MSPCIFormer: A Multi-Scale Patching Channel-Independent Transformer for Cryptocurrency Price Forecasting

Huali Zhao | **Martin Crane** | **Marija Bezbradica**

[1]School of Computing, ADAPT Research Centre, Dublin City University, Glasnevin, Dublin9, Ireland

Correspondence
Corresponding author Huali Zhao, School of Computing, ADAPT Research Centre, Dublin City University, Glasnevin, Dublin9, Ireland.
Email: huali.zhao3@mail.dcu.ie

Abstract

Forecasting cryptocurrency prices remains challenging due to extreme volatility, regime-dependent dynamics, and unstable cross-asset correlations. Statistical methods such as ARIMA and GARCH assume stationarity and linear dependence structures, making them inadequate for capturing non-linear temporal patterns in high volatile cryptocurrency data. Conventional machine learning methods often require hand-crafted features and fail to capture the sequential temporal dependencies inherent in price series. Recurrent deep learning approaches such as recurrent neural networks (RNNs) and LSTMs address the issues but suffer from limited parallelization, vanishing gradients, and difficulty in learning multi-scale temporal patterns. The advances in Transformer have demonstrated strong capability in capturing long-range temporal dependencies through self-attention while enabling parallel computation. However, canonical Transformer still struggle with noisy, volatile financial time series due to computational complexity and sensitivity to irrelevant temporal patterns. These limitations motivate the exploration of Transformer-based architectures. Our study makes three key contributions. First, we propose MSPCIFormer, a novel Transformer-based architecture that integrates multi-scale patching with channel-independent (CI) modeling to capture heterogeneous temporal dynamics while mitigating noise from time-varying inter-asset correlations. Second, we conduct comprehensive experiments comparing MSPCIFormer with state-of-the-art Transformer models and strong time-series forecasting baselines, evaluated using both statistical and economic metrics across multiple forecasting horizons. Third, we establish a unified evaluation framework incorporating both Hold-out and Walk-forward evaluation framework with economical metrics for regime-robustness testing. Empirical results demonstrate that MSPCIFormer achieves the best or tied-best predictive accuracy among all Transformer-based baselines across three cryptocurrency assets, while maintaining competitive and stable performance across diverse market regimes.

KEYWORDS
Cardano, Cryptocurrency price forecasting, Transformer, Multi-scale Patching, Channel-independent processing

1 | INTRODUCTION

Since their inception with the white paper of Satoshi Nakamoto (Nakamoto, 2008) proposing a decentralized, peer-to-peer electronic cash system, Cryptocurrencies have transitioned from experimental digital currencies to mainstream financial assets (Wątorek et al., 2023). They have been increasingly adopted by hedge funds and institutional investors. PwC's 2025 Annual Global Crypto Hedge Fund Report indicates that 55% of traditional hedge funds surveyed has exposure to digital assets, up from 47% in 2024, reflecting a growing trend in institutional adoption (PWC, 2025). As of 18 December 2025, Bitcoin has reached a global market capitalization of $1.7 trillion (CoinMarketCap, 2025), reflecting its dominance and mainstream acceptance. However, Bitcoin's reliance on the energy-intensive Proof-of-Work (PoW) consensus mechanism raises significant environmental concerns. In contrast, newer cryptocurrencies have adopted eco-friendly consensus protocols to balance innovation with sustainability. Among these, Cardano (ADA) has emerged as a leading alternative, with a market capitalization of $13 billion, ranking it among the top 10 cryptocurrencies as of 18 December 2025 (CoinMarketCap, 2025). ADA stands out as an eco-friendly cryptocurrency, leveraging a Proof-of-Stake (PoS) protocol that achieves substantial energy efficiency while ensuring scalability and security (Jayavarma et al., 2025). With PoS, users can stake ADA tokens for network validation and governance, thereby promoting a decentralized and energy efficient ecosystem that is attractive to investors seeking sustainable alternatives to comparable Proof-of-Work coins (Iuga et al., 2024). In addition, research (Anderson, 2023) identifies a pronounced lead-lag dynamic between Bitcoin and Cardano, in which Bitcoin consistently anticipates subsequent movements in ADA with a quantifiable delay. Furthermore, a recent study B. Chen & Sun (2024) shows that ADA acts as net exporter of both skewness and kurtosis spillovers, suggesting that its return distributions significantly influence the broader market. These studies suggest that ADA is embedded within a broader information transmission structure in the cryptocurrency market, making it an informative asset for examining price predictability and cross-asset spillover behavior.

Accurate price forecasting for ADA is thus essential to mitigate risks and optimize returns in its volatile market. However, a recent survey reports (Yousaf et al., 2025) that among the 234 cryptocurrency forecasting studies published from 2016 to 2023, only 1.8% of deep learning-based forecasting studies cover ADA, highlighting a substantial gap in the existing literature.

To bridge this gap, it is essential to understand the various methodologies used to forecast cryptocurrency prices. Our review of the literature categorizes the prevailing cryptocurrency forecasting methodologies into three primary types of frameworks: statistical models (Zhu, 2023; H. Yildirim & Bekun, 2023), conventional machine learning algorithms (Chaudhary & Saroj, 2023; Alsini et al., 2024; Z. Zhou et al., 2023), and deep learning architectures (Seabe et al., 2023; Kiram et al., 2025; Omole & Enke, 2024; Bourday et al., 2024). While statistical models like ARIMA remain widely used in time series forecasting, they possess inherent structural constraints. Notably, these models are typically built on the assumptions of linearity and stationarity (Tang & Xie, 2025; Madhulatha & Ghori, 2025) and necessitate domain-specific configurations such as trend, seasonality and differencing, making the modeling process time consuming and error-prone (Li et al., 2020). To mitigate these structural deficiencies, conventional machine learning (Z. Zhou et al., 2023; Alsini et al., 2024) and deep learning frameworks (Kiram et al., 2025; Omole & Enke, 2024) have been introduced. Despite their potential, machine learning and deep learning models frequently yield forecasting discrepancies due to the inherent complexities of cryptocurrency time series such as high volatility and regime[†] shifts (Kehinde et al., 2025), and vanishing and exploding gradients, which restrict their ability to capture long-term temporal dependencies (Li et al., 2020; T. Zhou et al., 2022; Zerveas et al., 2021).

Introduced by Vaswani et al. (Vaswani et al., 2017), the Transformer architecture replaced recurrence and convolution with a parallelizable multi-head self-attention mechanism that is capable of modeling long-term dependencies, enabling it to capture temporal patterns without the gradient vanishing or exploding issue. After Transformer-based models success in natural language processing (Devlin et al., 2019) and computer vision (Dosovitskiy et al., 2021), studies (Nie et al., 2023; Y. Liu et al., 2024; Cai et al., 2023) have increasingly demonstrated their effectiveness in time series forecasting. In this context, recent studies have shown that patching techniques can improve the accuracy and computational efficiency of Transformers. However, despite the demonstrated success of Transformer architectures in cryptocurrency price forecasting (Singh & Bhat, 2024; Kehinde et al., 2025; B. Yildirim & Taskiran, 2024), only a small proportion of existing studies (approximately 1% (Yousaf et al., 2025)) employ Transformer-based models and the application of patching techniques in cryptocurrency forecasting remains relatively underexplored and has not yet been systematically evaluated, revealing a notable gap in the existing literature. In addition, existing studies suggest that many cryptocurrency forecasting models exhibit strong performance under backtesting, yet may not generalize well in real-world application (Yousaf et al., 2025; Ang et al., 2026). This highlights the limited adoption of forward-testing frameworks and economically meaningful evaluation measures. These limitations underscore the need for a systemic investigation into Transformer-based models under realistic evaluation frameworks to better understand their robustness and to enable their effective adaption and optimization for cryptocurrency markets.

This study addresses the aforementioned gaps through the following contributions:

- We propose a Multi-Scale Patching Channel-Independent Transformer (MSPCIFormer) for cryptocurrency forecasting. The proposed method enhances the canonical Transformer through multi-scale patching strategy that captures diverse temporal patterns and reduces computational complexities. In addition, its channel-independent processing is shown to mitigate the noise introduced by time-varying cross-cryptocurrency relationships (Wątorek et al., 2025) in the data.
- We propose a unified evaluation framework for financial time series forecasting that integrates hold-out validation, walk-forward validation, and both statistical and economic evaluation metrics to enable reproducible and practically relevant evaluation. Within this framework, MSPCIFormer is comprehensively evaluated against state-of-the-art (SOTA) Transformer-based architectures and established time series forecasting baselines. Furthermore, computational efficiency is assessed through training time per epoch, inference latency, and memory consumption, providing a holistic evaluation of predictive accuracy, robustness, and practical deployment feasibility.
- Beyond BTC and ETH, which are primary focus of most existing cryptocurrency forecasting studies (Yousaf et al., 2025), we extend our experiments to ADA price data, providing a more comprehensive evaluation of model generalisability across diverse cryptocurrency market dynamics.

[†] Cryptocurrency market regimes are defined based on volatility: Low volatility (price movement is stable and predictable), high volatility (risker but with greater return potential) and distress (sharp downward movements)(Agakishiev et al., 2025).

The remainder of this paper is structured as follows: Chapter 2 reviews the existing studies on cryptocurrencies price prediction; outlines the weakness of the canonical Transformer for time series forecasting; and surveys techniques adopted by SOTA Transformer-based time series forecasting models to address the limitations. Chapter 3 details the architecture of our proposed model. Chapter 4 describes the experiments setup. Chapter 5 presents a comprehensive evaluation of MSPCIFormer against SOTA Transformer-based models and established time series forecasting models for cryptocurrency price forecasting. Finally, Chapter 6 summarizes the main findings and outlines directions for future research.

2 | RELATED WORK

This chapter reviews the existing literature on cryptocurrency price forecasting, outlining the limitations of statistical methods, conventional machine learning models, and deep learning approaches. Although the canonical Transformer has demonstrated strong capability in capturing long-term temporal dependencies without suffering from the vanishing-gradient issues inherent in recurrent architectures (Vaswani et al., 2017), Section 2.2 highlights its limitations when applied to time-series forecasting. Sections 2.3, 2.4, and 2.5 then examine SOTA Transformer-based time series forecasting architectures and survey the techniques employed to enhance the Transformer's performance in this domain. Despite its theoretical advantages, the Transformer remains relatively underexplored in the context of cryptocurrency forecasting, leaving an important gap in the current literature.

2.1 | Cryptocurrency price forecasting methodologies

The growing interest from both institutional and individual investors has made cryptocurrency price forecasting a significant research topic. Numerous studies have employed a wide range of models to capture temporal patterns in cryptocurrency data. These approaches can broadly be categorized into statistical methods, conventional machine learning techniques, and deep learning models.

Statistical models are mathematical frameworks that produce quantifiable predictions by leveraging statistical assumptions and probabilistic processes (Otabek & Choi, 2024). ARIMA is one of the most widely used statical models in cryptocurrency forecasting. It utilizes its own lagged values and lagged forecast error to facilitate the forecasting of future values (Otabek & Choi, 2024). Azari's study (Azari, 2019) examined the effectiveness of ARIMA for cryptocurrency forecasting using Bitcoin closing price data in USD from September 2015 to September 2018. The results show that ARIMA performs well in short-term forecasting, particularly when the price is stable. However, for long-term forecasting, the model is error-prone. Similarly, Yang (Si, 2022) conducted a comparative analysis of ARIMA models with different hyperparameter configurations using daily Bitcoin closing prices from January 2017 to September 2022. The study found that ARIMA consistently struggle to accommodate price fluctuations and is more effective for short-term forecasting than for longer-term predictions. Latif et al. (Latif et al., 2023) compared the ARIMA model and the Long Short-Term Memory (LSTM) model using one year of BTC closing price data at 10-minute intervals, spanning December 21, 2020, to December 21, 2021. The results showed that the LSTM model outperformed ARIMA, achieving an accuracy of 99.73%. These studies underscore the limitations of classical statistical models in capturing the nonlinear dynamics and extreme volatility inherent in cryptocurrency markets. Their shortcomings highlight the need for more robust approaches to achieve improved forecasting accuracy and stability.

To overcome the limitations of statistical models, the use of machine learning techniques in cryptocurrency research has increased in recent years (Ren et al., 2022). Chaudhary and Saroj (Chaudhary & Saroj, 2023) compared linear regression (LR), support vector machines (SVM), and decision trees (DT) for forecasting the next 5 days of BTC, ETH, DOGE, and BCH closing prices. Their experiments used daily price data from 31 December 2016 to 21 May 2022 for BTC, ETH, and DOGE, and from 23 July 2017 to 20 May 2022 for BCH. LR served as a linear baseline, SVM as a non-linear kernel-based model, and DT as a simple tree-based learner. The results show that LR delivered the most consistent performance and achieved the highest accuracy for BTC (98.61%). SVM performed best on BCH (92.92%) but poorly on DOGE, indicating sensitivity to data characteristics. DT produced stable but generally inferior predictions. Overall, the study suggests that while non-linear ML models can be advantageous under certain conditions, simple linear models may still provide strong and reliable forecasts for more stable cryptocurrencies such as Bitcoin. Similarly, Acosta et al. (Acosta & Arana, 2023) evaluated the forecasting performance of SVM, LSTM, and Bidirectional LSTM (BiLSTM) models using daily Cardano data, including open, close, high, low, and volume metrics, from January 2021 to April 2023. Their findings indicate that BiLSTM outperforms SVM and LSTM for prediction horizons of two and three weeks, achieving lower error rates, while all three models exhibit comparable accuracy for one-day, one-week, and one-month forecasts. Alsini et al. (Alsini et al., 2024) evaluated a Bagged Tree (BT) ensemble for generating cryptocurrency buy signals using 15-minute OHLCV data for BTC, ETH, ADA, and BNB from November 2018 to November 2021. After constructing features from raw prices and technical indicators, the BT model was compared against DT, KNN, neural networks, and random forests (RF). The results show that BT achieved leading or near-leading performance across all signal types, including 100% accuracy for RSI_buy_signal (tied with DT and RF), 95.73% for MACD_buy_signal (best overall), and 89.39% for BB_buy_signal (second to RF). These findings indicate that BT is an effective method for producing reliable trading signals in volatile cryptocurrency markets.

While traditional machine learning methods have improved cryptocurrency forecasting, deep learning approaches have emerged as a powerful alternative (Alnami et al., 2025). Ahmadi et al. (Ahmadi et al., 2024) compared Multi-Layer Perceptrons (MLP) and Simple Recurrent

Neural Networks (SimpleRNN) for ADA, BTC, ETH, BNB, and XRP, using dialy price data from 2017 to December 2023, incorporating features like simple moving averages (SMA), exponential moving averages (EMA), and trading volume. They found MLP achieved a lower MSE (0.000296) than SimpleRNN (0.000779) for ADA price prediction, highlighting the potential of feedforward neural networks. Using daily prices of ten major cryptocurrencies, including ADA, from November 2017 to September 2022, AlMadany et al. (AlMadany et al., 2024) evaluated a range of models: ARMA, GARCH, EGARCH, LSTM, and hybrid models such as EGARCH-LSTM. Their results indicate that hybrid models, particularly EGARCH-LSTM, slightly outperform others across coins, though gains are marginal for ADA. Penmetsa et al. (Penmetsa & Vemula, 2023) conducted a comparative analysis of LSTM and Transformer models for next day price prediction of BTC, ETH, and LTC, using the preceding 10 days of data as input. The dataset comprised 1463 days of daily open, high, low, close prices, and trading volume for these cryptocurrencies, starting from June 16, 2019 to June 16 2023, covering many momentous events such as Covid lockdowns, the outbreak of the Russia-Ukraine, trade disputes between US and China. They found that the Transformer model outperformed LSTM, and the inclusion of momentum and volatility indicators further improved the performance of both models. Yildirim et al. (B. Yildirim & Taskiran, 2024) proposed the Optuna-Based Optimized Transformer (OBOT) for Bitcoin price forecasting. Before applying the vanilla Transformer model, an Optuna-based optimization process was conducted to select appropriate hyperparameters. The study utilized a dataset of historical Bitcoin closing prices, recorded at 4-hour intervals, from January 1, 2020, to January 1, 2024. OBOT was compared with ARIMA, ARIMAX, eXtreme Gradient Boosting (XGBoost), Artificial Neural Networks (ANN), Recurrent Neural Networks (RNN), LSTM, Stacked LSTM, Bi-Directional LSTM, and Temporal Convolutional Network (TCN). The results demonstrated that the proposed method outperformed all other models, achieving an Root mean Squared Error (RMSE) of 0.0079 and a Mean Absolute Error (MAE) of 0.0122.

