditional models with fixed memory windows, allows it to learn long-term trend evolution, sequential causality, and regime shifts. These capabilities are crucial for trading strategies like momentum detection, breakout forecasting, and volatility modeling [20, 9].

The combination of self-attention for pattern discovery and autoregressive modeling for sequential prediction positions transformers as highly promising for time series forecasting. Compared to classical approaches like ARIMA or even LSTMs, which are often constrained in their capacity to model long-term dependencies, transformers offer greater flexibility and representational power.

However, despite their transformative impact in NLP, the application of transformer architectures to financial forecasting, particularly in the context of technical analysis, remains relatively nascent. Time series data introduces unique challenges: non-stationarity, low signal-to-noise ratios, regime switching, and the lack of a natural "vocabulary" or semantic hierarchy typical in language data [12].

This study seeks to explore whether transformer-based models can internalize the stylized facts, heuristics, and pattern-based reasoning that underpin technical trading. Specifically, it evaluates whether these models can capture the types of temporal dynamics that human traders use to identify support / resistance levels, momentum shifts, or volatility breakouts, assessing the practical viability of applying transformer architectures to model and predict structured behavior in financial markets.

2.4 Transformers for Financial Time Series

Although transformer architectures have become central in time-series forecasting research, their application to large-scale financial markets,particularly equity price prediction, remains limited. Much of the existing literature focuses on data domains such as weather, traffic, and energy systems, which are typically characterized by smoother dynamics, higher signal-to-noise ratios, and deterministic seasonal patterns. In contrast, financial time series are shaped by heterogeneous agents, reflexivity, and rapid feedback loops, leading to non-stationarity, regime shifts, and substantial noise. These characteristics make financial forecasting both more complex and less predictable than many natural processes.

A defining feature of financial markets is their sensitivity to human behavior. Unlike physical systems governed by stable laws, markets are influenced by collective psychology, sentiment, and institutional incentives. Technical analysis (TA) builds directly on this premise, asserting that price movements reflect recurring behavioral patterns, such as momentum, support / resistance levels, and volatility breakouts that can be identified and leveraged for trading decisions. This assumption distinguishes financial time series from those in natural sciences governed by mechanistic causality. Accordingly, applying transformer-based models - originally designed for symbolic and language data - to financial forecasting requires addressing fundamentally different sources of structure: ones driven not by physical laws but by cognitive and strategic behavior.

Despite these challenges, transformer models offer appealing properties for modeling financial time series. Their ability to capture long-range dependencies, model temporal context flexibly, and scale to high-dimensional inputs makes them a promising candidate for learning patterns in complex market environments governed by human behavior.

In recent literature, three major research directions have emerged around applying transformers in time-series forecasting: (1) the use of LLMs for zero-shot forecasting; (2) architecture-level innovations for time-series adaptation; and (3) benchmarking studies evaluating their empirical competitiveness.

Gruver et al. [7] show that instruction-tuned LLMs can perform zero-shot time-series forecasting directly from raw numeric sequences. Without any gradient-based fine-tuning, models like GPT-3.5 produce competitive one-step and multi-step forecasts across various domains, including equity indices and currency rates. The underlying ability is attributed to two key features: the autoregressive training objective, which implicitly teaches next-value prediction, and the model's encoded knowledge of real-world structures acquired during pre-training. While LLMs generally underperform domain-specific models on raw error metrics, they offer substantial practical benefits - most notably, the ability to make forecasts without training data. In the financial context, this opens up possibilities for rapid prototyping, exploratory scenario analysis, and priors for Bayesian-style pipelines.

At the architectural level, transformer variants have been tailored specifically for long-horizon time-series forecasting. Liu et al. [10] propose iTransformer, which introduces an inverted attention mechanism where queries attend to groups of keys and values extracted from non-overlapping patches of the input. This not only reduces computational complexity from $\mathcal{O}(L^2)$ to $\mathcal{O}(L\,p^{-1})$, where p is the patch length, but also improves empirical performance on long-range benchmarks such as ETTm1. Similarly, earlier work by Nie et al. (PatchTST) demonstrates that simply dividing the input into patches and removing positional encodings can outperform more intricate attention sparsification methods. Other models, such as Autoformer and FEDformer, decompose time series into trend and seasonal components before applying attention, making them particularly relevant for financial assets that exhibit periodic structure, such as foreign exchange and commodities. These innovations collectively highlight the importance of domain-aware tokenization and sparsity in adapting transformers for forecasting financial data.

