A black rectangle with a black background

Description automatically generated with low confidenceLogo, company name

Description automatically generated

DAEN 690

Project Report

Jashwanth Raj Gowlikar

Tharuni Myda

Ratan Rakesh Maddi

Venkata Krishna Reddy

Spring 2024

Multivariate Timeseries Forecasting with Generative AI

**About the Cover**

Professor Berlin is an instructor at the George Mason University College of Engineering and Computing, Volgenau School of Engineering, MS Data Analytics Engineering (DAEN) program. He began working with the DAEN program as an adjunct faculty member in 2012 and became a fulltime faculty member in 2016. He is a passionate contributor to the program and a devoted mentor to his students.

His passion for new value creation is built on over 50 years of professional experience – innovating and advocating for innovators applying leading-edge digital solutions to mission challenges. He has served with outstanding teams in various roles, including senior strategy executive, consultant, and mentor; applied information and systems technologist; collaborative leader; computer scientist, and public policy entrepreneur.

He serves as a strategy advisor and mentor to public and private sector innovators and entrepreneurs and as a public speaker (emerging challenges, innovation opportunities, and ethics). His core interests include public policy, high-performance computing, cyber, emerging big data, health informatics, and digital economy and governance challenges.

In addition to teaching and mentoring, Professor Berlin seeks new engagements with high-quality, core-value-centered innovation teams – collaborating to address societal and market challenges with cyber-physical and policy innovation. Specifically, sustainable solutions can be delivered at the intersection of innovative value creation, human aspiration, and strategic vision.

Professor Berlin is a graduate of the US Air Force Academy with a Bachelor of Science in Computer Science and Mathematics as well as a graduate of the University of Texas at Austin with a Master of Arts in Computer Science. He is also attended the USAF Air Command and Staff College for leadership and strategy training.

Contents

Table of Contents

[Abstract 1](#_Toc165318565)

[Problem Definition 3](#_Toc165318566)

[1.1 Background 3](#_Toc165318567)

[1.1.1 Introduction to forecasting 3](#_Toc165318568)

[1.1.2 Limitations of traditional forecasting methods 3](#_Toc165318569)

[1.1.3 Introduction to time series analysis 4](#_Toc165318570)

[1.1.4 Evolution of machine learning in forecasting 4](#_Toc165318571)

[1.1.5 Generative AI and its role in forecasting 4](#_Toc165318572)

[1.1.6 Detailed explanation of multivariate time series 5](#_Toc165318573)

[1.1.7 Advanced forecasting techniques 5](#_Toc165318574)

[1.1.8 Practical applications in various industries 5](#_Toc165318575)

[1.1.9 Ethical considerations and challenges 6](#_Toc165318576)

[1.1.10 Future of forecasting with AI 6](#_Toc165318577)

[1.1.11 Conclusion and project justification 6](#_Toc165318578)

[1.2 Problem space 7](#_Toc165318579)

[1.3 Research 8](#_Toc165318580)

[Solution space 10](#_Toc165318581)

[1.4 Project Objectives 10](#_Toc165318582)

[Primary User Stories 10](#_Toc165318583)

[Product Vision 11](#_Toc165318584)

[Scenario #1 11](#_Toc165318585)

[Scenario #2 11](#_Toc165318586)

[Section 2: Datasets 11](#_Toc165318587)

[2.1 Overview 11](#_Toc165318588)

[2.2 Field Descriptions 12](#_Toc165318589)

[2.3 Data Context 12](#_Toc165318590)

[2.4 Data Conditioning 12](#_Toc165318591)

[2.5 Seasonality and Stationarity in Time Series Forecasting 13](#_Toc165318592)

[2.5.1 Fourier Analysis 13](#_Toc165318593)

[2.5.2 Augmented Dickey-Fuller Test 14](#_Toc165318594)

[2.5.1 Kwiatkowski-Phillips-Schmidt-Shin 14](#_Toc165318595)

[2.5.2 Synthesis of Findings 14](#_Toc165318596)

[2.6 Data Quality Assessment 15](#_Toc165318597)

[2.7 Other Data Sources 15](#_Toc165318598)

[2.8 Storage Medium 16](#_Toc165318599)

[2.9 Storage Security 16](#_Toc165318600)

[2.10 Storage Costs 16](#_Toc165318601)

[Section 3: Algorithms & Analysis / ML Model Exploration & Selection 17](#_Toc165318602)

[3.1 Solution Approach 17](#_Toc165318603)

[3.1.1 Systems Architecture 17](#_Toc165318604)

[Systems Security 18](#_Toc165318605)

[3.1.2 Systems Data Flows 18](#_Toc165318606)

[Algorithms & Analysis 19](#_Toc165318607)

[3.2 Machine Learning 19](#_Toc165318608)

[**Model Exploration** 20](#_Toc165318609)

[Model Selection 23](#_Toc165318610)

[Section 4: Visualizations / ML Model Training, Evaluation, & Validation 23](#_Toc165318611)

[Overview 23](#_Toc165318612)

[4.1 Visualizations 23](#_Toc165318613)

[Machine Learning 25](#_Toc165318614)

[4.1.1 Hyper parameter tuning 25](#_Toc165318615)

[Model Training 26](#_Toc165318616)

[Model Evaluation 27](#_Toc165318617)

[Model Validation 27](#_Toc165318618)

[Section 5: Findings 30](#_Toc165318619)

[Section 6: Summary 31](#_Toc165318620)

[Section 7: Future Work 31](#_Toc165318621)

[Appendix A: Glossary 33](#_Toc165318622)

[Appendix B: GitHub Repository 34](#_Toc165318623)

[Overview 34](#_Toc165318624)

[GitHub Repository Link 34](#_Toc165318625)

[GitHub Repository Contents 34](#_Toc165318626)

[Appendix C: Risks 34](#_Toc165318627)

[Sprint 1 Risks 34](#_Toc165318628)

[Sprint 2 Risks 35](#_Toc165318629)

[Sprint 3 Risks 35](#_Toc165318630)

[Sprint 4 Risks 36](#_Toc165318631)

[Appendix D: Agile Development 37](#_Toc165318632)

[Scrum Methodology 37](#_Toc165318633)

[Sprint 1 Analysis 37](#_Toc165318634)

[Sprint 2 Analysis 37](#_Toc165318635)

[Sprint 3 Analysis 37](#_Toc165318636)

[Sprint 4 Analysis 37](#_Toc165318637)

[Sprint 5 Analysis 38](#_Toc165318638)

Table of Figures

[Figure 1: Fourier Analysis of ADANIENT's Stock Price Data: Visualizing Frequency Domain Characteristics 14](#_Toc165330013)

[Figure 2: System architecture 19](#_Toc165330014)

[Figure 3: A schematic system of data flows, from the input to the output. 19](#_Toc165330015)