Although the machine learning methods and deep learning methods have shown their potential, they demonstrate forecasting discrepancies due to the high volatility and regime shifts features in cryptocurrency market (Kehinde et al., 2025), and the inherent vanishing and exploding gradients problems (Li et al., 2020; T. Zhou et al., 2022; Zerveas et al., 2021). Built upon self-attention mechanism, Transformer-based models are able to capture complex temporal patterns while mitigating gradient instability (Kehinde et al., 2025; Zerveas et al., 2021). Studies such as (Penmetsa & Vemula, 2023; B. Yildirim & Taskiran, 2024) have demonstrated the effectiveness of Transformer models for cryptocurrency forecasting, however, the overall body of research applying Transformer-based architectures in this domain remains limited (Yousaf et al., 2025). In addition, although prior studies have examined ADA price forecasting (Alsini et al., 2024; Ahmadi et al., 2024), the majority of existing research focus primarily on BTC and ETH, leaving merging, eco-friendly altcoins like ADA underexplored. Moreover, the survey studies (John et al., 2024; Yousaf et al., 2025) emphasize that variations in datasets and evaluation metrics in prior research make rigorous and fair comparisons between forecasting models difficult. To mitigate the problem, the authors suggest a standardized evaluation framework that allows controlled and reproducible experimentation alongside open-source code to facilitate reproducibility, validation and further research. Following the suggestions, this study adopts a standardized time series forecasting (TSF) framework originally proposed by Zhou et al (H. Zhou et al., 2021), which has since been widely adopted in TSF research works (Nie et al., 2023; Y. Liu et al., 2024; Cai et al., 2023; T. Zhou et al., 2022; Wu et al., 2022). Building upon this framework, we further integrate walk-forward validation and economically meaningful evaluation metrics to facilitate evaluation of cryptocurrency forecasting models. To address the aforementioned gaps, we evaluate MSPCIFormer on BTC, ETH and ADA under the proposed evaluation framework using both statistical and economically meaningful evaluation metrics to comprehensively assess the models performance against SOTA benchmark methods.

2.2 | The vanilla Transformer model on time series forecasting

The core of the Transformer's success lies in its multi-head attention mechanism, which allows the model to selectively focus on any segment of the historical data, making it efficient on identifying recurring patterns with long-term dependencies (Li et al., 2020). However, the canonical Transformer faces limitations: quadratic computational complexity and insufficient handling of temporal patterns. The self-attention mechanism's dot product operates between token embeddings, which introduces quadratic complexity with respect to sequence length, making it computationally prohibitive for long time series. Moreover, the atomic nature of this operation hinders the model's ability to capture intrinsic information at varying time scales (T. Zhou et al., 2022). It is known that this can make the model prone to anomalies and bring underlying optimization issues (Li et al., 2020). To address these challenges, Transformer variants (Li et al., 2020; H. Zhou et al., 2021; Wu et al., 2022; T. Zhou et al., 2022; S. Liu et al., 2022; Zhang & Yan, 2023; Cai et al., 2023; Wang et al., 2024; Cai et al., 2023; Y. Liu et al., 2024), have been proposed, offering improvements in both computational efficiency and ability to capture temporal dynamics, thus enhancing Transformer's performance in time series forecasting.

Table 1 presents an overview of the enhancement strategies adopted by SOTA Transformer variants, including their attention computation complexities and the datasets utilized to assess their performance and effectiveness. Table 2 offers detailed descriptions of each dataset. While these SOTA models have been validated on diverse real-world datasets and shown to outperform various deep learning and machine learning baselines, it is noteworthy that their evaluation on cryptocurrency price data remains limited.

TABLE 1 Transformer-based models. The table is organized chronologically by publication date and ranked by the computational complexity of the attention mechanism.

Model	Method	Attention Complexity	Dataset (see Table 2)
LogSparse (Li et al., 2020)	Sparse attention	$\mathcal{O}(L(\log L)^2)$	Synthetic dataset; Electricity (fine); Traffic (fine); Traffic (coarse); Solar; Wind
Informer (H. Zhou et al., 2021)	Sparse attention with ProbSparse	$\mathcal{O}(L \log L)$	ETT; ECL; Weather
Autoformer (Wu et al., 2022)	Sparse attention with auto-correlation	$\mathcal{O}(L \log L)$	ETT; Electricity; Exchange; Traffic; Weather2; Illness
Fedformer (T. Zhou et al., 2022)	Frequency-enhanced attention	$\mathcal{O}(L)$	ETT; Electricity; Exchange; Weather2; Illness
Pyraformer (S. Liu et al., 2022)	Pyramidal attention module	$\mathcal{O}(L)$	Wind2; App Flow; Electricity2; ETT
Crossformer (Zhang & Yan, 2023)	Two-stage attention (TSA) with patching	$\mathcal{O}((L/P)^2)$	ETT; Weather; ECL; Illness; Traffic
MSGNet (Cai et al., 2023)	Patching with graph convolution	$\mathcal{O}((L/P)^2)$	Flight; Weather; ETT; Electricity; Exchange
TimeXer (Wang et al., 2024)	Patching with cross-variable attention	$\mathcal{O}((L/P)^2)$	ECL; Weather; ETT; Traffic
iTransformer (Y. Liu et al., 2024)	Channel-dependent attention with patching (M= length)	$\mathcal{O}(M^2)$	ETT; Electricity; Exchange; Traffic; Weather2; Solar; PEMS
MSPatch (Cao et al., 2025)	Multi-scale patching with hybrid patch convolution	$\mathcal{O}((L/P)^2)$	ETT; Weather; Solar; Electricity; Traffic

2.3 | Reducing computational complexity of Transformer

In an effort to mitigate the computational complexity of the attention mechanism in Transformers, LogSparse (Li et al., 2020) was proposed by Li et al. which introduces sparsity into the attention operation. As noted previously, in the standard attention mechanism, each data point attends to all previous data points, leading to quadratic complexity. In contrast, LogSparse restricts attention by allowing each data point to attend only to a subset of previous data points, with an exponentially increasing step size, in addition to attending to itself. This modification reduces the computational complexity from quadratic to $\mathcal{O}(L(LogL)^2)$.

Informer (H. Zhou et al., 2021) and Autoformer (Wu et al., 2022) have taken a further step by improving the complexity to $\mathcal{O}(L(logL))$. The Informer model introduced ProbSparse attention which allows each key to attend only to a limited number of the most significant queries, rather than all queries. This is accomplished by assessing a sparsity measure for each query and selecting only the top queries based on this metric. The Autoformer model achieves efficiency through its novel design that incorporate a decomposition strategy. It separates seasonal and trend components, allowing the model to focus on capturing long-range dependencies more efficiently. Instead of point wise attention, the Autoformer model introduced an Auto-Correlation mechanism that aggregates sub-series by similarity from underlying periods. Benefiting from the inherent sparsity and sub-series-level representation aggregation, This Auto-Correlation mechanism can simultaneously benefit the computation efficiency and information utilization (Wu et al., 2022).

More recently, Fedformer (T. Zhou et al., 2022) and Pyraformer (S. Liu et al., 2022) have achieved an even greater reduction, bringing the complexity down to $\mathcal{O}(L)$. Fedformer employs a decomposition strategy that separates seasonal and trend components, along with a frequency domain strategy that randomly selects a fixed number of Fourier components. This innovative combination allows Fedformer to achieve linear complexity while maintaining high accuracy. Pyraformer introduced Pyramidal Attention Module (PAM) which can be depicted as a pyramidal graph. In this architecture, each node a specific level of temporal resolution, forming a pyramidal graph. Attention scores are computed between nodes connected within this hierarchical structure, allowing Pyraformer to capture both inter-scale(between different resolutions) and intra-scale (within the same resolution) dependencies effectively, achieving sparsity in attention calculations.

2.4 | Patching - Enhancing temporal information modeling and reducing computational complexity for Transformer

Applying the vanilla attention mechanism to time series data, the dot product operation is applied on single time steps. However, a single time step in a time sequence does not have semantic meaning like a word

TABLE 2 Dataset description. This table presents detailed descriptions of the datasets utilized by the Transformer-based models.

Data	Description	References
Synthetic datase	Data generated using a set of sinusoidal signals.	(Li et al., 2020)
Electricity-f (fine)	15 min electricity usage data from 370 customers.	(Li et al., 2020)
Electricity-c (coarse)	As Electricity-c but for hourly intervals.	(Li et al., 2020)
Traffic-f (fine)	20 min occupancy data for 963 freeways in SF, CA.	(Li et al., 2020)
Traffic-c (coarse)	As Traffic-f (fine) but for hourly intervals.	(Li et al., 2020)
Solar	Hourly solar power data from 137 PV plants in AL from 01/2006-08/2006.	(Li et al., 2020)
Wind	Daily energy potential estimates from 28 countries 1986-2015, data as a percent of plant max output.	(Li et al., 2020)
M4	Consists of 414 hourly time series from the M4 competition.	(Makridakis et al., 2018)
ETT (Electricity transformer Temperature)	2 years of data from two Chinese counties, comprises: ETTh1&ETTh2(1 hour frequency). Data points comprise oil temperature and 6 associated power load features.	(H. Zhou et al., 2021)
ECL (Electricity Consuming Load)	Derived from electricity-c and electricity-f datasets (above, due to missing values in these) ECL has 2 years hourly electricity consumption data.	(H. Zhou et al., 2021)
Weather	Contains hourly observations of local climate data for approx 1600 US locations from 2010-2013.	(H. Zhou et al., 2021)
Electricity	Hourly consumption data from 321 customers 2012-2014.	(Wu et al., 2022)
Exchange	Daily data for 8 countries (Aus, NZ, UK, Canada, Switzerland, China, Japan, Singapore) 1990-2016.	(Lai et al., 2018)
Traffic	Hourly road occupancy freeway sensor data from SF Bay Area (from Caltrans).	Wu et al. (Wu et al., 2022)
Weather2	10 min meteorological data e.g. air temperature, humidity etc from 2020	(Wu et al., 2022)
Illness	Weekly CDC data for influenza-like illness (ILI) for 2002-2021.	(Wu et al., 2022)
Wind2	Hourly wind data from 1986-2015 as a percentage of a power plant's max output	(S. Liu et al., 2022)
App Flow	4 months of hourly max traffic flow for 128 systems across 16 data centers.	(S. Liu et al., 2022)
Electricity2	15 min electricity consumption data.	(S. Liu et al., 2022)
Flight	OpenSky data, granularity not explicitly stated.	(Cai et al., 2023)
PEMS	Consists of 4 sub-datasets recorded at 5 minute intervals with 358, 307, 883 and 170 time series.	(M. Liu et al., 2022)

in a sentence, thus extracting local semantic information is essential in analyzing their connections (Nie et al., 2023).

Inspired by the patching technique used in the Vision Transformer (ViT) (Dosovitskiy et al., 2021), which segments an image into patches and processes them as tokens in Transformers, a number of researchers (Nie et al., 2023; Wang et al., 2024; Chi et al., 2024) have introduced patching into their Transformer-based models by dividing the input time series into patches, treating each patch as an individual token. This approach not only enhances the ability to capture temporal patterns within time series data but also reduces computational complexity. Compared with previous Transformer-based models such as Autoformer (Wu et al., 2022), these models (Nie et al., 2023; Cai et al., 2023; Zhang & Yan, 2023; Wang et al., 2024) have achieved SOTA performance. Additionally, this technique has been applied to non-Transformer-based models, such as MLP-Mixer (Zhong et al., 2024), for similar benefits.

2.4.1 | Patch scale selection

CrossFormer (Zhang & Yan, 2023), PatchTST (Nie et al., 2023) and InjectTST (Chi et al., 2024), employ an exogenous parameter to determine the patch scale. Unlike CrossFormer, MLP-Mixer (Zhong et al., 2024), MSPatch (Cao et al., 2025) and MSGNet (Cai et al., 2023) use a list of varying patch scales, offering more flexibility in adapting to different temporal structures. A key distinction is that MSGNet utilizes Fast Fourier Transform to select patch scales adaptively, optimizing for different time series frequencies. On the other hand, iTransformer (Y. Liu et al., 2024) represents an extreme approach to patching, embedding each variable independently as a so-called 'variate' [‡] to capture global information.

2.5 | Channel-independence vs Channel-dependence

In addition to patching, PatchTST (Nie et al., 2023) and TimeXer (Wang et al., 2024) introduced CI into Transformer based time series modeling. As a multivariate time series is a multi-channel signal, it suits a Transformer input token as these can represent single or multiple channel data. CD refers to the latter case where the input token takes the vector of all time series features and projects it to the embedding space to mix information. On the other hand, CI means that each input token only contains information from a single channel (Nie et al., 2023).

PatchTST (Nie et al., 2023) highlights several benefits of CI for Transformer models in time series forecasting: 1. Adaptability: When passing through the data in a CI way through the Transformer model, Nie et al. observed that Transformer models can produce similar attention maps for unrelated but similar time series, indicating that this approach offers better adaptability across diverse data. 2. Data Efficiency: CI models demonstrate faster convergence with the same dataset size compared to CD models. This suggests that CD models require more data to achieve comparable convergence performance. 3. Reduced Overfitting: CI models tend to continue optimizing the loss while CD models are more prone to overfitting. This observation suggests that CI contributes to more robust learning and generalization during training.

Although CI tend to bring benefits for time series modeling, it has its limitations when compared with CD. Since CI treats multivariate time series as independent univariate signals, it may result in longer training times, lower scalability, and challenges the ability to leverage correlations between different variables in the dataset. Consequently, several works (Y. Liu et al., 2024; Zhang & Yan, 2023) continue to employ CD in Transformer based models.

Thus, in summary, while we can say that the recent advancements in Transformer based models have significantly improved time series forecasting, with techniques such as sparse attention (Wu et al., 2022) and patching strategies (Y. Liu et al., 2024) validated on real-world forecasting use cases such as predicting future traffic flow (Y. Liu et al., 2024) and electricity consumption (Wu et al., 2022). However, only a few studies have applied these innovative approaches to cryptocurrency price forecasting, particularly for altcoins.

This study bridges this gap by introducing the MSPCIFormer, designed to capture complex temporal and cross-cryptocurrency dynamics. Through a comparative analysis with SOTA Transformer-based models and deep learning models, we evaluate MSPFormer's effectiveness in forecasting BTC, ETH and ADA prices, advancing the application of advanced architectures to sustainable cryptocurrency markets.

3 | METHODOLOGY

3.1 | Dataset

According to CoinMarketCap[§], the top 10 cryptocurrencies, including ADA, BNB, BTC, DOGE, ETH, SOL, TRX, USDC, USDT and XRP, account for approximately 88.6% of the total market capitalization (USD 3.9 trillion as of 9 September 2025). Focusing on these assets therefore captures the vast majority of market dynamics, consistent with evidence that large market capitalization cryptocurrencies disproportionately influence overall market volatility (Korkusuz, 2025). Therefore, we gather the daily closing prices of these top 10 cryptocurrencies using data from Yahoo Finance[¶], covering 1 January 2020 to 1 January 2025. This period includes several episodes of heightened market turbulence, such as the COVID-19 pandemic (Kumar et al., 2023), US–China trade tensions (Upreti & Vásquez Callo-Müller, 2020; Sutter, 2025), the Terra–LUNA crash (Briola et al., 2023), and the Russian invasion of Ukraine (Kumar et al., 2023). Figure 1 presents the relative growth dynamics of the selected cryptocurrencies. Because Bitcoin's price level is substantially higher than the other assets, using raw prices would obscure corss-asset comparisons. To facilitate a clearer examination of co-movements and relative trends, the historical prices are presented in log-transformed form.

[‡] A variate token is used in iTransformer to denote a learned embedding that represents an individual time series in the context of a multivariate time series.