However, Zeng et al. [18] provide a sobering counterpoint, emphasizing that transformers are not universally superior. Their extensive benchmarking across eight public datasets reveals that, once proper normalization and scaling are applied, simple linear models or shallow multilayer perceptrons (MLPs) often outperform transformer variants. Their findings suggest that the key to good forecasting performance may lie more in pre-processing and data treatment than in model complexity. This is especially pertinent in finance, where low signal-to-noise ratios and frequent regime changes can cause overfitting in deep models. These insights serve as a caution against assuming that transformers are always the optimal choice, reinforcing the need for rigorous backtesting and economic validation, particularly when applying such models to trading strategies derived from technical analysis.

Despite the progress, several open challenges remain. First, financial markets are inherently non-stationary, and transformers trained under the assumption of stationarity may struggle during regime shifts unless explicitly adapted through online learning or meta-learning strategies. Second, most evaluation metrics focus on statistical loss functions such as MAE or MSE, while neglecting financially meaningful criteria like Sharpe ratio, maximum drawdown, or transaction-cost-adjusted returns. Third, data scarcity continues to be a constraint, especially in emerging markets or niche asset classes where long and clean historical series are unavailable. While synthetic data and transfer learning offer potential solutions, they remain underexplored in the financial context.

While transformers can learn from non-stationary data, structural changes in financial time series, such as regime shifts and volatility spikes, still challenge generalization, especially in zero-shot settings. Moreover, technical indicators often assume stationary distributions, and their interpretations may break down when the data evolves. For robust performance, attention to temporal segmentation and data regimes remains essential.

3. Empirical testing (plan)

This thesis investigates the forecasting capabilities of transformer-based models in financial time series by comparing two approaches:

1. **Zero-shot inference** using large language models (LLMs) deployed locally on CSC infrastructure
2. **A supervised domain-specific transformer**, trained from scratch on historical price data

The study focuses on the task of predicting short-term trading signals (e.g. Buy / Hold / Sell) from recent stock price movements, with consistent evaluation metrics across both modeling paradigms.

3.1 Task formulation

The predictive task is modeled as a classification problem based on future log-returns. The target labels are created by discretizing the return distribution. Two alternative schemes could / will be considered to enable detection of more robust price swings:

Option A: 3-Class Classification with Higher Thresholds

- **Buy**: Future return $\geq +3\%$
- **Sell**: Future return $\leq -3\%$
- **Hold**: Otherwise

This setup emphasizes high-confidence and strong-signal predictions. It simplifies modeling but may increase class imbalance due to fewer extreme movements.

Option B: 5-Class Classification (Granular Signal Tiers)

- **Strong Buy**: Future return $\geq +5\%$

- **Buy**: +1% to +5%
- **Hold**: −1% to +1%
- **Sell**: −5% to −1%
- **Strong Sell**: Future return ≤ −5%

This approach supports more nuanced signal calibration and enables analysis of model expressiveness across volatility levels, but may require more data and careful handling of class imbalance.

Both labeling schemes will be evaluated using consistent input-output structures across the supervised and zero-shot models.

3.2 Input Features

Models are tested in two phases to assess the contribution of different feature sets:

Phase 1: Price-only Inputs

- Input: 3–10 recent daily closing prices (see [7])
- Supervised model: uses normalized log-return vectors
- Zero-shot LLM: receives prices embedded in a natural language prompt

Phase 2: Feature-Augmented Inputs

Additional market variables are included to test the effect of feature enrichment:

- Daily trading volume
- Technical indicators (e.g., RSI, MACD)

In the supervised model, these are added as extra features. For LLMs, they are injected into the prompt as natural language, e.g.:

```
Volume:  31.2M
RSI: 67.4
MACD: +0.42
```

This design enables direct comparison of model behavior under sparse vs. enriched feature conditions.

