THE LONDON SCHOOL OF ECONOMICS AND POLITICAL SCIENCE

Application of Transformer Architecture to Signal Generation in the Mid-Frequency FX Spot Market

Department of Statistics 2024-25

Candidate Number
44026

Submitted for the Master of Science, London School of Economics
University of London

Abstract

This study evaluates encoder-only Transformers as generators of mid-frequency trading signals in the foreign-exchange spot market. Hourly OHLC data for six major pairs (EUR/USD, GBP/USD, USD/JPY, USD/CAD, AUD/USD, USD/CHF) recorded between 1 January 2013 and 31 December 2023 are cast into a three-class problem that jointly encodes direction and magnitude of the next 12-hour move. Three design axes are examined: (i) context length (24 h, 72 h, 120 h), (ii) positional encoding (local versus fully learnable), and (iii) imbalance-aware loss (Focal Loss versus weighted cross-entropy). A class-weighted XGBoost ensemble trained on flattened sequences serves as a non-sequential baseline.

We evaluate models with a custom Signal-Accuracy metric that scores predictions only at the moments a trade is taken, penalising costly errors and ignoring intervals of inactivity. Under this measure, the strongest Transformer - configured with a 72-hour context window, local positional encoding, and Focal-Loss weights 0.28/0.36/0.36 - attains a Signal-Accuracy of 1.17 while issuing trade signals on 12.5 % of all price bars (its "activity" level). The benchmark XGBoost model records accuracy score of 3.70 while signalling on 9.2 % of bars.

Back-tests on the 2021–2023 out-of-sample window show that Transformer signals generate modest yet low-risk returns (+1.41 % EUR/USD; +0.23 % AUD/USD), maintain annualised volatility below 3.4 %, and cap drawdowns at –8.2 %, though they generally trail the gradient-boosted benchmark.

The evidence confirms that self-attention layers can recover coherent directional structure, but their incremental edge hinges on locality-biased encodings and calibrated loss functions. Transformers therefore contribute stable, low-risk signal components but are not, in isolation, a substitute for well-tuned tree ensembles. The systematic exploration of context, positional priors, and loss design presented here establishes a transparent benchmark for future work that couples richer multi-modal inputs or ensemble stacking to unlock the full potential of attention mechanisms in FX trading.

Keywords: Foreign-exchange (FX) trading; mid-frequency signals; encoder-only Transformers; self-attention; positional encoding; imbalance-aware loss (Focal Loss); XGBoost benchmark; time-series classification

Acknowledgements

Reflecting on my academic journey at the London School of Economics, I am filled with a deep sense of gratitude. This experience has been both profoundly rewarding and intellectually demanding, and I could not have reached this point without the unwavering support of those closest to me.

First and foremost, I would like to thank my wife, whose patience, encouragement, and daily support have been a constant source of strength throughout this journey. I am equally grateful to my parents, who have given everything to support my professional and personal growth. Their belief in me has been the foundation of all my achievements.

I would also like to extend my sincere thanks to my supervisor, Dr. Tengyao Wang, for his insightful guidance and thoughtful supervision. His expertise and feedback have been instrumental in shaping the direction and quality of this dissertation.

List of Figures

List of Tables

Contents

1. Introduction

The Foreign Exchange (FX) spot market represents the largest and most liquid financial marketplace on a global scale, facilitating the immediate exchange of currencies, typically executed within a two-business-day time frame (Bank for International Settlements, 2022). This market operates in an over-the-counter (OTC) manner, which means the absence of a centralised exchange, with trading activities occurring continuously in international financial centres around the clock. Key participants include commercial and central banks, hedge funds, multinational corporations, and individual traders. These entities engage in transactions using electronic platforms and dealer networks with the objectives of hedging against currency risk, speculating on currency price fluctuations, or fostering international trade and investment. FX trading is characterised by deep liquidity, high volatility, and continuous accessibility. The most frequently traded currency pairs, such as EUR/USD and USD/JPY, dominate global trading volumes, whereas less common, or "exotic" pairs present opportunities with elevated risk. The spot market offers greater flexibility and immediacy compared to the FX futures market, where contracts are standardised and traded on regulated exchanges (Hull, 2018), although it is less transparent and regulated.

In recent years, algorithmic and data-driven trading has grown rapidly in the FX spot market (Chan, 2013). A key component of these approaches is signal generation, which involves identifying potential trading opportunities using various indicators. These signals can be derived from technical patterns, including moving averages of historical price trajectories, economic indicators such as interest rates or inflation statistics, or sentiment extracted from financial news sources. The frequency of these signals plays a crucial role in shaping trading strategies. Low-frequency signals often reflect broader economic trends, while higher-frequency signals focus on exploiting short-term price movements.

Machine learning has played an increasingly important role in the generation of these financial signals, particularly over the past two decades. Initially, simple statistical models, such as linear regression, were used to identify patterns in financial time series data. As computational power and data availability improved, more complex algorithms such as decision trees, support vector machines, and ensemble methods such as Random Forest became common in trading systems. These models allowed for greater flexibility in capturing non-linear relationships and interactions in market data. More recently, neural networks, especially deep learning models, have gained popularity due to their ability to process large volumes of data and uncover subtle patterns across multiple input sources. Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks have been used to capture temporal dependencies in price series, while Convolutional Neural Networks (CNNs) have found applications in feature extraction from alternative data. This evolution reflects the growing

complexity and adaptability of signal generation systems in modern financial markets.

Building on this trajectory, Transformer models, originally developed for natural language processing (Vaswani et al., 2017), are now being explored for time series forecasting in finance (Zhang et al., 2022), due to their ability to model sequential data using a self-attention mechanism. This allows the model to assess the relevance of each data point within the context of the entire sequence, effectively capturing long-range dependencies without relying on recurrence. As a result, Transformers offer notable advantages over traditional Deep Learning models like LSTMs, particularly in terms of parallelisation and reduced computational cost.

The primary objective of this dissertation is to address the problem of generating reliable trading signals in the mid-frequency FX spot market using deep learning architectures, specifically Transformers, with a focus on mid-frequency predictions based on historical hourly data. The study utilises Open, High, Low and Close (OHLC) prices for six major currency pairs: EUR/USD, GBP/USD, USD/JPY, USD/CAD, AUD/USD, and USD/CHF spanning the period from January 1, 2013, to December 31, 2023. Despite the widespread success of Transformer models in various domains, their application in FX markets remains limited, with existing research predominantly focused on equities, commodities, and indices. A recent contribution by (Fischer et al., 2024) demonstrates the potential of state-of-the-art Transformer models with time embeddings in FX spot prediction, suggesting that further exploration in this domain is timely and warranted.

To address this gap, this study introduces a set of equally important and novel contributions that collectively differentiate it from the existing literature. First, the forecasting task is formulated as a classification problem, where labels are carefully engineered to indicate both the direction and magnitude of price movements in the next n steps, allowing the model to produce a probabilistic forecast that supports informed trading decisions. Second, the study investigates the impact of varying context window sizes to determine the optimal historical input length to capture meaningful market dynamics. Third, it explores the comparative effectiveness of different positional encoding strategies; namely Local Positional Encoding and Learnable Positional Encoding, addressing the inherent limitation of Transformers in processing sequential information. Finally, the research evaluates the influence of advanced loss functions, specifically Focal Loss and Weighted Cross-Entropy, which are designed to handle class imbalance and improve model robustness.

The effectiveness of these modelling choices is assessed using a dual evaluation framework that combines proprietary statistical metrics with realistic backtest procedures to capture both predictive precision and practical economic value. As a benchmark, the study includes a comparative analysis with XGBoost, a widely used and well-established model, to assess whether Transformers provide a meaningful improvement in both predictive power

and trading performance within the FX spot market context.

The remainder of this dissertation is organised as follows. Section 2 presents a review of the literature that focusses on existing research related to the application of machine learning models in the generation of financial signals. Section 3 provides a detailed overview of the data, including its sources, exploratory data analysis, and the methodology used to label signals. In Section 4, we explore various architectures of the Transformer model and compare their performance with a baseline XGBoost model. Section 5 evaluates the practical utility of the models by backtesting the signals generated by the best-performing Transformer architecture and the baseline. Finally, Section 6 summarises the findings and outlines potential directions for future research.

2. Related Work

2.1 Traditional Models for Financial Time Series Forecasting

Forecasting financial time series has long been a challenging task due to noisy, non-stationary data and changing market regimes. Traditionally, statistical models such as ARIMA and GARCH were popular for modelling asset prices and volatility. During the 2010s, machine learning techniques became increasingly popular; in particular, tree-based ensemble methods like XGBoost (eXtreme Gradient Boosting) emerged as preferred approaches for predictive modelling in the financial sector (Iskandar et al., 2024). XGBoost can capture complex non-linear feature interactions and has outperformed linear models like ARIMA in stock price prediction, particularly for short-term forecasts. However, these ensemble methods treat data as independent and identically distributed (i.i.d.) and require extensive feature engineering to incorporate temporal dynamics. Although XGBoost offers strong predictive power, it demands careful hyperparameter tuning to avoid overfitting, and its results can be less interpretable without additional tools. In practice, features such as lagged returns, technical indicators, and macroeconomic variables must be manually constructed for tree-based models to use any time-dependent patterns.

By the late 2010s, deep learning approaches became increasingly popular for financial forecasting (Zhang et al., 2023). In particular, recurrent neural networks (RNNs) like Long Short-Term Memory (LSTM) networks were widely adopted to model sequential dependencies in price data. LSTMs, with their gating mechanisms, address vanishing gradients and can learn long-term dependencies better than simple RNNs. Many studies reported that LSTM-based models outperform traditional methods in capturing the complex patterns of asset returns. For instance, Deep Momentum Networks (DMNs) introduced by Zhang et al. (2023) optimised trading signals using an LSTM to maximise Sharpe ratio, significantly outperforming classical momentum strategies from the early 2000s onwards. LSTMs thus became a default choice for financial time series tasks, from stock trend prediction to FX rate forecasting, often yielding superior accuracy over methods like ARIMA or XGBoost when sufficient data was available.

Despite their success, RNNs and LSTMs have notable limitations. They process sequences sequentially, which makes training and inference slower and hinders capturing very long-range dependencies due to their finite memory. Even with gating, LSTMs can effectively "forget" older information once market regimes change. For example, an LSTM-based trading model may struggle when a new regime (e.g., a volatility spike or structural break) occurs because the model's state is reset and it has difficulty linking the information before the regime change with the dynamics after the change. In effect, LSTM is excellent

at learning local or recent patterns, but less adept at maintaining global context over long time horizons (Lim et al., 2021). This drawback can lead to suboptimal performance during periods of regime change or when the long-term historical context is important.

Additionally, while LSTMs can in principle learn complex patterns, they sometimes underperform or require substantial data to beat simpler models in noisy financial data sets. For example, a recent study on high-frequency limit order book data found that transformer-based models provided only marginal accuracy gains over LSTMs to predict raw price sequences, and LSTMs actually performed more consistently when predicting price differences or movements (Bilokon and Qiu, 2025). These mixed results highlight that traditional models such as gradient boosting and LSTMs remain strong benchmarks, and improvements from more complex architectures are context dependent. It is against this backdrop that the Transformer architecture has emerged as a promising advancement in time series forecasting.

2.2 Transformer Architectures in Financial Forecasting

The Transformer architecture, originally developed for natural language processing, has revolutionised sequence modelling by using an attention mechanism to capture dependencies across an entire sequence in parallel (Vaswani et al., 2017). Instead of processing data step by step (as RNNs do), Transformers use self-attention to directly attend to all time steps and learn which past observations are most relevant for forecasting the future. This ability to access all historical time points simultaneously allows Transformers to model long-range patterns and interactions more effectively than LSTMs in many cases (Lim et al., 2021).

As noted by Wood et al. (2021), unlike sequential LSTMs that are customised for local processing, an attention-based model has a direct connection to all previous time steps, allowing the learning of longer-term dependencies. This key innovation helps overcome the limitation of LSTM of forgetting older information, as the Transformer can revisit any previous data point through learnt attention weights. Moreover, Transformers are highly parallelizable, making training faster on modern hardware and allowing them to scale to large datasets or long sequences more efficiently than RNNs.

Researchers have begun to adapt Transformers to financial time series with promising results. Zhang et al. (2023) provide a comprehensive review of recent deep learning models for price forecasting and identify Transformers as one of the advanced architectures gaining traction due to their strong sequence modelling capabilities. Early applications of Transformers in finance had to address challenges unique to time series data, for example, handling temporal ordering and continuous time differences (unlike words in a sentence). One solution has been to incorporate time embeddings or positional encodings to give the model a sense of temporal order.

