A Tailored WGAN-GP Approach to Synthesise Realistic Stock Market Sequences

Alexander Mak-Pearce

2119504

Project Dissertation

Swansea University
Prifysgol Abertawe

Department of Computer Science

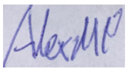
Adran Gyfrifiadureg

6th May 2025

Declaration

Statement 1

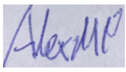
This work has not been previously accepted in substance for any degree and is not being concurrently submitted in candidature for any degree.

Signed  Alexander Mak-Pearce (2119504)

Date 6th May 2025

Statement 2

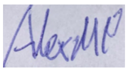
This thesis is the result of my own investigations, except where otherwise stated. Other sources are acknowledged by citations giving explicit references. A bibliography is appended.

Signed  Alexander Mak-Pearce (2119504)

Date 6th May 2025

Statement 3

The University's ethical procedures have been followed and, where appropriate, ethical approval has been granted.

Signed  Alexander Mak-Pearce (2119504)

Date 6th May 2025

Abstract

We propose a tailored Wasserstein GAN with gradient penalty for financial time-series synthesis and forecasting. By curating a multimodal dataset including OHLCV data for Amazon and peers, macro proxies, and engineered features like multi-scale indicators and Fourier-smoothed volatility we gain an intricate full picture of the stock market. Our model, comprised of a 1D-CNN critic and LSTM generator, uses adversarial training to learn realistic price sequences across regimes.

Using Bayesian optimisation for hyper-parameter tuning and a walk-forward validation scheme, we were able to keep our model calibrated to recent data, allowing capture of current trends. Our results show WGAN-GP stands up well to benchmark LSTM and ARIMA forecasters, even outperforming them in some metrics, showing particular promise in predicting turbulent market trends. This framework not only shows a capability in alleviating data scarcity by generating diverse scenarios but also delivers an adaptive, interpretable tool for trend analysis and risk management.

Contents

List of Figures

List of Tables

1 Introduction

Financial time-series forecasting in the modern day is critical to many applications, from high-frequency trading and portfolio allocation to systemic risk monitoring [73]. However, achieving proficient, robust forecasting is an evolving challenge within the reality of financial markets, which are described by constant change and uncertainty.

With the surge in artificial intelligence research and the ever growing presence of AI tools in our lives, we will be exploring how this trend can be extended in a financial context. Generative Adversarial Networks (GANs) have revolutionised fields from image synthesis to text generation by learning complex data distributions through adversarial training. This work proposes a Wasserstein GAN with Gradient Penalty (WGAN-GP) and a walk-forward calibration scheme to generate realistic stock sequences of current trends. Addressing historical and modern challenges in analysing financial time-series.

1.1 Motivation

Algorithmic trading now dominates global equity markets, executing over 78% of all U.S. equity trading volume as of 2012 [41, 27], with systems capable of processing terabyte-scale datasets in milliseconds. As this method of trading becomes exponentially larger it is more important to understand what drives these algorithms. For investors, they want to know which indicators predict trends and push trade. For researchers to innovate forecasting further they want a deeper heuristic understanding of market algorithms.

A further motivation supplied by forecasting innovations is data scarcity. This issue poses real challenges to training in machine learning models of many contexts, with finance being no expectation [46, 57]. Financial time-series are often short compared to other domains like computer vision or natural language processing, as companies can quickly emerge and dissolve. Furthermore, extreme market events (crashes and bubbles) occur infrequently yet significantly impact forecasting models. If we are working with a dataset that is too short to include a sufficient number of these events, it becomes difficult for models to learn how to predict or respond effectively [57, 23]. Furthermore, financial datasets are highly noisy due to unpredictable factors like investor sentiment, geopolitical events, and speculative trading. This noise can obscure meaningful patterns within the data [74]. If we want to build models that accurately forecast the market, given any condition, we need to find a method to robustly generate/synthesise realistic multi-regime stock market data. Allowing models to be fed with diverse financial time-series data is key to future model innovation.

Current methods of time-series forecasting have demonstrated several limitations in complex, real-world environments. Existing architectures remain brittle when confronted with cross-sector volatility. Approaches like ARIMA and standalone LSTMs work well under stable market conditions, yet struggle when confronted with turbulent events, as evidenced by their poor performance during the COVID-19 market crash [89]. This pushes us to look to new non-linear avenues of forecasting. Advances in the field of GANs have provided this new perspective on how complex data distributions can be learned and generated reliably. We would like to explore this framework and its application to time-series forecasting, as we believe it to have strong transferable skills, where the challenging environment of the stock market would provide a good test.

1.2 Aims and Objectives

We aim to build a forecasting model that captures both the natural and social phenomena present in the stock market, natural in terms of statistically quantifiable patterns and social as the market is highly influenced by human behaviour, sentiment, and external shocks. All of these factors being key to a robust accurate model [2]. Using this deep market knowledge we aim to provide a solution aiding in resolving data scarcity issues, through using realistic time-series data generation.

This study establishes three core innovations to achieve this objective:

1. **Tailored WGAN-GP model:** That enhances stock price prediction accuracy by generating synthetic data that captures complex trends across varying market conditions. Showing strong performance in capturing complex dependencies and generating realistic highly non-linear data, when applied to other tasks, such as images. We see these strengths as ideal for resolving issues laid out in the motivation.

2. **Multimodal dataset construction:** By extracting features from market data, we aim to incorporate meaningful forecasting patterns to build a novel view of the financial environment that surrounds a given stock, aiding in model training. Fundamental metrics such as P/E ratios and growth measures augmented with Fourier-smoothed volatility indicators as suggested in [75]. Competitor dynamics, to model non-linear relationships between external factors and stock prices. Macroeconomic proxies, to provide a wider scale representation of market dynamics.

3. **Dynamic Adaptation:** Implement cross validation to ensure that the model remains adaptive to evolving market conditions. This approach hopes to reduce error fluctuations during regime shifts and enhance generalisation to unseen data.

By building a deep learning forecasting model we hope to enable investors to use the model as an interpretative tool to help understand trends in the market. Answering questions such as ”What makes a good trend indicator?” and ”What features give us insight about trends of various market conditions (bull/bear/crisis)?”. Using feature importance analysis we hope to add interpretability, aiding both researchers and investors in decision making when it comes to deconstructing market trends to aid in future model building and portfolio analysis.

1.3 Overview

In this paper, we will outline traditional statistical techniques, machine learning methods, and Generative Adversarial Networks in a literature review. To provide context behind the field of financial forecasting and give reason into the methods use to build our WGAN-GP model.

We provide details on our data collection methodology, looking into the sources and types of data used for training our model. With an understanding of how a GAN learns and using that within our use context, we were able to curate a diverse and insightful dataset.

Our method breaks down the steps from, data preprocessing, feature engineering process and the implementation of our WGAN-GP model. It also covers training details, such as the implementation of cross validation, and evaluation techniques, breaking down the hyper-parameter tuning.

We present the model's forecasting results, evaluating real-world applicability by benchmarking performance against traditional models and assessing robustness under varying market conditions.

To conclude we analyse the results and summarise the findings against our aims, proving the success and potential improvements of each area. Future work is presented, such as incorporating broader data sources and enhancing the adaptability of the model.

2 Literature Review

Stock market forecasting has been attempted for as long as the market has existed, as such, methods of doing so have evolved significantly over the decades. From conventional statistical approaches, before advancing to sophisticated machine learning and deep learning methodologies. This literature review section highlights the challenges in financial time-series forecasting, then provides a comprehensive examination of approaches used to tackle forecasting. By critically assessing the strengths and limitations of each method, applied to non-stationary and noisy data inherent in financial environments, we highlight how recent innovations are leading to more robust and adaptive forecasting. This gives light to models that not only improve predictive accuracy but also enhance interpretability and decision-making support.

2.1 Overview of Challenges in Financial Time-Series Forecasting

To understand the domain of financial time-series forecasting, we must talk about its challenges. Here we recognise three core challenges. First, non-stationarity and regime shifts; requiring models to adapt as statistical properties jump unpredictably. Second, data scarcity and noise; meaning models have limited examples to learn true underlying patterns, and within these examples noise makes it extremely hard to distinguish signal from trends. Third, interpretability and feature selection; it's paramount as stakeholders need transparent drivers for predictions, yet modern models are becoming black-boxes through heightened complexity.

Together, these issues frame the fundamental tension in financial forecasting: how to build adaptive, robust models that can navigate both everyday fluctuations and the rare, extreme tail events that define market risk. The following subsections unpack each challenge in turn, reviewing its origins, impacts on forecasting techniques, and implications for our WGAN-GP framework.

2.1.1 Non-Stationarity and Non-linear Regime Shifts

By the very nature of financial markets they exhibit non-stationary data. This means through the passing of time, properties of the data change, such as mean, variance and autocorrelation structure, to name a few. The non-stationary trait is exemplified in periods of shifting regimes, by this we mean large swings in 'data properties' occur as a result of a new trend. Regime shifts in financial markets are influenced by a variety of non-linear factors, including macroeconomic shocks, policy changes, and variations in investor sentiment, as documented by Ang and Timmermann [3]. Shifts driven by exogenous factors can occur abruptly and unpredictably, notable examples show sudden and drastic shifts in investor sentiment following an economic shock, such as the 2008 financial crash or the COVID-19 pandemic where there was a large market sell-off.

Another quirk of non-stationarity in financial data is the fat-tailed nature of it's distribution. By this we mean extreme deviations (heavy-tails) occur far more often than might be expected in a Gaussian distribution. A paper by Rama Cont [22] finds the unconditional return distributions across equities, commodities, and foreign exchange markets consistently show excess heavy-tailed behaviour. Invalidating Gaussian-based risk prediction, which underpins many statistical forecasting models.

A constantly evolving non-stationary dataset provides challenges for forecasting, as models need to adapt to a constantly changing landscape. Hamilton's foundational work [39] demonstrates how given a regime change, past data becomes a less reliable guide, leading models to become biased or entirely inaccurate forecasters. These issues of constantly changing data properties and fat tails violate assumptions which underpin most statistical methods, leading to further poor forecasting. Modern models don't have it any easier, these factors massively slow the convergence of estimators and handling this type of data increases computational and methodological complexity [82].

2.1.2 Data Scarcity and Noise

Financial datasets are typically short, especially when compared to domains like image or language processing where data can span millions of samples; even major stock indices rarely offer more than a few decades of high-quality, high-frequency data [26]. Being short means these datasets often consist of frequent minor fluctuations and leave rare yet impactful events, such as market crashes, under-represented [26]. Leaving important market characteristics absent in datasets means it is impossible for models to learn and forecast the full range of the market. Moreover, financial markets prescribe strict data governance standards that provide privacy and security, however, further limiting data availability [79].

Noise from transient market micro-events obscures underlying signals degrading forecasting accuracy. This noise inflates volatility estimates, induces spurious autocorrelation, and biases parameter estimation [20].

Analysing these studies shows a challenging environment for forecasting. Modern deep learning models become ever more data-hungry with increasing levels of non-linear analysis [38]. This becomes a larger issue as we need complete and diverse datasets to correctly model financial environments if we are to avoid overfitting to a small set of regimes.

2.1.3 Interpretability and Feature Selection

When it comes to forecasting data we make future predictions based on historical data. Choosing what data to analyse in order to build good predictions is a complex task, we want a realistic diverse dataset that represents the full financial environment, whilst also handling the aforementioned challenges with noise and non-stationarity. This paper from Liu et al. [57] describes the benefits of having an adequately large feature-set size, through using over 10,000 global stocks they found far greater robust forecasting for neural network models. However, while one may look at this and assume ”more is better” it's important to keep in mind the curse of dimensionality, as we add further features it increases complexity, possibly overwhelming models and inflating overfitting risk. A case made by Huang and Yang [43], which argues for significant critique when selecting features, shows how conflating a dataset with non-critical features greatly reduces predictive performance.

Interpretability is essential for trust and regulatory compliance in finance. The issue of feature selection pairs with the issue of interpretability, we want to train models on useful data and to understand what data models find useful we must be able to interpret the market. With deep learning forecasting it conceals internal reasoning, impeding direct comprehension of how inputs map to forecasts, as shown by Giudici et al. [31]. Recent advances in explainable AI (XAI), such as SHAP values [59] and attention mechanisms, are beginning to address these concerns but often at the cost of increased computational complexity.

2.2 Financial Time-Series Forecasting History

Time-series forecasting has its origins in mid-20th-century smoothing methods such as Brown's exponential smoothing and Holt-Winters method, these ideas were then unified by Box and Jenkins into the ARIMA framework. The 1980s introduced volatility modelling with ARCH [28] and GARCH, capturing non-stationary data that classical models could not address. In the early 2000s, machine-learning techniques offered flexible, non-parametric alternatives that often outperformed the statistical methods on moderate datasets [76]. The 2010s then saw deep learning architectures capture further long-range complex dependencies, albeit at the cost of substantial data and hyper-parameter tuning [61]. Most recently, generative adversarial networks have enabled realistic synthetic data generation and probabilistic forecasting, marking the cutting edge in scenario creation and tail-risk modelling [83]. This next section will delve into various models of each period and break down their uses, with reference to examples.

2.2.1 Statistical techniques

Early efforts typically involved decomposing time series with an aim to separate systematic patterns from noise to find trends in data. A major early influence is the Holt-Winters method [19] which extends simple exponential smoothing by adding trend and seasonal components to produce forward-looking predictions. It is intrinsically interpretable as its trend and seasonal factors each have clear economic meaning, investors can trace the contribution of past observations via the smoothing recursion. However, due to its nature of assuming fixed smoothing parameters it struggles to adjust to shifting regimes without manually updating. On noisy series the exponential weighting dampens the impact of noise, while this approach is effective during stable trading, it leads to difficulty distinguishing between volatile regimes and outliers. Another early approach is the Census II X-11 method [87] which is less of a forecaster, instead being used as a tool designed to reveal underlying economic trends free of seasonal distortions, using a hierarchy of moving-average filters chosen according to data characteristics.

Building on these ideas of exponential smoothing and iterative seasonal-adjustment filters came the next innovation of the traditional statistical methods, one that is still favoured to this day. The Autoregressive Integrated Moving Average (ARIMA) [15], combines autoregressive, differencing, and moving average operations to statistically capture and forecast underlying trends. One of the improvements ARIMA brings is its systematic parameter estimation, where Holt-Winters requires manual tuning of constants ARIMA uses numerical optimisation to jointly estimate AR, I, and, MA parameters. This differencing provides great improvement in adjusting to trends, a step to combating non-stationary issues. A further innovation over Holt-Winters comes in the moving average term, which models the error process as a finite-duration linear filter of past shocks. This method of building up a picture of the noise from past behaviour, as opposed to constant pure white noise, gives a far better distinction between noise and signal in volatile circumstances. Due to its computational efficiency and interpretability ARIMA, and its variants demonstrate strong utility when it comes to capturing short to medium-term price movements. Combining this with the inherent interpretability, ARIMA is still favoured to this day as a forecasting tool. Providing a strong benchmark to test modern forecasting approaches against.

Despite the improvements over early approaches, ARIMA lacks the sophistication to fully capture complex market dynamics [53]. By depending on heuristic hyper-parameter tuning and simple MA noise components, it can misrepresent structured errors in real world data. Furthermore, The core components of ARIMA are linear, capturing only linear dependencies between lags and residuals, therefore neglecting the non-linear dynamics that are highly present in the stock market. ARIMA is also a static model, meaning it lacks adaptability, as noted by Hyndman & Khandakar [44], making updating the model to current regimes a challenging resource intensive task. This missing complexity begs the question "Can machine learning fill this gap?".