What about supervised model?? Arrays, which in prices-only mode are half empty, or re-train the whole thing?? Talk with Teemu Roos!

3.3 Zero-Shot Inference Using LLMs

Zero-shot experiments are conducted using locally deployed open-source models (e.g., LLaMA 2, Mistral) on CSC Puhti supercomputer. Prompts follow this format:

> Here are the closing prices for Company X:
> 2024-06-17: $194.50
> 2024-06-18: $197.80
> 2024-06-19: $199.10
> 2024-06-20: $198.20
> 2024-06-21: $200.35
>
> Based on this trend, what is the next trading action?
> A. Buy
> B. Hold
> C. Sell
>
> Answer:

For 5-class classification, options are extended to include `Strong Buy` and `Strong Sell`.

Output Mapping: Model completions are mapped to class labels based on exact match or soft regex matching.

Confidence Estimation: When logits are accessible, softmax or entropy is used; otherwise, prompt sampling with temperature control or ensemble prompting approximates uncertainty.

3.3.1 Supervised Transformer Model

A decoder-only transformer (NanoGPT-scale) is trained on the same data:

- Input: Normalized return sequences with optional feature augmentation
- Output: Softmax over 3 or 5 classes depending on the labeling scheme
- Loss: Cross-entropy

This model provides natively calibrated probabilities and serves as a domain-specific baseline.

3.4 Evaluation Metrics

Classification Metrics

- Accuracy
- Precision, Recall, F1-score (macro + per-class)
- Confusion matrix

Confidence-Aware Metrics

- Accuracy by confidence decile
- Top-k accuracy (where applicable)
- Expected Calibration Error (ECE)

Financial Metrics (Optional)

- Cumulative return (from backtested signals)
- Sharpe ratio
- Maximum drawdown

3.5 Infrastructure and Tooling

Experiments are run on the CSC Puhti supercomputer:

- **Supervised model training**: PyTorch / PyTorch Lightning
- **LLM inference**: Hugging Face Transformers, `llama.cpp`, text-generation-webui
- **Data and metrics**: pandas, NumPy, scikit-learn, matplotlib

API-based LLM access is avoided entirely to eliminate cost and latency overhead. All models run locally, enabling reproducibility and access to internal logits where possible.

3.6 Summary Table

Aspect	**Supervised Transformer**	**Zero-Shot LLM**
Training	Supervised	None
Input	Normalized returns (+features)	Natural language prompt
Label Space	3 or 5 classes	3 or 5 text labels
Confidence	Softmax score	Logits or sampling
Compute	CSC GPU training	CSC LLM inference
API Cost	None	None
Evaluation	Classification + financial	Same

Table 3.1: Comparison of modeling approaches

4. Figures and Tables

4.1 Figures

Figure 4.1 gives an example how to add figures to the document. Remember always to cite the figure in the main text. There are many ways to cite, for example: University of Helsinki has a nice logo (see Fig. 4.1).

Figure 4.1: University of Helsinki flame-logo for Faculty of Science.

4.2 Tables

Table 4.1 gives an example how to report experimental results. Remember always to cite the table in the main text. There are many ways to cite, for example: The results are as expected (see Table 4.1).

Table 4.1: Experimental results.

Koe	1	2	3
A	2.5	4.7	-11
B	8.0	-3.7	12.6
$A + B$	10.5	1.0	1.6

5. Citations

5.1 Citations to literature

References are listed in a separate .bib-file. In this case it is named `bibliography.bib` with the following content:

```
@article{einstein,
    author =        "Albert Einstein",
    title =         "{Zur Elektrodynamik bewegter K{\"o}rper}. ({German})
        [{On} the electrodynamics of moving bodies]",
    journal =       "Annalen der Physik",
    volume =        "322",
    number =        "10",
    pages =         "891--921",
    year =          "1905",
    DOI =           "http://dx.doi.org/10.1002/andp.19053221004"
}

@book{latexcompanion,
    author    = "Michel Goossens and Frank Mittelbach and Alexander Samarin",
    title     = "The \LaTeX\ Companion",
    year      = "1993",
    publisher = "Addison-Wesley",
    address   = "Reading, Massachusetts"
}

@misc{knuthwebsite,
    author    = "Donald Knuth",
    title     = "Knuth: Computers and Typesetting",
    url       = "http://www-cs-faculty.stanford.edu/%7Eknuth/abcde.html"
}
```

In the last reference url field the code `%7E` will translate into `~` once clicked in the final pdf.