[Figure 4: Histogram of close price values 24](#_Toc165330016)

[Figure 5: Normal Q-Q visualization 25](#_Toc165330017)

[Figure 6: Histogram of closing price values after standard scaling 25](#_Toc165330018)

[Figure 7: Normal Q-Q plot for the Normalized data 26](#_Toc165330019)

[Figure 8: The influence of different hyperparameters on the performance of the model 27](#_Toc165330020)

[Figure 9: The objective value (validation loss) along different hyper parameter investigations (Trials) 27](#_Toc165330021)

[Figure 10: Results of the model prediction after 50 epochs 29](#_Toc165330022)

[Figure 11: Results and loss graphs of the MSE and MAE calculations after 50 epochs 30](#_Toc165330023)

[Table : Glossary Table 33](#_Toc165330024)

[Table : Sprint 1 Risks 34](#_Toc165330025)

[Table : Sprint 2 Risks 35](#_Toc165330026)

[Table : Sprint 3 Risks 35](#_Toc165330027)

[Table : Sprint 4 Risks 36](#_Toc165330028)

[Table : Sprint 5 Risks 36](#_Toc165330029)

[Table : Sprint project dates 37](#_Toc165330030)

**This page intentionally left blank**

Abstract

Abstract

Accurate forecasting of multivariate time series data is crucial across numerous industries, from finance to healthcare, enabling informed decision-making and strategic planning, where multiple interdependent factors influence future outcomes. However, traditional forecasting models, ranging from basic regression techniques to more advanced methods like LSTM-RNNs, exhibit significant limitations in analyzing complex, high-dimensional data with intricate variable dependencies. These models are often constrained by their reliance on extensive training data volumes and their predominant focus on capturing linear relationships, failing to uncover the intricate non-linear patterns present in real-world multivariate time series. To overcome these limitations, this research explores the adaptation of advanced deep learning architectures, specifically transformer models, for multivariate time series forecasting. Transformer models leverage self-attention mechanisms to effectively process temporal sequences while accounting for multiple influential variables, enabling the capture of long-range dependencies and non-linear relationships with reduced data requirements. By employing the Patch Time series Transformer, tailored for time series tasks, we demonstrate the potential of this approach on a comprehensive stock market dataset spanning 100 companies with high-resolution 1-minute data over 7 years. Our findings showcase the transformer model's capability to generate accurate future predictions by accounting for the interdependencies across multiple variables, outperforming traditional techniques. This research underscores the transformative impact of cutting-edge deep learning architectures in facilitating precise multivariate forecasting, crucial for informed decision-making and strategic planning across business operations. Future work could investigate incorporating real-time streaming data as well as data fusion techniques to enable dynamic updating of the transformer model, resulting in more timely predictions.

**This page intentionally left blank**

Report

# Problem Definition

## Background

### Introduction to forecasting

The terms forecasting and prediction have conflicting definitions in different resources. Forecasting is defined as the systematic process of estimating future events based on analysis of trends and data, while prediction refers to the act of declaring a possible outcome under specific conditions. The predictability of an event is determined by the event itself, regardless of our ability to predict it [1]. In Forecasting Fundamentals, Sanders explains that forecasting is the process of making predictions about the future [2]. It involves predicting future events or trends based on historical data, current information, and statistical methods. The significance of forecasting lies in its ability to guide decision-making processes, enabling organizations and individuals to prepare and strategize effectively for the future.

Forecasting is an essential practice across numerous industries sectors, ranging from economics to meteorology. Chance is central to forecasting in all disciplines and reflects the obvious features that forecasts can, and often do, differ from outcomes [1].

The challenge is in being accurate. It has been shown that following a process that has been well established will help ensure forecasts with better accuracy [2]. Traditional forecasting methods are grounded in statistical techniques and historical data analysis. These methods include time series analysis, causal models, and qualitative techniques like expert judgment or market research. The forecaster’s job is difficult enough but is made all the harder by data being revised, so that we may only learn the correct values of today’s key macroeconomic variables at the time: a bad starting point will usually worsen the forecast accuracy [1].

### Limitations of traditional forecasting methods

Despite their widespread use, traditional forecasting methods have inherent limitations. A primary constraint is their reliance on historical data. These methods often assume that past patterns will continue, which may not always hold, especially in rapidly changing environments. The FasterCapital blog continues by pointing out the flaws with various computations of the moving average, exponential smoothing, and regression analysis, missing out on the delicate patterns and assumptions of a linear relationships [3].

These methods may not account adequately for sudden, unforeseen events, such as economic shocks or natural disasters. Such events can drastically alter trends and render predictions based on historical data inaccurate. Market conditions, legal and/or regulation changes are also overlooked external factors according to the FasterCapital blog [3].

The FasterCapital blog also explains the usage of the Delphi Method which relies on the experts and judgment of panel members. This type of forecasting is quite subjective.

There is a more complex approach called Activity Driver Analysis (ADA). This approach has advantages because it explores the cause-and-effect relationship within primarily financial data. However, according to FasterCapital blog, ADA too has its shortcomings [3]. If data is incomplete or inaccurate, the analysis may not be as accurate. Moreover, this type of analysis requires a high level of statistical analysis. Not every company has the personal resources or software to see this level of analysis through.

Particularly in newly emerging fields or capital ventured companies where there is brief history, there is a lack of data. Again, most traditional methods require historical data, preferably a significant amount of data.

### Introduction to time series analysis

Time series analysis is a cornerstone of traditional forecasting methods. It involves analyzing a series of data points, typically measured at successive points in time, to identify patterns and predict future values. Time series data is ubiquitous, found in daily stock market prices, monthly sales figures, yearly climate data, and more.

There are two main types of time series: univariate and multivariate. A univariate time series involves a single data sequence, such as daily temperatures or monthly sales of a product. In contrast, a multivariate time series consists of multiple data sequences that are often interrelated, such as the various economic indicators that impact stock market performance.

The analysis of time series is vital because it allows for the understanding and forecasting of temporal patterns and trends. It involves methods like moving averages, exponential smoothing, and Auto Regressive Integrated Moving Average (ARIMA) models. These techniques help in smoothing out short-term fluctuations and highlighting longer-term trends or cycles that are often observed in volatile data sources.

In univariate analysis, the focus is on understanding and forecasting based on the single observed series. This type of analysis is prevalent in situations where the interest lies in predicting the future values of a specific variable based on its past values.

Multivariate time series analysis, however, deals with understanding and forecasting based on the interrelationships among multiple variables. It is more complex but provides a more comprehensive view, essential in scenarios where variables influence each other. For instance, in financial markets, stock prices might be influenced by various factors like interest rates, inflation, and GDP growth.