2.2.2 Machine learning approaches

Advancements in machine learning have enhanced our ability to capture non-linear relationships, naturally questioning if their non-linear abilities could be implemented in a financial context. One shallow machine learning approach is Support Vector Machines (SVM) [85], SVMs use kernel functions to map input data into a high-dimensional space, where a maximum margin hyperplane is constructed to create non-linear decision boundaries in the original space, balancing complexity and error tolerance. A promising application of this method in a financial context is proposed in a paper by Kim [49] which achieves 57.83% accuracy on holdout data. Yet for comprehensive forecasting this paper is limited in its approach, as it only allows for binary classification, failing to capture the magnitude of price changes. Random Forests are another popular approach as shown by Bou-Hamad and Jamali [14]. By utilising an ensemble of decision trees trained on bootstrap samples with random feature selection, it has the advantage of mitigating overfitting. If an individual tree has low bias but high variance, the ensemble architecture preserves low bias and dramatically lowers variance through averaging many de-correlated trees, therefore reducing overall expected error.

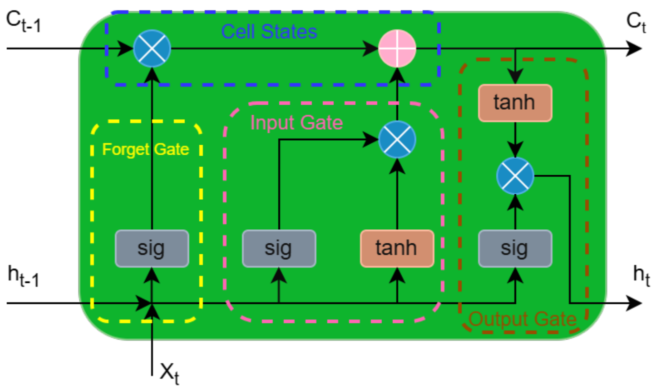
These shallow approaches operate effectively in data scarce environments [37], sometimes with only hundreds to low thousands of observations, especially if features are pre-engineered. This makes them ideal for situations where one can leverage domain expertise. However, we see these methods start to break down in higher-dimensional forecasting cases, such as predicting large global companies where their stock price is accountable to many complex features.

To handle more complex forecasting tasks we require more complex models, that can extract features from raw data and learn hierarchical representations. Deep learning models use more computing power to process high-dimensional data by building deep temporal connections and developing pattern recognition, that enables more accurate and robust stock price predictions. As demonstrated by Usmani et al. [84], by utilising a combination of attributes and ANNs to forecast stock market volatility. They concluded that multi-layer perceptron architectures outperformed alternative algorithms in their experimental settings.

Within the realm of deep learning, Recurrent Neural Networks (RNNs) have emerged as powerful tools for stock price forecasting due to their inherent ability to process sequential data and maintain the memory of previous inputs [55]. Studies show the RNN-based methodologies perform well, with Lu and Xu [58] demonstrating a modified time-series technique. Training the RNN on sliding windows, their model can adjust its forecast based on recent trading volumes which have a greater impact on current prices. A RNNs ability to learn over time can also lead to its greatest downfall, the issue of vanishing gradient. This phenomenon occurs during training when updates to the weights become so insignificant it leads to an exponential decrease in gradient magnitude [68]. That is, the network is unable to converge to a minimal loss.

This limitation provoked the development of more specialised architectures such as Long Short-Term Memory (LSTM) networks [42] and Gated Recurrent Units (GRU), proposed more recently by Cho et al [21]. These overcome the vanishing gradient problem through clever use of gating mechanisms that control information flow through the network [80].

Background: Long Short-Term Memory LSTM Networks are built from individual cells consisting of three parts, the ”Cell State”, the ”Hidden State” and the gates: “Forget”, “Input”, and “Output”.



- *sig*: Sigmoid layer
- *tanh*: Tanh layer
- $\otimes$: Pointwise multiplication
- $\oplus$: Pointwise addition
- $\mapsto$: Vector connections
- h_{t-1}: Previous hidden state
- X_t: Input
- C_{t-1}: Previous timestep
- C_t: Cell state information
- h_t: LSTM output

Figure 1: A single LSTM cell

The general idea of the LSTM is that information flows across cells through the cell state. Further information can be added or removed to this cell state via the various gates, composed of a sigmoid neural net layer and a pointwise multiplication operation. The sigmoid layer acts as a regulator, outputting a number between zero and one, which describes how much of each component should be let through. The initial gate involved is the forget gate, this as the name implies, should remove information deemed unnecessary. To provide an example in the context of the stock market, the cell state might contain information pertaining to fourth quarter closing prices of a given stock, moving through the time-series we are now looking for trends in the first quarter thus it would make sense for the forget gate to take in lower regard previous trends. The input gate is tasked with deciding what new information should be stored in the cell state. Here, like the forget gate, the sigmoid layer decides what information to update, this value is fed into a tanh layer that creates a vector of new candidate values that could be added to the state. Finally, we output a filtered version of our cell state, by use of the output gate. This uses a sigmoid layer to decide on what information is important to output, this result is pointwise multiplied with information from the cell state (passed through a tanh function) to produce the filtered output. Again, an example in reference to the stock market, the output might contain information about trends, momentum, or volatility patterns in context to the current state, that might be useful to the next cells.

To look at an example of this being applied in literature is Bao et al. WT–SAE–LSTM Framework [10]. Which decomposed the time-series via wavelet transforms (WT), learned hierarchical features using stacked autoencoders (SAE), and fed the de-noised representations into an LSTM for stock price forecasts. This method effectively reduced noise and captured complex trends across six market indices. The issue with this approach is that it adds complexity and many hyper-parameters, the multi-stage process increases the risk of overfitting due to separate WT/-SAE tuning. This might be improved by instead implementing trainable autoencoders, which have shown promise in similar tasks [48].

2.2.3 The Generative Adversarial Network framework

Generative Adversarial Networks (GANs), proposed by [33] traditionally have been used for image generation, due to their unique ability to learn complex data distributions through adversarial training. This however made them an ideal candidate for the financial sector [7], with their use representing one of the most innovative developments in stock market prediction methodology in recent years. Leveraging adversarial training to generate synthetic sequences that emulate the temporal dynamics and statistical properties of real financial time series.

The framework consists of a generator G and a discriminator D, optimised jointly through a minimax game. G synthesises plausible future price trajectories conditioned on historical data, while D distinguishes between real and generated sequences. This adversarial process encourages G to produce forecasts indistinguishable from real market behaviour. More generally this can be shown as, D and G playing a two-player minimax game.

Breaking down D's loss function:

$$\text{Loss}_D = -(\mathbb{E}_{x\sim p_{\text{data}}}[\log D(x)] + \mathbb{E}_{z\sim p_z}[\log(1 - D(G(z)))]). \tag{1}$$

This function is designed to penalise D when it makes incorrect classifications. It can be looked at in two components, loss on real data and loss on generated data. For real data the term $\mathbb{E}_{x\sim p_{\text{data}}}[\log D(x)]$ represents the expected log-likelihood of D correctly classifying real data as real. For the generated data the term $\mathbb{E}_{z\sim p_z}[\log(1 - D(G(z)))]$ represents the expected log-likelihood of correctly classifying generated data as fake.

Breaking down G's loss function:

$$\text{Loss}_G = \mathbb{E}_{z \sim p_z}[\log(1 - D(G(z)))] \tag{2}$$

This function is designed to penalise G when it fails to trick D. G wants $D(G(z))$ to be close to 1, so $1 - D(G(z))$ is close to 0 and the value is minimised.

A notable issue with this framework occurs when G optimises too much to trick D, known as mode collapse [54]. An example of this could be, if the discriminator learns that financial time-series tend to exhibit a positive trend in the period before Christmas, it would positively reward G for generating that trend. This teaches G to repeat the behaviour, provoking it to lose nuance to other factors beyond the rewarded trend, producing only a small subset of the real data distribution. Ultimately leading to a loss of generalisation and poor forecasting capabilities when market trends change.

The issue of mode collapse gave rise to the Wasserstein GAN [4] and later WGAN with Gradient Penalty (WGAN-GP) [35], which addresses weight clipping issues in vanilla WGAN such as capacity underuse and gradient explosion. An improved loss function makes use of the Wasserstein or Earth-Movers distance to measure distance the between the real and generated data distributions. This leads to more stable training as D (referred to as the critic in literature) produces a smooth gradient that does not vanish or explode, giving G meaningful signals to learn from. A successful implementation of this framework is [34], which managed to successfully estimate the state-of-charge in batteries better than both comparative CNN and Transformer models. A promising result that shows a loss that correlates closely with sample quality. Characteristics key to producing realistic synthetic stock market data.

2.3 Comparative Analysis and Synthesis

In summary throughout history many different techniques and models have been used, each with their individual strengths and weaknesses. Assessing the environment in which one is attempting to forecast is perhaps the most important step, as this will allow the person in question to select the best forecasting candidate to achieve a specific goal. Below is a table summarising our findings and analysis of various techniques applied to the financial world.

Model	Forecasting Challenges	Application Scenarios	Deficiencies
Holt–Winters	Handles smooth trend and seasonality via recursive smoothing but assumes fixed smoothing parameters	Seasonal demand forecasting, such as retail sales, inventory	Conflates noise with signal in volatile regimes; limited to univariate series; multiplicative seasonality can misbehave on low-level data
ARIMA	Explicitly models autocorrelation and noise via AR and MA terms, requires stationarity or manual differencing	Short-term economic forecasting	The linear framework cannot capture complex non-linear trends; sensitive to outliers
SVM	Robust generalisation under noise, handles non-stationarity via kernels	Stock-index regression, VaR forecasting	Computationally expensive on large datasets; degrades with high dimensionality without careful feature selection; kernel choice and hyperparameters critically affect performance
RNN (LSTM, GRU)	Captures non-linear temporal dependencies and evolving regimes; mitigates vanishing-gradient via gating	One-step and multi-step stock-price/index forecasting	Long training times; requires large labelled datasets, becoming sensitive to sequence length and over-fitting; black-box nature complicates interpretability
GAN	Learns complex distributions and rare events through adversarial training	Probabilistic forecasting and synthetic data augmentation for risk management	Training instability; requires very large datasets; difficulty enforcing financial constraints or interpretability; suffers from mode collapse

Table 1: Critical comparison of forecasting models across key challenges, applications, and deficiencies.

3 Data Collection

To select and build good datasets a thorough understanding of financial environments and how a GAN framework processes this environment is needed. The section focuses on building the environmental understanding, and the subsequent data collection methodology that forms a critical foundation for this study's empirical approach.

Amazon Inc. (AMZN) was chosen as the principal stock for this paper due to its broad reach across multiple sectors of the economy, from e-commerce to healthcare, ect., making it an ideal benchmark equity. As one of the largest constituents of the S&P 500, Amazon exhibits high liquidity and volatility-critical attributes for training generative models such as our GAN, which requires robust price signal variability to learn meaningful distributions.

To evaluate our model's robustness across distinct market conditions, two datasets were constructed. The first set contains data from 2010-2025 which we are calling the ”full context” dataset as it fully incorporates trends from the post-global Financial Crisis recovery, the COVID-19 pandemic, and current economic uncertainty due to the new US president. The second set is a ”truncated context” which captures a period of relative macroeconomic stability 2010-2020 (pre COVID-19). This allows us to evaluate whether incorporating additional turbulent-era data improves the model's ability to generalise across regimes.

3.1 Historical financial data

Daily stock data was retrieved via Yahoo Finance's *yfinance* API [5], a widely cited source in computational finance due to its accessibility, reproducibility, and compliance with adjusted closing price standards. Certain macroeconomic indicators were collected by use of *FRED* [30] which is an API from the Federal Reserve Bank of St.Louis that consists of hundreds of thousands of economic time-series.

Open-High-Low-Close-Volume (OHLCV) Amazon data is dual-purposed: it directly informs the generator's synthesis of Close price sequences while also serving as the basis for technical indicators. These derived technical features are useful as trend predictors and volatility smoothing, allowing WGAN-GP to gain perspective of traditional predictive and smoothing techniques.

Figure 2: Amazon close price history over our full dataset (2010-2025)

Competitor data was collected to help contextualise and enrich the picture in which Amazon operates. Daily OHLCV data was collected for five peers: Microsoft (MSFT), Alphabet (GOOGL), Apple (AAPL), Meta (META), and Walmart (WMT). This data helps disentangle Amazon-specific effects from broader market trends, and mitigates overfitting by anchoring predictions in cross-sectional variation, as seen in multiple studies, Safonov [77], Bandara et al. [9].

Macroeconomic Proxy Data was incorporated to provide conditional variables for the WGAN-GP model. As both networks learn to generate and classify realistic stock market data, it uses these conditional variables as auxiliary discriminators. Providing external information to ensure the predictions they are making align with broader economic contexts, enabling WGAN-GP to penalize syntheses that violate well-established market regularities.

- U.S. Dollar Index (DX-Y.NYB): Proxy for global trade dynamics and Amazon's international revenue exposure.
- Crude Oil Futures (CL=F): Impacts logistics costs and consumer discretionary spending.
- CBOE Volatility Index (VIX): Measures market-wide risk sentiment.
- Gold Futures (GC=F): Safe-haven asset reflecting macroeconomic uncertainty.
- U.S. 10-Year Treasury Yield: Risk-free rate benchmark for discounting cash flows.

These variables enable the generator to learn regime-specific price distributions such as, high-VIX vs low-VIX environments. Resulting in enhanced realism and robustness in adversarial training [62]. Furthermore, it helps to incorporate more social dynamics into the model as some of these features (i.e. U.S. Dollar Index, Gold Futures) give insight into social trends such as spending.

4 Methodology

This methodology section presents our full technical framework for implementing the WGAN-GP model for time-series forecasting. The feature engineering subsection explores technical indicators and Fourier transforms to supplement the dataset, aiding in extracting meaningful patterns from raw price data. In the data preprocessing subsection, we conduct feature analysis to understand relationships between variables. Discussing redundant features and potential multicollinearity issues affecting model training. Data split and normalisation establishes the foundational datasets for reliable model training, validation and evaluation. The GAN implementation subsection details both generator and discriminator architectures, adapted for financial time-series forecasting. The training subsection outlines the WGAN-GP training loop, a sophisticated approach that addresses mode collapse and convergence issues common in traditional GANs. Finally, the evaluation discusses hyper-parameter tuning and delves into the implementation of cross-validation techniques to ensure model robustness.

4.1 Feature engineering

Despite having a strong base, the dataset has some limitations. The subsection below highlights these limitations and the methods we employed to mitigate them.

High Noise and Volatility: Raw price data, particularly for high-frequency financial markets is inherently noisy and volatile. The drastic fluctuations in daily OHLCV values can obscure underlying patterns, making it difficult for the GAN's generator to learn the true data distribution, and for the discriminator to provide meaningful gradients. This high level of noise may lead to training instabilities and phenomena such as mode collapse [52].