[§] https://api.coinmarketcap.com/data-api/v3/cryptocurrency/listing, accessed 9 September 2025

[¶] `https://finance.yahoo.com`, accessed 9 September 2025

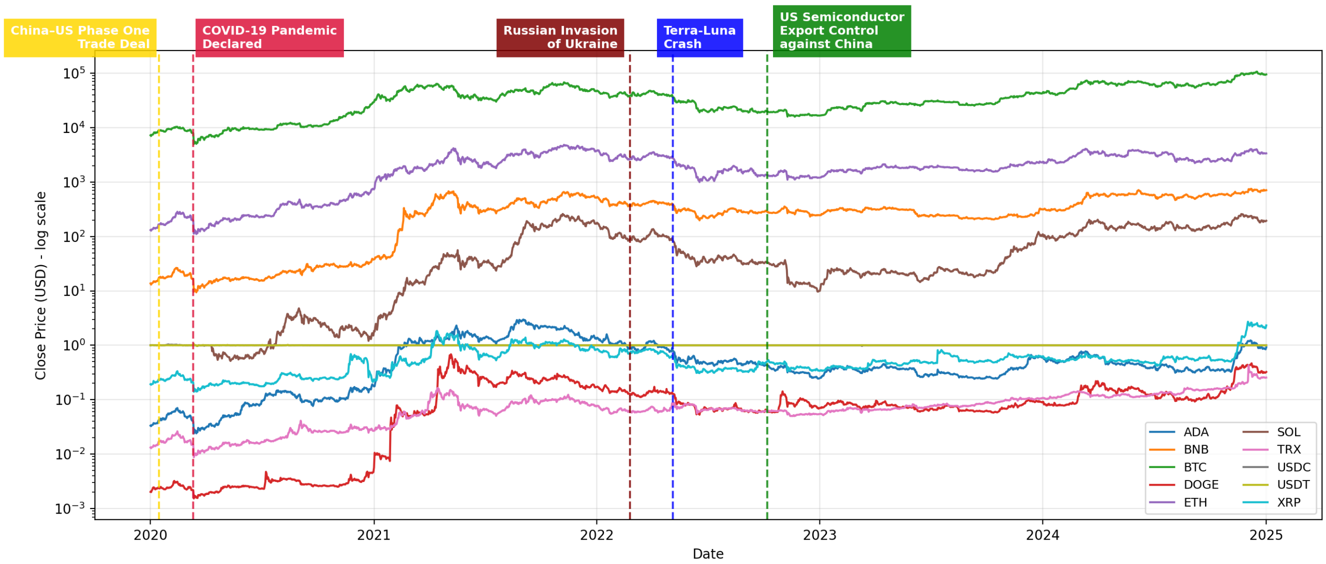

FIGURE 1 Historical price patterns of the top-10 market capitalization cryptocurrencies from 1st January 2020 to 1st January 2025. The panel displays the log-price (in USD) series to highlight relative growth dynamics.

Figure 1 shows that the stablecoins USDT and USDC appear as nearly flat lines, as expected given their design to maintain a fixed peg to fiat currency. Given evidence from existing work (Kristoufek, 2022) showing that stablecoins do not initiate price rallies in other cryptoassets, and in line with the study (Jaquart et al., 2022) that exclude stablecoins due to their pegged price behavior, we omit stablecoins from our ADA forecasting experiments.

In our dataset, SOL exhibits missing values prior to 10 April 2020, as the asset was launched in March 2020[#] and the earliest available trading data begin on 10 April 2020. Accordingly, observations before the date were removed to ensure a consistent sample across all assets. After the adjustments, the final dataset used in our experiments comprises 1728 daily time steps for ADA, BNB, BTC, DOGE, ETH, SOL, TRX, and XRP, covering the period from 10 April 2020 to 1 January 2025. The forecasting task leverages the interdependencies among these eight cryptocurrencies to enhance predictions of future ADA prices.

3.2 | Problem settings

Our goal is to predict the future values of one target coin (BTC, ETH and ADA) price based on the past time steps of ADA, BNB, BTC, DOGE, ETH, SOL, TRX, and XRP. Formally, given the historical observations $x_{1:t} \in R^{t \times N}$, where t denotes the number of past time steps, and N represents the number of variables (in our case, N is 8), we aim to forecast the next T time steps of the target coin's price, denoted as $\hat{x}_{t+1:t+T} \in R^{T \times 1}$.

[#] See Agatha Ozog, "Solana (SOL) was launched in March 2020 with an initial price of around \$0.22 per SOL," *Binance*, 2020. Available at: `https://www.binance.com/en/square/post/21381405543082`.

3.3 | MSPCIFormer

3.3.1 | Structure Overview

As illustrated in Figure 2, the proposed method consists of a configurable CI multi-scale patching layer, a Transformer encoder, and a learnable aggregation layer. The multi-scale patching layer transforms the input sequence into k channel-independent patched representations, where each representation corresponds to a patching scale defined in an exogenous scale list. Each patched sequence is then processed by a Transformer encoder to generate patching scale specific predictions. Finally, the aggregation layer assigns learnable weights to these predictions, determining the contribution of each scale to the final prediction. Algorithm 1 details the sequential steps of the proposed method.

3.3.2 | Multi-Scale Patching

As noted above, the Vision Transformer (Dosovitskiy et al., 2021) operates by segmenting images into patches and transforming each patch into a linear embedding (a fixed-length vector obtained passing the patch through a learnable linear layer) before feeding them into a Transformer for image classification. Inspired by this and more recent Transformer-based time series forecasting models (Cai et al., 2023; Zhang & Yan, 2023; Wang et al., 2024; Chi et al., 2024) which segment time series into temporal patches, MSPCIFormer adopts a patching technique to improve time series forecasting performance. By grouping data points at each time step into patches prior to processing by the Transformer encoding module, this approach improves the model's ability to capture temporal patterns while simultaneously reducing computation complexity. Comparing with the canonical attention mechanism, the

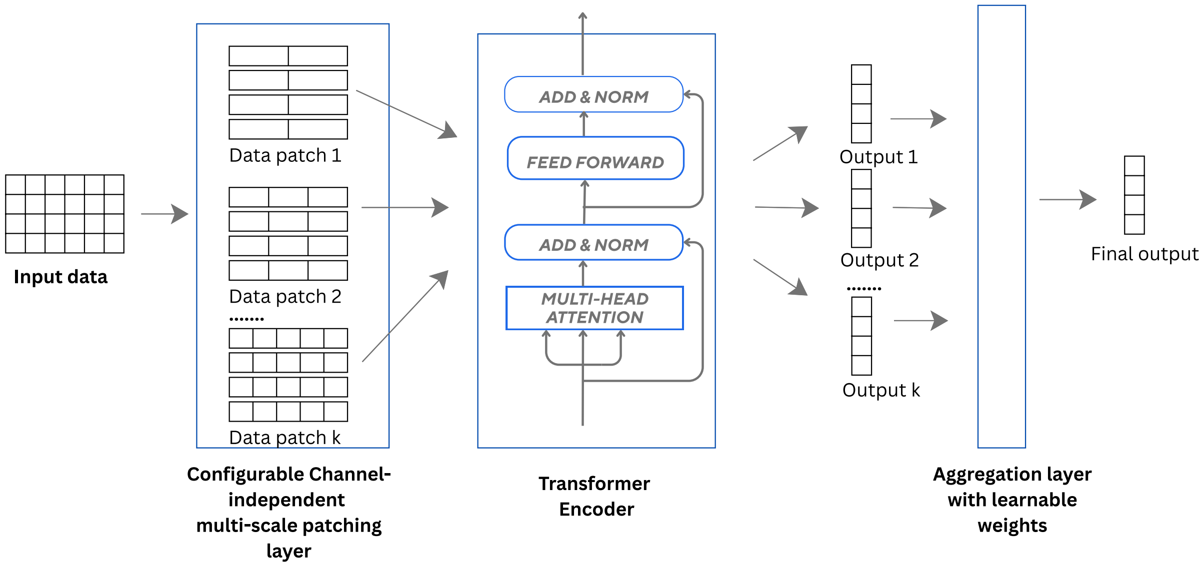

FIGURE 2 Architecture of MSPCIFormer. The input time series is transformed into k channel-independent patched representations, each corresponding to a patch scale from the predefined scale list. Each patched sequence is encoded by a Transformer to produce a scale-specific prediction. A learnable aggregation layer then weights and combines these predictions to form the final output.

computational complexity of the attention calculation drops from $\mathcal{O}(L)^2$ to $\mathcal{O}(L/P)^2$, where L denotes the sequence length and P represents the patch scale.

Instead of employing one single patch scale, MSPCIFormer utilizes a range of patch scales. Specifically, the input time series is segmented into patches across k distinct sizes in a list S, where k and S are hyperparameters determined based on the data's temporal characteristics. It enables the model to capture temporal patterns at multiple scales. The outputs from each scale are weighted by a learnable aggregation mechanism, which integrates them to generate the final prediction.

3.3.3 | Channel-Dependent Patching vs Channel-Independent Patching

As discussed previously, CI processing offers benefits such as adaptability, data efficiency, and reduced overfitting, while CD provides a better understanding of correlations between data channels. To assess whether explicitly modeling time-varying cross-asset relationships (CD) in the cryptocurrency dataset provides benefits over processing each channel independently (CI), we compared both approaches with the Transformer encoder model. The results show that the CI strategy consistently outperforms the CD alternative. Consequently, we adopt the CI formulation for the multi-scale patching layer.

3.3.4 | Transformer Encoder

We employ an encoder-only Transformer rather than the full encoder-decoder architecture used in canonical Transformer models. Unlike sequence-to-sequence tasks such as machine translation, where the decoder is required to generate outputs conditioned on both the encoder representation and previously generated tokens, our forecasting problem involves learning a direct mapping from a fixed window of historical observations to a fixed prediction horizon. In this setting, a decoder is unnecessary, as the encoded representations can be projected onto the forecasting horizon in a single step. The encoder-only design preserves the self-attention mechanism while avoiding the additional computational overhead associated with a decoder. This architectural choice is consistent with recent SOTA time series Transformer models, such as PatchTST (Nie et al., 2023) and iTransformer (Y. Liu et al., 2024), which have demonstrated the effectiveness of encoder-only Transformers for forecasting tasks.

After the input sequence is transformed into the patched representations by the multi-scale patching layer, it is processed by a multi-head self-attention mechanism within the Transformer encoder (see Figure 2). For each attention head $h = 1, \dots, H$, the input is projected into query, key, and value matrices[‖], as shown in Equations 1:

$$Q_h = x_i^d W_h^Q, \tag{1a}$$

$$K_h = x_i^d W_h^K, \tag{1b}$$

$$V_h = x_i^d W_h^V. \tag{1c}$$

where d is the Transformer model's feature dimension, W_h^Q and $W_h^K \in R^{D\times d_k}$, $W_h^V \in R^{D\times D}$ are learnable projection matrices. The attention scores are computed using the scaled dot-product attention, Equation 2:

$$\text{Attn}_h(Q_h, K_h, V_h) = \text{softmax}\left(\frac{Q_h K_h^\top}{\sqrt{d_k}}\right) V_h. \tag{2}$$

The outputs of all attention heads are concatenated and linearly transformed, Equation 3:

$$Z = \text{Concat}(\text{Attn}_1, \dots, \text{Attn}_H) W^O, \tag{3}$$

where $W^O \in R^{D\times D}$ is the output projection matrix. Finally, the Transformer encoder applies a feedforward network (FFN) with residual connections and layer normalization, Equation 4:

$$Z' = \text{LayerNorm}(Z + \text{FFN}(Z)). \tag{4}$$

After the Transformer encoder processes the input, the output is projected and reshaped back to match the input shape x_i. Finally, the Transformer encoder produces a prediction for each patched representation generated by the multi-scale patching layer. The aggregation layer then fuses these patching scale specific outputs by assigning learnable weights to each to obtain the final prediction.

Taken together, these architectural elements yield the overall parameterization of MSPCIFormer. Table 3 presents the corresponding parameter counts for the two patching scales employed in our implementation.

TABLE 3 Parameter counts of MSPCIFormer.

Patching Scale	Layer	Parameter Count
Patching scale 1	Encoding layer	224
	Transformer encoder	13456
	Projection head	520
Patching scale 2	Encoding layer	320
	Transformer encoder	13456
	Projection head	520
–	Aggregation layer	2
	Total	**269,858**

4 | EXPERIMENTS

Our dataset contains 1728 daily observations. To preserve temporal order, the data were partitioned into training, validation, and test sets using a 70%-10%-20% chronological split for a hold-out validation. This results in 1188 samples for training, 153 samples for validation, and 324 samples for testing. Such a partitioning strategy is standard practice in time-series forecasting and ensures that model evaluation reflects performance on future unseen data.

To evaluate the robustness of the proposed framework under evolving market conditions, we additionally perform a walk-forward validation experiment using 5 rolling evaluation folds. This evaluation protocol is widely regarded as more realistic for financial time series, as it preserves temporal ordering and periodically retrain the model on expanding windows, thereby capturing shifts in market regimes (Börjesson & Singull, 2020; Taheri Hosseinkhani, 2025).

To examine the effectiveness of the proposed method, we benchmark it against several SOTA Transformer-based time series forecasting models, including MSGNet (Cai et al., 2023), iTransformer (Y. Liu et al., 2024), and TimeXer (Wang et al., 2024), as well as widely adopted deep learning models in TSF: the recurrent LSTM model (Hochreiter & Schmidhuber, 1997), NBeats(Oreshkin et al., 2020), TimesNet(Wu et al., 2023) and the linear decomposition framework DLinear (Zeng et al., 2022).

Given the relatively small size of the dataset and the high parameter count of the models, there is an inherent risk of overfitting (Aliferis & Simon, 2024). To mitigate this risk, we employ an early-stopping strategy that monitors validation performance and terminates training once no further improvement is observed. We further incorporate a relatively high dropout rate to strengthen regularization. Model optimization is performed using the Adam optimizer, which adaptively adjusts learning rates to enhance training stability and accelerate convergence. Adam remains the default choice in deep learning due to its combination of momentum and per-parameter adaptive learning rates (Makinde, 2024), and it is consistently adopted in SOTA Transformer-based time series models such as PatchTST (Nie et al., 2023) and DeformableTST (Luo & Wang, 2024), where it has demonstrated stable and efficient convergence for long sequence forecasting tasks. For all the models, the key hyperparameters are tuned via a grid-search procedure implemented with Optuna[**], ensuring an appropriate balance between model complexity and predictive performance.

Following the SOTA Transformer-based time series architectures (Nie et al., 2023; Y. Liu et al., 2024; Wang et al., 2024; Cai et al., 2023) and

[‖] The concept of key, value, and query in the attention mechanism is analogous to information retrieval. Consider a library scenario: if you are searching for books on computers, your query is"computer books". The library's index cards, which list topics such as machine learning, data mining, and system design, serve as keys. The actual books corresponding to these indexed topics represents the values.

[**] Optuna is an open source Python library designed for hyperparameter optimization(Akiba et al., 2019)

Algorithm 1 Training and Inference Procedure for MSPCIFormer

```
Require: Training set D_train = {(X^(i), Y^(i))}_{i=1}^N, Evaluation
  set D_eval, Forecasting horizon H, Top-k patch scales k,
  Empirical scale list S_emp
Ensure: Trained model parameters (θ*, φ*) and predictions Ŷ,
  where
     θ denotes the Transformer encoder parameters
  responsible for temporal representation learning under
  CI setting;
     φ denotes the parameters of the aggregation layer.

  procedure TRAIN(D_train)
     for each epoch do
        for each (X, Y) do
           S ⇐ S_emp[0 : k]
           for each patch scale s ∈ S do
              P_s ⇐ MAKEPATCHES(X, s)
              Ĥ_s ⇐ TRANSFORMERENCODERCI(P_s, θ)
           end for
           Ŷ ⇐ AGGREGATE({Ĥ_s}_{s∈S}, φ)
           L ⇐ LOSS(Ŷ, Y)
           (θ, φ) ⇐ UPDATE(θ, φ, ∇L)
        end for
     end for
  end procedure

  procedure INFER(D_eval)
     for each (X, Y) in D_eval do
        S ⇐ S_emp[0 : k]
        for each scale s ∈ S do
           P_s ⇐ MAKEPATCHES(X, s)
           Ĥ_s ⇐ TRANSFORMERENCODERCI(P_s, θ*)
        end for
        Ŷ ⇐ AGGREGATE({Ĥ_s}_{s∈S}, φ*)
     end for
  end procedure
```

prior cryptocurrency price forecasting studies (Alnami et al., 2025; Kiram et al., 2025; Bourday et al., 2024), which predominantly employ MSE and MAE as performance metrics, we adopt MSE and MAE as primary statistical evaluation metrics. To complement these measures and capture additional aspects of forecasting performance, we also report RMSE and Mean Absolute Percentage Error (MAPE). Mean Directional Accuracy (MDA) is included to evaluate the model's ability to correctly predict market movement directions. In addition, the Sharpe ratio and Maximum Drawdown (MDD) are incorporated to assess the risk-adjusted return and the worst-case capital loss of each model's predicted trading signals, thereby providing a more comprehensive evaluation from economical utility perspective.