### Evolution of machine learning in forecasting

The landscape of forecasting has been revolutionized by Artificial Intelligence (AI) and Machine Learning (ML). ML's introduction marked a paradigm shift from traditional statistical methods to more dynamic and adaptable approaches. Early ML applications in forecasting were rudimentary, primarily focusing on pattern recognition in datasets. However, as ML technology advanced, its capabilities expanded, allowing for the analysis of complex and large datasets with greater accuracy and speed.

ML algorithms can identify non-linear patterns and relationships within data that traditional methods might overlook. These models learn from the data, continuously improving their accuracy with more input. This adaptive nature of ML makes it highly effective in environments where data patterns and relationships are complex and continually evolving.

### Generative AI and its role in forecasting

Generative AI, a subset of artificial intelligence, has emerged as a powerful tool in forecasting. Unlike traditional AI models that primarily focus on analyzing and classifying data, Generative AI specializes in creating new, synthetic data samples. These models, such as Generative Adversarial Networks (GANs), learn to generate statistically similar data, enabling them to predict future outcomes by extrapolating from existing patterns.

In forecasting, Generative AI's ability to simulate potential future scenarios and outcomes is invaluable. It can, for instance, generate simulated financial market scenarios based on historical data, allowing analysts to assess potential future market movements under different conditions. This capability is not just limited to financial forecasting but extends to other areas, like energy demand prediction and weather forecasting, where understanding a range of possible future scenarios is crucial.

### Detailed explanation of multivariate time series

Multivariate time series analysis is an advanced area of forecasting that deals with multiple interrelated variables over time. Unlike univariate time series, which analyzes a single variable, multivariate analysis considers the dynamics and interactions between several variables. According to Analytics Vidhya, each variable depends not only on its past values but also has some dependency on other variables [4]. This approach is crucial for understanding complex systems where multiple factors influence the outcome.

For instance, in financial forecasting, variables such as stock prices, interest rates, inflation rates, and economic indicators are interdependent, and their collective analysis provides a more accurate prediction than considering them in isolation. Similarly, in weather forecasting, variables like temperature, humidity, pressure, and wind speed interact with each other, and their joint analysis is necessary for accurate predictions. Even the federal government is exploring this new technology. The Defense Department is integrating multiple generative AI models into its series of global exercises that are intended to test out capabilities that could support the U.S. military’s Joint All-Domain Command and Control warfighting construct, according to a senior official, as reported by Jon Harper in the July 2023 episode of the Defense Scoop [5].

### Advanced forecasting techniques

Machine learning models like Random Forests and Neural Networks, which can process and analyze large datasets with complex relationships, can capture non-linear interactions between variables, which is a limitation of traditional statistical methods. In addition, deep learning techniques, especially Recurrent Neural Networks (RNNs) and Long Short-Term Memory networks (LSTMs), have been instrumental in predicting time series data. These models are adept at handling sequences of data, making them ideal for forecasting tasks where past data points significantly influence future ones. Although each of these categories has its pros and cons, the advent of advanced forecasting techniques, and leveraging generative models addressed some of the main issues, such as dependency on enormous datasets. A comprehensive literature review is presented in section 1.3 to better elaborate on the effectiveness of these methods.

### Practical applications in various industries

The application of advanced forecasting methods spans across various industries, showcasing their versatility and impact. In the financial sector, multivariate time series forecasting plays a crucial role in predicting stock market trends and aiding investors and policymakers in making informed decisions. Advanced algorithms can evaluate multiple factors at the same time, such as firm performance indicators and economic forecasts, resulting in more accurate market predictions than earlier models.

In meteorology, these forecasting techniques are instrumental in predicting weather patterns and climate changes. For example, the WRF-Fire model predicts wildfire spread using real-time topographic, meteorological, and satellite data, improving the efficiency of evacuation plans and resource allocation. By analyzing data from multiple sources, such as satellite imagery and atmospheric data, meteorologists can provide more precise weather forecasts, which are vital for agriculture, disaster management, and daily life. Since the late 1990’s, a multi-agency Weather Research and Forecasting (WRF) has been dedicated to the science of weather forecasting. Traditional models look at historical events. However, historical events are not always repeated. As noted in AMS Journals, historical trends are not always a good indicator of future performance [6]. The public domain numerical weather prediction model has various components and contributors. Even within the weather forecast modeling there are specialized areas. A wildland fire-behavior module, named WRF-Fire, was integrated into the Weather Research and Forecasting (WRF) public domain numerical weather prediction model [7].

The healthcare sector also benefits from advanced forecasting. One notable example is the application of AI as a diagnostic tool, which outperforms older methods such as the Modified Early Warning Score. AI systems have been trained to monitor electronic health records and can detect early warning indications of health problems, such as sepsis, allowing for fast medical intervention. Predictive models can analyze patient data, disease trends, and healthcare resource utilization, aiding in anticipating outbreaks, managing hospital resources, and improving patient care outcomes [8]. In Machine Learning and AI for Healthcare, a case study for image analysis is presented concerning diabetic patients, which describes the challenges of machine learning with images. The goal of the model was to predict the likelihood of complications related to diabetes within the extremities of the foot [8]. Additional case studies are reviewed in the textbook, including an assessment of a predictive AI model for personalized care such as behavioral patterns, including sleeping and bathroom routines utilizing sensors in the home to record activity. The goal of the modeling is to give context to caretakers, predicting where a resident “should” be and compare this to where they are for a given snip in time. This is complemented by the monitoring of body vitals to alert a caregiver if the situation seems of concern [8].

### Ethical considerations and challenges

The increasing reliance on AI-based forecasting models brings to the forefront various ethical considerations and challenges. One significant concern is the potential for inherent biases in these models. If the training data is biased or unrepresentative, the forecasts can perpetuate or amplify these biases, leading to unfair or unethical outcomes.

Privacy concerns are another critical issue. Forecasting models often require vast amounts of data, which may include sensitive personal information. Ensuring the privacy and security of this data is paramount to maintaining trust and complying with regulatory standards.

There is also the challenge of transparency and explainability. AI models, especially those based on complex algorithms like neural networks, can be opaque, making it difficult to understand how they arrive at certain predictions. This "black box" nature can pose challenges in sectors where understanding the reasoning behind predictions is crucial.

### Future of forecasting with AI

The future of forecasting with AI is poised for significant growth and innovation. Emerging trends include the integration of AI with big data analytics and the Internet of Things (IoT), which could further enhance the accuracy and scope of forecasting models. Machine learning algorithms are continually advancing, becoming more sophisticated and competent in handling more complex data sets.

Continuing research in this field focuses on developing more robust, transparent, and ethical AI models. Efforts are also being made to improve the interpretability of these models, making it easier for users to understand and trust their predictions.