Lack of Latent Feature Representation: The raw dataset primarily consists of price values and a handful of macroeconomic factors that do not explicitly capture market dynamics such as momentum, trend strength, or cyclic behaviour. Without statistical analysis of features, the GAN could struggle to differentiate between random price movements and systematic trends, which are essential for accurate forecasting [56].

Scale Differences: Data coming from different sources like Amazon's daily price or macroeconomic indicators, often exhibit different scales and units. This heterogeneity challenges the GAN's capacity to learn a single coherent representation of the market state, as the generator must simultaneously model distributions of disparate amplitude and dynamic range.

4.1.1 Technical indicators

Technical indicators refine our raw OHLCV data into interpretable trends. This provides key insights for our model as it captures structural patterns in the data, both handling outliers and providing a more generalised dataset allowing the GAN to converge faster based on macro trends.

Technical indicators were calculated using the *pandas_ta* library [47] that leverages the Pandas package and are as follows:

- Exponential Moving Averages: Short-term and medium-term, EMAs act as smoothed representations of price trends.
- Moving Average Convergence Divergence: Measures acceleration/deceleration of trends.
- Relative Strength Index: Quantifies overbought (>70) or oversold (<30) regimes.
- Bollinger Bands: Encodes volatility regimes. Narrowing bands (low σ) often precede breakout events, while price touching the upper/lower band signals mean price reversal.
- Stochastic Oscillator: Identifies cyclical turning points by comparing closing prices to recent high-low ranges.

Each indicator has a multi-scale representation, calculating each in 7/14/21-day spans, this means a triad spans intra-month, monthly, and quarterly horizons, allowing the GAN to disentangle high-frequency noise from longer structural trends [64, 6, 25]. Regime Conditioning indicators like Relative Strength Index and Bollinger Bands act as conditional variables, enabling the generator to change outputs based on volatility/momentum states for example, high Relative Strength Index $\rightarrow$ increased price reversal likelihood.

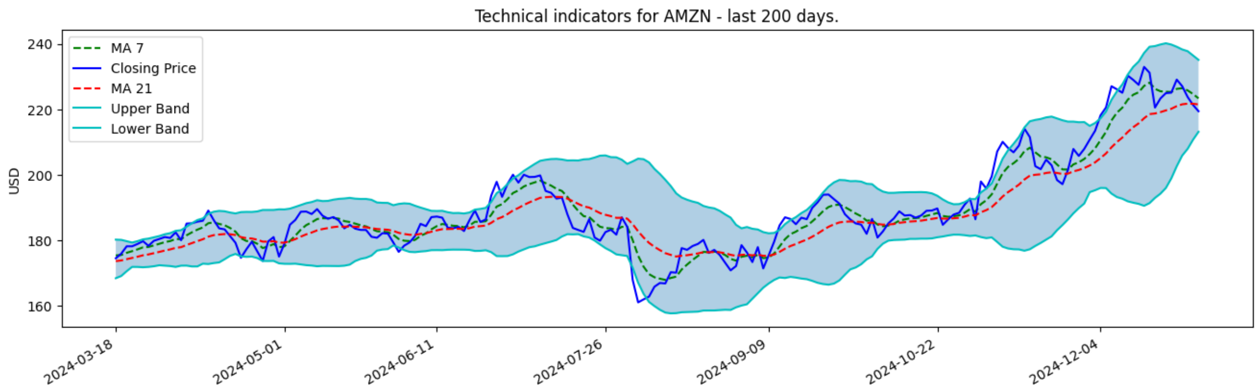


Figure 3: Graphical representation of Technical Indicators

4.1.2 Fourier transforms

Fourier transforms decompose a function into a set of many different sine waves. When combined, these sine waves approximate the original function. Its mathematical function is as follows:

$$G(f) = \int_{-\infty}^{\infty} g(t)e^{-i2\pi ft}\,dt \tag{3}$$

By approximating the price series with just a few key Fourier components we effectively filter out high-frequency noise, further clarifying short-term cycles and long-term trends that are not obviously apparent. This de-noised signal presents a smoother picture of the underlying market dynamics and reduces the risk of overfitting to random fluctuations [60]. The transformation often leads to a more stationary representation of the data, which benefits the training of our GAN by providing more stability. Stationary data helps the model learn the latent structure of the market without being shocked by time-dependent shifts in variance or mean.

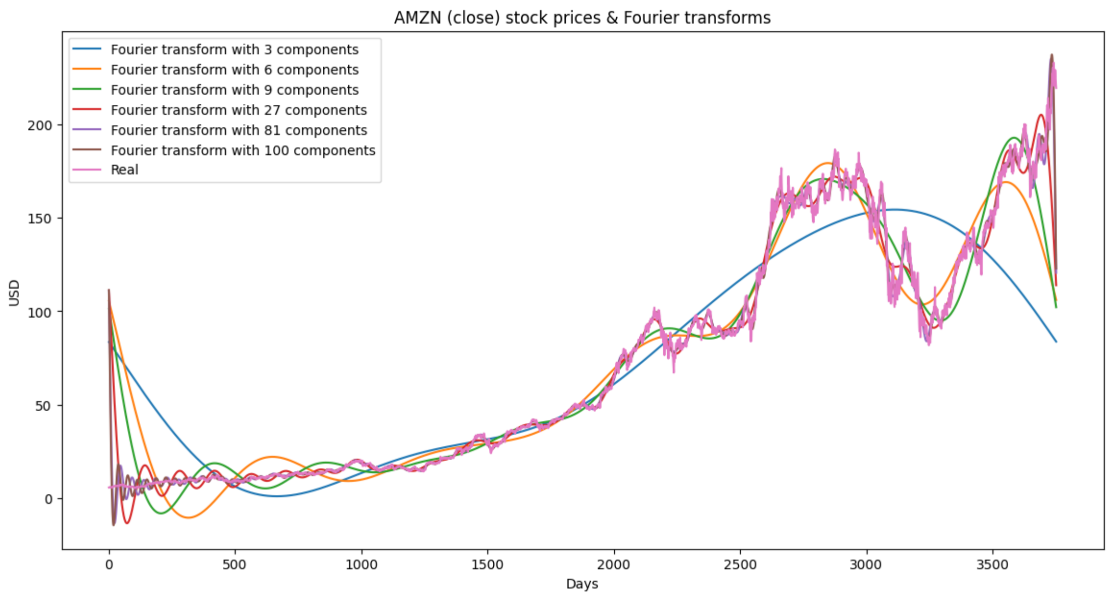


Figure 4: Various Fourier Transforms

4.2 Data Preprocessing

Our completed dataset is comprised of 42 features and 3774 rows. To handle missing or null data we used linear interpolation, to estimate missing gap values. If data was not present at the start or end of the set, a simple back-fill and front-fill were used to populate the row with the last known data.

4.2.1 Feature analysis

Gaining insights into pairwise relationships between features is useful to identify redundant features so that we can help guide our feature engineering and data preprocessing.

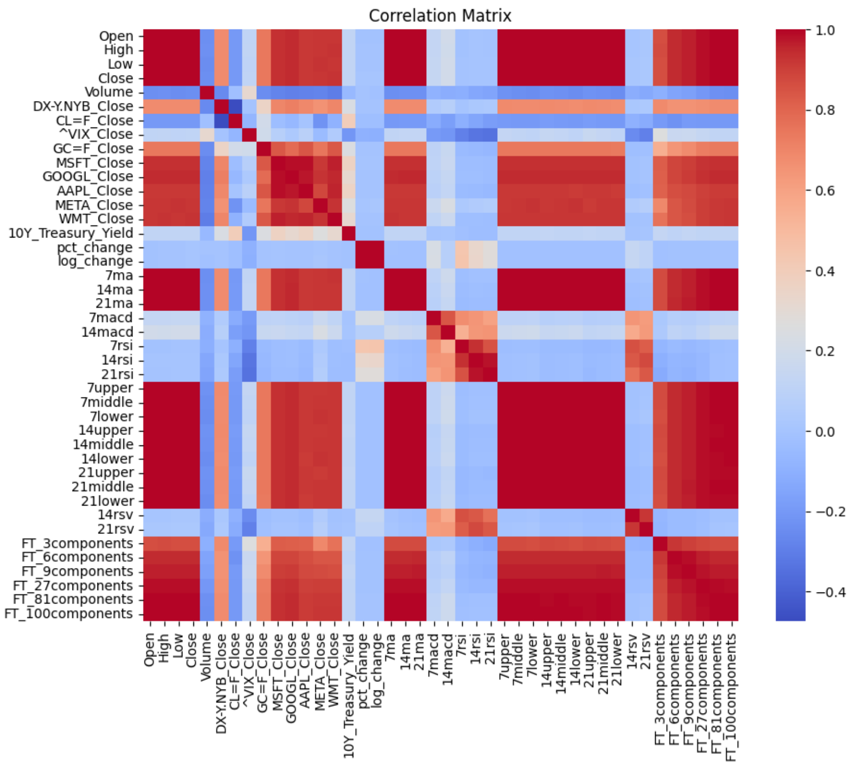


Figure 5: Correlation Matrix of features

We observe that many of the features show a high correlation with one another. This suggests some features are conveying similar information about the underlying data pattern, which can lead to several challenges. Multicollinearity can inflate the variances of the coefficient estimates, making them unstable and difficult to interpret. When features are highly correlated, it becomes challenging to disentangle their individual contributions.

One could argue many of the issues observed in this dataset could be remedied by using principle component analysis (PCA). PCA can be employed to reduce dimensionality and obtain important variables with the largest variation in datasets while minimising information loss. However, In the context of our WGAN-GP, PCA can hinder forecasting capability. PCA is inherently a linear method and may not capture the complex non-linear relationships present in our data, discarding subtleties that are crucial for high-fidelity forecasting [72]. Financial markets are non-stationary, and the underlying structure shifts over time. PCA components derived from historical data might not be stable when new, evolving market conditions come into play. A better method might be one such as non-linear Autoencoder Network, proposed in this paper[72], due to its effectiveness in accounting for non-linear variance. However, this fell out of the scope of our research.

Understanding the impact of input features is crucial to validate the dataset and model's credibility. It allows us to compare the model's behaviour against financial theory, assessing whether the features have similar economic relevance in the real and modelled world. Furthermore, understanding this provides interpretability which is key for informing stakeholders' decisions if this model is to be used in real-world applications.

4.2.2 Data split and Normalisation

We perform a few critical actions to adjust the shape and rearrangement our dataset, to optimise the training of our GAN and provide unbiased testing.

First, we split the whole dataset into training (70%), testing (15%) and validation (15%) sets. This is a fundamental step that isolates our data to be used in each distinct stage, so we can guarantee that our model is evaluated without bias, avoid data leakage, and ensure that the learning process generalises well to unseen examples. The training set is dedicated to the iterative process where both the generator and discriminator parameters are updated to generate and discern financial data, thus it is the largest segment. The validation set is used for fine-tuning the generator with walk forward prediction, and making iterative decisions regarding the hyper-parameters. The testing set acts as completely unseen data from which we can get an unbiased evaluation of the trained model's performance.

Next scaling/normalising and reshaping is done as Neural network models, including GANs, are sensitive to the magnitude and distribution of their inputs [50]. In our case, simple min-max scaling was applied between the range of 0 to 1. This helps align data values with the activation functions that are present in the LSTM framework, which inherently expects inputs to fall within a limited range. Furthermore, it minimises the disparity among feature magnitudes, the result of this being faster convergence by mitigating issues related to exploding or vanishing gradients [68]. In addition, reshaping the data to match the input dimensions expected by the network is performed.

Finally arranging datasets into sliding windows is key for processing time-series data, as shown by Norwawi [65]. Transforming the continuous sequence into overlapping segments, each representing a fixed time interval takes into account the relationship between adjacent data points. Furthermore, it provides the generator with the sequential context necessary for predicting future time steps, essentially setting the number of days the model forecasts ahead.

4.3 GAN Implementation

The GAN model has been designed to forecast the stock market through a hybrid design, combining recurrent neural networks and convolutional neural networks (CNN). The generator G's philosophy is to capture temporal dependencies in the time-series using a stack of Long Short-Term Memory (LSTM) layers. While, the discriminator D uses one-dimensional CNN layers that extract identities of small trends, which in turn form larger patterns, ultimately training itself to differentiate between real and synthetic sequences.

To implement the model we used the PyTorch API [69] for its efficient GPU utilisation, which is highly favourable when building and training large networks such as these. In addition to this, it is highly supported in the research community, making it easy to support novel frameworks such as the one we implemented.

4.3.1 Generator

Three stacked LSTM layers were utilised with progressively reduced hidden states (1024, 512 and 256). The initial high number of hidden states was chosen as our features, representing Amazon's stock environment, have a great deal of interdependence. To provide an example: finding trends in gold futures can reveal clues about the state of the wider economy (i.e, during times of economic instability banks stockpile gold as a hedge against inflation). This of course will have an impact on Amazon's stock price, therefore must be accounted for; to capture this level of detail a deep model with a high number of hidden states is needed. This ideology promotes G to capture long-term dependencies with the initial layer. The subsequent layers serve to refine and condense the long-term dependencies into a more compact representation, effectively filtering out noise and less relevant details. This is key in volatile regimes where distinguishing noise from signal is a tricky task, an area often misrepresented by traditional models. Each LSTM layer is initialised with zero-hidden states to avoid bias toward initial conditions.

The LSTM output is passed through three fully connected layers with sizes 128, 64, and 1 respectively. This allows us to project the high dimensional temporal features into a scalar forecast, enabling us to generate a close price, to feed to our D.

Dropout Regularisation is applied to help prevent over-fitting by randomly zeroing a subset of activations during training. Forcing the model to learn redundant representations allows for greater generalisation of unseen data [78]. In addition to this, dropout encourages the network to develop more robust and distributed feature representations by removing a reliance on specific neurons.

4.3.2 Discriminator

The discriminator implements three One-Dimensional Convolution Layers. Traditionally CNNs are involved with tasks related to images, as they exhibit strong classifying and feature extraction capabilities. One might for instance want to detect images of cars, for this, an initial CNN layer would detect edges, a second layer would piece together its shape and a third layer might recognise finer details like wheels. We use a 1d-CNN for its effectiveness on spatial data, which implies that data points in proximity to one another to have a stronger relationship than those that are farther apart. This concept applies to time-series data as well, especially in the role of a discriminator where we are attempting to discern real from synthetic data. In our example, each data point corresponds to each successive day, it's reasonable to expect days that are closer together have a greater connection to each other. Our model has three layers of size 32, 64 and 128, so in a time-series forecasting setting, the initial layer starts by detecting small trends, then progressively get passed through the layers to build a picture of larger patterns in the Amazon close price. This ability to build knowledge of what makes up a close price is ideal for discriminating real from fake data [29].

Batch normalisation is applied after the second convolution layer, this stabilises training by reducing internal covariate shift [45]. Some literature suggests layer normalisation as the preferred method of normalisation, however, we found it to reduce forecasting capabilities. Most likely due to layer normalisation individual sample approach that could be insensitivity to batch-level data variations, however more research needs to be done on our model.

Leaky ReLU is used as an Activation function with $\alpha = 0.01$. This stands to reduce gradient sparsity in early layers. In the later layers, RelU is used to encourage non-linear feature recombination.

The feature maps are then flattened and passed to three fully connected layers that transform the extracted features into a scalar output, representing the sequence's authenticity probability.

4.4 Training

Training starts by initialising the networks and spitting the feature-set, the first set being our features or input data and the second being close prices or target data, both these sets are then loaded into batches.