As expressed in Equations 5 and 6, MSE quantifies the average squared deviation between predicted values and ground truth, whereas MAE measures the average magnitude of absolute errors. Lower values of both metrics correspond to better predictive performance. Notably, MSE penalizes more on large errors due to the squaring of residuals, while MAE provides a more direct measure of the typical forecasting error (Adhikari & Agrawal, 2013).

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{5}$$

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}\left|y_i - \hat{y}_i\right| \tag{6}$$

As shown in Equation 7, RMSE is defined as the square root of the MSE. By applying the square root, RMSE expresses the forecasting error in the same units as the target variable, thereby enhancing interpretability and often making it a preferred metric over MSE (Hyndman & Koehler, 2006).

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{7}$$

As defined in Equation 8, MAPE expresses the mean absolute error as a percentage of the actual values. This metric is intuitive and easily interpretable, as it indicates the average proportional deviation of predictions from the ground truth. However, MAPE can become unstable when the true values are close to zero, which may lead to disproportionately large percentage errors (Jierula et al., 2021).

$$\mathrm{MAPE} = \frac{100\%}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \tag{8}$$

Equation 9 defines MDA which measure the ability of a forecasting model to correctly predict the direction of cryptocurrency price movements by comparing the predicted price direction (upward or downward) to the actual cryptocurrency price direction(Bouteska et al., 2024).

$$\mathrm{MDA} = \frac{1}{N}\sum_{t=1}^{N}\mathbb{I}\left[\mathrm{sign}(y_t - y_{t-1}) = \mathrm{sign}(\hat{y}_t - y_{t-1})\right] \tag{9}$$

Sharpe ratio is a widely adopted financial evaluation metric used to assess the risk-adjusted performance metric(Amédée-Manesme & Barthélémy, 2022). It is calculated by subtracting the risk-free rate of return from the average return of the investment and dividing the result by the investment's standard deviation (Jabbar & Jalil, 2024). A higher Sharpe Ratio indicates a more desirable risk-adjusted return. In Equation 10, $E[R]$, R_f and $\sigma(R)$ denotes the average strategy returns, risk-free rate, and standard deviation of returns respectively. T represents the annualized factor. Given that cryptocurrency markets trade continuously on a 24/7 basis, we adopt $T = 365$ to compute the annualized Sharpe ratio.

$$\mathrm{Sharpe} = \frac{E[R] - R_f}{\sigma(R)} \times \sqrt{T} \tag{10}$$

MDD can be defined as the worst cumulative loss in a given period(Choi, 2021). Following an extremely large fall (or a long sequence of small falls) in market prices, an investor (specially retirees) may decide to sell valuable positions irrespective of market conditions for fear of even larger losses(de Melo Mendes & Lavrado, 2017). As shown in Equation 11, MDD is calculated with V_t which denotes cumulative portfolio value at time t and Peak_t which represents the highest value reached up to time t.

$$\text{MDD} = \max_t \frac{\text{Peak}_t - V_t}{\text{Peak}_t} \tag{11}$$

5 | RESULTS

5.1 | Hold-Out Validation

Table 4, 5 and 6 presents the experimental results for ADA, BTC and ETH price forecasting across all evaluation metrics and prediction horizons under the hold-out evaluation. To provide a model performance comparison across all metrics simultaneously, Figures 3, 4, and 5 present radar charts that summaries each model's performance on ADA, BTC, and ETH respectively under hold-out validation. Each axis corresponds to one normalized evaluation metric, with the outer boundary denotes the best-performing model on that metric. Values are averaged across all forecasting horizons, providing a holistic comparison of model strengths and weaknesses across both statistical and financial dimensions. A larger enclosed polygon area indicates superior overall performance.

From the charts, it can be seen that the model rankings are asset-dependent: NBeats dominates on ADA and ETH, while MSPCIFormer and TimeXer achieve the largest polygons on BTC, indicating that no single model generalities optimally across all cryptocurrency markets.

On ADA, model performance is horizon-sensitive. MSPCIFormer achieves the best MSE and MAE at pred=1, consistent with its multi-scale patching mechanism capturing various scale oscillatory patterns that are most salient at short horizons. As the horizon extends to pred=3–7, DLinear and NBeats edge ahead, while MSPCIFormer falls to the lower half of the ranking. This degradation at medium horizons is attributable to the fixed patch sizes in MSPCIFormer's patching module becoming increasingly misaligned with the growing prediction window. Our ablation experiments confirm this diagnosis, using bigger patching scales improves hold-out MSE for both ADA and ETH, and substantially improves walk-forward Sharpe Ratio across most horizons. At pred=15, DLinear maintains rank 1, confirming that simple linear models remain strong competitors at extended horizons. Regarding financial metrics, the Sharpe Ratio is largely negative across all models, reflecting the adverse 2022–2024 post-peak market regime. NBeats is the exception, achieving the only notably positive Sharpe at pred=1 (+1.311) and pred=5 (+1.216). On MDD, MSPCIFormer records among the highest values across horizons, consistent with its negative Sharpe. NBeats achieves the best average MDD on ADA, suggesting its basis-function decomposition produces more conservative return trajectories. TimesNet and iTransformer exhibit erratic MDD profiles (0.442–0.914),

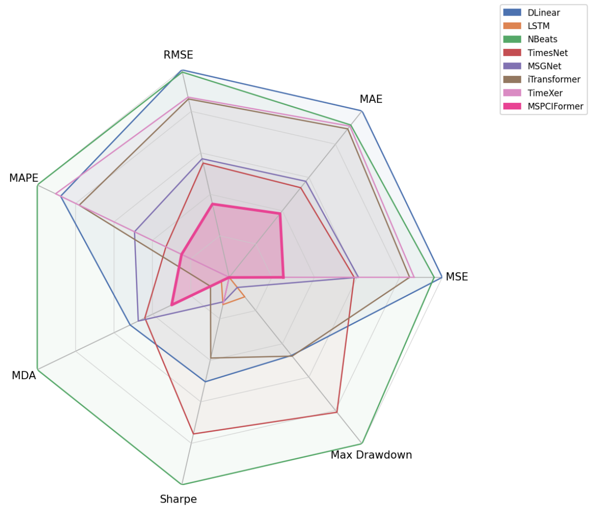

FIGURE 3 Performance radar chart for ADA under hold-out validation, where the outer boundary denotes the best-performing model. Values are averaged across prediction horizons of 1, 3, 5, 7, and 15 days. A larger enclosed area indicates superior overall performance across both statistical and financial metrics.

indicating sensitivity to the specific market regime sampled at each horizon. MDA hovers near 0.50 across all models, indicating near-random directional predictability on ADA.

On BTC, the competitive landscape shifts markedly. MSPCIFormer achieves the best MSE at pred=1 and pred=3, with TimeXer taking the lead at pred=5 and MSPCIFormer recovering to rank 1st at pred=7. At pred=15, TimesNet emerges as the dominant model, suggesting its FFT-based multi-period decomposition is particularly well-suited to BTC's long-horizon macro-cycle structure. Financially, MSPCIFormer and TimeXer achieve positive Sharpe at pred=1 alongside DLinear, while TimesNet leads at pred=15. MSPCIFormer records the lowest MDD at pred=1 and maintains competitive values across all horizons, indicating superior capital preservation for BTC forecasting.

On ETH, NBeats consistently achieves the largest radar polygon, leading on both MSE and Sharpe at pred=[1, 3, 15]. MSPCIFormer performs competitively at pred=1 and recovers to 2nd at pred=5, but degrades notably at pred=3, where its fixed patch granularity interacts poorly with ETH's price dynamics at that specific horizon. This horizon-sensitive pattern (strong at pred=1, degraded at medium horizons, partially recovering at pred=5) is consistent across both ADA and ETH, but is less pronounced on BTC is because BTC's stronger trend momentum is more aligned with the model's patch granularity even at medium horizons. The cross-asset pattern therefore points to an interaction between MSPCIFormer's fixed patch configuration and the price dynamics of each asset.

TABLE 4 Hold-out validation - ADA forecasting performance across prediction horizons using an input window of 5 days. The best result for each horizon is shown in bold, and the second-best result is underlined.

Horizon	Model	MSE	MAE	RMSE	MAPE	MDA	Sharpe	Max Drawdown
1	DLinear	<u>0.00188</u>	<u>0.02685</u>	<u>0.04333</u>	0.26750	<u>0.48986</u>	-0.36021	0.68864
	LSTM	0.00354	0.03679	0.05952	0.34936	0.46957	-1.99703	0.89300
	NBeats	0.00192	0.02803	0.04387	0.26203	**0.50435**	**1.31079**	<u>0.59551</u>
	TimesNet	0.00239	0.02979	0.04887	0.38209	0.47246	-0.27759	0.64467
	MSGNet	0.00257	0.03081	0.05066	0.33500	0.47536	-1.74642	0.88674
	iTransformer	0.00205	0.02817	0.04525	0.28431	0.46087	<u>0.40738</u>	**0.52396**
	TimeXer	0.00216	0.02887	0.04644	**0.25447**	0.46667	-1.77659	0.86629
	MSPCIFormer	**0.00187**	**0.02668**	**0.04319**	<u>0.25695</u>	0.48986	-1.44125	0.82287
3	DLinear	**0.00413**	**0.03904**	**0.06426**	<u>0.36378</u>	0.46939	-0.82116	0.78355
	LSTM	0.00677	0.05119	0.08226	0.53535	0.47522	-1.87676	0.85792
	NBeats	<u>0.00416</u>	0.04066	<u>0.06446</u>	**0.35630**	**0.52187**	<u>0.23923</u>	<u>0.59737</u>
	TimesNet	0.00501	0.04323	0.07079	0.51874	<u>0.49271</u>	**1.18560**	**0.44817**
	MSGNet	0.00488	0.04221	0.06983	0.43371	0.49271	-1.53148	0.85605
	iTransformer	0.00427	<u>0.03948</u>	0.06535	0.37772	0.47230	-1.35618	0.79402
	TimeXer	0.00428	0.04019	0.06539	0.36796	0.48688	0.22814	0.66178
	MSPCIFormer	0.00465	0.04121	0.06817	0.41899	0.47813	-2.17533	0.88706
5	DLinear	0.00678	0.05039	0.08234	0.51256	0.47801	-1.69466	0.88721
	LSTM	0.00725	0.05269	0.08513	0.54277	0.48094	-0.13713	0.63213
	NBeats	**0.00610**	**0.04840**	**0.07808**	**0.46785**	**0.50440**	**1.21593**	<u>0.59551</u>
	TimesNet	0.00787	0.05389	0.08869	0.59966	0.47214	-0.76783	0.73849
	MSGNet	0.00757	0.05292	0.08701	0.57904	0.49560	-2.23013	0.90445
	iTransformer	<u>0.00639</u>	0.04889	<u>0.07991</u>	0.49046	0.49853	<u>0.81611</u>	**0.44185**
	TimeXer	0.00646	<u>0.04866</u>	0.08036	<u>0.48854</u>	0.46921	-2.17338	0.91473
	MSPCIFormer	0.00952	0.06072	0.09757	0.67588	<u>0.50147</u>	-1.38356	0.79342
7	DLinear	**0.00848**	**0.05729**	**0.09210**	<u>0.57884</u>	**0.50442**	<u>0.32812</u>	<u>0.52879</u>
	LSTM	0.01204	0.06764	0.10974	0.77729	0.46903	-1.92010	0.86093
	NBeats	<u>0.00862</u>	<u>0.05788</u>	<u>0.09284</u>	**0.57136**	0.49558	**1.01355**	**0.39415**
	TimesNet	0.01031	0.06278	0.10152	0.63574	<u>0.50147</u>	-0.28332	0.67613
	MSGNet	0.00994	0.06132	0.09969	0.64940	0.49558	-0.91959	0.72724
	iTransformer	0.00955	0.05964	0.09775	0.63053	0.47493	-2.31844	0.90951
	TimeXer	0.00919	0.05830	0.09585	0.61115	0.48378	-2.10359	0.93782
	MSPCIFormer	0.01089	0.06420	0.10437	0.71000	0.48083	-1.72348	0.84146
15	DLinear	**0.01985**	**0.08382**	**0.14088**	1.02861	**0.51057**	<u>-0.40455</u>	0.70113
	LSTM	0.02605	0.09478	0.16142	1.17046	0.48338	-1.72218	0.84599
	NBeats	<u>0.02087</u>	0.08628	<u>0.14448</u>	<u>1.00722</u>	<u>0.50151</u>	-0.45470	<u>0.65263</u>
	TimesNet	0.02155	0.08879	0.14682	**1.00480**	0.50151	**0.36868**	**0.59626**
	MSGNet	0.02188	0.08946	0.14792	1.02755	0.48640	-1.39880	0.79555
	iTransformer	0.02108	0.08613	0.14520	1.03711	0.48036	-1.94695	0.91393
	TimeXer	0.02096	<u>0.08553</u>	0.14476	1.01036	0.46526	-1.91625	0.87592
	MSPCIFormer	0.02503	0.09278	0.15821	1.13712	0.46828	-2.59171	0.91026

5.2 | Walk-Forward Validation: 5-Folds

For the walk-forward validation, Tables 7, 8, and 9 report the models performance comparison across all forecasting horizons for all evaluation metrics. Similarly, Figures 6, 7, and 8 present an at-a-glance model performance comparison using radar charts with averaged performance metrics across all forecasting horizons.

From the radar charts, it can be observed that MSPCIFormer achieves highly competitive results for all three coins, representing a marked improvement over its hold-out performance and confirming that its generalization capability is better captured under multi-regime evaluation. Additionally, the Sharpe and MDD axes expand noticeably compared to the hold-out charts for ADA and ETH, indicating that walk-forward folds sample more diverse market conditions which are favorable to MSPCIFormer to generate positive returns. Furthermore, no single model dominates all axes across all assets, reinforcing that the statistical-financial performance trade-off is persistent regardless of validation protocol.

On ADA, MSPCIFormer achieves the most consistent statistical accuracy across all horizons, ranking 1st on MSE across all forecasting horizons, a meaningful reversal from hold-out results where it fell to the lower half at medium horizons. This cross-regime consistency suggests that its multi-scale patching captures ADA's oscillatory structure robustly across the various market phases. Regarding financial metrics, Sharpe Ratio improves substantially for all models relative to hold-out, reflecting the inclusion of more diverse regimes across folds. iTransformer achieves the best Sharpe at pred=3, while MSGNet leads at pred=5 and pred=7. MSPCIFormer's Sharpe remains modest ([−0.796 -

TABLE 5 Hold-out validation - BTC forecasting performance across prediction horizons using an input window of 5 days. The best result for each horizon is shown in bold, and the second-best result is underlined.