Fischer et al. (2024) introduce a state-of-the-art Transformer with time embeddings for FX rate prediction, demonstrating that Transformers can indeed be applied effectively to sequential financial data. Their work emphasises that Transformers are suitable for FX forecasting in general, but performance gains are most pronounced when the model is fed rich multivariate inputs. In other words, a Transformer given only a single price series may perform on par with an LSTM, but when given cross-sectional data (multiple currency pairs or related asset features) and multiple covariates, the Transformer's ability to attend across many features can lead to outperformance of LSTMs. This finding aligns with the intuition that attention can leverage relationships across different inputs (e.g., correlations between currency pairs or between a currency and a commodity price) better than models focused on one sequence at a time.

Several Transformer-based models tailored to time series forecasting have been proposed in the recent literature. Notable examples include the Temporal Fusion Transformer (TFT) by Lim et al. (2021), which integrates gating mechanisms for static features and provides interpretable forecasts in economics, and the Informer/Autoformer series by Zhou et al. (2021), which improve efficiency for long time series. In quantitative finance specifically, Zhang et al. (2024) introduced the "Quantformer," a custom Transformer architecture for stock trend prediction. By leveraging transfer learning from a language task (sentiment analysis), their Quantformer incorporated textual market sentiment into a financial model. This approach exploits the Transformer's strength in handling diverse data: the model was first trained to understand sentiment from news, then fine-tuned to predict stock returns, thereby blending textual signals with price data. The result was a significant improvement in the predictive accuracy of trading signals, as the added sentiment knowledge helped the model anticipate market movements. Empirically, Quantformer outperformed a broad set of 100 factor-based strategies (traditional quant factors such as value, momentum, etc.), producing higher returns with lower portfolio turnover and a more robust signal half-life.

One of the reasons Transformers can excel in financial forecasting is their ability to capture multiple patterns or regimes concurrently. Through the use of multi-head attention, a Transformer can attend to different aspects of the input data in parallel. Every attention head can focus on a unique pattern. For example, one head could monitor a long-term trend, while another could identify short-term mean reversion variations. Fischer et al. (2024) illustrate this by noting that separate attention heads could be responsible for different market characteristics in FX data, 'e.g., momentum, reversal patterns, support, and resistance levels'. Wood et al. (2021) similarly designed their Momentum Transformer with multiple attention heads to 'capture concurrent regimes or temporal dynamics at different timescales'. This means that a single model can simultaneously account for both slow-moving trends and fast oscillations in the market. Such capability is valuable in financial markets where regimes of-

ten overlap; for instance, a long-term bull trend can coexist with short-lived pullbacks. Traditional single-threaded models (such as LSTM) might struggle to disentangle these, whereas an attention-based model can isolate them in different heads.

Another advantage observed is that these architectures can be made inherently interpretable despite their complexity. Because attention weights explicitly show which past time steps or features the model is focussing on, researchers can extract insights about what the model 'thinks' is important for a prediction. For example, the Momentum Transformer provides insight into which past returns or technical factors are most influential for its predictions. This transparency is a significant benefit in an industry that demands understanding of model decisions for risk management and regulatory compliance.

2.3 Applications in Mid-Frequency FX Trading Strategies

A key focus of the current literature and the focus of this dissertation, is the practical use of Transformers for signal generation in mid-frequency trading, particularly in the FX (foreign exchange) market. Mid-frequency strategies operate on a time scale between high-frequency trading (milliseconds to seconds) and long-term investment (months to years). In FX, this might involve signals that trade on the order of hours, days, or weeks, capturing medium-term trends or mean-reverting moves without the ultra-tight latency constraints of high-frequency trading. This domain is well suited for advanced forecasting models: there is enough data granularity to train complex models, yet signals often persist long enough for models to exploit them before they disappear.

One practical application is in the design of momentum-based trading signals. Momentum (or time series momentum) strategies follow the principle 'trend is your friend', long on assets that have been rising and short on those falling, in the expectation that these trends will continue in the near future (Wood et al., 2021). Such strategies are prominent in FX and managed futures funds, as currency markets often exhibit trending behaviour driven by macroeconomic flows and carry trades. Deep learning models have been used to enhance momentum strategies by forecasting whether a trend will persist. For example, the Momentum Transformer proposed by Wood et al. (2021) explicitly targeted time series momentum trading. This architecture, an LSTM augmented with attention, learnt to predict an optimised position (long/short) for each asset by considering both its recent returns and longer-term context. The result was an improved momentum strategy that outperformed standard momentum benchmarks. The model's ability to look further back in time via attention allowed it to detect when a momentum signal was reliable versus when a trend was likely to break. Notably, the Momentum Transformer maintained strong performance even after accounting for transaction costs, whereas traditional momentum strategies often suffer once realistic

trading costs are included. This indicates that the Transformer's signals were not excessively noisy or flipping too frequently; it was capturing robust trends worth trading. In fact, during the volatile COVID-19 period (a major regime change), the hybrid model adapted better than a pure LSTM, underscoring how attention helps handle regime shifts in markets. Multiple attention heads allowed the model to concurrently manage different time horizons, effectively blending a momentum strategy (following longer-term trends) with a mean reversion strategy on shorter horizons. This is significant because in practice traders often combine strategies to balance each other; a model that can internally do so is highly valuable. Wood et al. (2021) report that their LSTM-based DMN could take advantage of both momentum and a 'fast mean reversion strategy' simultaneously, and the addition of attention only strengthened this capability by making the model more regime-aware and interpretable.

Mean reversion signals form the other side of the coin in many mid-frequency strategies, including FX. Mean reversion in FX might involve betting that an overextended move will reverse (for example, if a currency spiked far above its average range, one might short it expecting a pullback). These strategies, often called 'contrarian' or 'follow the loser' strategies, assume that what rises quickly will fall and vice versa (Chan, 2013). Traditional mean reversion indicators include RSI (Relative Strength Index) or Bollinger Band deviations, which indicate when an instrument is overbought or oversold. Machine learning models can improve mean reversion strategies by learning complex conditions under which reversals occur. For instance, a model might learn that a currency pair mean-reverts only when volume is low and a certain macro condition holds, but trends otherwise. A nonlinear pattern that is hard to code as a simple rule. Transformers are particularly adept at this kind of pattern recognition because they can incorporate many input features and long histories to decide if the current situation matches a past reversal scenario. In practice, a Transformer could generate a reversal signal by forecasting a negative return after a sharp rise, effectively telling the trading system to go against the recent price movement. Such signals can be integrated into portfolio strategies either on their own or as a complement to momentum signals, providing diversification (since momentum and reversal often offset each other). The ability of Transformers to handle multi-horizon forecasting is also useful here. A single model can predict short-term mean reversion moves and longer-term momentum trends as separate outputs, allowing a trader to allocate capital to each strategy appropriately.

Concrete examples from the literature demonstrate these applications. Zhang et al. (2024), although applied to stock selection, is conceptually similar to what one might do in FX: it created an 'investment factor' from a Transformer's output that can be seen as a trading signal. This factor effectively distilled various predictive features (including momentum from prices and sentiment from news) into a single time series used to rank stocks. In FX, one can imagine training a Transformer to output a score for each currency indicating

expected future return; this score becomes the trading signal (long the highest scores and short the lowest, for example). The Quantformer's success of beating 100 well-known quant factors, underlines the potential of Transformer-based signals to capture what traditional signals might miss.

Similarly, Fischer et al. (2024) tested a Transformer on FX spot rates of major currency pairs (EUR/USD, USD/JPY, etc.) and found it reliably predicted the direction when sufficient features were included. Their model could be used for a daily trading strategy that goes long or short each currency pair based on the predicted price movement, essentially using the Transformer's forecast as a signal. They report that the Transformer model, when given multivariate input (including other assets and technical indicators), could outperform an LSTM benchmark on FX return prediction. This suggests that in a mid-frequency context, a well-designed Transformer can serve as the core of a trading strategy, providing signals that encapsulate both momentum and reversal patterns detected across a rich dataset.

3. Data

This dissertation focusses on the six most liquid and widely traded FX spot pairs worldwide. EUR/USD, GBP/USD, USD/JPY, USD/CAD, AUD/USD, and USD/CHF. These currency pairs are selected due to their deep liquidity, which ensures narrow bid-ask spreads, minimal transaction costs, and reliable execution. Their global prominence also means they are less prone to pricing inefficiencies caused by low market participation, thereby allowing for easier execution of large orders, an important consideration for corporate and institutional investors seeking to minimise market impact.

The raw data for each currency pair was obtained from HistData.com (HistData.com, 2025), comprising price records in a one-minute interval over a period from 1 January 2013 to 31 December 2023. Each observation includes the Open, High, Low, and Close prices, as well as the corresponding timestamp.

To facilitate computational efficiency and allow the Transformer model to learn from longer temporal dependencies, the data set was resampled at an hourly frequency. Resampling was conducted using the following aggregation logic: for each hour, the Open is the first price of the interval, the High is the maximum observed price, the Low is the minimum, and the Close is the final price. The transformation significantly reduces data dimensionality, making it more tractable for iterative model training and allowing for the detection of extended patterns and potential momentum or reversal signals across broader horizons.

Subsequent to resampling, any rows containing missing values were removed. These gaps stemmed primarily from periods of no trading activity on weekends or occasional lapses in data recording. Deletion was preferred over forward filling or interpolation, as the latter can introduce misleading synthetic continuity, particularly over nontrading intervals that would cause the model to infer artificial flatlines or erroneous persistence. In contrast, omission of incomplete observations preserves the integrity of the temporal sequence and avoids encoding spurious signals, while only resulting in minor discontinuities across adjacent trading periods.

3.1 Directional Signal Labelling Framework

Given that the Transformer is a supervised machine learning model, it requires well-defined input-output pairs for training. In this application, the input consists of sequences of historical price movements, specifically the Open, High, Low, and Close prices sampled at hourly intervals for each currency pair. The output, or target label, traditionally could be a future price point, such as the next Close value, thereby framing the task as a regression problem.

However, for the purpose of this dissertation, the problem has instead been formulated as a classification task.

This decision is based on several practical and methodological considerations. Most notably, a classification framework allows the model to output a probability distribution over discrete market direction categories (e.g., ”up”, ”down”, or ”neutral”), rather than a single continuous value. This probabilistic output offers a richer interpretive layer, enabling threshold-based decision making. For example, trades can be conditionally executed only when the model's predicted probability for a certain class (e.g., a price increase) exceeds a confidence threshold. This setup introduces a mechanism for filtering out low-confidence predictions, potentially reducing the frequency of false signals and curbing over trading, a common source of unnecessary transaction costs in high-frequency strategies.

Furthermore, classification is more aligned with the nature of many real-world trading decisions, which are inherently categorical: buy, sell or hold. By discretising market outcomes and training the model to recognise directional patterns rather than exact price levels, the learning process becomes less sensitive to noise and minor fluctuations, which can often dominate short-term price movements but are not actionable in practice. Additionally, from a risk management perspective, the probabilistic outputs of a classifier can be incorporated into broader portfolio allocation or position-sizing frameworks, allowing for dynamic adjustment based on model confidence.

To convert continuous price data into a suitable classification format, labels were generated based on the concept of maximum expected movement within a fixed forward-looking window. Specifically, for each hourly observation, a 12-hour forward return window was used to calculate the maximum absolute price movement that could occur relative to the current Close price. This was implemented using a custom function that iterates over every timestamp t. For each t the algorithm looks one to twelve hours ahead, computes the forward-looking percentage return for every horizon, and keeps the return whose magnitude is greatest while preserving its sign.

$$\text{Forward return:} \quad \Delta_h(t) = \frac{P_{t+h} - P_t}{P_t} \times 100\%, \quad h \in \{1, \ldots, 12\};$$

$$\text{Horizon of the largest move:} \quad h^* = \arg\max_h |\Delta_h(t)|;$$

$$\text{Signed extreme 12-hour return:} \quad M_t = \Delta_{h^*}(t).$$

Thus, M_t is the *absolute-max move*: a single signed percentage change that is equal to the largest absolute change observed in the 12-hour window. Positive M_t values indicate the

greatest upward move and negative M_t values indicate the greatest downward move over the entire horizon. This approach captures not just the directional drift but the full extent of market volatility over a short- to medium-time horizon, aligning with the interests of intraday and swing trading strategies that rely on meaningful price excursions to justify execution.

Once this forward movement metric was obtained, the empirical distribution of these values was independently analysed for each pair of currencies. The 10th and 90th percentiles of the absolute return distribution served as dynamic thresholds for labelling. These thresholds were chosen to isolate significant directional movements while filtering out noisy or neutral behaviour, ensuring that the model focusses on the most volatile samples. Observations whose maximum forward movement fell below the 10th percentile were labelled 2, indicating a significant downward movement. In contrast, those above the 90th percentile were labelled 1, which signifies a substantial upward move. All remaining observations, whose movements fell between the 10th and 90th percentiles, were labelled as 0, denoting a neutral class with insufficient directional conviction.