The Adam optimiser was selected for both G and D. This optimiser combines ideas from both RMSProp and momentum to produce adaptive learning rates for each parameter [51]. It is ideal for our GAN as it effectively handles the sparse gradients and noisy updates that can be present in the dynamic environment of getting two networks to converge.

4.4.1 WGAN-GP Training Loop

G and D are trained over a fixed number of epochs (10,000). The training process involves an iterative optimisation procedure where G and D are updated adversarially, based on a min-max game with their loss functions.

Training D: The process involves using a Wasserstein GAN with Gradient Penalty (WGAN-GP) [4] framework as we found it to greatly improve stability and convergence. This framework proposes using the Wasserstein or Earth-Mover distance to produce a value (loss) function for the discriminator (otherwise called a critic). The informal definition of this distance is the lowest cost of mass transportation required to convert the distribution q into the distribution p, where the cost is equal to mass times the transport distance. This distance is computed via the Kantorovich-Rubinstein duality, which only holds when D is 1-Lipschitz. This must be enforced if we are to ensure that small changes in the input lead to proportionally small changes in the output.

To uphold the 1-Lipschitz constraint a Gradient Penalty is applied, we propose the following function to do so. First, we interpolate samples by linearly combining real and synthetic data points (from G) using a random weight (epsilon). This results in continuous data points between real and generated distributions. We then calculate the gradient of D's output with respect to the interpolated samples. The gradient represents the rate of change in D's output as the input varies. The important final step enforcing the Lipschitz constraint lies in penalising D if the ℓ_2-norm of the gradient deviates from 1, by enforcing this throughout training we will encourage D to have a gradient that lies within 1-Lipschitz. This gradient penalty function can be described in the following equation:

$$\mathrm{GP} = \lambda \ \mathbb{E}_{\hat{x}\sim\mathbb{P}_{\hat{x}}}\Big(\|\nabla_{\hat{x}} D(\hat{x})\|_2 - 1\Big)^2. \tag{4}$$

where:

- λ: The gradient penalty coefficient, which scales the penalty term.
- $\mathbb{E}_{\hat{x}\sim\mathbb{P}_{\hat{x}}}[\cdot]$: Expectation over the interpolated distribution $\mathbb{P}_{\hat{x}}$.
- $\hat{x} = \epsilon\, x + (1-\epsilon)\,\tilde{x}$: Interpolated sample between a real data point $x \sim \mathbb{P}_r$ and a generated sample $\tilde{x} \sim \mathbb{P}_g$.
- $D(\hat{x})$: The critic (discriminator) output evaluated at the interpolated sample $\hat{x}$.
- $\|\nabla_{\hat{x}} D(\hat{x})\|_2$: The ℓ_2-norm of the critic's gradient vector, measuring its magnitude.

This approach was shown to provide a more stable and effective training process, preventing D becoming too confident or making abrupt judgments. Which allows G to produce a wider range of more realistic samples. In addition, it helped reduce mode collapse. This is an issue where G learns to mimic the exact traits D uses to discern real from synthetic data, collapsing into this state G will only produce a limited variety of samples.

Training G: This happens at the final stage of the loop and is based on the output/gradient of D. This ensures G gains a useful understanding of how good its synthetic data samples were, pushing generated samples toward regions the critic deems more "real". Updating its weights, allows it to generate higher quality samples that can trick and maximise D's score next epoch.

4.5 Evaluation

One of our main challenges implementing this model was dialling in the hyper-parameters. It is a particularly poignant issue for WGAN-GP with its intricate loss functions, as stability and convergence of the networks are acutely sensitive to its hyper-parameters [71], meaning small changes can dictate success or failure. The next section details the strategies we used to tackle this and improve forecasting capabilities to unseen data.

In summary we initially trained on hardcoded hyper-parameters, we reviewed the model's performance on each dataset split to display its strengths and weaknesses, given varying market conditions. This allowed us to have a baseline so we could understand how various hyper-parameter tunings were affecting the model. Once we achieved the desired forecasting capability of our tuned model we then optimised it to current trends through walk-forward validation techniques.

4.5.1 Hyper-Parameter Tuning

Hyper-parameter optimisation is necessary to achieve network convergence and push the best performance out of our model. We used Ray-Tune with Optuna's Bayesian search [1] to automate this process, making it faster and more efficient. Ray-tune is a tool for distributed computing, meaning it runs many tests in parallel, making use of multiple systems. Optuna uses a Bayesian search algorithm, which provides efficient guessing when given a parameter value range. It starts by trying a few values, and then uses past results to predict which values might work better next, focusing on the most promising areas. This saves time compared to examining every potential combination by striking a balance between investigating new choices and taking advantage of known, beneficial ones.

Each run evaluated how well the model predicted on a testing set, using the root mean squared error as the metric. With deep learning models there are a large number of possible hyper-parameters to optimise, this number is doubled when building a GAN as you are training two networks.

There an interesting dynamic when increasing the number of hyper-parameters to tune, it may result in a stronger model as suggested by Won et al. [88]. However, this exponentially increases the complexity and runtime of the optimisation algorithm, meaning we have to reduce the number of total runs. Resulting in the range of values tested for each hyper-parameter being reduced. Through experimenting with including and excluding various hyper-parameters, (number of layers, hidden layers, dropout rate, ect.) we found the most influential to be:

- *Learning rate* for the generator and discriminator, influencing the speed of convergence.
- *Batch size* affecting training stability.
- *Lambda term* regulating the gradient penalty.
- *Weight decay* regulating the optimiser to prevent over-fitting.

The search space for these hyper-parameters was defined in ranges and samples were selected using log-uniform distribution, apart from batch size where there are provided choices. For example, the learning rate was sampled between 1e-5 and 1e-3. The final hyper-parameters that were found to optimise GAN training were: learning rate: 2.3027e-05, batch size: 128, weight decay: 0.0002 and lambda(GP): 5.3031.

4.5.2 Cross Validation

When training our model if you were to pick a single training/testing split, this chosen split will have a marked effect on its evaluation. This means the performance of the model could be greatly overestimated or underestimated due to random variations in the data. Furthermore, there is a greater risk of overfitting, resulting in poor generalisation to unseen data, if the model is tuned to perform well on just one specific hold-out set. Cross validation mitigates these risks by partitioning the dataset into windows and averaging the performance across them.

Unlike traditional datasets where samples are assumed to be independent and identically distributed, stock market data is a time series which has a natural ordering. This provides challenges if we were to implement standard cross validation methods (such as k-fold), as we must ensure the model is always trained on past data and validated on future data to preserve causality.

Walk forward optimisation is a cross validation technique applied to validate our trading strategy (data generation), by training on historical in-sample data and testing on subsequent out-of-sample data in a rolling window [67]. In our context, we will be improving the generator by fine-tuning its parameters to new unseen data. The simple algorithm is as follows.

```
For each validation_sample i:
  y_pred = generator(current_window)

  If (i % fine_tune_interval == 0):

    generator.train()

    For each epoch:
      loss = compute_loss(generator(current_window), val_y[i])
      update_model(model, loss)

  current_window = current_window[1:] + [val_x[i]]
```

Here we see G generate a prediction using the current window of data. If the iteration meets the fine tuning interval, we switch G to training mode. This was done so as to not over-fit our model to the validation set. For a given number of epochs, we compute the mean squared error loss between the prediction and the actual target values, updating G using the Adam optimiser.

This form of optimisation becomes especially important for keeping this model up-to-date. If this model were to be used for real-world deployment, it's key that it continues to adapt to changing market environments as trends are constantly moving.

Updating the model to unseen data is a delicate process as we want to give importance to new trends, yet still preserve underlying market movements. This means, as with defining and training the model, it is similarly important to do hyper-parameter tuning in respects to our walk-forward validation algorithm. Walk Forward hyper-parameters:

- *Fine tuning interval* determining how often to update the model during the walk-forward process.
- *Epochs* controlling the extent of each update.
- *Learning rate* within low bounds to ensure stable updates.

The resulting configure was: fine tune interval: 5, epochs: 4, learning rate: 9.3099e-06. This was found to be the correct balance to enable generalised forecasting, given our model applied to each dataset. We recommend performing frequent hyper-parameter tuning as the market evolves to keep the model up to date with current trends.

After we dialled in our hyper-parameters we were able to achieve a marked performance increase in all datasets, and for the unseen (testing) dataset there was a 32.53% improvement in forecasting accuracy.

5 Results and Discussion

The following section presents a comprehensive analysis of the WGAN-GP framework's performance forecasting Amazon Inc. (AMZN) stock prices. By integrating walk-forward validation, hyper-parameter optimisation, and comparative benchmarking against ARIMA and LSTM baselines, this study demonstrates significant advancements in modelling financial time series under both stable and turbulent market regimes.

The primary metric we chose to optimise/measure the model on was root mean squared error (RMSE) with further metrics being, mean average error (MAE) and mean absolute percentage error (MAPE). MAPE measures the average percentage error, useful for understanding relative accuracy, especially when the scale of the data varies. RMSE emphasises larger errors due to squaring, making it sensitive to outliers, it's also expressed in the same units as the target variable (dollars), which makes the error magnitude directly interpretable thus useful to investors comparing it to other models. MAE provides the average absolute error, offering a straightforward measure of prediction accuracy.

5.1 WGAN-GP forecasting results

The final WGAN-GP model results in the following forecasts on each dataset.

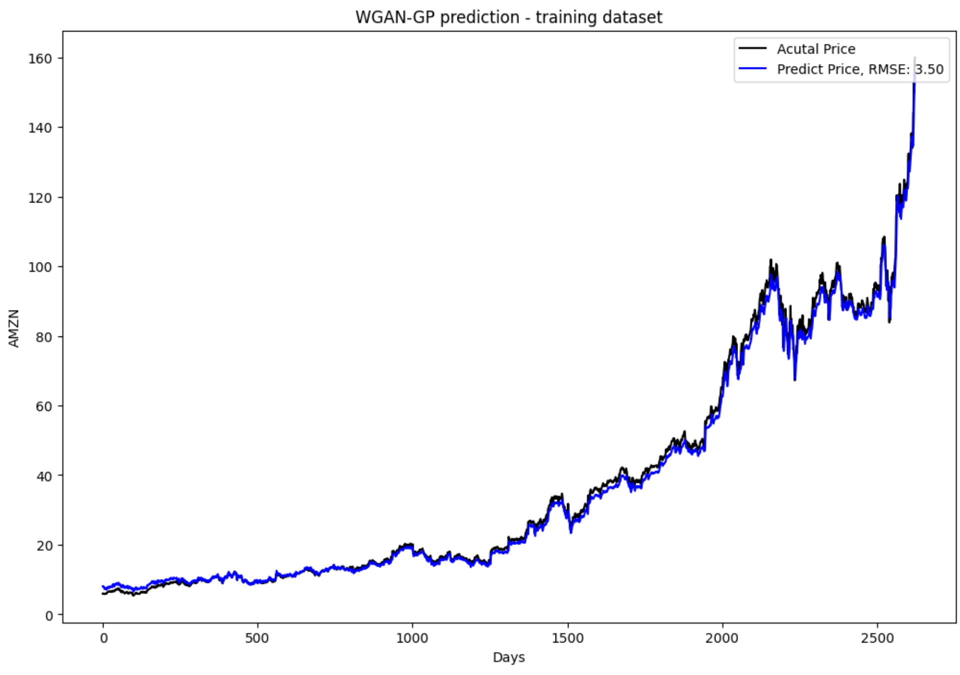


Figure 6: Training forecast

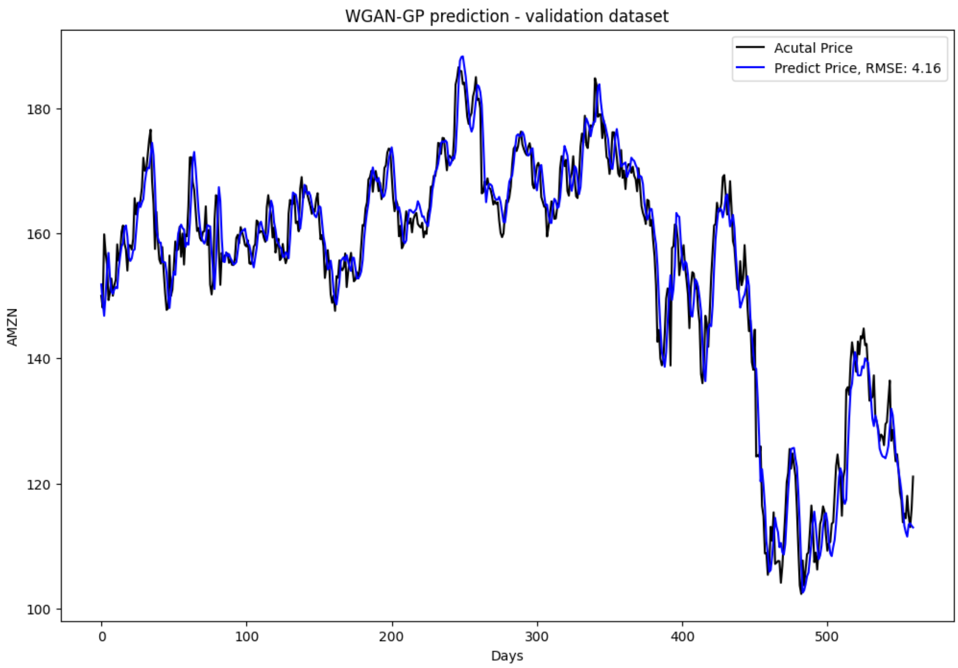


Figure 7: Validation forecast

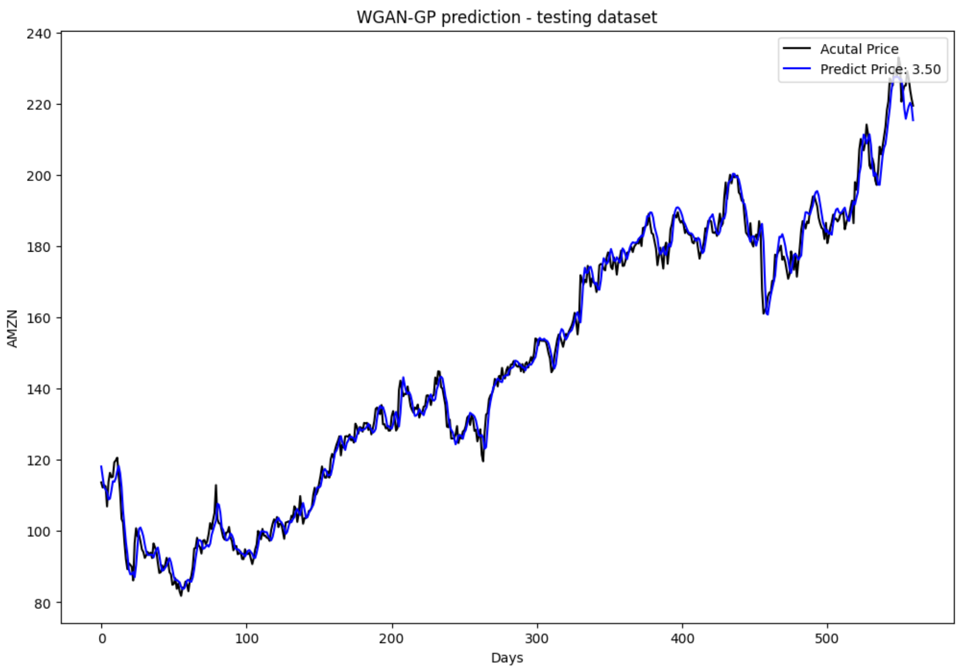


Figure 8: Testing forecast

Looking at the model trained on our "full dataset" we see strong forecasting performance across the training, validation and testing sets with RMSE values of 3.50, 4.16 and 3.50 respectively. This result suggests that the model is able to generalise well to unseen data. The parity between training and testing errors implies limited overfitting which is a notable achievement given the volatility of financial data.