Horizon	Model	MSE	MAE	RMSE	MAPE	MDA	Sharpe	Max Drawdown
1	DLinear	0.01545	0.08975	0.12429	0.03843	<u>0.53043</u>	**2.08213**	<u>0.28440</u>
	LSTM	0.01869	0.10080	0.13672	0.04343	0.48406	-0.54950	0.46534
	NBeats	0.01678	0.09353	0.12955	0.03962	0.47536	-1.69884	0.66584
	TimesNet	0.01644	0.09432	0.12823	0.04060	0.51304	-0.15527	0.35056
	MSGNet	0.01618	0.09217	0.12720	0.03959	0.49275	-0.76701	0.55647
	iTransformer	0.01645	0.09486	0.12826	0.04046	0.51304	0.83352	0.30434
	TimeXer	<u>0.01544</u>	<u>0.08919</u>	<u>0.12427</u>	<u>0.03818</u>	**0.53333**	1.10886	0.35983
	MSPCIFormer	**0.01539**	**0.08909**	**0.12406**	**0.03806**	0.53043	<u>1.18030</u>	**0.26019**
3	DLinear	0.03106	0.12930	0.17623	0.05541	0.50146	-0.31338	<u>0.41575</u>
	LSTM	0.03409	0.13596	0.18465	0.05888	0.47813	-0.62224	0.45916
	NBeats	0.03801	0.13939	0.19496	0.05754	**0.54810**	<u>0.89037</u>	0.54269
	TimesNet	0.03263	0.13261	0.18063	0.05662	0.51020	-0.52318	0.48281
	MSGNet	0.03196	0.13217	0.17878	0.05686	0.51312	-0.46318	0.49025
	iTransformer	0.03180	0.13160	0.17834	0.05665	0.47230	-0.77757	0.62723
	TimeXer	<u>0.03049</u>	<u>0.12777</u>	<u>0.17460</u>	<u>0.05493</u>	<u>0.53353</u>	**1.03309**	**0.29743**
	MSPCIFormer	**0.03023**	**0.12726**	**0.17386**	**0.05466**	0.51312	0.24107	0.48223
5	DLinear	0.04656	0.15986	0.21579	0.06843	0.45161	-1.48445	0.66087
	LSTM	0.05230	0.17196	0.22869	0.07465	0.50733	-0.24568	0.46229
	NBeats	0.05957	0.18025	0.24407	0.07458	0.46921	-2.01715	0.71050
	TimesNet	0.04740	0.16412	0.21771	0.07069	**0.53372**	**1.07445**	<u>0.35142</u>
	MSGNet	0.04742	0.16068	0.21777	0.06975	0.50440	0.32067	**0.33464**
	iTransformer	0.04637	0.15959	0.21533	0.06891	0.47507	-0.68138	0.68725
	TimeXer	**0.04453**	**0.15583**	**0.21103**	**0.06737**	0.51026	-1.36098	0.64876
	MSPCIFormer	<u>0.04554</u>	<u>0.15757</u>	<u>0.21339</u>	<u>0.06788</u>	<u>0.52493</u>	<u>0.39560</u>	0.53486
7	DLinear	0.06240	0.18286	0.24979	<u>0.07860</u>	0.46608	-1.72758	0.70181
	LSTM	0.07061	0.19605	0.26572	0.08523	0.47788	-0.66387	0.51288
	NBeats	0.06941	0.19143	0.26347	0.08112	0.46313	-2.18119	0.76553
	TimesNet	0.06406	0.18672	0.25310	0.08075	<u>0.51917</u>	-0.46180	0.61550
	MSGNet	0.06810	0.19099	0.26095	0.08236	0.51327	**0.95864**	**0.35570**
	iTransformer	<u>0.06145</u>	<u>0.18277</u>	<u>0.24790</u>	0.07923	0.48378	-1.24684	0.61564
	TimeXer	0.06378	0.18720	0.25255	0.08116	**0.52802**	<u>0.35044</u>	<u>0.40514</u>
	MSPCIFormer	**0.06074**	**0.18127**	**0.24645**	**0.07841**	0.51917	-0.21882	0.43380
15	DLinear	0.13466	0.26872	0.36695	0.11413	0.50151	-1.02520	0.63271
	LSTM	0.14356	0.27546	0.37889	0.11809	0.50755	-0.47250	0.47971
	NBeats	0.17337	0.29747	0.41637	0.11926	**0.52568**	<u>0.66650</u>	**0.42321**
	TimesNet	**0.12075**	**0.25633**	**0.34749**	**0.11088**	<u>0.52568</u>	**1.58750**	<u>0.45933</u>
	MSGNet	0.12775	0.26237	0.35742	0.11244	0.48036	-0.69772	0.61617
	iTransformer	0.12846	0.26272	0.35841	0.11267	0.46828	-1.02723	0.57225
	TimeXer	<u>0.12604</u>	<u>0.25930</u>	<u>0.35502</u>	0.11129	0.50755	-0.24830	0.48483
	MSPCIFormer	0.12693	0.25947	0.35627	<u>0.11114</u>	0.50755	-0.09290	0.50670

+0.643]), placing it mid-table on financial performance. On MDD, iTransformer achieves the best capital preservation at pred=3 and pred=5, while MSGNet leads at pred=7. Similar to the Sharpe, MSPCIFormer's drawdown ([0.338 - +0.436]) is moderate and competitive with the field.

On BTC, MSPCIFormer achieves a large polygon across statistical and financial axes with competitive evaluation metrics. MSPCIFormer achieves the best MSE at pred=1 and pred=3, remaining competitive at pred=5. At pred=7, TimeXer and NBeats lead marginally. At pred=15, iTransformer achieves the best MSE with MSPCIFormer ranking 4th. Among financial metrics, MSPCIFormer records the highest MDA across all forecasting horizons, suggesting its multi-scale patching captures BTC price momentum effectively. NBeats and DLinear is leading the board with best Sharpe across all forecasting horizons. On MDD, NBeats records the best capital preservation at pred=1 and 2nd place at pred=7, while DLinear is leading at pred=[3, 5, 15]. MSPCIFormer maintains low MDD at pred=1, making it the best capital-preserving Transformer at short horizons on BTC.

On ETH, MSPCIFormer ranks 1st on MSE at pred=1 and pred=5, and remains comparative with the best performer NBeats and DLinear on other prediction horizons. MDA for MSPCIFormer is competitive across all horizons, reaching the highest MDA in the ETH walk-forward results at pred=15, tied with iTransformer. Sharpe Ratio is more dispersed across models than on ADA or BTC. TimesNet achieves the highest Sharpe at pred=5 and pred=7, while NBeats leads at pred=1, MSGNet at pred=3 and DLinear at pred=15. MSPCIFormer's Sharpe is positive but

TABLE 6 Hold-out validation - ETH forecasting performance across prediction horizons using an input window of 5 days. The best result for each horizon is shown in bold, and the second-best result is underlined.

Horizon	Model	MSE	MAE	RMSE	MAPE	MDA	Sharpe	Max Drawdown
1	DLinear	0.00910	0.06671	0.09540	0.06670	0.53623	0.90908	0.35589
	LSTM	0.00995	0.07205	0.09975	0.07241	0.48116	-0.23894	0.67433
	NBeats	**0.00861**	**0.06608**	**0.09280**	**0.06568**	0.52754	**2.00341**	**0.32393**
	TimesNet	0.00987	0.07330	0.09937	0.07339	0.48986	0.33027	0.44428
	MSGNet	0.00945	0.06856	0.09720	0.06877	0.51884	0.89530	0.48221
	iTransformer	0.00929	0.06850	0.09636	0.06823	0.53333	0.61093	0.33664
	TimeXer	0.00930	0.06824	0.09644	0.06786	**0.55072**	0.65853	0.49455
	MSPCIFormer	0.00888	0.06610	0.09421	0.06615	0.50435	0.01740	0.59203
3	DLinear	0.01783	0.09511	0.13351	0.09602	**0.55977**	**1.84915**	**0.30927**
	LSTM	0.02949	0.12545	0.17173	0.12712	0.50437	-0.68449	0.62674
	NBeats	**0.01688**	**0.09322**	**0.12991**	**0.09410**	0.53061	1.27005	0.49337
	TimesNet	0.01903	0.09808	0.13795	0.09809	0.52770	1.01178	0.59355
	MSGNet	0.01927	0.09840	0.13881	0.09878	0.49563	-1.08334	0.76017
	iTransformer	0.01815	0.09619	0.13471	0.09665	0.51020	-0.99429	0.65845
	TimeXer	0.01797	0.09506	0.13404	0.09630	0.48397	-0.98618	0.73662
	MSPCIFormer	0.02134	0.10521	0.14607	0.10625	0.52770	-0.41630	0.59360
5	DLinear	0.02751	0.11756	0.16586	0.11864	0.48974	0.60372	0.41423
	LSTM	0.03733	0.13876	0.19321	0.13886	**0.55425**	0.96362	0.44324
	NBeats	**0.02589**	**0.11579**	**0.16089**	**0.11645**	0.53959	**1.45826**	**0.31916**
	TimesNet	0.02990	0.12290	0.17291	0.12448	0.51613	0.33650	0.50591
	MSGNet	0.02928	0.12094	0.17112	0.12064	0.53372	-0.34099	0.65999
	iTransformer	0.02876	0.12086	0.16959	0.12111	0.53959	1.29206	0.36128
	TimeXer	0.02957	0.12241	0.17195	0.12198	0.52493	0.01186	0.54726
	MSPCIFormer	0.02736	0.11731	0.16540	0.11792	0.48974	-1.11829	0.70767
7	DLinear	0.03762	0.13928	0.19397	0.13917	0.49853	-0.71817	**0.60227**
	LSTM	0.04488	0.15517	0.21185	0.15431	**0.50737**	-0.93891	0.66786
	NBeats	**0.03631**	**0.13666**	**0.19055**	**0.13493**	0.48968	-0.96742	0.63725
	TimesNet	0.04050	0.14468	0.20126	0.14446	0.49558	**-0.36426**	0.64632
	MSGNet	0.04018	0.14464	0.20044	0.14307	0.48673	-2.51403	0.84099
	iTransformer	0.03922	0.14238	0.19805	0.14115	0.46608	-1.21224	0.75314
	TimeXer	0.03779	0.13976	0.19441	0.13895	0.50147	-1.00446	0.75409
	MSPCIFormer	0.04159	0.14735	0.20393	0.14555	0.50737	-0.72841	0.61606
15	DLinear	0.07581	0.20648	0.27534	0.20240	0.47432	-0.66705	0.50653
	LSTM	0.08482	0.22040	0.29124	0.21312	0.50151	-0.83590	0.66202
	NBeats	**0.06947**	**0.19305**	**0.26358**	**0.18690**	**0.54381**	**2.00307**	**0.29866**
	TimesNet	0.07934	0.21151	0.28168	0.20848	0.53474	-0.24145	0.59329
	MSGNet	0.08483	0.22191	0.29126	0.22118	0.50453	-1.00999	0.70393
	iTransformer	0.08608	0.22334	0.29339	0.21595	0.50151	-1.02689	0.68167
	TimeXer	0.08610	0.22240	0.29344	0.21461	0.49547	-1.29006	0.72812
	MSPCIFormer	0.07633	0.20529	0.27627	0.19932	0.51964	-0.70298	0.58447

modest, reflecting the same pattern seen on ADA: strong statistical accuracy with limited directional trading signal. On MDD, NBeats achieves the best capital preservation at pred=1, while LSTM leads at pred=3 and pred=5. MSPCIFormer's MDD is competitive, achieved the lowest at pred=1 when comparing with other Transformer-based models.

5.3 | Computational Efficiency: Training Time, Peak Memory, and Inference Latency

Table 10 summaries the average epoch training time, peak inference memory, and per-sample inference latency for all eight models, averaged across all coins and prediction horizons.

Training epoch time measures the time in seconds to complete one full pass over the training set. It reflects how quickly a model can be retrained on new data. In our results, NBeats is the fastest to train, followed by LSTM and MSPCIFormer. TimesNet is the most expensive at 3.96 s/epoch, reflecting its FFT-based multi-period decomposition and heavier convolutional backbone. MSPCIFormer's epoch time is the third fastest among all models and notably faster than the other Transformer baselines. This efficiency advantage is attributable to the CI architecture, which processes each feature independently and avoids the quadratic attention cost over channel dimensions. Walk-forward epoch times are consistently lower than hold-out for all models, as the walk-forward training windows cover a smaller portion of the dataset on average than the 70% hold-out training split.

TABLE 7 Walk-forward validation - ADA forecasting performance across prediction horizons using an input window of 5 days. The best result for each horizon is shown in bold, and the second-best result is underlined.

Horizon	Model	MSE	MAE	RMSE	MAPE	MDA	Sharpe	Max Drawdown
1	DLinear	<u>0.00058</u>	0.01609	<u>0.02267</u>	0.22288	0.46667	-0.23919	0.38942
	LSTM	0.00069	0.01687	0.02473	0.23547	**0.50435**	<u>0.19837</u>	<u>0.37441</u>
	NBeats	0.00059	<u>0.01593</u>	0.02280	<u>0.22171</u>	0.49710	-0.38254	0.39725
	TimesNet	0.00070	0.01745	0.02493	0.22929	0.46812	-0.82440	0.45782
	MSGNet	0.00069	0.01710	0.02464	0.24798	0.46377	-1.00164	0.47263
	iTransformer	0.00059	0.01610	0.02293	**0.21839**	0.47826	**0.72245**	**0.36229**
	TimeXer	0.00063	0.01607	0.02331	0.23104	<u>0.50435</u>	-0.11982	0.43705
	MSPCIFormer	**0.00056**	**0.01557**	**0.02235**	0.22267	0.48551	-0.79598	0.43601
3	DLinear	0.00112	0.02202	0.03161	<u>0.27296</u>	0.48382	0.27343	0.36623
	LSTM	0.00135	0.02390	0.03426	0.29107	**0.53529**	0.27010	0.38209
	NBeats	0.00114	0.02310	0.03214	0.27602	0.50000	0.72140	0.36963
	TimesNet	0.00119	0.02268	0.03232	0.29453	0.49265	0.29501	0.37863
	MSGNet	0.00116	0.02192	0.03184	0.28767	<u>0.52059</u>	<u>0.86428</u>	0.37536
	iTransformer	<u>0.00111</u>	<u>0.02151</u>	<u>0.03113</u>	0.27501	0.51176	**1.39200**	**0.31432**
	TimeXer	0.00114	0.02179	0.03143	0.28368	0.50735	0.53897	0.35839
	MSPCIFormer	**0.00105**	**0.02099**	**0.03040**	**0.27104**	0.49853	-0.17398	<u>0.34838</u>
5	DLinear	0.00188	0.02913	0.04106	**0.31165**	0.47761	-0.42867	0.38784
	LSTM	0.00205	0.02911	0.04164	0.36064	0.49701	0.59015	0.34512
	NBeats	<u>0.00167</u>	0.02746	0.03883	0.32089	0.51045	-0.04375	0.39982
	TimesNet	0.00176	0.02767	0.03901	0.32045	0.48955	<u>1.27503</u>	0.34533
	MSGNet	0.00171	0.02666	0.03826	0.33381	0.50896	**1.28072**	**0.29530**
	iTransformer	**0.00165**	0.02637	0.03817	0.31731	<u>0.52090</u>	1.17975	<u>0.30230</u>
	TimeXer	**0.00165**	<u>0.02609</u>	<u>0.03786</u>	<u>0.31633</u>	0.50299	0.27544	0.37637
	MSPCIFormer	**0.00165**	**0.02604**	**0.03783**	0.31894	**0.52239**	0.57325	0.34602
7	DLinear	0.00237	0.03288	0.04617	**0.36233**	0.49242	-0.30333	0.40195
	LSTM	0.00276	0.03400	0.04871	0.38923	**0.52727**	0.45213	0.35424
	NBeats	0.00243	0.03343	0.04721	0.37140	0.47576	-0.43952	0.45768
	TimesNet	0.00245	0.03187	0.04569	0.39701	0.46212	-0.32087	0.41555
	MSGNet	0.00233	0.03117	0.04466	0.38051	0.50909	**1.73921**	**0.24790**
	iTransformer	<u>0.00225</u>	<u>0.03063</u>	<u>0.04412</u>	0.37606	0.50758	<u>0.96258</u>	<u>0.32538</u>
	TimeXer	0.00241	0.03135	0.04535	<u>0.36259</u>	0.50152	0.20617	0.32919
	MSPCIFormer	**0.00221**	**0.03034**	**0.04378**	0.36303	<u>0.52273</u>	-0.02501	0.40613
15	DLinear	0.00520	0.05101	0.06896	**0.35409**	0.49032	0.49284	0.34108
	LSTM	0.00542	0.04866	0.06795	0.42728	0.48387	-0.93175	0.43531
	NBeats	0.00496	0.04872	0.06708	<u>0.36563</u>	**0.52258**	<u>0.99147</u>	**0.30977**
	TimesNet	0.00475	0.04625	0.06425	0.37573	0.50323	-0.10465	0.36214
	MSGNet	0.00480	0.04581	0.06410	0.37179	0.50323	**1.15391**	<u>0.31821</u>
	iTransformer	<u>0.00459</u>	<u>0.04499</u>	<u>0.06302</u>	0.38619	0.51452	0.07839	0.39869
	TimeXer	0.00501	0.04744	0.06580	0.38586	0.49839	0.00323	0.39456
	MSPCIFormer	**0.00454**	**0.04495**	**0.06296**	0.38090	<u>0.51613</u>	0.64450	0.34570

Peak inference memory measures the maximum GPU memory allocated in megabytes during a model's forward pass at inference timeGao et al. (2020). It reflects the worst-case memory usage when the model is deployed for prediction. In our expriment results, peak memory usage during inference is dominated by TimesNet, which is 2.4 times higher than the second-highest model LSTM. All other models cluster tightly between 17–23 MB. MSPCIFormer's peak memory of 20.7 MB is comfortably within the low-memory cluster, confirming that its multi-scale patching introduces negligible memory overhead over simpler Transformer architectures. TimesNet's high memory footprint may limit its applicability in resource-constrained deployment settings such as shared-GPU cloud environments.