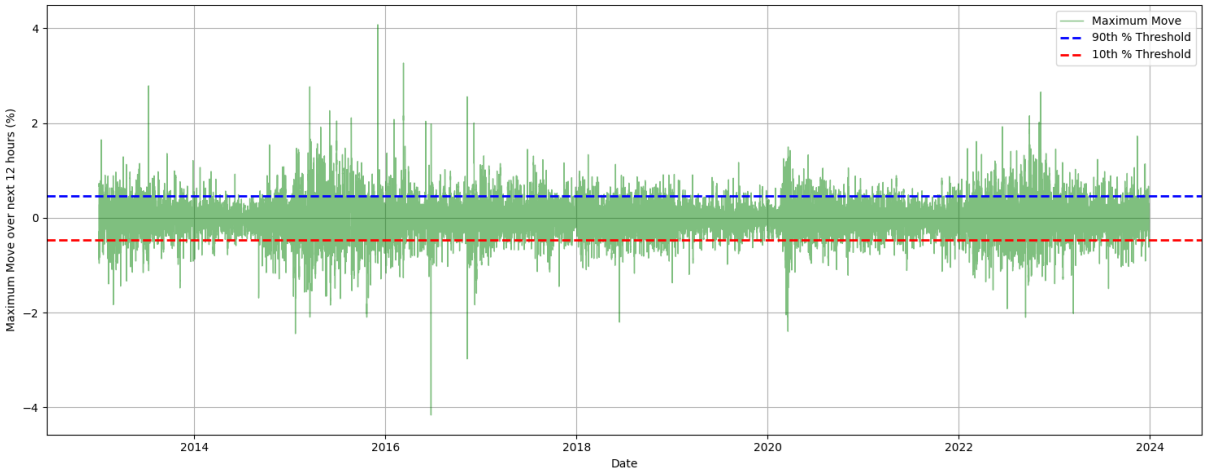

Figure 3.1: Forward 12-Hour Maximum Price Move for EUR/USD

This trichotomous labelling strategy has several advantages. Firstly, it biases learning toward time-steps that are more likely to contain useful trading cues, and away from those characterised by market indecision or background noise. Secondly, it accounts for the unique volatility and behaviour of each currency pair by applying thresholds locally rather than globally. This per pair calibration is essential, as different FX instruments exhibit different microstructural characteristics, liquidity profiles, and sensitivity to macroeconomic events.

3.2 Dataset Partitioning, Scaling, and Sequence Generation

Before training the models, a structured data preprocessing pipeline was implemented to ensure consistency, model stability, and generalisation across the six currency pairs. The pipeline consisted of four primary steps: chronological dataset partitioning, standardisation, sequence generation, and final dataset concatenation.

First, the data set for each currency pair was split into three subsets: training, validation, and testing based on a fixed chronological ratio of 40%-20%-40%. This method preserves the temporal structure of the time series and avoids data leakage, which could occur if future information were inadvertently introduced into the training process. As an example, for the EUR/USD pair, index boundaries were calculated based on the total number of observations, and slicing was performed using `pandas.DataFrame.iloc`, ensuring non-overlapping and time-consistent splits *(see Figure 3.2)*. The same methodology was applied uniformly to the other five pairs.

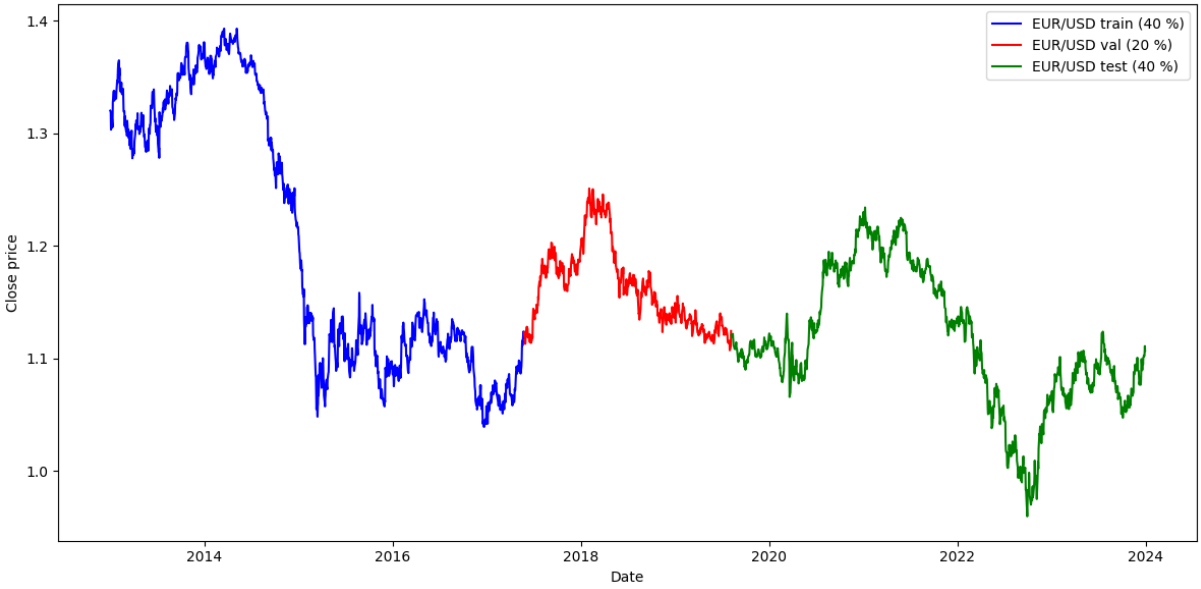

Figure 3.2: Chronological Train / Validation / Test split for EUR/USD

Following the split, a `StandardScaler` was fitted exclusively to the training subset and subsequently used to transform the three partitions (training, validation and test). This step is crucial given that deep learning models such as Transformers are sensitive to the scale of input features. By normalising the Open, High, Low, and Close prices to have unit variance and zero mean (as learnt from the training set), the model is better positioned to learn structural patterns and relationships between variables, rather than being influenced by their absolute magnitudes. Moreover, since different currency pairs (e.g., USD/JPY vs. GBP/USD) exist

on vastly different price scales, standardisation mitigates scale-induced biases that could impede convergence or distort attention scores within the Transformer layers.

Subsequently, fixed-length sequences were generated from the scaled data. For each sample, a sliding-window approach was used to extract input sequences of 24, 72 or 120 consecutive hourly observations. The label corresponding to each sequence was taken from the final row of that window, thus preserving causal ordering. This approach ensures that the model only learns from the information available up to time t, aligning with the constraints of the real world market. Feature selection was consistently applied across pairs, focusing on OHLC inputs, while the labels reflected the classification structure described above.

Finally, to train a single unified model capable of capturing generalisable patterns across currency pairs, the individual training, validation, and test sets from all six FX instruments were concatenated. This aggregation improves the model's exposure to diverse market dynamics and reduces the risk of overfitting to idiosyncratic patterns found in any pair, promoting greater applicability and robustness in forecasting directional movements.

4. Methodology

4.1 Transformer Model Architecture

This section outlines the deep learning model architecture used for classifying directional signals in the mid-frequency FX spot market. The architecture is inspired by the Transformer framework introduced by Vaswani et al. (2017), but is tailored specifically for the characteristics of multivariate financial time-series data. In this work, only the encoder component of the Transformer is employed, thereby framing the task as a sequence-to-vector modelling problem. This design choice aligns with the objective of extracting a fixed-length representation from variable-length input sequences for classification purposes. The model outputs a probability distribution across three directional classes: upward, neutral, and downward movements.

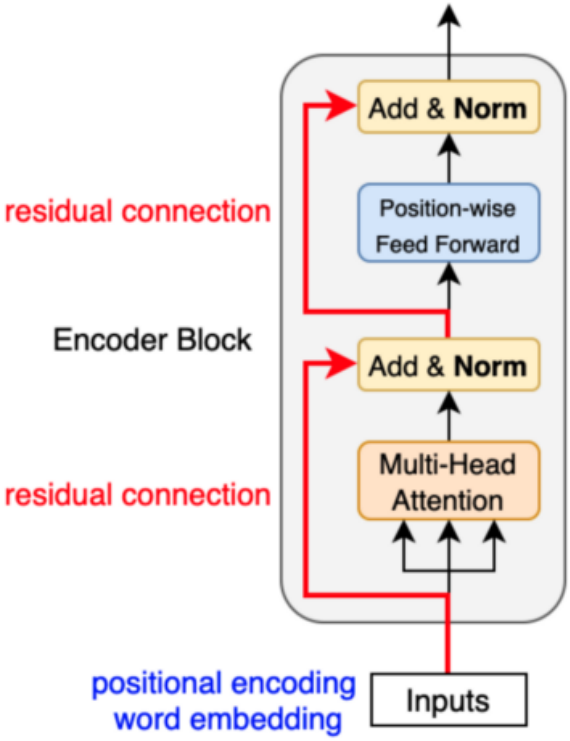

Figure 4.1: Encoder Layer of Transformer Architecture

4.1.1 Input Representation and Feature Expansion

Let the input be denoted by $\mathbf{X} \in \mathbb{R}^{T \times d}$, where $T = 72$ corresponds to the number of hourly time steps in each sequence (equivalent to a 3-day window), and $d = 4$ represents the four input features: Open, High, Low, and Close (OHLC) prices. The study evaluates the impact of different context window lengths on forecasting performance by experimenting with input sequence lengths of 24, 72, and 120 hours. These variants allow the model to capture short-term, medium-term, and extended temporal dependencies, respectively.

The input is passed through two fully connected (dense) layers:

$$\mathbf{H}_1 = \texttt{ReLU}(\mathbf{X}\mathbf{W}_1 + \mathbf{b}_1), \quad \mathbf{W}_1 \in \mathbb{R}^{d \times 8d} \tag{4.1}$$

$$\mathbf{H}_2 = \texttt{ReLU}(\mathbf{H}_1\mathbf{W}_2 + \mathbf{b}_2), \quad \mathbf{W}_2 \in \mathbb{R}^{8d \times 16d} \tag{4.2}$$

These layers serve to non-linearly project the input into a higher-dimensional space $\mathbf{H}_2 \in \mathbb{R}^{T \times D}$ with $D = 16d = 64$. This expansion enhances the model's representational capacity, enabling it to learn richer feature interactions and hierarchical abstractions.

To improve convergence in deep networks, all dense layers in the model use **He normal initialisation** (He et al., 2015), formally:

$$W_{ij} \sim \mathscr{N}\left(0, \sqrt{\frac{2}{n_{\text{in}}}}\right) \tag{4.3}$$

where n_{in} is the number of input units to the layer. This initialisation strategy helps prevent vanishing or exploding gradients, particularly in networks using ReLU activations.

4.1.2 Positional Encoding Variants

Following the embedding of input features, an additional step is required to inject information about temporal order into the model. Transformer models are inherently permutation-invariant and lack a built-in mechanism to capture the sequential ordering of input tokens. In the context of financial time series forecasting, where temporal structure is critical, positional encoding plays a vital role in informing the model about the relative or absolute positions of observations. This study investigates two distinct positional encoding strategies tailored to the characteristics of mid-frequency FX data: Local Positional Encoding (LPE) and Learnable Positional Encoding (LPE*).

Local Positional Encoding (LPE).

Local Positional Encoding combines trainable embeddings with a convolutional filter to emphasise short-range temporal locality. Intuition says that in financial markets, local patterns such as microtrends, breakouts, or short-term reversals often carry strong predictive value.

Formally, a learnable matrix $P \in \mathbb{R}^{T \times D}$ is constructed, where T is the sequence length and D is the model dimension. A one-dimensional convolution with kernel size $k = 3$ is applied to P, producing a smoothed encoding $L \in \mathbb{R}^{T \times D}$. This is added element-wise to the projected input representation H_2 to obtain the position-aware input:

$$Z_0 = H_2 + \mathrm{ReLU}(\mathrm{Conv1D}(P; k)) \tag{4.4}$$

This approach biases the model toward learning position-relative patterns within a local window.

Learnable Positional Encoding (LPE*).

As an alternative, this dissertation also explores a fully learnable positional encoding scheme in which each position in the input sequence is assigned a unique, trainable embedding vector, without any convolutional smoothing or locality constraint.

A positional matrix $P \in \mathbb{R}^{T \times D}$ is initialised from a zero-mean normal distribution and updated during training. The encoded input is then computed as a simple additive combination:

$$Z_0 = H_2 + P \tag{4.5}$$

This formulation allows the model to flexibly learn arbitrary positional dependencies, potentially capturing long-range or non-local temporal relationships that may be overlooked by local encoding methods. This is particularly useful in financial contexts where meaningful interactions can occur at variable and often non-contiguous time intervals.

Comparison

The two encoding strategies reflect different inductive biases. LPE imposes a structural prior that nearby time steps are more informative, effectively acting as a regulariser that promotes local temporal awareness. In contrast, LPE* delegates the learning of positional importance entirely to the model, enabling it to discover complex temporal dependencies without predefined assumptions. Both approaches are integrated into otherwise identical

Transformer architectures and compared empirically in Section 5 to assess their suitability for directional signal prediction in the mid-frequency FX spot market.