The WGAN-GP framework's gradient penalty provided enhancements to the training stability. However, the relative underperformance with the validation set could indicate evidence of residual mode collapse and overfitting to the training set. Forecasting highly volatile data, present in the validation dataset (covering COVID-19, a period of economic uncertainty) is a challenging task for any forecasting model. We expect this challenge was heightened due to highly volatile data being under-represented in the training dataset, meaning the WGAN-GP was ill-exposed to the extremisms that embody this type of data.

5.2 Dataset/Feature analysis

To analyse if the dataset we have built is being used effectively we are using SHAP (SHapley Additive exPlanations). It provides a mathematically rigorous way to quantify how much each feature contributes to the model's prediction for a specific instance, either increasing or decreasing the output compared to a baseline average [59].

The beeswarm plot shows feature impacts across samples. We observe the features ranked from top to bottom by their mean absolute SHAP values for the entire dataset. Each instance of a point on the graph represents a row in the dataset, distributed across the x-axis according to their SHAP value. The colour bar corresponds to the raw values, so high values appear red and low values are blue.

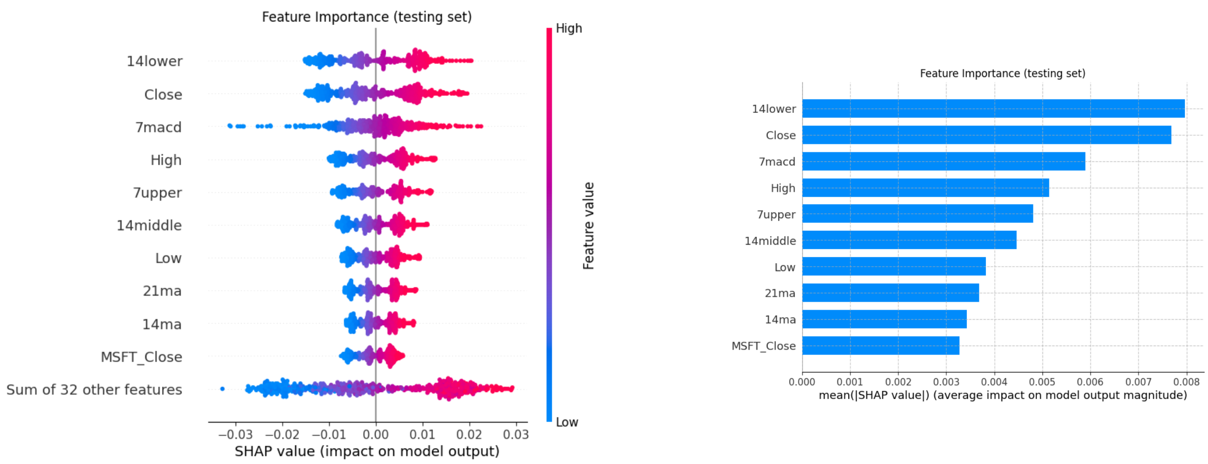


Figure 9: Top 10 feature importance across the testing dataset (in respect to SHAP values) displayed with a beeswarm and summary plot

The feature hierarchy highlights the model's reliance on technical indicators and competitor relationships, alongside raw price data. The raw price data (Close, High, Low), as fundamental features, are situated near the top as ultimately the close price is the one we are trying to predict. The prevalence of technical indicators is supported by financial theory [40], indicators such as Bollinger Bands, MACD, and moving averages (MA) are widely used in quantitative finance.

We also see a number of the same indicators calculated over different day periods, such as 14 day MA and 21 day MA. Financial volatility tends to cluster [24], during a volatility spike, short-window volatility measures will spike, while long-window averages remain smoother. Thus, having both horizons helps a model "see" when volatility is building or receding. Likewise, crosses of short/long moving averages often signal a change of market regime (from bull to bear or vice versa). Our WGAN-GP finding significance in this multi-timeframe analysis is a promising find, because it shows how the model can leverage basic technical indicators to produce highly complex market behaviours, resulting in realistic data generation.

The summary plot shows relatively small discrepancies in feature importance, with no one feature leading the pack. Indicating that during the testing period (a time of relative economic stability post COVID-19) the model takes a diversified approach to trend analysis. Using a number of technical indicators plus competitor and macroeconomic features (most of which fall within the top 20) means the model is less likely to jump to any extreme conclusions given a spike in any one single feature.

Performing the same SHAP analysis on our model over the more turbulent (COVID-19) validation period we see a slightly different result.

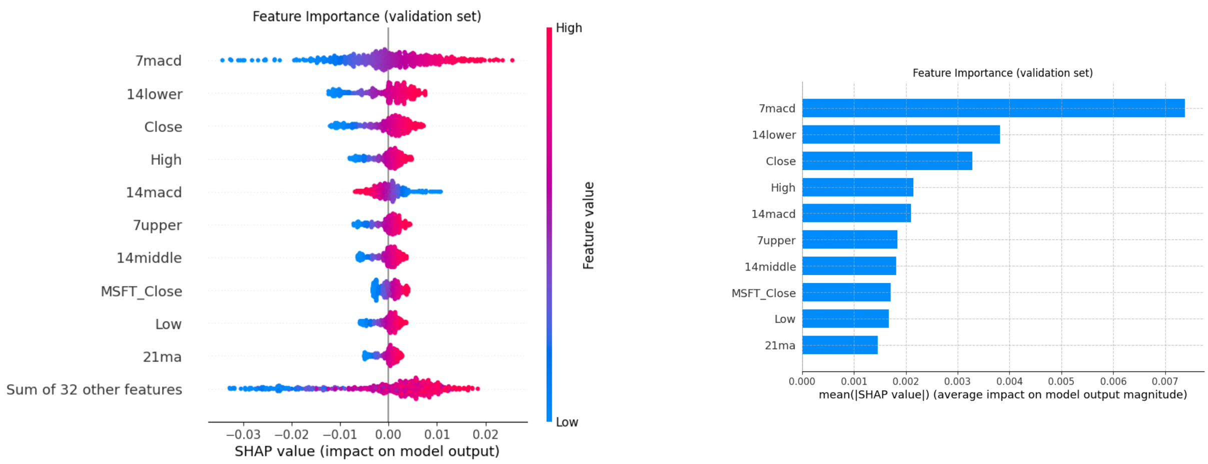


Figure 10: Top 10 feature importance across the validation dataset (in respect to SHAP values) displayed with a beeswarm and summary plot

Here we see the week-long MACD leads the feature importance by some margin. This is a technical indicator that financial analysis use to identify potential trend changes and momentum shifts by comparing two exponential moving averages of the stock price. Being selected as an important feature falls in line with financial literature which looks towards signifiers of a momentum shift in times of high volatility [16], such as the COVID-19 pandemic. Analysing the beeswam plot we notice the SHAP values for '7macd' are both wide and consistently impactful across numerous observations. This suggests that no matter deviations in raw value MACD should be highly regarded as an important feature for forecasting stock price, given a volatile market.

Ideally, to gain further understanding and better analyse the interpretability of our model, we would like to examine SHAP measures across multiple regimes of unseen data. This would provide us with a depth of insight into the model's feature importance process in an unbiased way.

5.3 Model optimisation

Ray Tune's Bayesian optimisation framework allowed us to systematically fine-tune both the hyper-parameters of our WGAN-GP model and the parameters governing the walk-forward function. These improvements highlight that WGAN-GP's adaptability is significantly enhanced through a carefully calibrated, dynamic optimisation process. An approach that is supported by recent advances in both Bayesian tuning techniques and robust time-series validation strategies [81].

5.3.1 Walk-Forward optimisation impact

We employed walk-forward validation to simulate the evolving market conditions inherent in time series data. This approach recalibrates the model frequently using the most recent data observations. After we dialled in our hyper-parameters we were able to achieve a marked performance increase in our testing and validation datasets with our walk-forward algorithm, greatly improving forecasting accuracy.

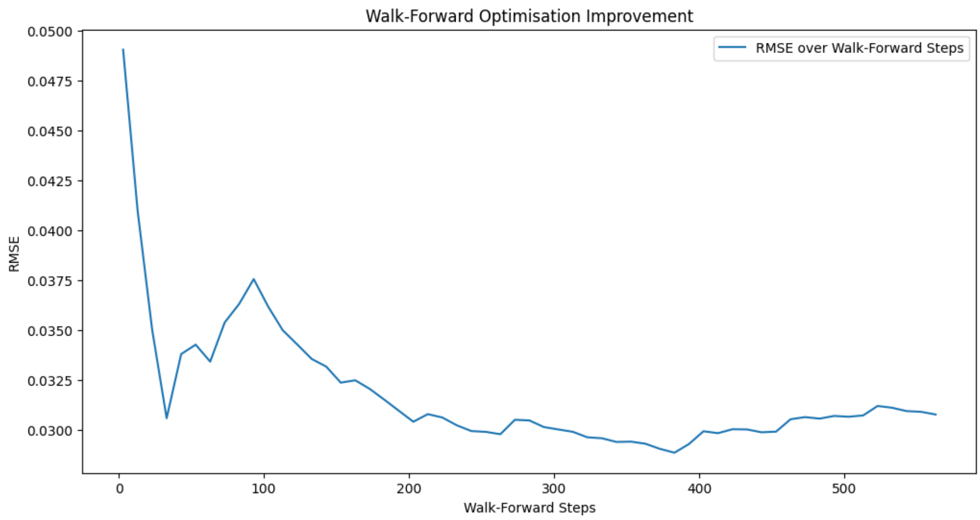


Figure 11: RMSE improvement in the validation set through the walk-forward period

Observing the trend in RMSE throughout the walk-forward period we see that it generally reduces over time, with fluctuations as the model over-compensates then corrects for these predictions. This indicates the model is attempting to continually improve itself, yet may be struggling during extreme market volatility and abrupt regime shifts.

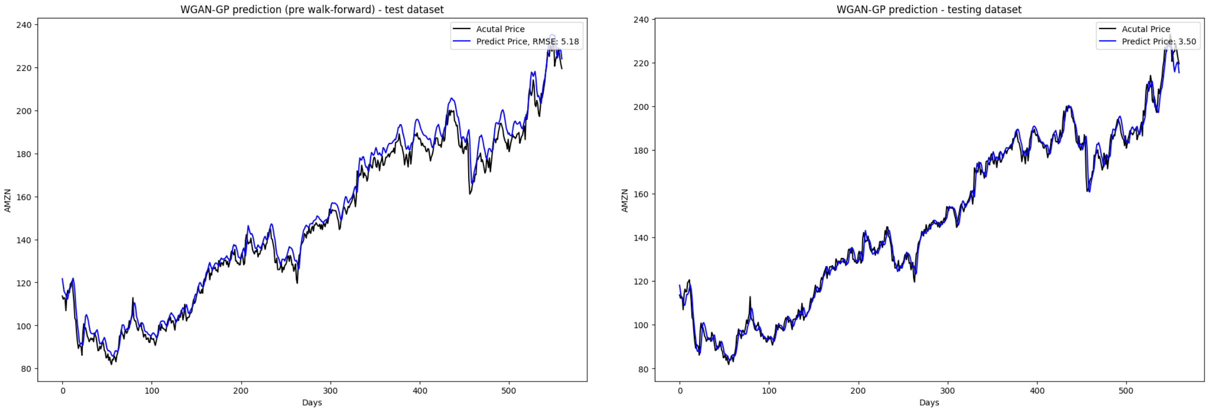


Figure 12: Comparison of testing set forecast, before and after walk-forward validation. *Left*: pre walk-forward, RMSE: 5.18. *Right*: post walk-forward, RMSE: 3.50

.

On the unseen (testing) dataset there was a RMSE improvement of 32.53%. By iteratively fine-tuning the generator the model dynamically adapts to emerging market regimes, such as the 2020 COVID-19 crash, while avoiding overfitting. Our finding is in line with arguments made by Bergmeir & Benítez [11] who advocate for time-series validation techniques to preserve temporal causality. We see this as a promising result because it shows how walk-forward validation can adapt highly complex non-linear models such as ours to handle the challenge of non-stationary data, and for financial markets; adaptability to shifting regimes. Traditionally a challenging point for static models such as ARIMA as noted by Hyndman & Khandakar [44].

Interestingly by looking at the changes in forecasting accuracy over the training dataset we see a different story.

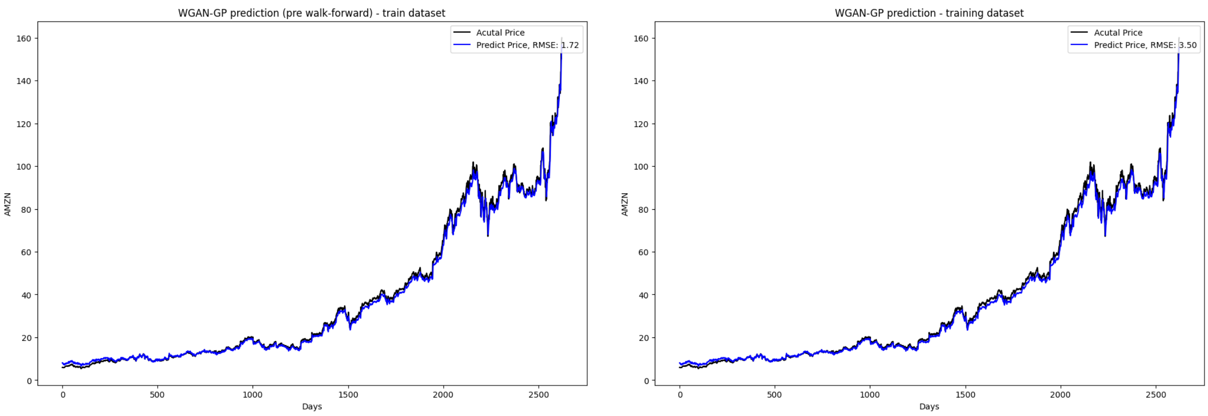


Figure 13: Comparison of training set forecast, before and after walk-forward validation. *Left*: pre walk-forward, RMSE: 1.72. *Right*: post walk-forward, RMSE: 3.50

.

This result shows a decreased forecasting performance. Although it looks bad on paper, what we are seeing is walk-forward validation accounting for overfitting to the training set. Exposing the model to highly volatile data present in the validation set, promoted an encoding of factors that predict volatile trends, as seen by its improvement in the latter two datasets. So when the re-trained generator forecasts on the training set again, it performs worse due to being a more generalised model, but now it has the advantage of being more resilience to volatility and noise. Whilst generalisation is what we are aiming to achieve, meaning better forecasting on unseen data, this result also shows there is room for improvement in the pre walk-forward model. If we have less overfitting to start with, cross validation methods have a greater chance of further optimising forecasting accuracy.