Inference latency measures the average time in milliseconds to process a single input sample during the model's forward pass. It is a critical deployment constraint in automated trading systems, where predictions must be generated within strict time windows (Bhuiyan et al., 2025). MSPCIFormer achieves the fourth-lowest latency under hold-out validation, behind only NBeats, LSTM and iTransformer, and ranks first under walk-forward validation, making it the most latency-efficient Transformer under walk-forward and the second-most under hold-out. DLinear and TimesNet exhibit the highest inference latencies among all models. TimesNet's latency is expected, reflecting the computational cost of its FFT-based period detection and multi-scale inception convolution operations. However, DLinear's high latency is counter-intuitive for a model comprising only linear layers. The cause lies in its channel-individual decomposition configuration, selected by hyperparameter tuning for predictive accuracy. Rather than applying a single shared linear mapping across all channels, DLinear maintains independent trend

TABLE 8 Walk-forward validation - BTC forecasting performance across prediction horizons using an input window of 5 days. The best result for each horizon is shown in bold, and the second-best result is underlined.

Horizon	Model	MSE	MAE	RMSE	MAPE	MDA	Sharpe	Max Drawdown
1	DLinear	0.00487	0.04109	0.05826	**0.09742**	0.50290	-0.30175	0.27052
	LSTM	0.00518	0.04455	0.06140	0.10802	0.49710	0.33608	0.27217
	NBeats	0.00478	0.04127	0.05762	0.09838	0.51884	**1.99890**	**0.19512**
	TimesNet	0.00506	0.04269	0.05978	0.10187	0.50725	0.12178	0.26402
	MSGNet	0.00509	0.04251	0.06021	0.10350	0.54058	0.55542	0.29624
	iTransformer	0.00494	0.04315	0.05915	0.10204	0.47101	0.14974	0.33885
	TimeXer	0.00471	0.04076	0.05796	0.10114	0.53623	-0.45719	0.31433
	MSPCIFormer	**0.00462**	**0.04023**	**0.05707**	0.09744	**0.54348**	0.65412	0.25443
3	DLinear	0.00893	0.05764	0.08093	0.13306	0.53382	0.78431	**0.20409**
	LSTM	0.01011	0.06252	0.08683	0.14888	0.49559	0.01120	0.26985
	NBeats	0.00856	0.05634	0.07887	**0.12502**	0.50735	**1.09039**	0.20904
	TimesNet	0.00926	0.05859	0.08174	0.13404	0.49559	-0.05651	0.29062
	MSGNet	0.00858	0.05649	0.07956	0.13270	0.52794	0.98712	0.24203
	iTransformer	0.00869	0.05723	0.08032	0.13211	0.48824	-0.59276	0.36253
	TimeXer	0.00855	0.05640	0.08001	0.13358	0.51029	-0.46731	0.33105
	MSPCIFormer	**0.00834**	**0.05511**	**0.07811**	0.12884	**0.54559**	0.13699	0.28060
5	DLinear	0.01294	0.07054	0.09778	**0.15960**	0.50149	**1.52367**	**0.22283**
	LSTM	0.01543	0.07922	0.10797	0.18989	0.51343	0.32858	0.24831
	NBeats	0.01390	0.07487	0.10227	0.16813	0.51045	1.18836	0.22402
	TimesNet	0.01325	0.07169	0.09918	0.16716	0.51045	0.46700	0.26902
	MSGNet	0.01312	0.07129	0.09872	0.16724	0.49851	-0.68749	0.34824
	iTransformer	0.01300	0.07078	0.09862	0.16648	0.48209	-0.46005	0.35648
	TimeXer	0.01305	0.07044	0.09920	0.16915	0.51343	-0.67836	0.31194
	MSPCIFormer	**0.01266**	**0.06895**	**0.09701**	0.16346	**0.52537**	0.28584	0.26304
7	DLinear	0.01899	0.08700	0.11885	0.19462	0.49545	0.25656	0.27857
	LSTM	0.01893	0.08745	0.12018	0.21249	0.51061	-1.23794	0.33717
	NBeats	0.01790	0.08531	0.11643	**0.19087**	0.50152	**1.43526**	0.22586
	TimesNet	0.01815	0.08630	0.11705	0.20462	0.48788	-0.28531	0.29889
	MSGNet	0.01775	0.08383	0.11507	0.20268	0.52879	0.89366	**0.20909**
	iTransformer	0.01766	0.08287	0.11506	0.19735	0.49697	0.42682	0.26152
	TimeXer	**0.01697**	**0.08083**	**0.11293**	0.19489	0.52576	-0.01096	0.29282
	MSPCIFormer	0.01726	0.08235	0.11459	0.20479	**0.53939**	0.03985	0.27031
15	DLinear	0.03271	0.11806	0.16107	**0.28151**	0.51129	**1.19039**	**0.20789**
	LSTM	0.03417	0.11855	0.16210	0.30146	0.50323	-0.79412	0.28752
	NBeats	0.03779	0.12830	0.17262	0.30377	0.49839	-0.77091	0.29000
	TimesNet	0.03327	0.11789	0.16044	0.28774	0.49194	-0.55371	0.29318
	MSGNet	0.03509	0.11814	0.16278	0.29442	0.50645	-0.29051	0.27515
	iTransformer	0.03256	0.11478	0.15914	0.28759	0.53710	0.24307	0.27081
	TimeXer	**0.03151**	**0.11306**	**0.15704**	0.28292	0.52419	-0.64299	0.28694
	MSPCIFormer	0.03541	0.11726	0.16315	0.28405	**0.54516**	0.65532	0.26170

and seasonal layers for each of the 8 input channels, applied sequentially during the forward pass. This sequential per-channel execution causes inference time to scale linearly with the number of channels.

6 | CONCLUSION AND FUTURE WORK

This study proposes MSPCIFormer, a novel Transformer-based architecture that integrates multi-scale patching to capture the regime-dependent temporal dynamics inherent in cryptocurrency markets, and CI modeling to learn feature-specific patterns across multiple temporal scales. We evaluated its effectiveness on ADA, BTC and ETH price forecasting over multiple prediction horizons, benchmarking against several SOTA Transformer based time series models as well as established deep learning baselines, including LSTM, DLinear, NBeats and TimesNet. To further assess the model's generalization, we conduct walk-forward validation. Both statistical and economical meaningful metrics are included for the experiments in this study. The results collectively demonstrate that no single model achieves uniformly superior performance across all assets, prediction horizons, validation protocols, and evaluation criteria. Rather, model rankings are systematically shaped by the interplay of four factors: the prediction horizon relative to the model's designs, the market regime captured by the evaluation window, the choice between statistical accuracy and financial utility metrics, and the computational constraints of the deployment environment.

The prediction horizon relative to the model's designs: For ADA and ETH forecasting, MSPCIFormer exhibits a consistent horizon-dependent performance pattern: it ranks at or near the top at pred=1

TABLE 9 Walk-forward validation - ETH forecasting performance across prediction horizons using an input window of 5 days. The best result for each horizon is shown in bold, and the second-best result is underlined.

Horizon	Model	MSE	MAE	RMSE	MAPE	MDA	Sharpe	Max Drawdown
1	DLinear	0.00372	0.03721	0.05334	1.24833	0.50870	-0.30541	0.34671
	LSTM	0.00413	0.04165	0.05722	1.20606	0.49420	0.20351	0.34295
	NBeats	0.00345	0.03741	0.05240	1.27661	**0.54348**	**1.67314**	**0.21978**
	TimesNet	0.00433	0.04004	0.05684	1.27072	0.52464	1.50698	0.26578
	MSGNet	0.00373	0.03787	0.05417	**1.10341**	0.52464	0.86412	0.27112
	iTransformer	0.00363	0.03831	0.05358	1.48177	0.50725	0.85079	0.30018
	TimeXer	0.00358	0.03700	0.05316	1.22503	0.52899	0.04100	0.29248
	MSPCIFormer	**0.00341**	**0.03606**	**0.05174**	1.18138	0.53768	0.82271	0.25349
3	DLinear	**0.00646**	**0.04986**	**0.07099**	1.11460	0.52647	0.75698	0.27430
	LSTM	0.00835	0.05578	0.07909	1.65116	0.50882	0.75567	**0.24038**
	NBeats	0.00652	0.05045	0.07175	1.15624	0.50735	-0.35368	0.35027
	TimesNet	0.00689	0.05142	0.07326	1.01002	0.53088	1.36107	0.27028
	MSGNet	0.00699	0.05145	0.07362	**0.71783**	**0.55735**	**1.45837**	0.28505
	iTransformer	0.00674	0.05184	0.07287	1.08729	0.51765	1.08155	0.25664
	TimeXer	0.00655	0.05013	0.07170	1.00359	0.53529	0.57374	0.26156
	MSPCIFormer	0.00657	0.04998	0.07170	1.08384	0.53971	0.54238	0.27347
5	DLinear	0.01018	0.06313	0.08900	1.23466	0.50896	0.56250	0.27422
	LSTM	0.01202	0.06908	0.09670	1.51163	**0.55970**	1.55565	**0.21182**
	NBeats	0.00996	0.06358	0.08861	1.18622	0.51940	0.96750	0.25424
	TimesNet	0.01009	0.06225	0.08812	1.11689	0.53433	**2.03313**	0.25476
	MSGNet	0.01042	0.06330	0.08953	1.09892	0.50448	0.45741	0.26756
	iTransformer	0.01007	0.06291	0.08845	1.36359	0.52985	1.31346	0.24065
	TimeXer	0.01001	0.06213	0.08802	**1.02367**	0.52388	0.97941	0.22263
	MSPCIFormer	**0.00981**	**0.06125**	**0.08712**	1.15393	0.52836	0.65025	0.29581
7	DLinear	0.01454	0.07495	0.10431	1.92245	0.54091	1.25825	0.21572
	LSTM	0.01793	0.08156	0.11451	1.89130	0.51364	0.29941	0.26690
	NBeats	**0.01338**	0.07479	0.10318	1.94241	0.49697	0.38404	0.28897
	TimesNet	0.01440	0.07370	0.10377	**1.75685**	**0.55303**	**2.55445**	**0.19042**
	MSGNet	0.01412	0.07372	0.10378	1.77525	0.51364	-0.69215	0.34112
	iTransformer	0.01351	0.07303	0.10234	1.94946	0.49394	0.41017	0.30746
	TimeXer	0.01344	**0.07188**	**0.10153**	1.76508	0.53636	1.02212	0.23835
	MSPCIFormer	0.01359	0.07285	0.10271	1.77666	0.51515	-0.00639	0.34367
15	DLinear	0.02535	0.10114	0.13953	**3.13485**	0.52581	**1.09948**	0.24467
	LSTM	0.02830	0.10750	0.14642	3.54939	0.51613	0.47437	0.23867
	NBeats	**0.02469**	0.10314	0.14183	3.23694	0.50161	1.07973	0.22714
	TimesNet	0.02623	0.10277	0.14176	3.49300	0.51613	-0.28966	0.30869
	MSGNet	0.02687	0.10500	0.14274	3.16696	0.49677	-0.59400	0.34053
	iTransformer	0.02605	0.10224	0.14091	3.71418	**0.53871**	0.84206	0.25382
	TimeXer	0.02584	0.10133	0.14013	3.50480	0.53387	0.78244	**0.21697**
	MSPCIFormer	0.02539	**0.10039**	**0.13914**	3.58061	0.53871	0.59942	0.25451

TABLE 10 Computational Efficiency Summary. Epoch time and latency are averages; peak memory is the observed maximum. The metrics are averaged across ADA, BTC, ETH and all prediction horizons.

Model	Epoch Time HO (s)	Epoch Time WF (s)	Peak Mem HO (MB)	Peak Mem WF (MB)	Latency HO (ms/sample)	Latency WF (ms/sample)
DLinear	1.601	1.260	17.34	17.34	4.727	5.028
LSTM	0.214	0.140	80.80	80.80	1.744	2.850
NBeats	0.141	0.104	21.23	21.23	1.720	2.607
TimesNet	3.955	3.125	193.77	194.90	5.280	3.794
MSGNet	1.300	1.024	21.83	21.83	3.088	2.644
iTransformer	0.520	0.429	20.66	20.66	1.840	2.577
TimeXer	0.379	0.294	22.93	22.93	2.021	2.634
MSPCIFormer	0.348	0.265	21.05	20.73	1.875	2.513

and degrades relative to baselines at pred=[3, 7], before partially recovering at pred=15 under hold-out evaluation. This degradation at medium horizons is attributable to the fixed patch sizes in MSPCI-Former's patching module becoming increasingly misaligned with the growing prediction window. Our ablation experiments support this

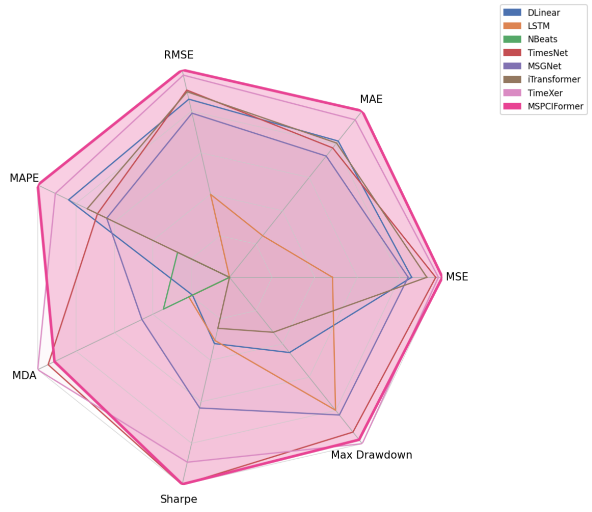

FIGURE 4 Performance radar chart for BTC under hold-out validation, where the outer boundary denotes the best-performing model. Values are averaged across prediction horizons of 1, 3, 5, 7, and 15 days. A larger enclosed area indicates superior overall performance across both statistical and financial metrics.