References are created using command `\cite{einstein}`, showing as [?]. Other examples: [?, ?].

Citations should be arranged in alphabetical order by author, using the default style `abbrv`.

5.2 Crossreferences

Appendix A on page 27 contains a code example.

6. From tex to pdf

In Linux, run `pdflatex filename.tex` and `bibtex filename` repeatedly until no more warnings are shown. You should use `pdflatex` when compiling your document.

7. Conclusions

It is good to conclude with some insightful discussion.

Bibliography

[1] S. B. Achelis. *Technical Analysis from A to Z.* McGraw Hill, New York, 2001.

[2] T. Bollerslev. Generalized autoregressive conditional heteroskedasticity. *Journal of Econometrics*, 31(3):307–327, 1986.

[3] W. Brock, J. Lakonishok, and B. LeBaron. Simple technical trading rules and the stochastic properties of stock returns. *The Journal of Finance*, 47(5):1731–1764, 1992.

[4] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, et al. Language models are few-shot learners. *Advances in Neural Information Processing Systems (NeurIPS)*, 33:1877–1901, 2020.

[5] D. A. Dickey and W. A. Fuller. Distribution of the estimators for autoregressive time series with a unit root. *Journal of the American Statistical Association*, 74(366):427–431, 1979.

[6] E. F. Fama. Efficient capital markets: A review of theory and empirical work. *The Journal of Finance*, 25(2):383–417, 1970.

[7] N. Gruver, R. Jagtap, F. Vieira, and S. H. Bach. Large language models are zero-shot time series forecasters. In *Advances in Neural Information Processing Systems*, volume 36, pages 19622–19635, 2023.

[8] N. Jegadeesh and S. Titman. Returns to buying winners and selling losers: Implications for stock market efficiency. *The Journal of Finance*, 48(1):65–91, 1993.

[9] B. Lim and S. Zohren. Time-series forecasting with deep learning: A survey. *Philosophical Transactions of the Royal Society A*, 379(2194):20200209, 2021.

[10] Y. Liu, Y. Liu, Z. Zhang, Z. Zhang, and P. Xu. itransformer: Inverted transformers are effective for time series forecasting. *arXiv preprint*, 2023.

[11] J. J. Murphy. *Technical Analysis of the Financial Markets.* New York Institute of Finance, 1999.

[12] Y. Ni. Revisiting technical analysis in the era of algorithmic trading. *Journal of Financial Data Science*, 6(1):33–48, 2024.

[13] I. K. Nti, A. F. Adekoya, and B. A. Weyori. A systematic review of fundamental and technical analysis of stock market predictions. *Artificial Intelligence Review*, 53(4):3007–3057, 2020.

[14] K. Olorunnimbe and H. Viktor. Deep learning in the stock marketâa systematic survey of practice, backtesting, and applications. *Artificial Intelligence Review*, 56(3):2057–2109, 2023.

[15] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever. Improving language understanding by generative pre-training. *OpenAI Technical Report*, 2018. https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf.

[16] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever. Language models are unsupervised multitask learners. *OpenAI Blog*, 2019. https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf.

[17] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems (NeurIPS)*, 30, 2017.

[18] A. Zeng, X. Guo, B. Liu, J. Huang, et al. Are transformers effective for time series forecasting? In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 10974–10982, 2023.

[19] Y. Zhang, Y. Zhu, L. Zhang, and Z. Jiang. Transformer-based models for financial time series forecasting: A survey. *Expert Systems with Applications*, 216:119442, 2023.

[20] T. Zhou, Y. Liu, W. Zhang, Y. Yu, and J. Wang. Informer: Beyond efficient transformer for long sequence time-series forecasting. *Proceedings of AAAI*, 35(12):11106–11115, 2021.

Appendix A. Code example

Program code can be added as appendix:

```
#!/bin/bash
text="Hello World!"
echo $text
```