4.1.3 Transformer Encoder Layers

Once the input sequence has been embedded and enriched with positional information, it is passed through a stack of Transformer encoder layers to model complex temporal dependencies. The model contains a stack of identical transformer encoder blocks $L = 6$. Each block comprises two sublayers:

- Multi-head self-attention with residual connection.
- Position-wise feedforward network with residual connection.

Multi-Head Self-Attention

The core operation of the Transformer is the multi-head self-attention mechanism, which enables the model to learn dependencies across all positions in the input sequence. This is especially useful in financial time series, where relevant historical context for a given event may occur at varying temporal distances.

In the multi-head self-attention module, the input sequence $\mathbf{X} \in \mathbb{R}^{T \times D}$ is linearly projected into queries, keys, and values using distinct learned weight matrices:

$$\mathbf{Q}_i = \mathbf{X}W_i^Q, \quad \mathbf{K}_i = \mathbf{X}W_i^K, \quad \mathbf{V}_i = \mathbf{X}W_i^V, \quad i = 1, \ldots, h \tag{4.6}$$

where $W_i^Q, W_i^K, W_i^V \in \mathbb{R}^{D \times d_k}$ are the learnable projection matrices for the i-th attention head, and h denotes the number of heads. For each head, the scaled dot-product attention is computed as:

$$\text{head}_i = \texttt{Attention}(\mathbf{Q}_i, \mathbf{K}_i, \mathbf{V}_i) = \texttt{softmax}\left(\frac{\mathbf{Q}_i \mathbf{K}_i^\top}{\sqrt{d_k}}\right)\mathbf{V}_i \tag{4.7}$$

Each head learns to attend to different aspects of the sequence. For example, one head may focus on short-term reversals while another captures longer-term trends. The outputs from all heads are concatenated and projected back into the model dimension D using an output projection matrix $W^O \in \mathbb{R}^{hd_k \times D}$:

$$\text{MultiHead}(\mathbf{X}) = \texttt{Concat}(\text{head}_1, \ldots, \text{head}_h)W^O \tag{4.8}$$

The matrix W^O is a learnable linear transformation that merges the multi-head outputs into a single vector of the original model dimensionality, allowing integration of diverse attention patterns.

Pre-Normalisation, Residual Connections, and Dropout

To stabilise training and improve gradient flow, a *pre-normalisation* scheme is used in each sublayer. Specifically, Layer Normalisation is applied *before* the attention and feedforward computations rather than after, as this empirically results in better optimisation dynamics in deep Transformer models. Given the input to the l-th encoder block, $\mathbf{Z}_{l-1} \in \mathbb{R}^{T \times D}$, the self-attention sublayer is computed as:

$$\mathbf{A}_l = \texttt{MultiHeadSelfAttention}(\texttt{LayerNorm}(\mathbf{Z}_{l-1})) \tag{4.9}$$

$$\mathbf{Z}'_l = \mathbf{Z}_{l-1} + \texttt{Dropout}(\mathbf{A}_l) \tag{4.10}$$

The addition of $\mathbf{Z}_{l-1}$ in the second line constitutes a *residual connection*, which directly forwards the input and adds it to the attention output. This architectural mechanism, introduced by He et al. (2016), mitigates the vanishing gradient problem, preserves the identity mapping, and enables the construction of deeper networks by facilitating more stable convergence.

The `Dropout` operation, applied after the attention output, serves as a regularisation technique to prevent overfitting. During training, dropout randomly zeroes a proportion p of the attention output elements. This stochasticity forces the model to rely on distributed representations rather than memorising specific attention paths, which is especially beneficial in data regimes prone to noise or overfitting, such as financial time series.

Formally, the pre-normalised residual self-attention block can be expressed as:

$$\mathbf{Z}'_l = \mathbf{Z}_{l-1} + \texttt{Dropout}_{0.1}\left(\texttt{MultiHeadSelfAttention}\left(\texttt{LayerNorm}(\mathbf{Z}_{l-1})\right)\right) \tag{4.11}$$

where $\texttt{Dropout}_{0.1}(\cdot)$ denotes dropout with a probability $p = 0.1$. This combination of pre-normalisation, residual connectivity, and dropout results in a more stable, expressive, and regularised sequence processing block.

Feedforward Subnetwork

Following the multi-head self-attention mechanism, each encoder block applies a position-wise feedforward neural network (FFN) to each time step independently. This subnetwork introduces additional non-linearity and enables the model to learn higher-order interactions among features. The FFN consists of two dense layers with a ReLU activation in between:

$$\mathbf{F}_l = \texttt{ReLU}(\texttt{LayerNorm}(\mathbf{Z}'_l)\mathbf{W}_{f1} + \mathbf{b}_{f1}) \tag{4.12}$$

$$\mathbf{F}_l = \mathbf{F}_l\mathbf{W}_{f2} + \mathbf{b}_{f2} \tag{4.13}$$

where:

- $\mathbf{W}_{f1} \in \mathbb{R}^{D\times f}$ and $\mathbf{W}_{f2} \in \mathbb{R}^{f\times D}$ are the weight matrices of the two layers (with $f = 256$ as the intermediate feedforward dimension),
- $\mathbf{b}_{f1}, \mathbf{b}_{f2}$ are the bias vectors,
- and `ReLU` denotes the rectified linear unit activation function.

As with the attention block, a pre-normalisation strategy is used. Specifically, LayerNorm is applied to the input $\mathbf{Z}'_l$ prior to the FFN, improving gradient stability and optimisation performance.

Residual Connection and Dropout

A residual connection adds the FFN output back to the input $\mathbf{Z}'_l$, and a `Dropout` layer is applied to reduce overfitting. The complete formulation of the feedforward sublayer is:

$$\mathbf{Z}_l = \mathbf{Z}'_l + \texttt{Dropout}_{0.2}(\mathbf{F}_l) \tag{4.14}$$

where $\texttt{Dropout}_{0.2}(\cdot)$ denotes dropout with a rate of 0.2. This setup, pre-normalisation, two-layer FFN, residual connection, and dropout, enables the network to maintain stable and expressive representations while preserving gradient flow throughout training.

4.1.4 Global Pooling and Classification Head

After processing through L Transformer layers, the final sequence representation $\mathbf{Z}_L \in \mathbb{R}^{T \times D}$ is aggregated via global average pooling:

$$\mathbf{v} = \frac{1}{T} \sum_{t=1}^{T} \mathbf{Z}_L[t] \tag{4.15}$$

This reduces the temporal sequence into a single vector $\mathbf{v} \in \mathbb{R}^D$, representing the learned summary of the input. It is passed through a final dense layer with softmax activation to output the probability vector over the 3 classes:

$$\hat{\mathbf{y}} = \texttt{Softmax}(\mathbf{v}\mathbf{W}_c + \mathbf{b}_c) \tag{4.16}$$

4.1.5 Loss Function

Two distinct loss functions were investigated to address the imbalanced nature of class labels: Focal Loss and Weighted Cross-Entropy. Both approaches aim to enhance the model's sensitivity to the minority classes (labelled 1 and 2), which correspond to upward and downward directional movements, respectively.

Focal Loss

Focal Loss is a modified form of cross-entropy loss designed to address class imbalance by reducing the relative loss contribution from well-classified examples and focussing training on harder, misclassified cases. It achieves this by modulating the standard cross-entropy loss with a factor that down-weights predictions with high confidence, thereby encouraging the model to focus on samples that are more difficult to classify. The loss is defined as:

$$\mathscr{L}_{\text{focal}} = -\sum_{c=0}^{2} \alpha_c (1 - \hat{y}_c)^{\gamma} \log(\hat{y}_c) \cdot \mathbf{1}_{\{y=c\}} \tag{4.17}$$

where α is the class weighting vector and γ is the focusing parameter.

Weighted Cross-Entropy

In an alternative configuration, a Weighted Cross-Entropy loss was employed using class weights computed from the training set distribution. These weights were further adjusted

to assign greater importance to the minority classes. Specifically, classes 1 and 2 were up-weighted using a scaling factor of 3 and 4. The final class weights were applied as:

$$\mathscr{L}_{\text{wce}} = -\sum_{c=0}^{2} w_c \log(\hat{y}_c) \cdot \mathbf{1}_{\{y=c\}} \tag{4.18}$$

where w_c denotes the adjusted class weights.

This dual-loss experimentation enables a comparative evaluation of how loss function design influences classification performance under class imbalance.

4.1.6 Model Configuration

The hyperparameter configuration, including components and parameters that remained constant across all model architectures, is summarised in Table 4.1.

Table 4.1: Transformer Model Configuration

Parameter	**Value**
Input features d	4 (OHLC)
Expanded feature dim D	64
Transformer layers L	6
Attention heads h	4
Feedforward dimension	256
Optimizer	Adam
Learning rate	0.001

4.2 XGBoost Baseline Model

To benchmark the performance of the Transformer architecture, an XGBoost classifier is implemented as a baseline model. XGBoost is a gradient-boosted decision tree ensemble that has demonstrated robust performance in a variety of financial prediction tasks, particularly where the input data can be effectively represented in tabular form (Zhang et al., 2023). In contrast to the sequence-based nature of Transformers, XGBoost operates on fixed-size vector inputs, making it a suitable comparator for assessing the added value of temporal modelling via attention mechanisms.

4.2.1 Input Flattening and Feature Preparation

To conform to XGBoost's expected input format, each input sequence of shape $(n_{\text{samples}}, T, d)$ was flattened into a two-dimensional matrix of shape $(n_{\text{samples}}, T \times d)$, where $T = 72$ and $d = 4$ represent the sequence length and number of features (OHLC), respectively. This transformation results in a 288-dimensional feature vector per sample, effectively discarding the temporal structure of the input while retaining full information content.

4.2.2 Handling Class Imbalance

To address the class imbalance inherent in the label distribution, class-specific weights were manually defined: class 0 (neutral) was assigned a baseline weight of 1.0, while minority classes 1 (upward) and 2 (downward) were both assigned higher weights. These weights were used to compute a sample-weight vector for the training set, thereby amplifying the contribution of minority class samples to the model's objective function. The weighted training data was incorporated using XGBoost's native `DMatrix` structure.

4.2.3 Model Configuration and Training

The XGBoost model was trained using a multi-class softmax objective function (`multi:softprob`) with the following hyperparameters:

Table 4.2: XGBoost Model Configuration

Parameter	**Value**
Objective	Multi-class classification with softmax probabilities
Number of classes	3
Evaluation metric	Multi-class log loss
Max depth	6
Learning rate (η)	0.1
Subsample ratio	0.8
Column subsample ratio	0.8
Number of boosting rounds	200
Early stopping rounds	10

Validation was conducted using a held-out validation set without sample weighting. Early stopping was applied based on validation log-loss to prevent overfitting. The resulting model serves as a baseline for comparison against Transformer-based architectures, especially in evaluating the necessity and effectiveness of temporal attention mechanisms for directional signal classification.

4.3 Results

4.3.1 Softmax Output Distribution Across Classes

An essential component in the evaluation of classification models, is the analysis of predicted class probability distributions. Rather than relying solely on the final class label, examining the full softmax output vector provides a deeper understanding of the confidence of the model and the degree of separation between class boundaries. This is particularly relevant in scenarios involving directional trading signals, where the distinction between high-confidence and low-confidence predictions can significantly impact trading decisions and associated risk. A well-calibrated model should allocate the probability mass in a manner that reflects the true uncertainty of the market state, enabling more informed threshold-based execution logic. Furthermore, this analysis may reveal biases such as systematic over-representation of the majority (neutral) class or failure to confidently predict minority classes, both of which are common in imbalanced financial datasets. Visualising the probability distributions across the 'upward', 'neutral', and 'downward' classes facilitates the identification of such patterns and supports a more granular assessment of the behaviour of the model. In the following, we present comparative visualisations of the predicted probability distributions across different Transformer architectures, each trained under distinct configurations of context window length, positional encoding, and loss function, to evaluate the impact of these design choices on model calibration and predictive confidence.