5.4 Comparative contextualisation

Comparative analysis is key for situating the WGAN-GP model within the broader forecasting landscape, for this, we chose ARMIA, LSTM and our WGAN-GP with a truncated dataset. Achieving low MAPE, RMSE and MAE values across diverse market conditions is challenging due to ever changing trends, therefore cross-analysing each model's performance helps us to gauge their strengths and limitations. Identifying our model's position within the broader landscape, and assessing the conditions under which it excels, enables us to better understand its optimal application, for both data generation and investor insight.

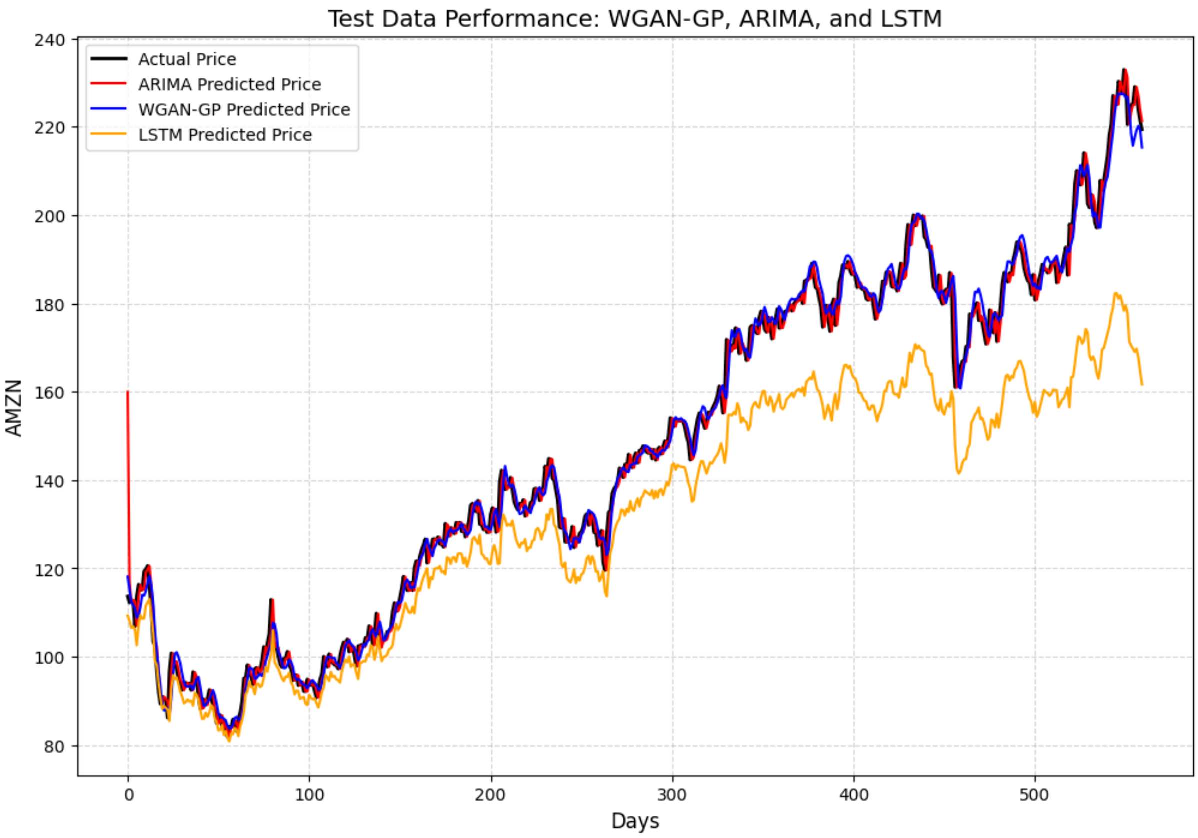


Figure 14: Comparison of WGAN-GP, ARIMA and LSTM forecasting models on the test dataset

Model	MAPE	RMSE	MAE
WGAN-GP (Truncated Dataset)	4.39%	8.92	7.07
WGAN-GP (Full Dataset)	1.84%	**3.50**	2.60
ARIMA	**1.60%**	3.52	**2.24**
LSTM	4.10%	10.45	7.05

Table 2: Comparison of forecasting models performance metrics

Our WGAN-GP model performs similarly or better compared to other models tested. This suggests it handles the complexity of stock market data effectively, especially when trained on diverse data spanning economic stability and volatility.

The LSTM model is a naive implementation, so whilst some studies have managed to gain strong results from optimising this method [63]. Here we are using the same basic network that our generator makes use of, to gauge improvement that the competitive nature of the critic in the WGAN-GP framework provides. As seen the results the full WGAN-GP far outperforms LSTM, showcasing and validating the possible performance increases from using a WGAN-GP framework.

Comparing our GAN's trained on the full dataset truncated dataset, we observe the full training performs better across all metrics. This suggests that using more data significantly enhances performance, likely due to capturing more recent trends and patterns, which is crucial for financial time series with evolving dynamics. This aligns with time series forecasting principles, which argues for the importance of data diversity [38] (periods such as COVID-19 or the 2008 financial crisis) or else models trained on stable periods may struggle during turbulent times. The improvement is particularly notable in RMSE and MAE, indicating fewer large errors and better average accuracy with the full dataset. A potential issue in this finding comes in the temporal overlap between the two datasets. Since the truncated dataset is a subset of the full dataset, overlapping periods might lead to redundant information being used. An alternative method would be to have two distinct non-overlapping datasets, meaning isolation of each model's training, allowing evaluation of performance to determine if differences in performance metrics are due solely to model improvements. However, this requires a much larger dataset, which as we have discussed is not always available in financial time-series.

ARMIA is often used as a baseline forecasting model in academic research, as despite its linear nature it still performs competitively by capturing complex trends, this is reinforced in our findings. Comparing our WGAN-GP to ARMIA we can see ARMIA pulls ahead in MAPE and MAE metrics, indicating ARMIA's more consistent average errors, but WGAN-GP's marginally better RMSE suggests fewer large errors.

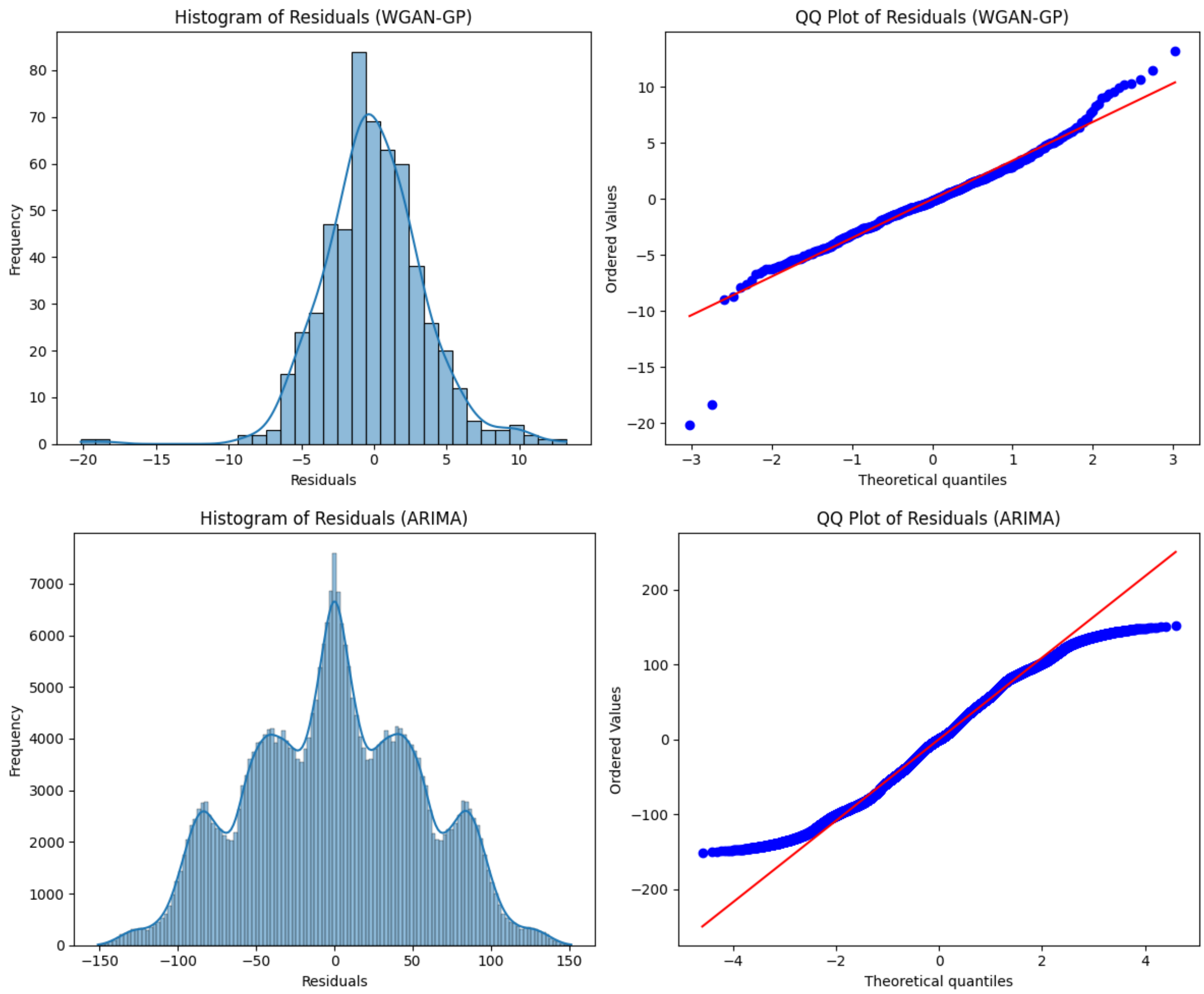


Figure 15: Comparison of WGAN-GPs (top) and ARIMAs (bottom) distributions of forecast errors (residuals)

The QQ plots reveal differences in error distribution. ARIMA's slightly S-shaped plot and non-normal distribution suggest that it might struggle with capturing certain asymmetries in stock price movements. Our WGAN-GP displays a mostly straight QQ plot with the presence of heavy tails. The bulk of residuals approximate a normal distribution, however the extreme residuals occur more frequently (also shown in the histogram). This presence of heavy tails indicates the utility of a WGAN-GP framework. As the Wasserstein algorithm minimises the Earth-Mover distance, it penalises discrepancies across the entire distribution, including tails, thereby enforcing the generator to allocate probability mass/importance to rare regions. This behaviour can make WGAN-GP better at modelling rare events, like sudden price drops or surges, predicting extreme movements more accurately. As an overall forecaster this is important because financial returns often exhibit fat tailed distributions, where extreme movements occur with higher probability than under the normal curve. Whilst this is a positive result, it is important to be aware that without proper hyper-parameter tuning this effect could push the model into overestimating extreme events.

6 Conclusion and Future Work

We set out to build a robust time-series forecasting model, that can aid with the data scarcity issue by generating realistic synthetic financial time-series data. We also imposed a further goal of showing that our model can provide insightful information as to which features of the financial environment provide important details about current trends in the stock market.

This study presented a novel WGAN-GP framework that leveraged deep learning, to model complex data distributions allowing us to generate realistic stock market predictions. Ultimately improving stock price forecasting robustness, particularly when trained on diverse datasets encompassing economic disruptions.

Building a dataset, curated to provide context about the chosen stock (Amazon Inc.), and the wider market involving macroeconomic and competitor data. Supplementing this with numerous technical indicators and further volatility smoothing features such as Fourier transforms. This set up our model with the possibility to learn highly non-linear patterns through having access to this diverse information, whilst avoiding becoming overly reliant on noisy data, leading to reduced overfitting and better generalisation. Whilst our methods produced a strong dataset we observed certain multicollinearity issues, leading to a possible limitation in the usefulness of certain features.

We built a GAN based on an LSTM generator and 1dCNN discriminator, trained using Wasserstein distance upheld by a Gradient Penalty function. Resulting in a generator model that can effectively produce realistic synthetic stock market data, that makes use of the powerful LSTM cell framework to capture significant trends. Meanwhile, loss from the Wasserstein distance regulates overfitting through its nuanced approach to calculating the discriminator (critic) loss. Despite facing challenges with model complexity and hyper-parameter tuning, we overcame this by way of Bayesian search techniques that allowed us to dial in the framework.

The generator was shown to prove more effective when retrained/updated on current data through walk-forward optimisation. Continuous retraining with the most up-to-date trends in the stock market proved to allow the model to generalise better to unseen current data. However, from analysing the distribution of residuals we acknowledge there are still improvements to be made in this area. Both within walk-forward validation and more up-to-date cross validation techniques [36, 66] that could provide improved current-trend forecasting, better than the results we achieved.

In conclusion, this model proves a strong step to resolving the issues motivating this project. Our model generates realistic time-series financial data within a RMSE of 3.50 and a MAPE of 1.84%. Enabling it to build a convincing dataset of synthetic stock market data, particularly excelling in forecasting turbulent periods. This strength can be used in alleviating data scarcity, particularly by generating realistic data for extreme market events, which goes under-represented in many financial datasets [26]. Training future forecasting models and algorithmic trading by providing more complete datasets. Another motivation was to create an interpretable model to highlight market trends for investors' scrutiny, we believe our model shows promise in being a tool for understanding the market. Our SHAP analysis shows our model consolidates important features that predict trends, supported in literature. By providing investors with this knowledge it allows them to make more informed decisions on portfolio management.

6.1 Future Work

Building on these findings, several research directions merit investigation to further enhance the robustness, interpretability and practical utility of our WGAN-GP framework.

Extended Metric Framework: We chose RMSE to be the primary metric to optimise our model against, as for financial forecasting it's often beneficial to penalise large errors more heavily. It's also a useful metric with regard to interpretability as it expresses errors in the same units as the forecasted variable, meaning it is a great metric when evaluating against other forecasting models [8]. However, solely relying on one metric is risky as RMSE's sensitive to outliers meaning a few extreme mispredictions can disproportionately affect the metric. Especially when it comes to WGAN-GP optimisation it is important to have metrics which can fully capture market dynamics. WGAN-GP's strength lies in generating probabilistic scenarios so in future works we suggest supplementing RMSE with metrics like CRPS (Continuous Ranked Probability Score) to evaluate distributional accuracy [12].

Methodological Extensions: Employing an ensemble architecture could allow for a model to deploy specialised generators for bull/bear/crisis markets. Allowing greater coverage and specialisation of various market regimes is ideal for capturing deeper non-linear relationships, reducing overall overfitting. Our results show use of both our WGAN-PG and ARMIA cover a wide range of market conditions, finding a way to comprehensively combine them could result with a powerful model in respect to forecasting both stable and unstable markets. Another angle to improve WGAN-GP is through using transformer networks [86], which self-attention mechanisms complement the Wasserstein critic's distributional matching by enabling the generator to capture both local patterns and global sequence structures

Regime Specific models: As the field of AI develops we are seeing a trend to increasingly specific models (e.g language models tuned for creative writing[32] vs professional writing [70]). This ideology could be very beneficial in financial contexts where we see a number of regimes present throughout time, each with their unique challenges and a model best fitted to forecast it. Furthermore, having specific models for specific use cases expands explainability as it allows investors to narrow down which factors build up a given regime.