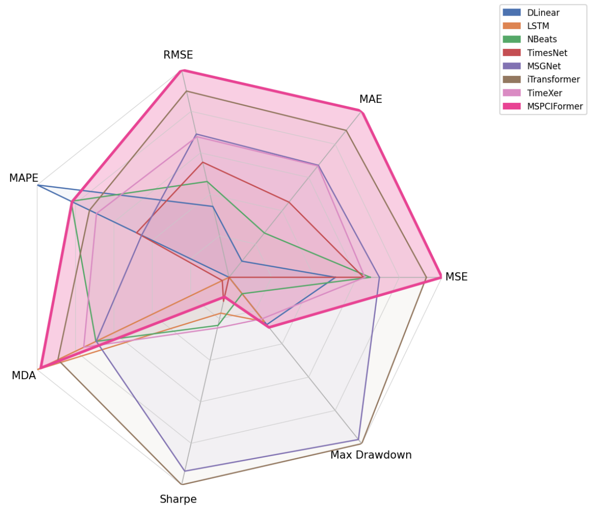

FIGURE 6 Performance radar chart for ADA under walk-forward validation, where the outer boundary denotes the best-performing model. Values are averaged across prediction horizons of 1, 3, 5, 7, and 15 days. A larger enclosed area indicates superior overall performance across both statistical and financial metrics.

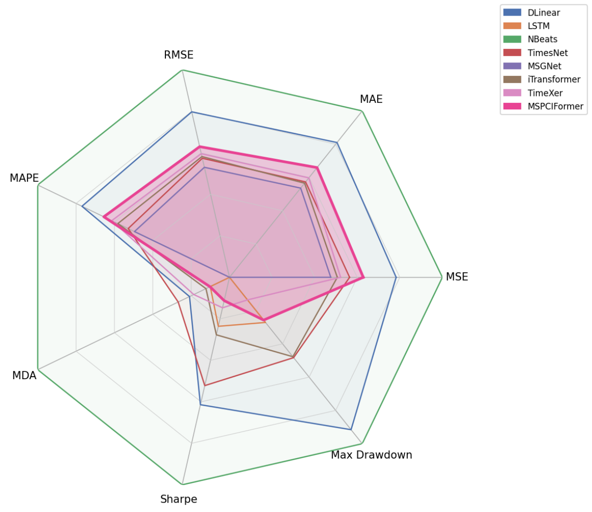

FIGURE 5 Performance radar chart for ETH under hold-out validation, where the outer boundary denotes the best-performing model. Values are averaged across prediction horizons of 1, 3, 5, 7, and 15 days. A larger enclosed area indicates superior overall performance across both statistical and financial metrics.

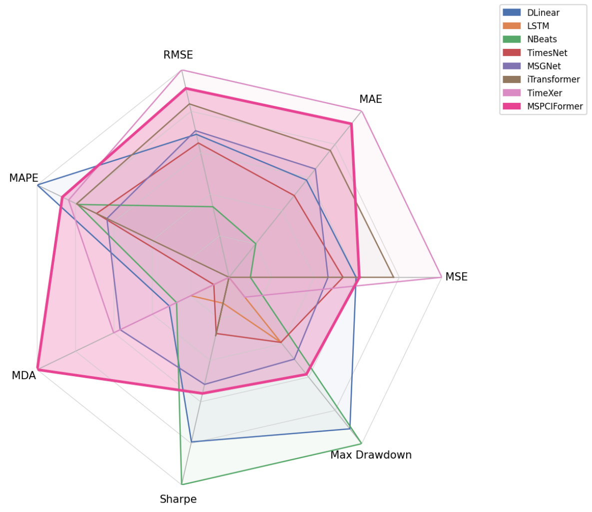

FIGURE 7 Performance radar chart for BTC under walk-forward validation, where the outer boundary denotes the best-performing model. Values are averaged across prediction horizons of 1, 3, 5, 7, and 15 days. A larger enclosed area indicates superior overall performance across both statistical and financial metrics.

explanation by increasing patch granularity consistently closes the medium-horizon performance gap. This finding has a direct architectural implication: adaptive or learned patch scale selection that adjusts with the prediction horizon would likely close the performance gap further. We leave this as a direction for future work.

The market regime captured by the evaluation window: A central contribution of this paper is the dual validation framework, and the results justify its necessity. Hold-out and walk-forward rankings diverge substantially: MSPCIFormer ranks last or near-last on several ADA and ETH hold-out horizons but consistently top-1 or top-2 on the same metrics under walk-forward evaluation. Conversely, TimesNet and NBeats show strong hold-out Sharpe but more variable walk-forward behavior.

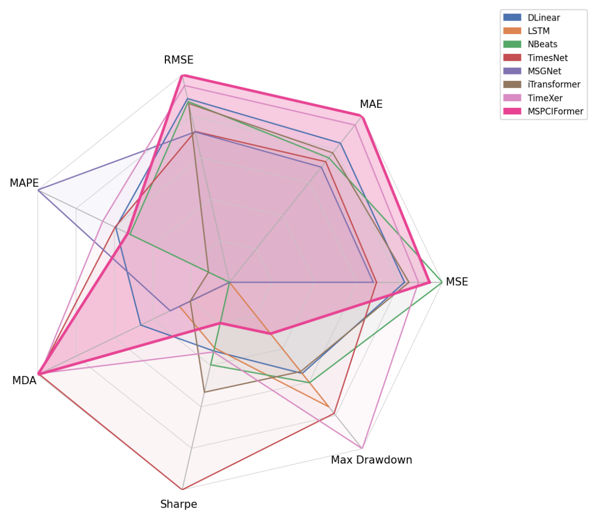

FIGURE 8 Performance radar chart for ETH under walk-forward validation, where the outer boundary denotes the best-performing model. Values are averaged across prediction horizons of 1, 3, 5, 7, and 15 days. A larger enclosed area indicates superior overall performance across both statistical and financial metrics.

This divergence arises because the static hold-out test set falls predominantly within a single market regime, while walk-forward evaluation distributes test windows across various regimes. The practical implication is significant: a model selected on hold-out alone may have been regime overfitted. Walk-forward validation provides a more honest estimate of deployment performance.

The choice between statistical accuracy and financial utility metrics: A consistent finding across all three assets is that MSE and Sharpe Ratio rankings are poorly correlated. The most striking example is NBeats, which achieves the best MSE across multiple ETH horizons yet shows negative Sharpe at pred=7 under hold-out. Conversely, TimesNet ranks poorly on MSE at pred=7 for ETH walk-forward, yet achieves the highest Sharpe. This divergence arises because MSE measures the magnitude of price level prediction error, while Sharpe depends on the model's ability to correctly identify the direction and magnitude of returns at the next step (Jabbar & Jalil, 2024). This result motivates the dual evaluation framework adopted in this study and suggests that papers reporting only statistical metrics are providing an incomplete picture of model utility in trading contexts.

The computational constraints of the deployment environment: The computational efficiency results reveal a meaningful trade-off between computational cost and predictive accuracy. TimesNet achieves the highest Sharpe Ratio in the study (+2.554 for ETH walk-forward at pred=7), but at a substantial cost: approximately 11 times slower training than MSPCIFormer, 9 times higher peak memory than the average model, and the highest inference latency. For resource-constrained deployment, TimesNet's cost profile is prohibitive. MSPCIFormer achieves competitive or best-in-class statistical accuracy across most assets and horizons under walk-forward evaluation, while maintaining low training cost and memory, and the lowest walk-forward inference latency among all baselines. NBeats offers faster training and comparable peak memory footprint to MSPCIFormer, but delivers less competitive statistical accuracy, particularly at short prediction horizons under walk-forward evaluation. DLinear's surprisingly high inference latency despite being a linear model is a practical reminder that inference cost cannot be inferred from model complexity alone. Empirical experiments reveals the cause to be its channel-individual decomposition configuration which maintains independent trend and seasonal layers for each of the 8 input channels.

While MSPCIFormer has demonstrated strong performance on BTC hold-out validation and walk-forward validation, several areas remain open for future work:

1. Investigating learned patch scale selection mechanism such as the multi-scale weighting mechanism used in Pathformer (P. Chen et al., 2024) to dynamically adjusts the contribution of each patch scale based on the input's temporal dynamics and the target forecast horizon, rather than relying on a fixed patch configuration determined at training time. Cai et al. (Cai et al., 2023) used a Mixture of Experts (MoE) mechanism to prioritize scale-specific representations during fusion, while Zhang et al. (Zhang et al., 2024) combined the token representations obtained from different scales through flattening, concatenation, and subsequent linear layer.
2. Enhancing the model's contextual awareness by integrating sentiment information such as cryptocurrency related financial news (Hossain et al., 2024) and aggregate market sentiment measures such as the Fear & Greed Index (Lee, 2025).
3. Examine different approaches like cross-attention for fusing time series information and sentiment information. For example, TimeXer (Wang et al., 2024) utilized a cross-attention mechanism to fuse endogenous and exogenous information.

FINANCIAL DISCLOSURE

This research was conducted with the partial financial support of Taighde Éireann-Research Ireland under Grant Agreement No. 13/RC/2106_P2 at ADAPT, the Research Ireland Centre for AI-Driven Digital Content Technology at DCU funded through the Research Ireland Research Centres Programme.

SUPPORTING INFORMATION

Code: All the code used in this paper is available online: `https://github.com/cmajorsolo/MSPCIFormer`

References

Acosta, R. P., & Arana, A. Z. (2023, 12). Cardano cryptocurrency price from twitter. a prediction algorithm from machine learning. *SSRG International Journal of Electronics and Communication Engineering*, *10*, 33-44. doi: 10.14445/23488549/ijece-v10i12p104

Adhikari, R., & Agrawal, R. K. (2013). *An introductory study on time series modeling and forecasting.* Retrieved from https://arxiv.org/abs/1302.6613

Agakishiev, I., Härdle, W. K., Becker, D., et al. (2025). Regime switching forecasting for cryptocurrencies. *Digital Finance*, *7*, 107–131. Retrieved from https://doi.org/10.1007/s42521-024-00123-2 doi: 10.1007/s42521-024-00123-2

Ahmadi, F., Eshlaghi, A. T., & Radfar, R. (2024). Examining and comparing the efficiency of mlp and simplernn algorithms in cryptocurrency price prediction. *Management Strategies and Engineering Sciences*, *6*(3), 121-137. Retrieved from http://193.36.85.187:8092/index.php/mses/article/view/89 doi: https://doi.org/10.61838/msesj.6.3.12

Akiba, T., Sano, S., Yanase, T., Ohta, T., & Koyama, M. (2019). Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th acm sigkdd international conference on knowledge discovery & data mining* (p. 2623–2631). New York, NY, USA: Association for Computing Machinery. Retrieved from https://doi.org/10.1145/3292500.3330701 doi: 10.1145/3292500.3330701

Aliferis, C., & Simon, G. (2024). Overfitting, underfitting and general model overconfidence and under-performance pitfalls and best practices in machine learning and ai. In G. J. Simon & C. Aliferis (Eds.), *Artificial intelligence and machine learning in health care and medical sciences: Best practices and pitfalls* (pp. 477–524). Cham: Springer International Publishing. Retrieved from https://doi.org/10.1007/978-3-031-39355-6_10 doi: 10.1007/978-3-031-39355-6_10

AlMadany, N. N., AlMadany, O., Naymat, G. A., & Maghyereh, A. (2024). Forecasting cryptocurrency returns using classical statistical and deep learning techniques. *International Journal of Information Management Data Insights*, *4*(2), 100251. Retrieved from https://www.sciencedirect.com/science/article/pii/S2667096824000405 doi: https://doi.org/10.1016/j.jjimei.2024.100251

Alnami, H., Mohzary, M., Assiri, B., & Zangoti, H. (2025). An integrated framework for cryptocurrency price forecasting and anomaly detection using machine learning. *Applied Sciences*, *15*(4). Retrieved from https://www.mdpi.com/2076-3417/15/4/1864 doi: 10.3390/app15041864

Alsini, R., Abu Al-Haija, Q., Alsulami, A. A., Alturki, B., Alqurashi, A. A., Mashat, M. D., … Alhebaishi, N. (2024, May). Forecasting cryptocurrency's buy signal with a bagged tree learning approach to enhance purchase decisions. *Frontiers in Big Data*, *7*, 1369895. Retrieved from https://doi.org/10.3389/fdata.2024.1369895 doi: 10.3389/fdata.2024.1369895

Amédée-Manesme, C.-O., & Barthélémy, F. (2022). Proper use of the modified sharpe ratios in performance measurement: rearranging the cornish fisher expansion. *Annals of Operations Research*, *313*(2), 691–712.

Anderson, B. (2023). A tick-by-tick level measurement of the lead-lag duration between cryptocurrencies: The case of bitcoin versus cardano. *Investment Management & Financial Innovations*, *20*(1), 174.

Ang, Y., Wang, Q., Huang, Q., Bao, Y., Xi, X., Tung, A. K. H., … Huang, Z. (2026). CTBench: Cryptocurrency time series generation benchmark. In *The fourteenth international conference on learning representations.* Retrieved from https://openreview.net/forum?id=RzT2sombPD

Azari, A. (2019). Bitcoin price prediction: An arima approach. *arXiv preprint arXiv:1904.05315*.

Bhuiyan, M. S. M., Rafi, M. A., Rodrigues, G. N., Mir, M. N. H., Ishraq, A., Mridha, M., & Shin, J. (2025). Deep learning for algorithmic trading: A systematic review of predictive models and optimization strategies. *Array*, *26*, 100390. Retrieved from https://www.sciencedirect.com/science/article/pii/S2590005625000177 doi: https://doi.org/10.1016/j.array.2025.100390

Bourday, R., Aatouchi, I., Kerroum, M., & Zaaouat, A. (2024). Cryptocurrency forecasting using deep learning models: A comparative analysis. *HIGHTECH AND INNOVATION JOURNAL Учредители: Ital Publication*, *5*(4), 1055–1067.

Bouteska, A., Abedin, M. Z., Hajek, P., & Yuan, K. (2024). Cryptocurrency price forecasting – a comparative analysis of ensemble learning and deep learning methods. *International Review of Financial Analysis*, *92*, 103055. Retrieved from https://www.sciencedirect.com/science/article/pii/S1057521923005719 doi: https://doi.org/10.1016/j.irfa.2023.103055

Briola, A., Vidal-Tomás, D., Wang, Y., & Aste, T. (2023). Anatomy of a stablecoin's failure: The terra-luna case. *Finance Research Letters*, *51*, 103358. Retrieved from https://www.sciencedirect.com/science/article/pii/S1544612322005359 doi: https://doi.org/10.1016/j.frl.2022.103358

Börjesson, L., & Singull, M. (2020). Forecasting financial time series through causal and dilated convolutional neural networks. *Entropy*, *22*(10). Retrieved from https://www.mdpi.com/1099-4300/22/10/1094 doi: 10.3390/e22101094

Cai, W., Liang, Y., Liu, X., Feng, J., & Wu, Y. (2023). *Msgnet: Learning multi-scale inter-series correlations for multivariate time series forecasting.* Retrieved from https://arxiv.org/abs/2401.00423

Cao, Y., Tian, Z., Guo, W., & Liu, X. (2025). Mspatch: A multi-scale patch mixing framework for multivariate time series forecasting. *Expert Systems with Applications*, *273*, 126849. Retrieved from https://www.sciencedirect.com/science/article/pii/S0957417425004713 doi: https://doi.org/10.1016/j.eswa.2025.126849

Chaudhary, D., & Saroj, S. K. (2023). Cryptocurrency price prediction using supervised machine learning algorithms. *ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal*, *12*(1), e31490. Retrieved from `https://adcaij.usal.es/index.php/ADCAIJ/article/view/31490` Regular Issue. doi: 10.14201/adcaij.31490

Chen, B., & Sun, Y. (2024). Risk characteristics and connectedness in cryptocurrency markets: New evidence from a non-linear framework. *The North American Journal of Economics and Finance*, *69*, 102036. doi: https://doi.org/10.1016/j.najef.2023.102036

Chen, P., ZHANG, Y., Cheng, Y., Shu, Y., Wang, Y., Wen, Q., ... Guo, C. (2024). Pathformer: Multi-scale transformers with adaptive pathways for time series forecasting. In *The twelfth international conference on learning representations.* Retrieved from `https://openreview.net/forum?id=lJkOCMP2aW`

Chi, C., Wang, X., Yang, K., Song, Z., Jin, D., Zhu, L., ... Feng, J. (2024). *Injecttst: A transformer method of injecting global information into independent channels for long time series forecasting.* Retrieved from `https://arxiv.org/abs/2403.02814`

Choi, J. (2021). Maximum drawdown, recovery, and momentum. *Journal of Risk and Financial Management*, *14*(11), 542.