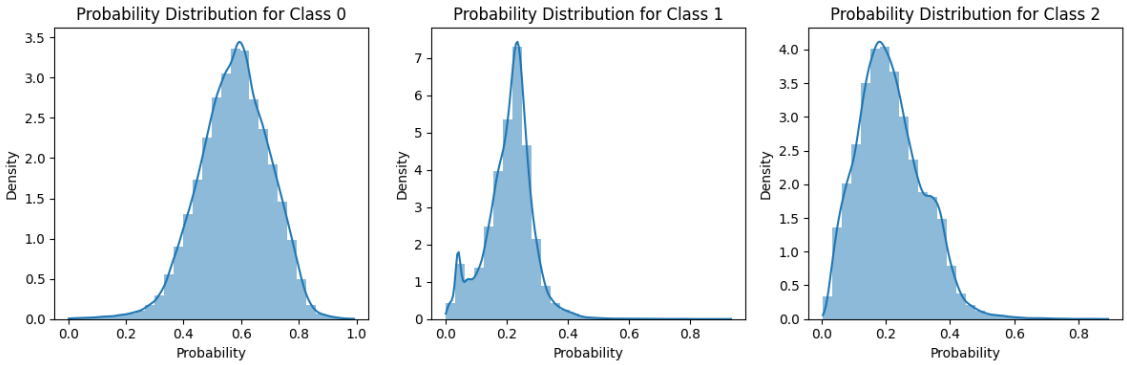

Figure 4.2: Context Window 24, Local Positional Encoding, Focal Loss

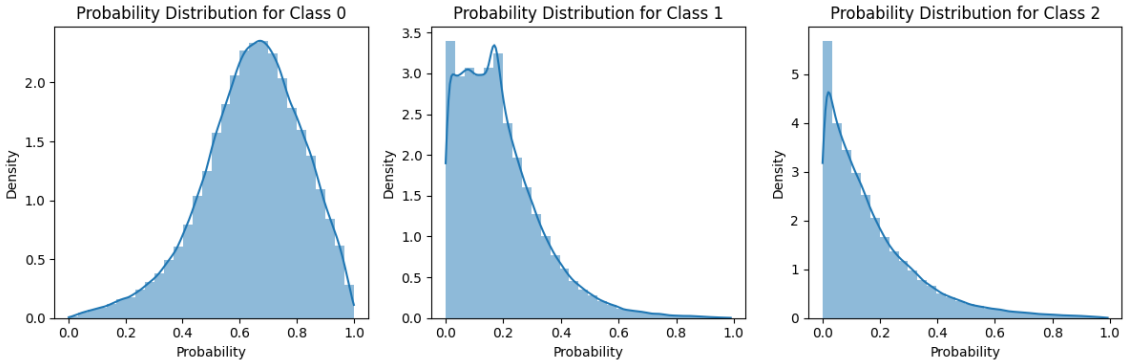

Figure 4.3: Context Window 72, Local Positional Encoding, Focal Loss

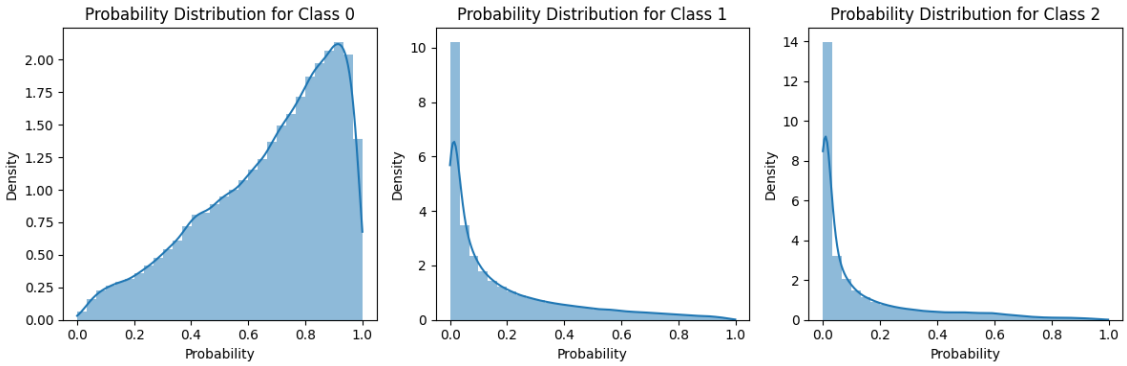

Figure 4.4: Context Window 120, Local Positional Encoding, Focal Loss

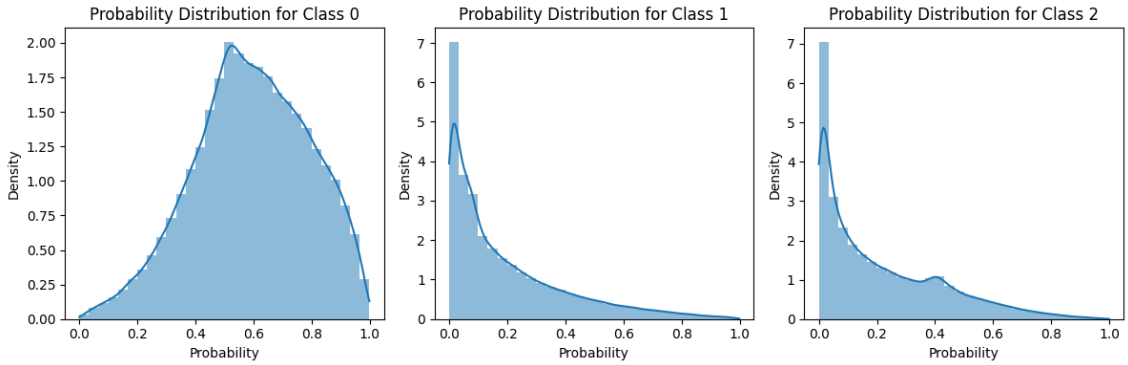

Figure 4.5: Context Window 72, Learnable Positional Encoding, Focal Loss

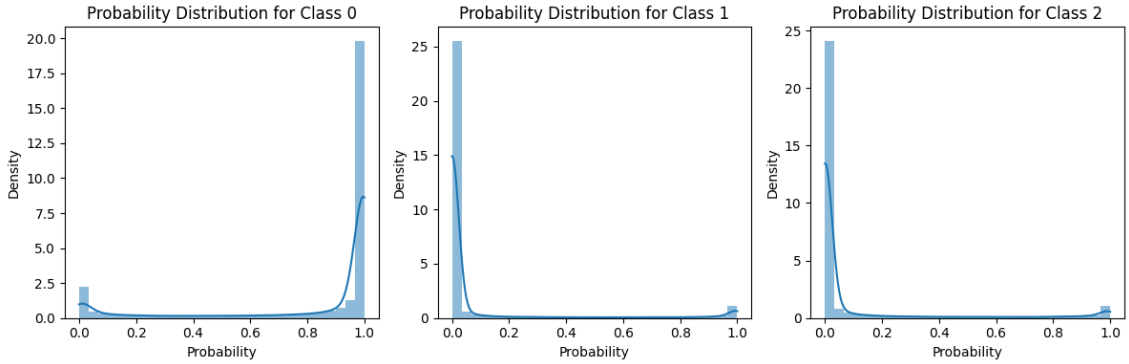

Figure 4.6: Context Window 72, Local Positional Encoding, Weighted Cross Entropy Loss

From the figures presented above, we observe notable variations in the distribution of softmax output across the three directional classes, which provide insight into the influence of architectural and loss function choices on model calibration and confidence. A particularly clear distinction emerges between the outputs generated under Focal Loss and those produced using Weighted Cross-Entropy. As illustrated in Figure 4.6, model trained with Weighted Cross-Entropy exhibit highly polarised probability distributions, where predicted probabilities are overwhelmingly concentrated near the extrema of the interval $[0, 1]$. This behaviour suggests a pronounced overconfidence, as the model tends to assign near-certain probabilities to its predictions, leaving little room for uncertainty or ambiguity. In contrast, the Focal Loss configuration produces distributions that are more dispersed, with a wider spread of values across the unit interval. This more gradual allocation of probability mass indicates that the model retains a degree of caution in its classification decisions, which may be advantageous in noisy or volatile market regimes.

Further differences are evident when comparing positional encoding schemes. The variant employing Learnable Positional Encoding (LPE*), shown in Figure 4.5, results in distributions that are heavily right-skewed for classes 1 (upward) and 2 (downward) with fat-tailed behaviour. This suggests that while the model occasionally exhibits high confidence in these directional predictions, it more frequently assigns relatively modest probabilities, indicating a nuanced treatment of directional signals under this encoding.

Lastly, the effect of the length of the context window is most pronounced in the model's treatment of the neutral class (label 0). As seen in Figures 4.2, 4.3, and 4.4, under the configuration using Focal Loss and Local Positional Encoding, increasing the context window length leads to a marked shift in the output distribution toward higher-confidence predictions for class 0. This implies that with more historical information, the model becomes increasingly confident in identifying periods of market indecision, thereby reinforcing the neutral prediction. This behaviour aligns with the intuition that extended context can improve the model's ability to identify the absence of significant directional momentum.

4.3.2 Training and Validation Loss Progression

An examination of the training and validation loss over successive epochs provides critical insight into the learning dynamics of the Transformer models employed in this study. Training loss reflects the model's performance on the dataset it directly learns from, while validation loss offers an external check on generalisation by measuring performance on previously unseen data. Tracking both metrics allows one to identify overfitting, underfitting, or instability during optimisation. A well-behaved training process typically shows a gradual and smooth reduction in both losses, with a stabilising gap between the two. Conversely, a widening divergence, particularly where training loss continues to decrease while validation loss plateaus or increases, can indicate that the model is memorising training data at the expense of generalisation. This is of particular importance in financial contexts, where noise and non-stationarity can lead to overconfident models that perform poorly in live trading environments. Furthermore, the choice of loss function and its sensitivity to class imbalance may directly influence not only the convergence speed but also the alignment between the learnt representations and the downstream classification objective. Consequently, this subsection presents the evolution of the loss curves for each Transformer configuration, evaluating how the architectural and optimisation choices affect the model's ability to learn meaningful and generalisable patterns from the directional FX data.