Data Enhancements: Alternative data integration such as dynamic semantic relationships from news flow, as demonstrated by [13] could enrich the dataset. Their framework improved forecasting accuracy by 40.3% by modelling entity correlations in Twitter news data over 12 years. Applying this to a conditional-GAN framework [62] may better capture latent market influencers, improving forecast prediction in turbulent periods. We would also like to see our methods and model applied to other areas of the stock market beyond Amazon Inc. specifically more data-scarce environments, to show how it stands up to a wider range of market behaviours. A further improvement we recognise can come from addressing multicollinearity issues, traditionally fruitful methods such as Autoencoder-Based Encoding [18] could be a strong non-linear way to resolving this problem. Or perhaps more intricate solutions could be adapted to our WGAN-GP framework, like Correlation-Embedded Attention which integrates a correlation penalty into attention modules. This forces the network to weigh inputs inversely to their mutual correlations, applied to LSTMs for FX forecasting, this method mitigated multicollinearity without dropping features and improved out-of-sample returns[17].

References

[1] Takuya Akiba et al. *Optuna: A Next-generation Hyperparameter Optimization Framework*. 25th July 2019. DOI: 10.48550/arXiv.1907.10902. arXiv: 1907.10902 [cs]. URL: http://arxiv.org/abs/1907.10902. Pre-published.

[2] Isik Akin and Meryem Akin. 'Behavioral Finance Impacts on US Stock Market Volatility: An Analysis of Market Anomalies'. In: *Behavioural Public Policy* (13th Mar. 2024), pp. 1–25. ISSN: 2398-063X, 2398-0648. DOI: 10.1017/bpp.2024.13. URL: https://www.cambridge.org/core/journals/behavioural-public-policy/article/behavioral-finance-impacts-on-us-stock-market-volatility-an-analysis-of-market-anomalies/D1CEF34141D03D8BECB2AE42467166B3?utm_source=chatgpt.com.

[3] Andrew Ang and Allan Timmermann. 'Regime Changes and Financial Markets'. In: *Annual Review of Financial Economics* 4 (Volume 4, 2012 1st Oct. 2012), pp. 313–337. ISSN: 1941-1367, 1941-1375. DOI: 10.1146/annurev-financial-110311-101808. URL: https://www.annualreviews.org/content/journals/10.1146/annurev-financial-110311-101808.

[4] Martin Arjovsky, Soumith Chintala and Léon Bottou. *Wasserstein GAN*. 6th Dec. 2017. DOI: 10.48550/arXiv.1701.07875. arXiv: 1701.07875 [stat]. URL: http://arxiv.org/abs/1701.07875. Pre-published.

[5] Ran Aroussi. *Ranaroussi/Yfinance*. 10th Apr. 2025. URL: https://github.com/ranaroussi/yfinance.

[6] *Assessing the Impact of Technical Indicators on Machine Learning Models for Stock Price Prediction*. URL: https://arxiv.org/html/2412.15448v1.

[7] Zhao B*. 'Improved Generative Adversarial Network Stock Price Prediction Method'. In: *Open Access Journal of Data Science and Artificial Intelligence* 2.1 (2024). ISSN: 2996671X. DOI: 10.23880/oajda-16000111. URL: https://medwinpublishers.com/OAJDA/improved-generative-adversarial-network-stock-price-prediction-method.pdf.

[8] Prakash Balasubramanian et al. 'A Systematic Literature Survey on Recent Trends in Stock Market Prediction'. In: *PeerJ Computer Science* 10 (31st Jan. 2024), e1700. ISSN: 2376-5992. DOI: 10.7717/peerj-cs.1700. URL: https://peerj.com/articles/cs-1700.

[9] Kasun Bandara et al. *Sales Demand Forecast in E-commerce Using a Long Short-Term Memory Neural Network Methodology*. 11th Aug. 2019. DOI: 10.48550/arXiv.1901.04028. arXiv: 1901.04028 [cs]. URL: http://arxiv.org/abs/1901.04028. Pre-published.

[10] Wei Bao, Jun Yue and Yulei Rao. 'A Deep Learning Framework for Financial Time Series Using Stacked Autoencoders and Long-Short Term Memory'. In: *PLOS ONE* 12.7 (14th July 2017), e0180944. ISSN: 1932-6203. DOI: 10.1371/journal.pone.0180944. URL: https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0180944.

[11] Christoph Bergmeir and José M. Benítez. 'On the Use of Cross-Validation for Time Series Predictor Evaluation'. In: *Information Sciences*. Data Mining for Software Trustworthiness 191 (15th May 2012), pp. 192–213. ISSN: 0020-0255. DOI: 10.1016/j.ins.2011.12.028. URL: https://www.sciencedirect.com/science/article/pii/S0020025511006773.

[12] Jonathan Berrisch and Florian Ziel. 'Distributional Modeling and Forecasting of Natural Gas Prices'. In: *Journal of Forecasting* 41.6 (2022), pp. 1065–1086. ISSN: 1099-131X. DOI: 10.1002/for.2853. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/for.2853.

[13] Amirhosein Bodaghi and Jonathan J. H. Zhu. 'A Big Data Analysis of the Adoption of Quoting Encouragement Policy on Twitter during the 2020 U.S. Presidential Election'. In: *Journal of Computational Social Science* 7.2 (19th May 2024), pp. 1861–1893. DOI: 10.1007/s42001-024-00291-6. URL: http://www.scopus.com/inward/record.url?scp=85193419864&partnerID=8YFLogxK.

[14] Imad Bou-Hamad and Ibrahim Jamali. 'Forecasting Financial Time-Series Using Data Mining Models: A Simulation Study'. In: *Research in International Business and Finance* 51 (1st Jan. 2020), p. 101072. ISSN: 0275-5319. DOI: 10.1016/j.ribaf.2019.101072. URL: https://www.sciencedirect.com/science/article/pii/S027553191830761X.

[15] George E. P. Box et al. *Time Series Analysis: Forecasting and Control*. John Wiley & Sons, 29th May 2015. 709 pp. ISBN: 978-1-118-67492-5. Google Books: rNt5CgAAQBAJ.

[16] Giovanni Campisi, Silvia Muzzioli and Bernard De Baets. 'A Comparison of Machine Learning Methods for Predicting the Direction of the US Stock Market on the Basis of Volatility Indices'. In: *International Journal of Forecasting* 40.3 (1st July 2024), pp. 869–880. ISSN: 0169-2070. DOI: 10.1016/j.ijforecast.2023.07.002. URL: https://www.sciencedirect.com/science/article/pii/S0169207023000729.

[17] Jireh Yi-Le Chan et al. 'A Correlation-Embedded Attention Module to Mitigate Multicollinearity: An Algorithmic Trading Application'. In: *Mathematics* 10.8 (8 Jan. 2022), p. 1231. ISSN: 2227-7390. DOI: 10.3390/math10081231. URL: https://www.mdpi.com/2227-7390/10/8/1231.

[18] Jireh Yi-Le Chan et al. 'Mitigating the Multicollinearity Problem and Its Machine Learning Approach: A Review'. In: *Mathematics* 10.8 (8 Jan. 2022), p. 1283. ISSN: 2227-7390. DOI: 10.3390/math10081283. URL: https://www.mdpi.com/2227-7390/10/8/1283.

[19] C. Chatfield. 'The Holt-Winters Forecasting Procedure'. In: *Journal of the Royal Statistical Society. Series C (Applied Statistics)* 27.3 (1978), pp. 264–279. ISSN: 0035-9254. DOI: 10.2307/2347162. JSTOR: 2347162. URL: https://www.jstor.org/stable/2347162.

[20] Rong Chen, Zhixin Kang and Lan Zhang. 'Forecasting Return Volatility in the Presence of Microstructure Noise'. In: *Statistics and Its Interface* 3.2 (2010), pp. 145–157. ISSN: 19387989, 19387997. DOI: 10.4310/SII.2010.v3.n2.a2. URL: https://link.intlpress.com/JDetail/1806634564384698370.

[21] Kyunghyun Cho et al. *Learning Phrase Representations Using RNN Encoder-Decoder for Statistical Machine Translation*. 3rd Sept. 2014. DOI: 10.48550/arXiv.1406.1078. arXiv: 1406.1078 [cs]. URL: http://arxiv.org/abs/1406.1078. Pre-published.

[22] R. Cont. 'Empirical Properties of Asset Returns: Stylized Facts and Statistical Issues'. In: *Quantitative Finance* 1.2 (1st Feb. 2001), pp. 223–236. ISSN: 1469-7688. DOI: 10.1080/713665670. URL: https://doi.org/10.1080/713665670.

[23] Rama Cont. 'Empirical Properties of Asset Returns: Stylized Facts and Statistical Issues'. In: *QUANTITATIVE FINANCE* ().

[24] Rama Cont. 'Volatility Clustering in Financial Markets: Empirical Facts and Agent-Based Models'. In: *Long Memory in Economics*. Ed. by Gilles Teyssière and Alan P. Kirman. Berlin, Heidelberg: Springer, 2007, pp. 289–309. ISBN: 978-3-540-34625-8. DOI: 10.1007/978-3-540-34625-8_10. URL: https://doi.org/10.1007/978-3-540-34625-8_10.

[25] Akash Deep et al. *Risk-Adjusted Performance of Random Forest Models in High-Frequency Trading*. Working paper. arXiv.org, Feb. 2025. URL: https://EconPapers.repec.org/RePEc:arx:papers:2412.15448.

[26] Tom Djaaleb and Julien Romero. 'Evaluating and Finetuning Models For Financial Time Series Forecasting'. In: (13th Oct. 2023). URL: https://openreview.net/forum?id=53gU1BASrd&utm_source=chatgpt.com.

[27] Samuel K.-B. Dzidzornu and Sonia D. and Tamakloe. 'Stylized Statistical Properties in Bloc-Based Financial Markets'. In: *Research in Statistics* 3.1 (31st Dec. 2025), p. 2486172. ISSN: null. DOI: 10.1080/27684520.2025.2486172. URL: https://doi.org/10.1080/27684520.2025.2486172.

[28] Robert Engle. 'Risk and Volatility: Econometric Models and Financial Practice'. In: *The American Economic Review* 94.3 (2004), pp. 405–420. ISSN: 0002-8282. JSTOR: 3592935. URL: https://www.jstor.org/stable/3592935.

[29] Hassan Ismail Fawaz et al. 'Deep Learning for Time Series Classification: A Review'. In: *Data Mining and Knowledge Discovery* 33.4 (1st July 2019), pp. 917–963. ISSN: 1384-5810, 1573-756X. DOI: 10.1007/s10618-019-00619-1. arXiv: 1809.04356 [cs]. URL: http://arxiv.org/abs/1809.04356.

[30] *Federal Reserve Economic Data | FRED | St. Louis Fed*. URL: https://fred.stlouisfed.org/.

[31] Paolo Giudici et al. 'Explainable Artificial Intelligence Methods for Financial Time Series'. In: *Physica A: Statistical Mechanics and its Applications* 655 (1st Dec. 2024), p. 130176. ISSN: 0378-4371. DOI: 10.1016/j.physa.2024.130176. URL: https://www.sciencedirect.com/science/article/pii/S037843712400685X.

[32] Carlos Gómez-Rodríguez and Paul Williams. *A Confederacy of Models: A Comprehensive Evaluation of LLMs on Creative Writing*. 12th Oct. 2023. DOI: 10.48550/arXiv.2310.08433. arXiv: 2310.08433 [cs]. URL: http://arxiv.org/abs/2310.08433. Pre-published.

[33] Ian J. Goodfellow et al. *Generative Adversarial Networks*. 10th June 2014. DOI: 10.48550/arXiv.1406.2661. arXiv: 1406.2661 [stat]. URL: http://arxiv.org/abs/1406.2661. Pre-published.

[34] Xinyu Gu et al. 'A Time-Series Wasserstein GAN Method for State-of-Charge Estimation of Lithium-Ion Batteries'. In: *Journal of Power Sources* 581 (15th Oct. 2023), p. 233472. ISSN: 0378-7753. DOI: 10.1016/j.jpowsour.2023.233472. URL: https://www.sciencedirect.com/science/article/pii/S0378775323008480.

[35] Ishaan Gulrajani et al. *Improved Training of Wasserstein GANs*. 25th Dec. 2017. DOI: 10.48550/arXiv.1704.00028. arXiv: 1704.00028 [cs]. URL: http://arxiv.org/abs/1704.00028. Pre-published.

[36] Ishaan Gulrajani et al. *Improved Training of Wasserstein GANs*. 25th Dec. 2017. DOI: 10.48550/arXiv.1704.00028. arXiv: 1704.00028 [cs]. URL: http://arxiv.org/abs/1704.00028. Pre-published.

[37] Necla Gunduz and Ernest Fokoue. *Robust Classification of High Dimension Low Sample Size Data*. 3rd Jan. 2015. DOI: 10.48550/arXiv.1501.00592. arXiv: 1501.00592 [stat]. URL: http://arxiv.org/abs/1501.00592. Pre-published.

[38] Faizal Hafiz, Jan Broekaert and Akshya Swain. 'Evolution of Neural Architectures for Financial Forecasting: A Note on Data Incompatibility during Crisis Periods'. In: *Annals of Operations Research* 346.2 (1st Mar. 2025), pp. 1011–1025. ISSN: 1572-9338. DOI: 10.1007/s10479-024-06098-y. URL: https://doi.org/10.1007/s10479-024-06098-y.

[39] James D. Hamilton. 'A New Approach to the Economic Analysis of Nonstationary Time Series and the Business Cycle'. In: *Econometrica* 57.2 (1989), pp. 357–384. ISSN: 0012-9682. DOI: 10.2307/1912559. JSTOR: 1912559. URL: https://www.jstor.org/stable/1912559.

[40] Yufeng Han et al. *Technical Analysis in the Stock Market: A Review*. 21st May 2021. DOI: 10.2139/ssrn.3850494. Social Science Research Network: 3850494. URL: https://papers.ssrn.com/abstract=3850494. Pre-published.

[41] Brian C. Hatch et al. 'Algorithmic Trading and Firm Value'. In: *Journal of Banking & Finance* 125 (1st Apr. 2021), p. 106090. ISSN: 0378-4266. DOI: 10.1016/j.jbankfin.2021.106090. URL: https://www.sciencedirect.com/science/article/pii/S0378426621000480.

[42] Sepp Hochreiter and Jürgen Schmidhuber. 'Long Short-Term Memory'. In: *Neural Computation* 9.8 (1997), pp. 1735–1780. ISSN: 1530-888X. DOI: 10.1162/neco.1997.9.8.1735.

[43] Chengyue Huang and Yahe Yang. *Time Series Feature Redundancy Paradox: An Empirical Study Based on Mortgage Default Prediction*. 23rd Dec. 2024. DOI: 10.48550/arXiv.2501.00034. arXiv: 2501.00034 [q-fin]. URL: http://arxiv.org/abs/2501.00034. Pre-published.

[44] Rob J. Hyndman and Yeasmin Khandakar. 'Automatic Time Series Forecasting: The Forecast Package for R'. In: *Journal of Statistical Software* 27 (29th July 2008), pp. 1–22. ISSN: 1548-7660. DOI: 10.18637/jss.v027.i03. URL: https://doi.org/10.18637/jss.v027.i03.

[45] Sergey Ioffe and Christian Szegedy. *Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift*. 2nd Mar. 2015. DOI: 10.48550/arXiv.1502.03167. arXiv: 1502.03167 [cs]. URL: http://arxiv.org/abs/1502.03167. Pre-published.