The potential for AI in forecasting extends to virtually every sector, from predicting consumer behavior in retail to anticipating cybersecurity threats. As technology advances, these models will probably become an even more integral part of decision-making processes in various industries.

### Conclusion and project justification

In conclusion, the advancement of forecasting methods, particularly through usage of the Natural Language Processer of Transformers in generative AI for a multivariate time series analysis, represents a significant leap in our ability to predict and prepare for future events. The limitations of traditional forecasting methods, such as dependency on large amount of data, limited adaptability to different scenarios, limitations with memory storage in machine learning models, and lower accuracy.

There are multiple algorithms used for predicting values with multiple variables involved. Long Short-Term Memory or LSTM uses massive data sets which take a great deal of resources in computing power and can be costly. This approach is taken with Reoccurring Neural Networks (RNN), because on its own there is a limitation of memory. Combining RNN with LSTM extends the memory, but this requires more learning to process time, according to instructional on-line coursework from MIT. Natural Language Processing Transformers modeling offers these qualities, capable of handling complex, large-scale data and providing more nuanced predictions within out the limitations of memory or significant processing time.

This project is justified by the need for more advanced forecasting methods that can adapt to the complexities and dynamics of modern datasets. The potential benefits of these methods span across numerous industries, enhancing decision-making processes, and contributing to more efficient and effective planning and strategy. As such, the continued development and application of AI in forecasting are not only beneficial, but essential for future advancements in various sectors.

## Problem space

Across diverse sectors, there is a critical need for reliable multivariate forecasting to support the decision-making process. Whether it's deciding to carry an umbrella based on weather predictions, allocating resources for municipal services, or planning financial budgets in corporate settings, the ability to accurately forecast outcomes based on a multitude of variables is essential. In the modern world, there is a diversity of issues that affect the accuracy of future predictions even when cutting-edge technologies have been used. A significant issue is the dynamic and the complicated relationships between various variables in the data. It's challenging to manage risks, plan finances, and make wise investment decisions in the financial industry because of the complex relationships between factors like market conditions, economic indicators, etc. A major challenge in multivariate time series forecasting is the size and quality of data being reviewed, as memory and speed are obstacles to modeling.

There are major issues related to the nature of the data that make using LSTM (Long Short-Term Memory) models for multivariate time series forecasting challenging. The fact that the datasets are small and fragmented, and have large dimensions, is one of the main challenges. Because LSTM models are built to function best with large volumes of data, they can discover complex relationships and patterns in the given data. These models, however, find it difficult to generalize and represent the underlying complexities when datasets are small.

Moreover, the high dimensionality datasets introduce even another level of complexity. Many factors or characteristics are included in high-dimensional data, and each influences the way the time series behaves. Large-scale multidimensional datasets are challenging for LSTM models to interpret and learn from because they are designed for sequential data. In the multivariate environment, this may lead to low performance and an inability to identify significant patterns.

In the Finance sector, time series forecasting is critical because it directly affects investment decisions, risk management, and financial planning. Predicting economic indicators, market trends, and currency changes allows for well-informed decision-making, minimizing financial losses and maximizing investment strategies. Encouraging economic growth, safeguarding investor assets, and preserving market stability all depend on finding a solution to the forecasting dilemma in finance.

The critical need for patient care and resource allocation in the healthcare sector pushes the need to have a multivariate time series forecasting model. Healthcare workers can plan for the changing healthcare needs of communities, foresee illness outbreaks, and efficiently distribute medical resources when they have accurate forecasts. By guaranteeing prompt and suitable responses to additional health concerns, improving forecasting capacities in healthcare lowers healthcare costs, improves the overall effectiveness of medical interventions, and ultimately saves lives.

In the environment sector, the purpose of creating efficient policies, reducing the impact of natural disasters, and guaranteeing the preservation of ecosystems, accurate forecasts of environmental variables, such as temperature, precipitation, and biodiversity, are essential. Overcoming these forecasting obstacles helps ensure that natural resources are managed responsibly, biodiversity is protected, and sustainable practices are developed to counteract the negative effects of climate change.

Inaccurate forecasting models can lead to poor decision-making due to the unpredictability of the future and the limitations of current prediction algorithms. In real-time applications, the shortcomings of existing methods become evident, and forecasting model accuracy becomes crucial for making informed decisions [[10].](https://blog.research.google/2023/04/recent-advances-in-deep-long-horizon.html) A company's ability to make decisions effectively and succeed overall in dynamic environments depends on its ability to navigate the challenging terrain of multiple interdependent variable predictions across various domains.

The need for precise decision-making, and the effectiveness of prediction tools, emphasize how important it is to understand the problem at hand. By considering these factors, we can create solutions that effectively tackle the challenges of forecasting numerous interdependent variables across different domains.

## Research

Multivariate time series prediction, a method pivotal for forecasting future values based on multiple time-dependent variables, demands a comprehensive approach. Its significance is particularly pronounced in sectors where interdependent variables critically influence decision-making. For instance, financial markets hinge on the accurate prediction of stock prices, which in turn rely on a complex interplay of variables like market trends, economic indicators, and company performance metrics. In meteorology, variables such as temperature, humidity, and wind speed intertwine to enable precise weather forecasting.

Generative AI has demonstrated remarkable potential in various fields. In transportation and AIOps, ScoreGrad [9], a continuous energy-based generative model, has been developed for multivariate probabilistic time series forecasting, proving particularly effective in transportation applications. It outperforms other models on various real-world datasets, underscoring its applicability in intelligent transportation systems. CDX-NET [10], a deep neural network, addresses aperiodic MTS in AIOps. It integrates Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), and attention mechanisms, demonstrating significant performance improvement in a real-world AIOps dataset. MARINA [11], a Multi-Layer Perceptron (MLP)-attention based model, has been proposed for multivariate time series analysis. It is versatile, suitable for forecasting and anomaly detection, and shows state-of-the-art performance in AIOps. A recent study, an innovative approach in long-term forecasting for multivariate time series, utilizes Seasonal-Trend decomposition based on Loess (STL), adeptly handling complex data by decomposing it into trend, seasonal, and residual components. This methodology underscores the importance of advanced decomposition techniques in managing the intricacies of interdependent variables in multivariate time series prediction, offering a nuanced perspective on forecasting challenge [12].