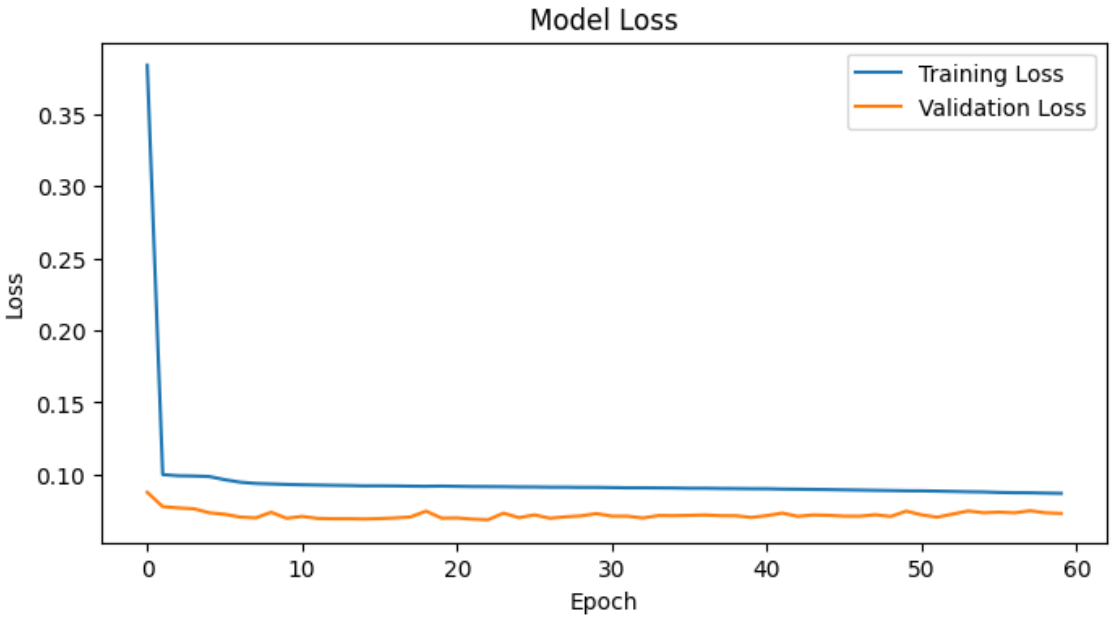

Figure 4.7: Context Window 24, Local Positional Encoding, Focal Loss

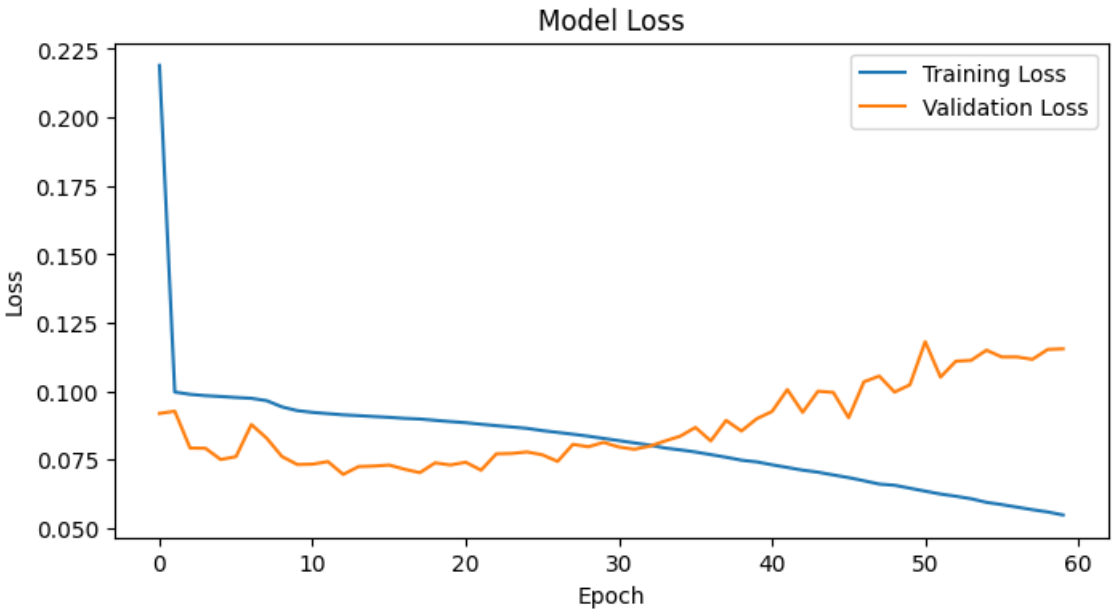

Figure 4.8: Context Window 72, Local Positional Encoding, Focal Loss

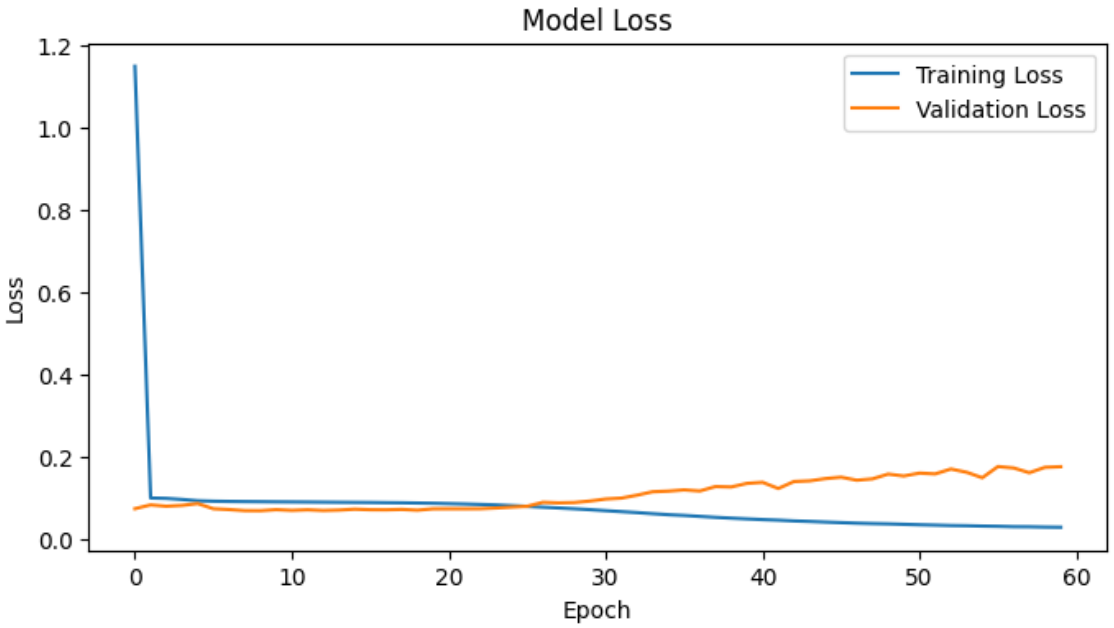

Figure 4.9: Context Window 120, Local Positional Encoding, Focal Loss

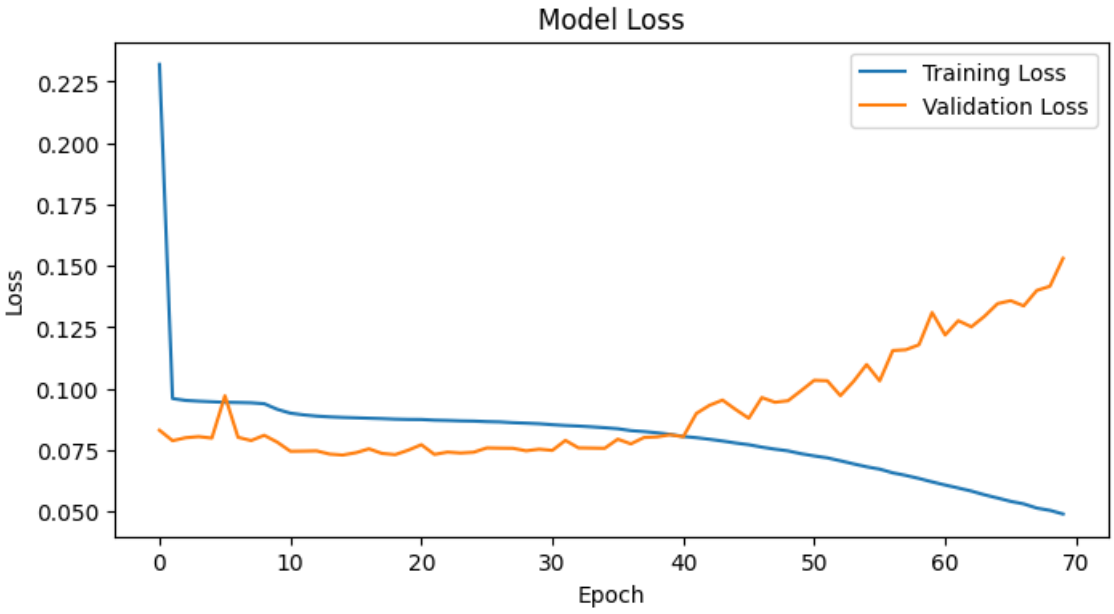

Figure 4.10: Context Window 72, Learnable Positional Encoding, Focal Loss

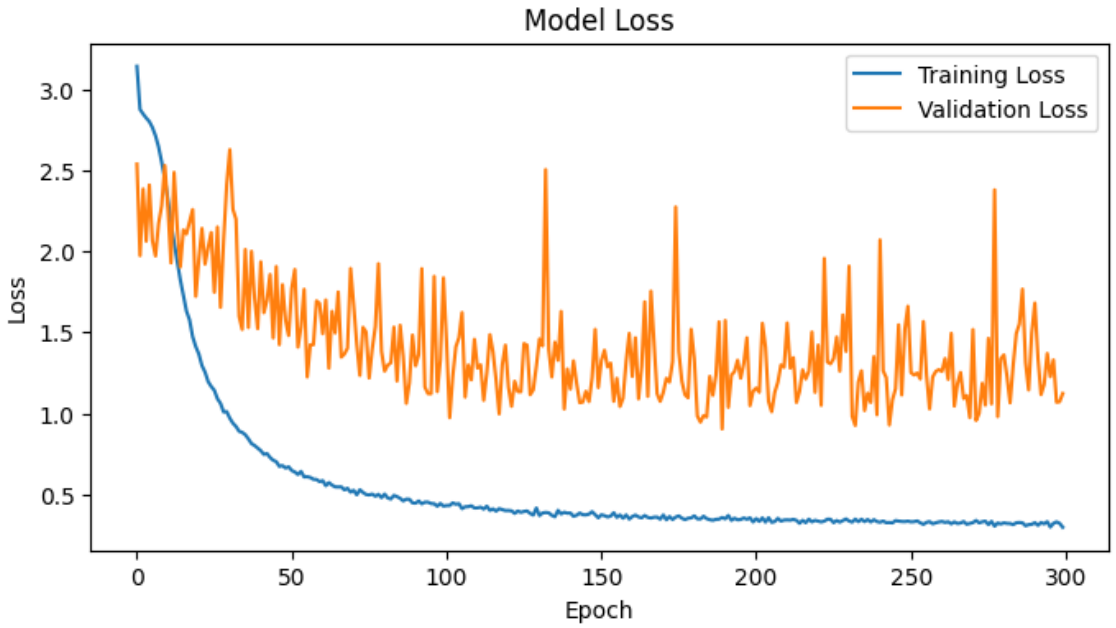

Figure 4.11: Context Window 72, Local Positional Encoding, Weighted Cross Entropy Loss

Visual inspection of the training and validation loss curves across the evaluated Transformer architectures reveals several noteworthy patterns that inform the stability, efficiency, and generalisation behaviour of each configuration. The model trained with Weighted Cross-Entropy loss (Figure 4.11) exhibits markedly slow convergence, requiring approximately 300 epochs to reach a plateau in training loss. This stands in stark contrast to models trained with Focal Loss. The extended training duration under Weighted Cross-Entropy may be attributable to the nature of the static weighting scheme, which imposes a fixed cost structure that is potentially misaligned with the evolving gradient landscape, thereby hindering optimisation efficiency.

Moreover, the validation loss in this configuration displays considerable volatility, manifesting as frequent and pronounced spikes across epochs. Such behaviour suggests that the model's generalisation is unstable and may be highly sensitive to minor fluctuations in the data distribution. This is a critical concern in financial applications, where robustness to noise and regime changes is essential.

In contrast, the model configured with the shortest context window (24 hours) and trained using Focal Loss (Figure 4.7) shows an anomalously flat loss trajectory on both the training and validation sets. This lack of meaningful reduction in loss over time raises concerns about the model's ability to learn signal-relevant patterns from such a limited temporal context. It suggests a failure to capture the underlying structure of the input data, possibly due to insufficient historical information for meaningful temporal abstraction.

Finally, configurations employing longer context windows (72 and 120 hours) in conjunction with Learnable or Local Positional Encoding under Focal Loss show early signs of overfitting. In these setups, the validation loss begins to increase noticeably after roughly 30 epochs, despite continued improvement in training loss. This divergence indicates that the model is beginning to memorise training-specific features at the expense of generalisation, a behaviour commonly associated with high-capacity models trained on limited or noisy data. Such findings underscore the importance of careful early stopping criteria and regularisation strategies when deploying deep Transformer architectures in financial time-series forecasting.

4.3.3 Model Selection and Signal Conversion Strategy

In practical trading applications, the probabilistic outputs generated by the Transformer model must be converted into discrete trading signals to enable actionable decision-making. In this study, the classification model assigns a probability vector $\hat{\mathbf{y}} = [\hat{y}_0, \hat{y}_1, \hat{y}_2] \in [0,1]^3$ to each input sequence, where each component represents the predicted likelihood of a neutral (0), upward (1), or downward (2) movement. To transform this into a single prediction, the primary method employed was the `argmax` function, i.e.,

$$\hat{c} = \arg\max_{i \in \{0,1,2\}} \hat{y}_i,$$

which selects the class with the highest predicted probability. Although alternative thresholding strategies were explored, such as issuing a prediction only when $\hat{y}_1 > \theta$ or $\hat{y}_2 > \theta$ for various thresholds $\theta \in (0.3, 0.9)$, these methods did not yield consistent improvements in predictive precision or trading performance. In fact, overly restrictive thresholds led to the omission of potentially profitable trades, while lenient thresholds increased exposure to noise.

A critical dimension of model evaluation is the trade-off between activity and precision. The *model activity rate* is formally defined as:

$$\text{Activity} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{1}(\hat{c}_i \in \{1,2\}),$$

where N is the total number of predictions and $\mathbf{1}(\cdot)$ is the indicator function. This expression quantifies the frequency with which the model generates directional trading signals. Given that the data set comprises approximately 10% upward (1), 10% downward (2), and 80% neutral (0) labels, a desirable activity range lies between 10% and 20%. An activity level

above 20% may indicate overtrading, while values below 10% suggest missed opportunities. Model activity was found to be highly sensitive to the loss function's class weighting scheme: insufficient penalisation resulted in a degenerate model predicting class 0 exclusively, whereas excessive weighting led to hyperactivity and reduced signal fidelity.

In addition to activity, a bespoke *signal accuracy* metric was introduced to assess the quality of directional forecasts. Since trading actions are only taken when the model predicts class 1 or 2, the metric is defined as:

$$\text{Accuracy} = \frac{TP - FP}{N_{\text{active}}},$$

where

- $TP = \sum_{i=1}^{N} \mathbf{1}(\hat{c}_i = 1 \wedge c_i = 1) + \mathbf{1}(\hat{c}_i = 2 \wedge c_i = 2)$ is the number of correct directional predictions,
- $FP = \sum_{i=1}^{N} \mathbf{1}(\hat{c}_i = 1 \wedge c_i = 2) + \mathbf{1}(\hat{c}_i = 2 \wedge c_i = 1)$ counts predictions that wrongly forecast the opposite direction,
- $N_{\text{active}} = \sum_{i=1}^{N} \mathbf{1}(\hat{c}_i \in \{1,2\})$ is the total number of directional predictions made.