[46] Md Khairul Islam et al. *Large Language Models for Financial Aid in Financial Time-series Forecasting*. 24th Oct. 2024. DOI: 10.48550/arXiv.2410.19025. arXiv: 2410.19025 [cs]. URL: http://arxiv.org/abs/2410.19025. Pre-published.

[47] Kevin Johnson. *Twopirllc/Pandas-Ta*. 1st May 2025. URL: https://github.com/twopirllc/pandas-ta.

[48] Gunho Jung and Sun-Yong Choi. 'Forecasting Foreign Exchange Volatility Using Deep Learning Autoencoder-LSTM Techniques'. In: *Complexity* 2021.1 (2021), p. 6647534. ISSN: 1099-0526. DOI: 10.1155/2021/6647534. URL: https://onlinelibrary.wiley.com/doi/abs/10.1155/2021/6647534.

[49] Kyoung-jae Kim. 'Financial Time Series Forecasting Using Support Vector Machines'. In: *Neurocomputing*. Support Vector Machines 55.1 (1st Sept. 2003), pp. 307–319. ISSN: 0925-2312. DOI: 10.1016/S0925-2312(03)00372-2. URL: https://www.sciencedirect.com/science/article/pii/S0925231203003722.

[50] Yang-Seon Kim et al. 'Investigating the Impact of Data Normalization Methods on Predicting Electricity Consumption in a Building Using Different Artificial Neural Network Models'. In: *Sustainable Cities and Society* 118 (1st Jan. 2025), p. 105570. ISSN: 2210-6707. DOI: 10.1016/j.scs.2024.105570. URL: https://www.sciencedirect.com/science/article/pii/S2210670724003962.

[51] Diederik P. Kingma and Jimmy Ba. *Adam: A Method for Stochastic Optimization*. 30th Jan. 2017. DOI: 10.48550/arXiv.1412.6980. arXiv: 1412.6980 [cs]. URL: http://arxiv.org/abs/1412.6980. Pre-published.

[52] Naveen Kodali et al. *On Convergence and Stability of GANs*. 10th Dec. 2017. DOI: 10.48550/arXiv.1705.07215. arXiv: 1705.07215 [cs]. URL: http://arxiv.org/abs/1705.07215. Pre-published.

[53] Vaia I. Kontopoulou et al. 'A Review of ARIMA vs. Machine Learning Approaches for Time Series Forecasting in Data Driven Networks'. In: *Future Internet* 15.8 (8 Aug. 2023), p. 255. ISSN: 1999-5903. DOI: 10.3390/fi15080255. URL: https://www.mdpi.com/1999-5903/15/8/255.

[54] Youssef Kossale, Mohammed Airaj and Aziz Darouichi. 'Mode Collapse in Generative Adversarial Networks: An Overview'. In: *2022 8th International Conference on Optimization and Applications (ICOA)*. 2022 8th International Conference on Optimization and Applications (ICOA). Oct. 2022, pp. 1–6. DOI: 10.1109/ICOA55659.2022.9934291. URL: https://ieeexplore.ieee.org/document/9934291.

[55] S. V. Jagadesh Kumar et al. 'Stock Price Prediction Using Recurrent Neural Network and Long-Short Term Memory'. In: *International Journal of Engineering Research & Technology* 13.10 (17th Nov. 2024). ISSN: 2278-0181. DOI: 10.17577/IJERTV13IS100114. URL: https://www.ijert.org/research/stock-price-prediction-using-recurrent-neural-network-and-long-short-term-memory-%20IJERTV13IS100114.pdf,%20https://www.ijert.org/stock-price-prediction-using-recurrent-neural-network-and-long-short-term-memory.

[56] Sohyeon Kwon and Yongjae Lee. *Can GANs Learn the Stylized Facts of Financial Time Series?* 13th Oct. 2024. DOI: 10.48550/arXiv.2410.09850. arXiv: 2410.09850 [q-fin]. URL: http://arxiv.org/abs/2410.09850. Pre-published.

[57] Chen Liu et al. *Global Neural Networks and The Data Scaling Effect in Financial Time Series Forecasting*. 20th Feb. 2025. DOI: 10.48550/arXiv.2309.02072. arXiv: 2309.02072 [econ]. URL: http://arxiv.org/abs/2309.02072. Pre-published.

[58] Minrong Lu and Xuerong Xu. 'TRNN: An Efficient Time-Series Recurrent Neural Network for Stock Price Prediction'. In: *Information Sciences* 657 (1st Feb. 2024), p. 119951. ISSN: 0020-0255. DOI: 10.1016/j.ins.2023.119951. URL: https://www.sciencedirect.com/science/article/pii/S0020025523015360.

[59] Scott Lundberg and Su-In Lee. *A Unified Approach to Interpreting Model Predictions*. 25th Nov. 2017. DOI: 10.48550/arXiv.1705.07874. arXiv: 1705.07874 [cs]. URL: http://arxiv.org/abs/1705.07874. Pre-published.

[60] Philippe Masset. *Analysis of Financial Time-Series Using Fourier and Wavelet Methods*. 24th Oct. 2008. DOI: 10.2139/ssrn.1289420. Social Science Research Network: 1289420. URL: https://papers.ssrn.com/abstract=1289420. Pre-published.

[61] Ebikella Mienye et al. 'Deep Learning in Finance: A Survey of Applications and Techniques'. In: *AI* 5.4 (4 Dec. 2024), pp. 2066–2091. ISSN: 2673-2688. DOI: 10.3390/ai5040101. URL: https://www.mdpi.com/2673-2688/5/4/101.

[62] Mehdi Mirza and Simon Osindero. *Conditional Generative Adversarial Nets*. 6th Nov. 2014. DOI: 10.48550/arXiv.1411.1784. arXiv: 1411.1784 [cs]. URL: http://arxiv.org/abs/1411.1784. Pre-published.

[63] Adil Moghar and Mhamed Hamiche. 'Stock Market Prediction Using LSTM Recurrent Neural Network'. In: *Procedia Computer Science*. The 11th International Conference on Ambient Systems, Networks and Technologies (ANT) / The 3rd International Conference on Emerging Data and Industry 4.0 (EDI40) / Affiliated Workshops 170 (1st Jan. 2020), pp. 1168–1173. ISSN: 1877-0509. DOI: 10.1016/j.procs.2020.03.049. URL: https://www.sciencedirect.com/science/article/pii/S1877050920304865.

[64] Seyed Mostafa Mostafavi and Ali Reza Hooman. 'Key Technical Indicators for Stock Market Prediction'. In: *Machine Learning with Applications* 20 (1st June 2025), p. 100631. ISSN: 2666-8270. DOI: 10.1016/j.mlwa.2025.100631. URL: https://www.sciencedirect.com/science/article/pii/S2666827025000143.

[65] Norita Md Norwawi. 'Sliding Window Time Series Forecasting with Multilayer Perceptron and Multiregression of COVID-19 Outbreak in Malaysia'. In: *Data Science for COVID-19* (2021), pp. 547–564. DOI: 10.1016/B978-0-12-824536-1.00025-3. pmid: null. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8988917/.

[66] David Opeoluwa Oyewola, Emmanuel Gbenga Dada and Juliana Ngozi Ndunagu. 'A Novel Hybrid Walk-Forward Ensemble Optimization for Time Series Cryptocurrency Prediction'. In: *Heliyon* 8.11 (1st Nov. 2022), e11862. ISSN: 2405-8440. DOI: 10.1016/j.heliyon.2022.e11862. URL: https://www.sciencedirect.com/science/article/pii/S2405844022031504.

[67] Robert Pardo. *Design, Testing, and Optimization of Trading Systems*. John Wiley & Sons, 26th Aug. 1992. 188 pp. ISBN: 978-0-471-55446-2. Google Books: LTmHPfs93_YC.

[68] Razvan Pascanu, Tomas Mikolov and Yoshua Bengio. *On the Difficulty of Training Recurrent Neural Networks*. 16th Feb. 2013. DOI: 10.48550/arXiv.1211.5063. arXiv: 1211.5063 [cs]. URL: http://arxiv.org/abs/1211.5063. Pre-published.

[69] Adam Paszke et al. *PyTorch: An Imperative Style, High-Performance Deep Learning Library*. 3rd Dec. 2019. DOI: 10.48550/arXiv.1912.01703. arXiv: 1912.01703 [cs]. URL: http://arxiv.org/abs/1912.01703. Pre-published.

[70] Shounak Paul et al. 'Pre-Trained Language Models for the Legal Domain: A Case Study on Indian Law'. In: *Proceedings of the Nineteenth International Conference on Artificial Intelligence and Law*. ICAIL '23. New York, NY, USA: Association for Computing Machinery, 7th Sept. 2023, pp. 187–196. ISBN: 9798400701979. DOI: 10.1145/3594536.3595165. URL: https://doi.org/10.1145/3594536.3595165.

[71] Henning Petzka, Asja Fischer and Denis Lukovnikov. 'On the Regularization of Wasserstein GANs'. In: International Conference on Learning Representations. 15th Feb. 2018. URL: https://openreview.net/forum?id=B1hYRMbCW&utm_source=chatgpt.com.

[72] Alexandra A. Portnova-Fahreeva et al. 'Linear and Non-linear Dimensionality-Reduction Techniques on Full Hand Kinematics'. In: *Frontiers in Bioengineering and Biotechnology* 8 (5th May 2020). ISSN: 2296-4185. DOI: 10.3389/fbioe.2020.00429. URL: https://www.frontiersin.orghttps://www.frontiersin.org/journals/bioengineering-and-biotechnology/articles/10.3389/fbioe.2020.00429/full.

[73] Mande Praveen et al. 'Financial Time Series Forecasting: A Comprehensive Review of Signal Processing and Optimization-Driven Intelligent Models'. In: *Computational Economics* (5th Mar. 2025). ISSN: 1572-9974. DOI: 10.1007/s10614-025-10899-z. URL: https://doi.org/10.1007/s10614-025-10899-z.

[74] *Quantitative Investing and the Limits of (Deep) Learning from Financial Data*. URL: https://www.capco.com/capco-institute/journal-47-digitization/quantitative-investing-and-the-limits-of-deep-learning-from-financial-data.

[75] Xiaohang Ren et al. 'Fourier Transform Based LSTM Stock Prediction Model under Oil Shocks'. In: *Quantitative Finance and Economics* 6.QFE-06-02-015 (2 2022), pp. 342–358. ISSN: 2573-0134. DOI: 10.3934/QFE.2022015. URL: http://www.aimspress.com/article/doi/10.3934/QFE.2022015.

[76] Francesco Rundo et al. 'Machine Learning for Quantitative Finance Applications: A Survey'. In: *Applied Sciences* 9.24 (24 Jan. 2019), p. 5574. ISSN: 2076-3417. DOI: 10.3390/app9245574. URL: https://www.mdpi.com/2076-3417/9/24/5574.

[77] Kirill Safonov. *Neural Network Approach to Demand Estimation and Dynamic Pricing in Retail*. 13th Dec. 2024. DOI: 10.48550/arXiv.2412.00920. arXiv: 2412.00920 [econ]. URL: http://arxiv.org/abs/2412.00920. Pre-published.

[78] Imrus Salehin and Dae-Ki Kang. 'A Review on Dropout Regularization Approaches for Deep Neural Networks within the Scholarly Domain'. In: *Electronics* 12.14 (14 Jan. 2023), p. 3106. ISSN: 2079-9292. DOI: 10.3390/electronics12143106. URL: https://www.mdpi.com/2079-9292/12/14/3106.

[79] Sudhi Sinha and Young M. Lee. 'Challenges with Developing and Deploying AI Models and Applications in Industrial Systems'. In: *Discover Artificial Intelligence* 4.1 (16th Aug. 2024), p. 55. ISSN: 2731-0809. DOI: 10.1007/s44163-024-00151-2. URL: https://doi.org/10.1007/s44163-024-00151-2.

[80] Kamilya Smagulova and Alex Pappachen James. 'Overview of Long Short-Term Memory Neural Networks'. In: *Deep Learning Classifiers with Memristive Networks: Theory and Applications*. Ed. by Alex Pappachen James. Cham: Springer International Publishing, 2020, pp. 139–153. ISBN: 978-3-030-14524-8. DOI: 10.1007/978-3-030-14524-8_11. URL: https://doi.org/10.1007/978-3-030-14524-8_11.

[81] Jasper Snoek, Hugo Larochelle and Ryan P. Adams. *Practical Bayesian Optimization of Machine Learning Algorithms*. 29th Aug. 2012. DOI: 10.48550/arXiv.1206.2944. arXiv: 1206.2944 [stat]. URL: http://arxiv.org/abs/1206.2944. Pre-published.

[82] Stoyan V. Stoyanov et al. 'Fat-Tailed Models for Risk Estimation'. In: *Working Paper Series in Economics* 30 (2011). URL: https://ideas.repec.org//p/zbw/kitwps/30.html.

[83] Shuntaro Takahashi, Yu Chen and Kumiko Tanaka-Ishii. 'Modeling Financial Time-Series with Generative Adversarial Networks'. In: *Physica A: Statistical Mechanics and its Applications* 527 (1st Aug. 2019), p. 121261. ISSN: 0378-4371. DOI: 10.1016/j.physa.2019.121261. URL: https://www.sciencedirect.com/science/article/pii/S0378437119307277.

[84] Mehak Usmani et al. 'Stock Market Prediction Using Machine Learning Techniques'. In: *2016 3rd International Conference on Computer and Information Sciences (ICCOINS)*. 2016 3rd International Conference on Computer and Information Sciences (ICCOINS). Aug. 2016, pp. 322–327. DOI: 10.1109/ICCOINS.2016.7783235. URL: https://ieeexplore.ieee.org/document/7783235.

[85] Vladimir Vapnik. *The Nature of Statistical Learning Theory*. Springer Science & Business Media, 19th Nov. 1999. 340 pp. ISBN: 978-0-387-98780-4. Google Books: sna9BaxVbj8C.

[86] Ashish Vaswani et al. *Attention Is All You Need*. 2nd Aug. 2023. DOI: 10.48550/arXiv.1706.03762. arXiv: 1706.03762 [cs]. URL: http://arxiv.org/abs/1706.03762. Pre-published.

[87] Pham Hoang Vuong et al. 'A Bibliometric Literature Review of Stock Price Forecasting: From Statistical Model to Deep Learning Approach'. In: *Science Progress* 107.1 (15th Mar. 2024), p. 00368504241236557. ISSN: 0036-8504. DOI: 10.1177/00368504241236557. pmid: 38490223. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC10943735/.

[88] Jonghyeon Won, Hyun-Suk Lee and Jang-Won Lee. 'A Review on Multi-Fidelity Hyperparameter Optimization in Machine Learning'. In: *ICT Express* 11.2 (1st Apr. 2025), pp. 245–257. ISSN: 2405-9595. DOI: 10.1016/j.icte.2025.02.008. URL: https://www.sciencedirect.com/science/article/pii/S2405959525000244.

[89] Pengxiang Xia. 'Investigating COVID-19 Influence on the VIX Index through ARIMA Model Predictions.' In: *Highlights in Business, Economics and Management* 36 (17th July 2024), pp. 328–334. ISSN: 2957-952X. DOI: 10.54097/eapfr142. URL: https://drpress.org/ojs/index.php/HBEM/article/view/23601.