In anomaly detection and maintenance within industrial multivariate time series data, Generative Adversarial Networks (GAN)s, Long Short-Term Memory (LSTM), and Variational Auto Encoders (VAE)s, have shown remarkable capabilities. Cross-Correlation Graph-Based Encoder–Decoder GAN framework was proposed and focuses on anomaly detection and localization in industrial settings, relevant to transportation and AIOps [13]. It effectively captures time-dependent and correlational features of multivariate time series. Additionally, an integrated generative model with bidirectional LSTM is proposed which intelligently detect anomalies in industrial Internet of Things (IIoT), in which captures time-series dependency and improves detection accuracy of industrial multidimensional time-series anomaly [14]. In addition, a GAN-based anomaly detection for power plants detection and localizing anomalies in multivariate time series data is presented, transforming data into 2D images to exploit encoder and decoder structures [15]. Also, in another work, stacking VAE with graph neural networks leverages a method for time-series anomaly detection [16]. It captures stable interrelation structures among multiple time series data channels, enhancing interpretability and detection accuracy. Furthermore, a multiscale wavelet graph autoencoder focusing on fault diagnosis and root cause discovery was proposed which integrates discrete wavelet transform into autoencoders, decomposing multivariate time series into multifrequency components [17]. It captures anomalies caused by changes in intervariable dependence, demonstrating effectiveness in industrial settings.

In healthcare, particularly in the synthesis and analysis of multivariate medical time series data, techniques like GANs and neural processes are enabling the creation of synthetic data and providing new ways to interpret complex medical datasets, paving the way for advanced research and privacy-preserving solutions. GAN has been employed to generate synthetic multichannel electrocardiogram signals [18]. This approach addresses privacy concerns in medical data sharing by creating high-quality synthetic time series data that can effectively substitute actual patient data while maintaining privacy. Additionally, heatmaps have been used for visual explainability of CNNs in multivariate time series problems in healthcare [19]. This method, applied to predicting the risk of in-hospital mortality, provides rational insights conforming to medical knowledge. A study highlights the potential of synthetic patient data generated by generative AI, like GANs, to revolutionize clinical research and protect patient privacy [20]. The paper discusses key uses of synthetic data in clinical research, data privacy, and medical education, along with ethical and practical concerns. Neural processes have been utilized to estimate missing values in clinical time-series data [21]. This approach adapts to the dynamics of available clinical data, demonstrating effectiveness compared to conventional methods. In another work, the use of GANs for generating realistic medical time series data is investigated, focusing on the generation of synthetic multichannel ECG signals [22]. The approach demonstrates the GAN's ability to create structurally similar and diverse data while ensuring privacy, by not relying on the real patients’ personal data.

One of the fields that received high interest in this regard is financial markets and stock prediction, leveraging the complexities of multivariate time series data. A framework that includes the temporal generative filters that implement a memory-based mechanism onto an LSTM network in an attempt to learn individual patterns per stock and hypergraph attentions for stock movement prediction, focusing on both multi-order and internal dynamics of stocks is presented [23]. Additionally, Score Grad, a multivariate probabilistic forecasting framework based on continuous energy-based generative models, applying to financial time series [9] is presented in the literature. Additionally, discussing generative models for augmenting financial datasets, focusing on synthesizing realistic time series that capture specific properties of financial data [24]. MMGAN-HPA for stock market prices prediction is a deep learning framework based on GANs with LSTM and CNN for improved stock market price prediction by overcoming challenges of hyperparameter tuning for GAN-based stock market prediction, with reinforcement learning and Bayesian optimization [25]. In another study, features selected by multiple techniques (logistic regression, support vector machine, and random forest) are combined and used in a deep generative model for better stock market forecasting performance [26]. Additionally, financial time series prediction with GGM-GAN integrates sparse Gaussian graph model information into GAN for forecasting stock prices in the Chinese A-share market [27]. In another study, ST-GAN for predicting stock trends is proposed which is a deep learning model that analyzes financial news texts and numerical data to predict stock trends using GAN [28]. Additionally, addresses existing GAN limitations in stock market applications, such as mode collapse and limited one-step prediction, by integrating financial domain knowledge, news context, and a multi-step attentive seq2seq learning network, Index GAN is proposed [29]. This model, which operates within a Wasserstein GAN framework, uniquely incorporates market sentiment and news analysis, reflecting the stock market's inherent characteristics and demonstrating enhanced predictive performance in real-world indices. Another innovative approach, Fin-GAN focuses on probabilistic forecasting and introduces a novel economics-driven loss function, repositioning GANs for classification tasks and enabling comprehensive conditional probability distributions for price returns [30]. This approach, validated through numerical experiments on equity data, achieves higher Sharpe Ratios compared to traditional models like LSTMs and ARIMA, offering nuanced and probabilistic financial forecasting [31]. In this study, Koochali et al. introduce ProbCast, a cutting-edge probabilistic model for multivariate time-series forecasting. Leveraging a Conditional Generative Adversarial Network framework, Precast is trained through adversarial training, aiming to learn the probability distribution of future values conditioned on historical data. Additionally, the researchers propose a unique framework that transforms deterministic models into superior probabilistic counterparts. The framework streamlines the GAN architecture search process, enhancing stability and efficiency. Experimental validation on two public datasets, electricity consumption, and exchange rates, demonstrates Podcast's outstanding performance, validating the effectiveness of the proposed methodology.

## Solution space

**Transformers for Multivariate Time Series Forecasting:**

The goal of this project is to develop a time series forecasting system that can accurately predict future stock prices and market trends. The solution will analyze historical price data, technical indicators, and related time series to output multi-step forecasts.

Our solution employs Transformer neural network architectures for multivariate time series forecasting on stock market data. The input data contains historical OHLCV prices, volumes, and technical indicators for major Indian stocks at a 1-minute. We want to use the self-attention mechanism in Transformers to model both long-term dependencies as well as local patterns in the multivariate input features. The data requires preprocessing, such as handling missing values, normalizing features, and engineering derived indicators. We will then feed the preprocessed data into Transformer models optimized through hyperparameter tuning. The models will be trained to predict future stock prices and indicator values in a multi-step forecasting setup. Evaluation metrics like RMSE, MAPE, and directional accuracy will guide model optimization. The end-to-end system ingests raw stock data, prepares it for modeling, trains deep Transformer networks, and outputs multi-step ahead forecasts for key stock market variables.

Research by Nie and Nguyen [29] showed that a specific type of Transformer called a PatchTST proved to be an effective means of long-term forecasting. PatchTST is short for Patch Time Series Transformers. According to the Towards AI blog, the transformer backbone of the model incorporates patching plus two versions of machine learning, supervised and self-supervised [32]. Patches have various qualities, such as overlapping and varying in length and stride. The stride represents the number of time steps that separate the starting points of consecutive patches [32]. The blog goes on to explain the tokenization process gives PatchTST its advantage. Each patch serves as an input token. This tokenization process lessens the number of tokens processed. The tokens are processed through an encoder which are subject to a self-supervised learning process. All these advantages yield superior performance to other transformers, such as N-BEATS and N-HITS [32].

## Project Objectives

Upon completion, the team will have a thorough understanding of how generative artificial intelligence (GenAI) is used specifically for time series. The solution will be a system that provides a faster and more accurate ability to predict future values given by multiple variables. The team brings value with time and effort in exploration into various machine learning algorithm/s. This effort includes reading up on research efforts already made on this topic, data selection, evaluation, and review, along with data model review and training.