This formulation offers a net measure of directional correctness relative to the full set of active signals, penalising the most costly misclassifications and ensuring alignment with practical trading objectives. The results of this evaluation are summarised in Table 4.3, which compares alternative architectures and loss function configurations.

Table 4.3: Performance Summary of Model Architectures

Model	**Positional Encoding**	**Loss (Params)**	**Context**	**Activity (%)**	**Accuracy**
1	Local Positional Encoding	FL (α=.22; .39; .39)	72	20.1	0.59
2	Local Positional Encoding	FL (α=.28; .36; .36)	72	12.5	1.17
3	Local Positional Encoding	FL (α=.28; .36; .36)	24	5.5	0.36
4	Local Positional Encoding	FL (α=.28; .36; .36)	120	21.6	0.35
5	Local Positional Encoding	WCE (adj=3)	72	62.0	0.88
6	Local Positional Encoding	WCE (adj=4)	72	15.9	0.04
7	Learnable Positional Encoding	FL (α=.28; .36; .36)	72	21.4	-0.06
XGBoost	–	–	–	9.2	**3.70**

From the comparative results summarised in Table 4.3, several important findings emerge regarding model behaviour and selection. In particular, the XGBoost baseline model achieves the highest accuracy score of 3.70 despite exhibiting the lowest activity among all evaluated models at 9.2%. This underscores the robustness of tree-based models when faced with class imbalance, although the limited activity suggests a relatively conservative approach that may fail to capitalise on higher-frequency trading opportunities. Among the Transformer-based architectures, the only configuration to produce a negative accuracy score is Model 7, which employs Learnable Positional Encoding in conjunction with Focal Loss. This result suggests that, within the present modelling setup, fully learnable positional encodings may be more prone to overfitting and may not generalise well to unseen market regimes. In contrast, Local Positional Encoding, which introduces structural inductive bias toward local temporal dependencies, consistently outperforms its learnable counterpart across metrics.

The impact of loss function tuning is also apparent in the comparison of weighted cross-entropy (WCE) configurations. An adjustment factor of 3 (Model 5) leads to excessive model activity of 62.0%, indicative of overtrading, while increasing the factor to 4 (Model 6) dramatically reduces activity to 15.9%. This sensitivity illustrates the delicate balance between encouraging the model to take directional positions and maintaining a disciplined signal cadence.

4.3.4 Activity–Accuracy Trade-offs Across Models

To assess the relationship between model activity and predictive accuracy, we examine both the XGBoost baseline and a range of Transformer variants across different configurations. In particular, XGBoost consistently delivers higher directional accuracy compared to all Transformer architectures, regardless of its activity level.

For XGBoost, the accuracy deteriorates markedly as activity increases. This pattern suggests that the model can only issue high-quality predictions in a relatively small subset of cases - specifically those where its confidence is highest. When the model is constrained to trade more frequently (i.e., take directional positions more often), it is likely forced to act on lower confidence signals. These additional signals appear to be suboptimal, leading to a degradation in performance.

In contrast, the relationship between activity and accuracy for Transformer-based models is less straightforward. Across different Transformer configurations-varying in context window size, positional encoding method, and loss function-the expected inverse correlation between activity and accuracy is not consistently observed. In some cases, increased activity coincides with marginal improvements in accuracy, while in others it leads to deterioration or neutral effects. This lack of a clear trend probably reflects the greater number of interact-

ing components within the Transformer architecture. Elements such as positional encoding schemes, attention span, regularisation mechanisms, and the sensitivity of custom loss functions (e.g., Focal Loss or Weighted Cross-Entropy) can collectively modulate the trade-off between activity and accuracy in complex and configuration-specific ways.

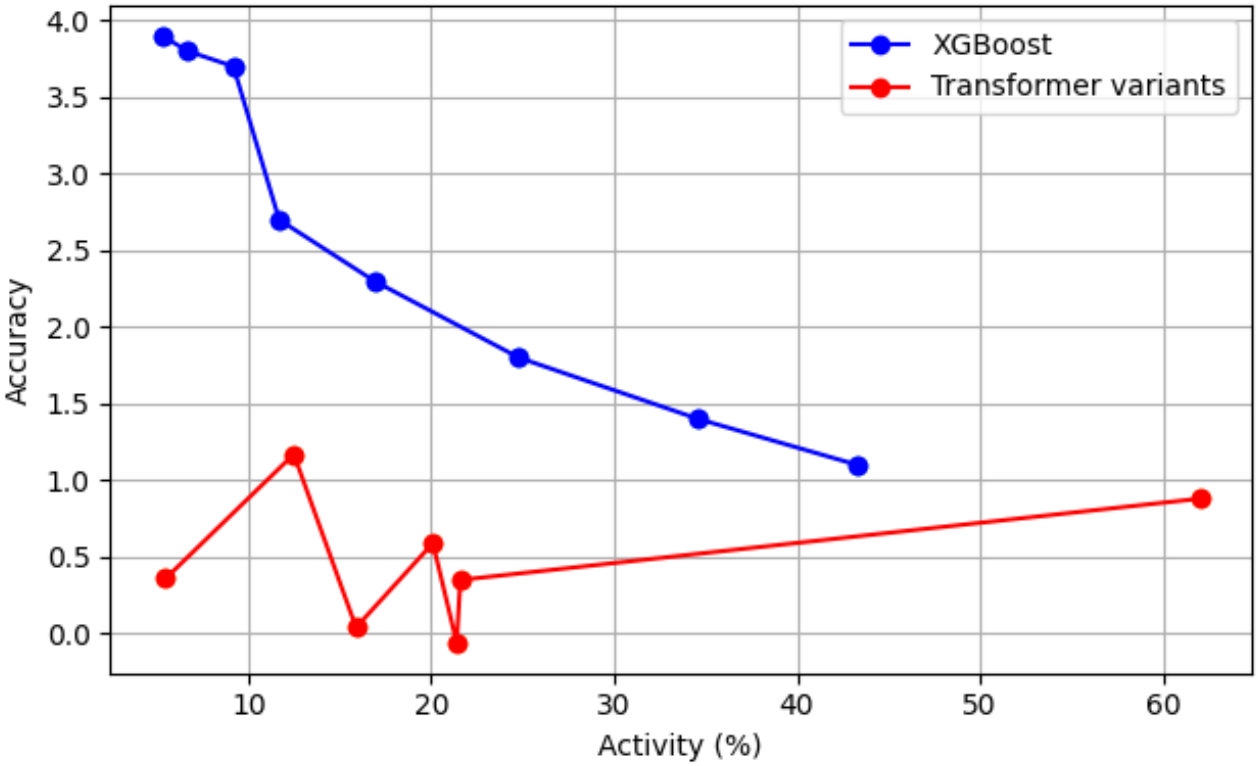

Figure 4.12: Model Activity vs Accuracy

Among all Transformer variants, Model 2 emerges as the best performing architecture, combining an optimal context window of 72, Learnable Positional Encoding, and Focal Loss (FL) with class weights $\alpha = [0.28, 0.36, 0.36]$. This configuration results in a balanced activity level of 12.5% and an accuracy score of 1.17, reflecting both the restriction in signal issuance and the superior directional discrimination. As such, Model 2 is selected for downstream evaluation via backtesting under simulated trading conditions.

5. Backtesting

Following the evaluation of model architectures using statistical metrics, it is essential to validate their practical utility through rigorous backtesting in a simulated trading environment. While classification accuracy and calibration provide useful diagnostic insights, the true value of any signal generation system lies in its ability to support profitable trading under realistic market conditions. To this end, the `Backtesting.py` Python library was employed to implement an event-driven backtest framework capable of simulating FX spot market trading strategies (Kījima and Contributors, 2025).

Consistent with the established data pipeline, only the test partition of the dataset was used for backtesting, ensuring strict temporal separation between model training and performance evaluation. The original test set was further divided into two segments: an *internal training set* spanning 2019-07-16 to 2020-12-31, used exclusively for optimising key trading parameters, and an *evaluation set* covering 2021-01-03 to 2023-12-29, reserved for out-of-sample performance assessment. This split was performed independently for each of the six currency pairs: EUR/USD, GBP/USD, USD/JPY, USD/CAD, AUD/USD, and USD/CHF.

The core trading strategy was designed to exploit the directional signals produced by the models. Given that each signal implicitly specifies an expected directional move over a 12-hour horizon, the principal hyperparameter subject to optimisation was the *stop-loss threshold*, which was calibrated per currency pair to maximise cumulative returns. The trading logic was structured as follows:

- For a signal classified as **1** (upward movement), enter a *long* position at the next hourly open. The position is closed upon reaching either:
    1. a predefined profit target equal to the 90th percentile of the 12-hour forward return distribution calculated during the signal labelling process,
    2. the optimised stop-loss threshold,
    3. or after 12 hours, whichever occurs first.
- For a signal classified as **2** (downward movement), enter a *short* position at the next hourly open. The position is closed upon reaching either:
    1. a predefined profit target equal to the 10th percentile of the 12-hour forward return distribution calculated during the signal labelling process,
    2. the optimised stop-loss threshold,
    3. or after 12 hours.
- For a signal classified as **0** (neutral), no trade is executed.

Each backtest simulation began with an initial capital of 10,000 units of the base currency. Realistic transaction costs of 0.002% per executed trade were incorporated to account for brokerage fees. The position size for each trade was set at 20% of the available capital at the time of trade initiation. This capital allocation strategy was chosen to mitigate risk through position sizing while allowing the strategy to pursue multiple trades concurrently in cases where signals were clustered in time. By allocating a fixed fraction of capital per trade, the strategy avoids overexposure to any single signal and ensures that capital remains available to respond dynamically to evolving market conditions. Importantly, all six currency pairs were optimised and evaluated independently to account for pair-specific volatility.

Table 5.1: Transformer Model 2 Backtest Performance Metrics Across Six Currency Pairs (2021–2023 Evaluation Period).

Metric	**USD/JPY**	**EUR/USD**	**GBP/USD**	**USD/CHF**	**AUD/USD**	**USD/CAD**
Return (%)	-0.52	1.41	-0.52	-1.82	0.23	-1.73
XGBoost Model Return (%)	-0.23	1.62	0.89	5.47	-0.21	-0.02
Return Annualized (%)	-0.14	0.38	-0.14	-0.39	0.06	-0.47
Annualized Volatility (%)	0.61	0.76	0.93	3.36	0.83	0.62
Sharpe Ratio	-0.22	0.49	-0.15	-0.14	0.08	-0.77
Max Drawdown (%)	-1.24	-1.05	-1.99	-8.21	-1.89	-2.16
Trades	347	524	504	721	337	285
Win Rate (%)	47.83	49.91	49.62	49.66	47.83	46.42

The results presented in Table 5.1 highlight several important observations regarding the out-of-sample trading performance of the directional signals generated by the Transformer models across six major currency pairs, with the XGBoost baseline returns included for comparison.

The return profile of the strategy was heterogeneous across currency pairs, with positive absolute returns observed only for EUR/USD (+1.41%) and AUD/USD (+0.23%) while the remaining pairs recorded small to moderate losses. Importantly, even in the strongest performing pair (EUR / USD), the Transformer model underperformed the XGBoost baseline, which achieved a higher return of +1.62%. The disparity is particularly stark in USD/CHF, where the Transformer recorded a -1.82% return versus a substantial +5.47% from XGBoost, underscoring that simpler models may still provide superior robustness under certain market regimes.

Despite these mixed outcomes in return generation, several encouraging signals emerge regarding the model's stability and risk characteristics. The strategy exhibited low annualised volatility across pairs (0.61%–3.36%), consistent with a controlled risk profile and disciplined capital allocation. In addition, maximum drawdowns remained contained (ranging from -1.05% to -8.21%), suggesting resilience to adverse market moves.

Turning to signal quality, the Sharpe ratios reveal limited risk-adjusted performance, with only EUR/USD (0.49) and AUD/USD (0.08) achieving positive Sharpe values. The remaining pairs exhibited negative ratios, confirming that while the Transformer generated directional signals of reasonable accuracy (win rates between 46.42% and 49.91%), the magnitude of profitable trades was insufficient to overcome transaction costs and occasional large losses.

Finally, the volume of trades (285–721 per pair) reflects an active strategy aligned with the objectives of mid-frequency trading. Although high trade counts can increase exposure to noise, consistent execution across pairs demonstrates that the Transformer architecture was capable of generating actionable signals across diverse market conditions.

To complement the quantitative backtest results presented in Table 5.1, this section also presents detailed visualisations of the trading performance for each of the six currency pairs. In each figure, the upper panel shows the equity curve, with key annotations indicating the equity peak, the final value, the maximum drawdown, and the duration of the drawdown. The middle panel displays the profit and loss of individual trades, where the green and red circles are scaled according to the magnitude and sign of trade outcomes. The bottom panel shows the OHLC price chart for the entire evaluation period (2021–2023), with trade entry and exit points overlaid as green and red markers. These visualisations provide insight into periods of profitability and drawdown, allowing a more detailed view into the real-world trading behaviour of the model beyond the aggregate performance metrics.