CoinMarketCap. (2025, Oct). *Cryptocurrency prices, charts and market capitalizations.* Retrieved from https://coinmarketcap.com/. Accessed: 10th Apr 2025.

de Melo Mendes, B. V., & Lavrado, R. C. (2017). Implementing and testing the maximum drawdown at risk. *Finance Research Letters*, *22*, 95-100. Retrieved from `https://www.sciencedirect.com/science/article/pii/S1544612317300156` doi: https://doi.org/10.1016/j.frl.2017.06.001

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the north american chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)* (pp. 4171–4186).

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., ... Houlsby, N. (2021). *An image is worth 16x16 words: Transformers for image recognition at scale.* Retrieved from `https://arxiv.org/abs/2010.11929`

Gao, Y., Liu, Y., Zhang, H., Li, Z., Zhu, Y., Lin, H., & Yang, M. (2020). Estimating gpu memory consumption of deep learning models. In *Proceedings of the 28th acm joint meeting on european software engineering conference and symposium on the foundations of software engineering* (p. 1342–1352). New York, NY, USA: Association for Computing Machinery. Retrieved from `https://doi.org/10.1145/3368089.3417050` doi: 10.1145/3368089.3417050

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

Hossain, M. F. B., Lamia, L. Z., Rahman, M. M., & Khan, M. M. (2024, November). *FinBERT-BiLSTM: A Deep Learning Model for Predicting Volatile Cryptocurrency Market Prices Using Market Sentiment Dynamics* (Papers 2411.12748). arXiv.org. Retrieved from `https://ideas.repec.org/p/arx/papers/2411.12748.html`

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

Iuga, I. C., Nerișanu, R.-A., & and, L.-L. D. (2024). Volatility and spillover analysis between cryptocurrencies and financial indices: a diagonal bekk and dcc garch model approach in support of sdgs. *Cogent Economics & Finance*, *12*(1), 2437002. Retrieved from `https://doi.org/10.1080/23322039.2024.2437002` doi: 10.1080/23322039.2024.2437002

Jabbar, A., & Jalil, S. Q. (2024). A comprehensive analysis of machine learning models for algorithmic trading of bitcoin. *arXiv preprint arXiv:2407.18334*.

Jaquart, P., Köpke, S., & Weinhardt, C. (2022). Machine learning for cryptocurrency market prediction and trading. *The Journal of Finance and Data Science*, *8*, 331-352. Retrieved from `https://www.sciencedirect.com/science/article/pii/S2405918822000174` doi: https://doi.org/10.1016/j.jfds.2022.12.001

Jayavarma, A., Parakkat Kesava Panikker, P., & Nair, M. G. (2025). Revolutionizing the energy sector: exploring diversified blockchain platforms for a sustainable future. *Frontiers in Blockchain*, *8*, 1544770.

Jierula, A., Wang, S., OH, T.-M., & Wang, P. (2021). Study on accuracy metrics for evaluating the predictions of damage locations in deep piles using artificial neural networks with acoustic emission data. *Applied Sciences*, *11*(5). Retrieved from `https://www.mdpi.com/2076-3417/11/5/2314` doi: 10.3390/app11052314

John, D. L., Binnewies, S., & Stantic, B. (2024). Cryptocurrency price prediction algorithms: A survey and future directions. *Forecasting*, *6*(3), 637–671. Retrieved from `https://www.mdpi.com/2571-9394/6/3/34` doi: 10.3390/forecast6030034

Kehinde, T., Adedokun, O., & Joseph, A. e. a. (2025). Helformer: an attention-based deep learning model for cryptocurrency price forecasting. *Big Data*, *12*. Retrieved from `https://doi.org/10.1186/s40537-025-01135-4`

Kiram, S., Ula, M., & Kurniawati. (2025). A comparative study of temporal convolutional network and gated recurrent unit for predicting *ethereum* prices. *Applied Engineering Innovation and Technology*, *2*(1), 1–8. Retrieved from `https://doi.org/10.62777/aeit.v2i1.55` doi: 10.62777/aeit.v2i1.55

Korkusuz, B. (2025). Coin impact on cross-crypto realized volatility and dynamic cryptocurrency volatility connectedness. *Financial Innovation*, *11*. doi: 10.1186/s40854-025-00881-x

Kristoufek, L. (2022). On the role of stablecoins in cryptoasset pricing dynamics. *Financial Innovation*, *8*(1), 37. Retrieved from `https://doi.org/10.1186/s40854-022-00343-8` doi: 10.1186/s40854-022-00343-8

Kumar, S., Jain, R., Narain, Balli, F., & Billah, M. (2023). Interconnectivity and investment strategies among commodity prices, cryptocurrencies, and g-20 capital markets: A comparative analysis during covid-19 and russian-ukraine war. *International Review of Economics Finance*, *88*, 547-593. Retrieved from https://www.sciencedirect.com/science/article/pii/S1059056023002046 doi: https://doi.org/10.1016/j.iref.2023.06.039

Lai, G., Chang, W.-C., Yang, Y., & Liu, H. (2018). Modeling long and short-term temporal patterns with deep neural networks. In *Sigir '18: The 41st international acm sigir conference on research development in information retrieval.* Retrieved from https://dl.acm.org/doi/10.1145/3209978.3210006 doi: 10.1145/3209978.3210006

Latif, N., Selvam, J. D., Kapse, M., Sharma, V., & Mahajan, V. (2023). Comparative performance of lstm and arima for the short-term prediction of bitcoin prices. *Australasian Accounting, Business and Finance Journal*, *17*(1), 256–276.

Lee, M. C. (2025). Temporal fusion transformer-based trading strategy for multi-crypto assets using on-chain and technical indicators. *Systems*, *13*(6). Retrieved from https://www.mdpi.com/2079-8954/13/6/474 doi: 10.3390/systems13060474

Li, S., Jin, X., Xuan, Y., Zhou, X., Chen, W., Wang, Y.-X., & Yan, X. (2020). *Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting.* Retrieved from https://arxiv.org/abs/1907.00235

Liu, M., Zeng, A., Chen, M., Xu, Z., LAI, Q., Ma, L., & Xu, Q. (2022). Scinet: Time series modeling and forecasting with sample convolution and interaction. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, & A. Oh (Eds.), *Advances in neural information processing systems* (Vol. 35, pp. 5816–5828). Curran Associates, Inc. Retrieved from https://proceedings.neurips.cc/paper_files/paper/2022/file/266983d0949aed78a16fa4782237dea7-Paper-Conference.pdf

Liu, S., Yu, H., Liao, C., Li, J., Lin, W., Liu, A. X., & Dustdar, S. (2022). Pyraformer: Low-complexity pyramidal attention for long-range time series modeling and forecasting. In *International conference on learning representations.* Retrieved from https://openreview.net/forum?id=0EXmFzUn5I

Liu, Y., He, T., Zhang, H., Wu, H., Wang, S., Ma, L., & Long, M. (2024). *itransformer: Inverted transformers are effective for time series forecasting.* Retrieved from https://arxiv.org/abs/2310.06625

Luo, D., & Wang, X. (2024). Deformabletst: Transformer for time series forecasting without over-reliance on patching. *Advances in Neural Information Processing Systems*, *37*, 88003–88044.

Madhulatha, T. S., & Ghori, D. M. A. S. (2025). Forecasting currency exchange rates using arima, ets and rnn: A machine learning perspective. *Journal of Theoretical and Applied Information Technology*, *103*(15).

Makinde, A. (2024). Optimizing time series forecasting: A comparative study of adam and nesterov accelerated gradient on lstm and gru networks using stock market data. *arXiv preprint arXiv:2410.01843*.

Makridakis, S., Spiliotis, E., & Assimakopoulos, V. (2018). The m4 competition: Results, findings, conclusion and way forward. *International Journal of Forecasting*, *34*(4), 802-808. Retrieved from https://www.sciencedirect.com/science/article/pii/S0169207018300785 doi: https://doi.org/10.1016/j.ijforecast.2018.06.001

Nakamoto, S. (2008). *Bitcoin: A peer-to-peer electronic cash system.* Retrieved from https://bitcoin.org/bitcoin.pdf

Nie, Y., Nguyen, N. H., Sinthong, P., & Kalagnanam, J. (2023). *A time series is worth 64 words: Long-term forecasting with transformers.* Retrieved from https://arxiv.org/abs/2211.14730

Omole, O., & Enke, D. (2024). Deep learning for bitcoin price direction prediction: models and trading strategies empirically compared. *Financial Innovation*, *10*(1), 117.

Oreshkin, B. N., Carpov, D., Chapados, N., & Bengio, Y. (2020). N-beats: Neural basis expansion analysis for interpretable time series forecasting. In *International conference on learning representations (iclr).* Retrieved from https://arxiv.org/abs/1905.10437 doi: 10.48550/arXiv.1905.10437

Otabek, S., & Choi, J. (2024). From prediction to profit: A comprehensive review of cryptocurrency trading strategies and price forecasting techniques. *IEEE Access*, *12*, 87039-87064. doi: 10.1109/ACCESS.2024.3417449

Penmetsa, S., & Vemula, M. (2023). Cryptocurrency price prediction with lstm and transformer models leveraging momentum and volatility technical indicators. In *2023 ieee 3rd international conference on data science and computer application (icdsca)* (p. 411-416). doi: 10.1109/ICDSCA59871.2023.10393319

PWC. (2025, November). *Annual global crypto hedge fund report 2025.* Retrieved from https://caymanfinance.ky/wp-content/uploads/2025/11/7th-Annual-Global-Crypto-Hedge-Fund-Report.pdf.

Ren, Y.-S., Ma, C.-Q., Kong, X.-L., Baltas, K., & Zureigat, Q. (2022). Past, present, and future of the application of machine learning in cryptocurrency research. *Research in International Business and Finance*, *63*, 101799. Retrieved from https://www.sciencedirect.com/science/article/pii/S0275531922001854 doi: https://doi.org/10.1016/j.ribaf.2022.101799

Seabe, P. L., Moutsinga, C. R. B., & Pindza, E. (2023). Forecasting cryptocurrency prices using lstm, gru, and bi-directional lstm: A deep learning approach. *Fractal and Fractional*, *7*(2). Retrieved from https://www.mdpi.com/2504-3110/7/2/203 doi: 10.3390/fractalfract7020203

Si, Y. (2022, 12). Using arima model to analyse and predict bitcoin price. *BCP Business Management*, *34*, 1210-1216. doi: 10.54691/bcpbm.v34i.3161

Singh, S., & Bhat, M. (2024). *Transformer-based approach for ethereum price prediction using crosscurrency correlation and sentiment analysis.* Retrieved from https://arxiv.org/abs/2401.08077

Sutter, K. M. (2025, August 22). *U.s. export controls and china: Advanced semiconductors* (CRS Report R48642). Congressional Research Service. Retrieved from https://www.congress.gov/crs_external_products/R/PDF/R48642/R48642.2.pdf

Taheri Hosseinkhani, N. (2025). Comparative analysis of financial performance among traditional quantitative, ai-based, and hybrid modeling approaches.

Tang, X., & Xie, Y. (2025). Exchange rate forecasting: A deep learning framework combining adaptive signal decomposition and dynamic weight optimization. *International Journal of Financial Studies*, *13*(3). Retrieved from https://www.mdpi.com/2227-7072/13/3/151 doi: 10.3390/ijfs13030151

Upreti, P. N., & Vásquez Callo-Müller, M. (2020, 03). Phase one us-china trade deal: What does it mean for intellectual property? *GRUR International*, *69*(4), 389-393. Retrieved from https://doi.org/10.1093/grurint/ikaa033 doi: 10.1093/grurint/ikaa033

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... Polosukhin, I. (2017). Attention is all you need. In I. Guyon et al. (Eds.), *Advances in neural information processing systems* (Vol. 30). Curran Associates, Inc. Retrieved from https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

Wang, Y., Wu, H., Dong, J., Qin, G., Zhang, H., Liu, Y., ... Long, M. (2024). Timexer: Empowering transformers for time series forecasting with exogenous variables. *Advances in Neural Information Processing Systems*. Retrieved from https://arxiv.org/abs/2402.19072 arXiv:2402.19072v4.

Wątorek, M., Bezbradica, M., Crane, M., Kwapień, J., & Drożdż, S. (2025). Filtering amplitude dependence of correlation dynamics in complex systems: application to the cryptocurrency market. *arXiv preprint arXiv:2509.18820*.

Wu, H., Hu, T., Liu, Y., Zhou, H., Wang, J., & Long, M. (2023). Timesnet: Temporal 2d-variation modeling for general time series analysis. In *International conference on learning representations (iclr)*. Retrieved from https://arxiv.org/abs/2210.02186 doi: 10.48550/arXiv.2210.02186

Wu, H., Xu, J., Wang, J., & Long, M. (2022). *Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting*. Retrieved from https://arxiv.org/abs/2106.13008

Wątorek, M., Kwapień, J., & Drożdż, S. (2023). Cryptocurrencies are becoming part of the world global financial market. *Entropy*, *25*(2). Retrieved from https://www.mdpi.com/1099-4300/25/2/377 doi: 10.3390/e25020377

Yildirim, B., & Taskiran, M. (2024). Optuna based optimized transformer model approach in bitcoin time series analysis. In *2024 26th international conference on digital signal processing and its applications (dspa)* (p. 1-6). doi: 10.1109/DSPA60853.2024.10510091

Yildirim, H., & Bekun, F. V. (2023). Predicting volatility of bitcoin returns with arch, garch and egarch models. *Future Business Journal*, *9*(1), 75. Retrieved from https://doi.org/10.1186/s43093-023-00255-8 doi: 10.1186/s43093-023-00255-8

Yousaf, M., Tariq, M., Jabbar, A., & Jalil, S. Q. (2025). A comprehensive survey of cryptocurrency forecasting: Methods, trends, and challenges. *SSRN*. Retrieved from https://doi.org/10.1007/s10462-024-11044-2 doi: 10.2139/ssrn.5234539

Zeng, A., Chen, M., Zhang, L., & Xu, Q. (2022). *Are transformers effective for time series forecasting?* Retrieved from https://arxiv.org/abs/2205.13504

Zerveas, G., Jayaraman, S., Patel, D., Bhamidipaty, A., & Eickhoff, C. (2021). A transformer-based framework for multivariate time series representation learning. In *Proceedings of the 27th acm sigkdd conference on knowledge discovery & data mining* (pp. 2114–2124).

Zhang, Y., Ma, L., Pal, S., Zhang, Y., & Coates, M. (2024). Multi-resolution time-series transformer for long-term forecasting. In S. Dasgupta, S. Mandt, & Y. Li (Eds.), *Proceedings of the 27th international conference on artificial intelligence and statistics* (Vol. 238, pp. 4222–4230). PMLR. Retrieved from https://proceedings.mlr.press/v238/zhang24l.html

Zhang, Y., & Yan, J. (2023). Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting. In *The eleventh international conference on learning representations*. Retrieved from https://openreview.net/forum?id=vSVLM2j9eie

Zhong, S., Song, S., Zhuo, W., Li, G., Liu, Y., & Chan, S. H. G. (2024). *A multi-scale decomposition mlp-mixer for time series analysis*. Retrieved from https://arxiv.org/abs/2310.11959

Zhou, H., Zhang, S., Peng, J., Zhang, S., Li, J., Xiong, H., & Zhang, W. (2021). *Informer: Beyond efficient transformer for long sequence time-series forecasting*. Retrieved from https://arxiv.org/abs/2012.07436

Zhou, T., Ma, Z., Wen, Q., Wang, X., Sun, L., & Jin, R. (2022). *Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting*. Retrieved from https://arxiv.org/abs/2201.12740

Zhou, Z., Song, Z., Xiao, H., & Ren, T. (2023). Multi-source data driven cryptocurrency price movement prediction and portfolio optimization. *Expert Systems with Applications*, *219*, 119600. Retrieved from https://www.sciencedirect.com/science/article/pii/S095741742300101X doi: https://doi.org/10.1016/j.eswa.2023.119600

Zhu, J. (2023). Bitcoin price prediction: Arima & sarima vs linear regression. *Advances in Economics, Management and Political Sciences*, *61*(1), 47–54.