## Primary User Stories

Where data analysis is a cornerstone, the need for an AI-driven platform becomes paramount. Such a platform should be adept and employ generative models to analyze historical multivariate time series data across various domains. Its primary utility lies in predictive analysis, adaptable to an array of industries, including finance, healthcare, and energy, among others.

Focusing on the financial sector as an example, this system should encompass comprehensive tools for delving into historical stock market data, thereby enabling analysis to forecast market trends with a high degree of accuracy. Internal and external stakeholders are interested in a dataset that includes financial indicators to be processed with a sophisticated system. Risk analysis can be assessed quickly and accurately via an AI-driven platform. Providing portfolio analysis for financial professionals and customers with self-managed accounts, this system will be highly valued.

There are other applications to a system that predict values in a multivariate time series. This platform is envisioned as a tool that equips professionals in diverse fields with advanced AI techniques. By enhancing decision-making, refining trend analysis, and improving forecasting capabilities, it promises to boost both efficiency and accuracy across various domains.

## Product Vision

### Scenario #1

**FOR** stakeholders, **WHO** require robust management and precise forecasting of energy production, consumption, and demand, **THE** Forecast AI **IS A** state-of-the-art multivariate time series forecasting tool powered by generative AI. **THAT** delivers highly accurate forecasts for both renewable energy generation and usage, thereby facilitating more efficient grid operations. **UNLIKE** traditional forecasting approaches that may lack capability to effectively process multiple variables simultaneously, **OUR PRODUCT** provides sophisticated analytics with real-time data processing, which inherently enhances decision-making processes. The potential **CAVEATS** include that the precision of forecasts may be influenced by the caliber of the input data and sudden shifts in weather conditions or other uncontrollable external elements.

### Scenario #2

**FOR** decision-makers and strategists, **WHO** seek to enhance their investment strategies and asset management, **THE** Market AI **IS A** cutting-edge stock market analysis and prediction tool powered by deep learning. **THAT** provides real-time insights and predictive analytics for stock performance, helping the investors to make informed decisions on buying, holding, or selling their assets. **UNLIKE** traditional analysis tools that depend on historical data and conventional statistical methods, **OUR PRODUCT** incorporates a diverse array of market indicators and news sentiment analysis to provide a more vibrant and proactive outlook. The potential **CAVEATS** include the inherent volatility of the stock market, regulatory changes, and the unpredictability of global economic events (including the political conflicts across the globe) which can all impact the accuracy of the forecasts.

# Datasets

## Overview

The dataset includes complete historical daily price records for the Nifty 100 stocks - the top 100 companies on the National Stock Exchange of India defined by market capitalization. Along with specific stock data, the collection encompasses complete information for the Nifty 50 and Nifty Bank, two crucial indices in the Indian financial markets. The Nifty Bank index comprises the top 12 liquid and highly capitalized banking stocks, while the Nifty 50 is a diversified 50-stock index that covers 13 economic sectors. With price and indicator values recorded at 1-minute intervals, the data has an exceptionally fine level of detail, offering an extensive overview of market trends and fluctuations.

The data's time range starts in February 2022 and provides a useful and relevant overview of the state of the market as of lately. The original size of the dataset is approximately 33 gigabytes. With an effective compression technique, the dataset has been reduced to 13 gigabytes without sacrificing accuracy, allowing for more efficient handling and storage. The OHLCV metrics - opening price (Open), highest price (High), lowest price (Low), closing price (Close), and trading volume (Volume)—are provided by the dataset for every stock and index. Besides offering a framework to evaluate market sentiment and anticipated price movements, these fundamental market indicators are essential for technical analysis.

55 calculated technical indicators, including moving averages, Bollinger Bands, the Relative Strength Index (RSI), the Moving Average Convergence Divergence (MACD), and others, have been included to the dataset. These indicators are essential resources for traders and analysts to forecast future market behavior. These indicators provide buying and selling signals, indicate out trends, and measure volatility and market momentum.

## Field Descriptions

Our dataset provides an extensive collection of financial data points for the well-known Nifty 100, Nifty 50, and Nifty Bank indices—all important gauges of the health of the Indian stock market. It captures the Open, High, Low, Close, and Volume (OHLCV) information at one-minute intervals for every listed entity. This demonstrates the highs and lows of the market from January 2015 to February 2022 in a detailed tapestry. We also have 55 advanced technical indicators to further enhance analytical usage. These include a wide range of indicators, such as oscillators that aid in identifying momentum and moving averages that smooth price movement. With their varied perspectives on the vitality of market conditions and trends, each one is a vital component of the market analysis.

## Data Context

Our selected time series dataset includes detailed real-time pricing and other technical indicators for the top 100 companies on the National Stock Exchange of India's NIFTY 50 and NIFTY Bank indexes. This dataset, acquired from Kaggle and released in February 2022, is particularly beneficial for people interested in stock price analysis or forecasting.

The dataset, currently covers January 2015 to February 2022, retrieves stock values at 1-minute intervals throughout the trading day, detailing every twist and turn in the market. This level of granularity provides an almost real-time view of market moves that daily reports cannot match. It includes not only the standard trading volumes and open, high, low, and close price data, but also 55 pre-calculated technical indicators such as the Relative Strength Index (RSI), Moving Average Convergence Divergence (MACD), Bollinger Bands, and various moving averages. These indicators are ready to use, liberating analysts from the strain of complicated calculations.

While the dataset's breadth is large, it is limited to stocks listed on the NIFTY indexes, providing an overview rather than an in-depth analysis of the Indian stock market. Integrating this dataset with economic statistics and news could provide more insight into the factors driving price fluctuations. Nonetheless, the dataset stands out for its wide range of practical applications.

## Data Conditioning

Data conditioning is essential for improving model accuracy in multivariate time series forecasting using generative AI. This is done thoroughly transforming, engineering, and cleaning input data as required.

**Data Transformation and Mutation:**

To prepare multivariate time series data for forecasting, the processes of data transformation and mutation are essential. To improve the data's quality and usefulness for analysis, alterations can be made to its format, structure, or values.

**Feature engineering:** The computation of the difference between the High and Low prices, as well as the Open and Close prices, provides valuable information on the volatility and directionality of the market during a certain time frame. The market's momentum and investor views throughout the time under examination are indicated by these designed traits, which are crucial for forecasting future movements.

**Indexing and Resampling:** Our approach to indexing and resampling strategy aims to streamline the minute-by-minute, high-frequency stock market data into more manageable 2-hour timeframes for analysis. This transformation helps uncover longer-term trends and patterns that minute-level data may have hidden, in addition to simplifying and managing the dataset. To make our stock market analysis more relevant and clearer and to gain more strategic insights into market behavior, we must resample over a time.