Figure 5.1: Backtest visualisation for USD/JPY (2021-2023)

Figure 5.2: Backtest visualisation for EUR/USD (2021-2023)

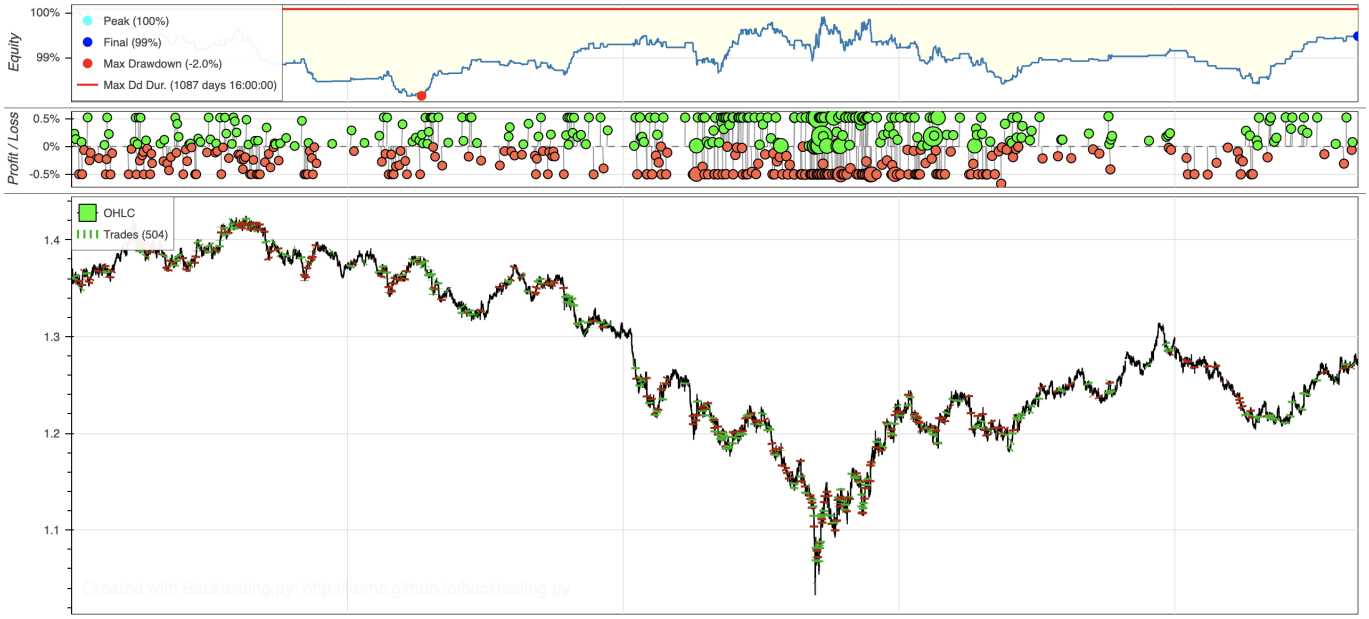

Figure 5.3: Backtest visualisation for GBP/USD (2021-2023)

Figure 5.4: Backtest visualisation for USD/CHF (2021-2023)

Figure 5.5: Backtest visualisation for AUD/USD (2021-2023)

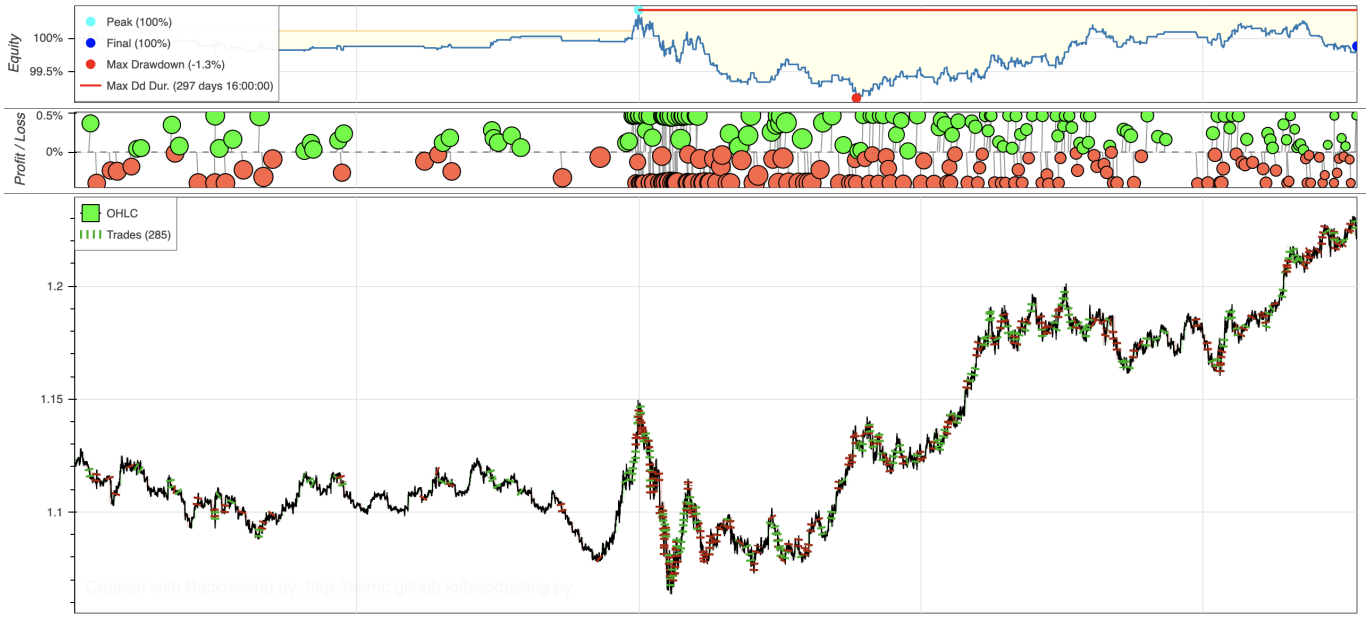

Figure 5.6: Backtest visualisation for USD/CAD (2021-2023)

6. Conclusion

The present study set out to discover whether an encoder-only Transformer augmented with carefully chosen positional encodings, context-window lengths, and imbalance-aware loss functions can deliver economically useful mid-frequency trading signals across the six most liquid FX spot pairs. The evidence collected paints a nuanced picture. On the one hand, attention layers demonstrably learn non-trivial directional structure: the final model (72-hour context, Local Positional Encoding, Focal-Loss weights 0.28/0.36/0.36) produced a net accuracy of 1.17 while trading only 12.5 % of available hours, a clear improvement over other Transformer variants. Yet a well-tuned XGBoost benchmark remained a formidable yardstick, recording a markedly higher accuracy of 3.70 on a comparably restrained activity level of 9.2 %. Back-tests confirmed that the Transformer's statistical edge does not automatically translate into superior returns: in the 2021–23 out-of-sample period only EUR/USD and AUD/USD generated small absolute gains (+1.41 % and +0.23 % respectively), and even the better of these still lagged the corresponding XGBoost strategy (+1.62 %). Nevertheless, the Transformer's risk remained tightly controlled, annualised volatility never exceeded 3.4 % and the deepest drawdown was a manageable –8.2 % so its signals could plausibly add diversification value in a broader portfolio.

Interpreting these findings yields three principal insights. *First*, architectural elegance alone is insufficient: the most dramatic performance swings arose from loss-function design and the attendant activity–precision balance, not from deeper stacks or larger embeddings. Weighted Cross-Entropy, when aggressively scaled, drove activity above 60 % and collapsed accuracy, whereas Focal Loss produced well-calibrated probabilities and steadier convergence. *Second*, inductive bias matters: conferring a modest locality prior through convolution-smoothed positional embeddings lifted every metric relative to fully learnable encodings, echoing market-microstructure intuition that most predictive content in hourly FX lives within short temporal radii. *Third*, while the model's absolute returns were modest, its shallow risk profile suggests attention-based signals can act as low-volatility sleeves within multi-factor systems.

Placed in the wider literature, these results temper recent enthusiasm for Transformers as universal "alpha machines". Bilokon and Qiu (2025) similarly reported only parity with LSTMs once features were restricted to prices, whereas Fischer et al. (2024) found large gains after enriching the input set with cross-sectional and temporal embeddings. The present work therefore supports the view that attention's comparative advantage emerges chiefly when heterogeneous information such as text, order flow, macro data are fused.

Alternative explanations do, however, warrant consideration. Because the experiment confined itself to OHLC prices, it is plausible that the self-attention mechanism was information-

starved; adding news sentiment or options-implied risk-reversals might unlock more of the model's capacity. Likewise, the trichotomous label, based on 10th/90th-percentile forward moves could have blunted nuanced directional gradients, and a multi-horizon or continuous-target formulation might prove more fertile. Finally, evaluation coincided with a post-pandemic tightening regime: alternative volatility states may reward the longer-range dependencies that attention captures.

Several limitations flow from these design choices. The data cover only six G10 pairs and hourly bars; emerging-market currencies or finer-grained intervals might exhibit larger inefficiencies. Transaction costs were modelled as a flat 0.002 %, ignoring slippage and funding, so live implementation risk is understated. Capital was siloed per pair, forfeiting potential risk-netting benefits, and stop-loss levels were tuned on an internal slice, opening the door to mild over-fitting.

These caveats point naturally to future work. Incorporating macro calendars, dealer-quoted order-book depths, or sentiment embeddings would test whether Transformers' cross-feature attention yields a clearer edge. Jointly predicting direction and expected magnitude through multi-task learning, experimenting with adaptive or relative-time encodings that respect irregular FX calendars, and stacking tree ensembles with attention models all represent promising avenues. Reinforcement-learning overlays could replace hard stop-loss rules with policies that optimise risk-adjusted return under realistic cost frictions, while rolling-window evaluations reaching back to pre-2008 regimes would stress-test robustness across structural breaks.

In closing, this dissertation shows that Transformer architectures can be moulded into stable mid-frequency FX signal generators, but also that their incremental edge over mature ensemble methods hinges on loss engineering, locality priors and critically richer information sets. Attention, in short, is a powerful lens; whether it becomes a decisive tool for FX traders will depend on the breadth and depth of the data passing through it.

Bibliography

Bank for International Settlements. Triennial central bank survey: Foreign-exchange turnover in april 2022, 2022. URL `https://www.bis.org/statistics/rpfx22.htm`. accessed 23 June 2025.

P. A. Bilokon and Y. Qiu. Transformers versus lstms for electronic trading. *arXiv pre-print*, 2025.

Ernest P. Chan. *Algorithmic Trading: Winning Strategies and Their Rationale*. John Wiley & Sons, Hoboken, NJ, 2 edition, 2013.

T. Fischer, M. Sterling, and S. Lessmann. Fx-spot predictions with state-of-the-art transformer and time embeddings. Working paper, Kühne Logistics University, Hamburg, 2024.

K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 1026–1034, Santiago, Chile, 2015. IEEE.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778. IEEE, 2016.

HistData.com. Free historical forex data. `https://www.histdata.com`, 2025. accessed 23 June 2025.

John C. Hull. *Options, Futures, and Other Derivatives*. Pearson, Harlow, 10 edition, 2018.

H. Iskandar, W. Xiong, and J. Smith. Arima and xgboost stock-market forecasting: A review. *International Journal of Scientific Research in Engineering and Technology*, 11(3):123–131, 2024.

Makoto Kījima and Backtesting-Py Contributors. Backtesting.py 1.3.6 [software]. `https://github.com/kczat/backtesting.py`, 2025. accessed 23 June 2025.

B. Lim, S. Zohren, and W. M. Ng. Time-series forecasting with the temporal fusion transformer. *Machine Learning and Knowledge Extraction*, 3(4):1000–1020, 2021.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*, 30:5998–6008, 2017.

K. Wood, S. Giegerich, S. Roberts, and S. Zohren. Trading with the momentum transformer: An intelligent and interpretable architecture. Working paper, University of Oxford, Oxford, 2021.

C. Zhang, N. N. A. Sjarif, and R. Ibrahim. Deep-learning models for price forecasting of financial time series: A review of recent advancements (2020–2022). *arXiv pre-print*, 2023.

Jindou Zhang, Weinan Zhang, Zengchang Qin, Yang Yang, and Weinan Wu. Transformer-based frameworks in financial time-series prediction: A survey. *arXiv pre-print*, 2022.

Z. Zhang, B. Chen, S. Zhu, and N. Langrené. From attention to profit: Quantitative trading strategy based on transformer. *OpenReview pre-print*, 2024. ICLR 2024 submission.

H. Zhou, S. Zhang, J. Hou, Y. Yang, X. Liu, and L. Wang. Informer: Beyond efficient transformer for long-sequence time-series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 11106–11115, 2021.