**Data Cleaning and Preprocessing:**

**Handling Missing Values:** Our dataset came clean from day 1 and as the graphical insights we produced in the presentation, there were no missing or null values.

**Removing Duplicates:** We performed analysis of the data using R studio and checked for duplicated and found none.

**Data Type Conversions:** For time series forecasting, it is essential to make sure that the appropriate data types are used, particularly for dates and times. All the date and time columns in our preprocessing will be converted to the proper standard date-time format, making procedures like resampling and time-based querying easier to understand and without errors.

## Seasonality and Stationarity in Time Series Forecasting

### 2.5.1 Fourier Analysis

The application of Fourier analysis [33]on ADANIENT's stock price data, as a representative of the dataset at hand, encompassing variables like open, close, high, and low prices, in addition to volume, has yielded crucial insights into the frequency domain characteristics of the data. As depicted in Figure 1, A significant finding is the pronounced peak at zero frequency across the open, close, high, and low prices. This distinct peak often indicates a long-term trend or average level within the time series, suggesting a stable long-term trajectory or a consistent average price level with minimal fluctuations over time. Such patterns are typical in stock market data, reflecting either sustained directional trends or stable price levels.

In contrast, the Fourier transform's linear appearance at non-zero frequencies suggest the absence of notable periodic or seasonal patterns. This lack of regular cyclical patterns implies that the data does not exhibit typical seasonal characteristics. However, the volume data presents a more variable pattern in the Fourier analysis, indicating potential cyclic behaviors but not necessarily regular seasonality. This variability in volume data can be attributed to factors like market sentiment and news events, which lead to fluctuations that do not conform to predictable seasonal patterns.

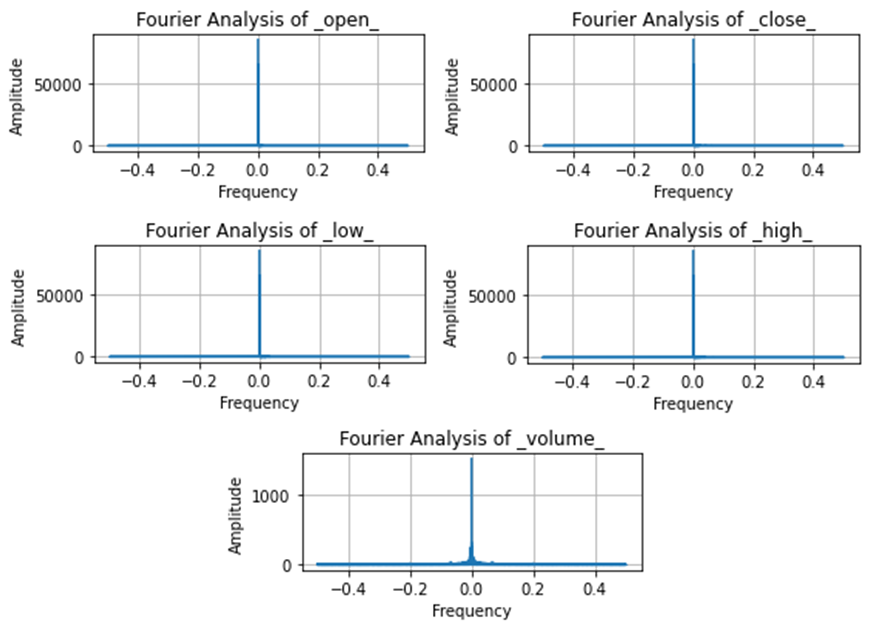


Figure 1: Fourier Analysis of ADANIENT's Stock Price Data: Visualizing Frequency Domain Characteristics

### 2.5.2 Augmented Dickey-Fuller Test

The Augmented Dickey-Fuller (ADF) test [34], a statistical method to assess the presence of a unit root in a time series, was conducted on the 'Close Prices' of ADANIENT's stock data. A code is developed in python and the ADF Test Statistic value of -1.812, along with a p-value of approximately 0.375, suggests the data's non-stationarity, as the p-value exceeds the common threshold of 0.05. Therefore, there is insufficient evidence to reject the null hypothesis, indicating the 'Close Prices' data's non-stationarity. With 2 lags and 997 observations used, the analysis is robust. This non-stationarity necessitates data transformations for accurate forecasting and suggests the potential applicability of models like ARIMA or LSTM networks, which are better suited for such data.

### Kwiatkowski-Phillips-Schmidt-Shin

The Kwiatkowski-Phillips-Schmidt-Shin (KPSS) test [35], conducted on the 'Close Prices' series, provides additional insights into the data's statistical properties. A code is developed in python and the KPSS Statistic of approximately 1.195 and a p-value of 0.01, which is below the 0.05 significance level, indicates strong evidence against the null hypothesis of stationarity. The test statistic significantly exceeds the 1% critical value of 0.739. With 19 lags used, the analysis is thorough. The KPSS test results, aligning with the ADF test findings, reinforce the conclusion of non-stationarity in the 'Close Prices' series. This suggests that standard stationary-based modeling techniques may not be suitable for this data, advocating for alternative approaches that accommodate non-stationary characteristics.

### Synthesis of Findings

Integrating the outcomes of the Fourier analysis with the ADF and KPSS tests reveals that ADANIENT's stock price data is characterized by non-stationary trends and lacks evident regular seasonal patterns. The dominant trend observed in the Fourier analysis aligns with the non-stationarity inferred from the ADF and KPSS tests. The absence of clear seasonal or cyclical patterns, typically associated with stationary data, is consistent across these analytical methods. This indicates the need for data modeling techniques that are suitable for non-stationary time series, focusing on capturing trends rather than assuming stationarity.

Incorporating seasonal attributes into the input features of advanced forecasting models involves a strategic enhancement of the dataset. Specifically, attributes such as the day of the week and month extracted from the data entries are appended to the existing feature set. This enrichment enables the model to recognize and learn from the underlying temporal patterns that might influence stock prices. For instance, in a LSTM network, these seasonal attributes can act as additional dimensions in the input data, providing the network with contextual temporal information. This assists the LSTM in capturing dependencies not just based on past stock prices, but also on seasonal trends. Similarly, in GANs and Transformer models tailored for time series, these seasonal indicators can guide the model in distinguishing and adapting to recurring patterns over different time frames. By methodically integrating these seasonal features, the forecasting model becomes more adjusted to subtle cyclical influences, potentially leading to enhanced prediction accuracy. This approach is particularly beneficial in datasets like stock prices, where primary trends are non-stationary, yet underlying seasonal patterns might still exert a significant influence.

## Data Quality Assessment

This is an exceptionally robust and fine dataset for time series modeling and analysis because of its comprehensive intraday coverage that extends more than seven years. Further enhancing accessibility is the pre-calculated technical indicator inclusion. Although there is a dearth of contextual data that could further enhance analysis, overall quality is excellent. However, the fundamental OHLCV and indicator data are excellent.

**Completeness**: The dataset contains OHLCV price data and technical indicators with no missing values for 100 stocks over 7+ years at 1-minute intervals. All expected data points are present.

**Uniqueness**: Each row contains a unique timestamp identifier and corresponding unique data values. There are no duplicate rows.

**Accuracy** Prices and technical indicators appear accurate and consistent based on spot checks. Calculated indicator values align with formulas.

**Atomicity**: Each row represents the smallest granular unit, one time-stamped observation. No values can be broken down further.

**Conformity**: Data types and formats are consistent across all stocks and timeframes.

## Other Data Sources

**Energy Dataset**

We assessed the energy consumption dataset during the dataset selection process for our generative AI multivariate time series forecasting project, but ultimately opted not to use it. Many factors affected this decision. Firstly, the stock market dataset, which is 26GB in size and gives a more comprehensive and detailed base for training our models, is larger than the energy usage dataset, which is only 129MB. In addition, the stock market dataset offers a more comprehensive view of market dynamics due to the presence of price, volume, and 55 technical indicators, whereas the energy consumption dataset primarily focuses on metrics related to household electricity usage, offering a narrower scope and diversity in features. To fully capture the intricacies of multivariate time series forecasting, a richer and more detailed dataset can be found by this larger dataset.

**Infectious Disease Dataset**

In addition, a review was conducted of a data set created by the Centers for Disease Control and Prevention (CDC), with research conducted by the National Institutes of Health (NIH). Similarly to the energy dataset, this dataset is smaller (under 145MB) but spliced into 2,468 files. The data was recorded at a daily rate of COVID infection occurrences, by county and by state. There are ten variables within this dataset, including COVID infections cases, deaths, hospitalization, and population. This data was used in research to predict hospitalizations over a four-week period.

## Storage Medium

It was found that storing the dataset on personal computers is a sufficient and reasonable approach. Despite the dataset's large size (33 gigabytes uncompressed and 13 gigabytes compressed) the consumption pattern enables the team to work efficiently without the requirement for cloud-based storage.

The approach does not demand that the entire dataset be actively engaged. Instead, chunks of the dataset are used to test and improve forecasting algorithms. This strategy decreases the load on storage resources because only subsets of the data during each analysis session need to be handled. Using a local machine would provide complete control over data management and backup operations, according to specific use case and workflow. It would also eliminate the recurring costs connected with cloud storage, especially as there is no need for extensive capabilities of such solutions for the project.

## Storage Security

Since the dataset is obtained from publicly available repositories, concerns related to data sensitivity, compliance, and legal considerations are less prominent. However, it is crucial to implement security measures for the dataset and its storage, focusing on integrity and consistency. It is crucial to keep the data from unauthorized external access. Common security practices for file storage, both locally and on servers, are found sufficient due to the non-sensitive nature of the project's data. These include standard antivirus and firewall protections typically installed on computers and servers and using VPN when accessing the dataset.

In other words, dataset access and permissions are particularly important, given that the project employs an agile methodology, which suggests that all team members should theoretically have access to the data, whether stored locally or on a server. Implementing role-based access control, through data version control tools, would be advantageous. However, considering the public nature of the dataset, a greater emphasis should be placed on securing and safeguarding the codes used for data manipulation and model training rather than spending resources and efforts on the dataset. GitHub's version control system is proposed for managing and securing these codes.

Another notable aspect is backup and recovery. The responsibility is distributed among team members, supplemented by the automatic backup system of the cluster server used for fast computing. The priority lies in ensuring frequent backups of the codes, which can further keep accessibility and collaboration throughout the project appropriately.

## Storage Costs

The primary storage cost consideration is related to handling the dataset, particularly regarding access to high-speed internet for downloading and re-uploading. Initially, costs are incurred from downloading the entire dataset from its source link and uploading it to the Hopper cluster server. Once the data is uploaded to the Hopper cluster, it is managed within the university’s infrastructure, which presumably covers these associated costs. Additionally, there is an aspect of data manipulation at the local level, where team members may work with compressed versions or selected portions of the dataset for model training purposes. This strategy minimizes the need for extensive storage space and consequently reduces potential costs. Storing the dataset on each team member's laptop incurs no direct financial costs, but it does require sufficient storage space on personal devices.

# Algorithms & Analysis / ML Model Exploration & Selection

## Solution Approach

### Systems Architecture

Our proposed system architecture for multivariate time series forecasting using generative AI models consists of several key components, as illustrated in Figure 2. After obtaining data from the user, the dataset should be processed initially. The following aspects are included in this stage:

A diagram of a system architecture

Description automatically generated

The system architecture describes the complete process that is followed. The end user uploads multivariate data in the form of a CSV file. This file is cleaned, missing value imputation, and merged according to the specifications. The data is then normalized and standardized to ensure that all values are on the same scale. This multivariate data is then used to train the Patch TST model. Once trained, the model is evaluated against the remaining dataset. This test produces predictions, which are subsequently utilized to make forecasts.

**Data Integration**

The data integration phase is a crucial step in the preprocessing pipeline, ensuring the quality and readiness of our multivariate time series dataset for forecasting with generative AI models.

**Handling Missing Values**

We performed a missing-ness map analysis using R libraries, which revealed that our stock data had no missing values. Given the high quality of our dataset, no imputation techniques were required.

**Outlier Detection and Management**

We employed z-score and box plot analysis to detect potential outliers in the dataset. While some volatile outliers were identified, we decided to retain them, as such volatility is inherent to stock data. Therefore, we did not exclude or transform any outliers, preserving the integrity of the dataset's underlying patterns and characteristics.

**Timestamp Management**

Ensuring accurate temporal analysis and alignment across the time series, we reviewed and formatted all timestamps to maintain consistency throughout the dataset. The timestamps were modified to include 2-hour time steps.

**Feature Engineering**

To capture temporal dependencies, a crucial aspect of time series forecasting models, we constructed lagged features. These lagged features enable the models to learn from past behavior and patterns, enhancing their predictive capabilities.

**Data Transformation**

To address any remaining null values, we implemented appropriate transformations, ensuring that our dataset did not contain any gaps that could lead to inaccuracies in prediction. A simple interpolation was adopted for this purpose.

**Data Format Standardization**

To facilitate efficient handling by machine learning libraries, we converted our data into formats compatible with our analytical tools, primarily pandas Data Frames and Numpy arrays.

**Normalization and Scaling**

Given the sensitivity of models to feature scales, we normalized our features using techniques such as min-max scaling and z-score normalization. This step ensures that each feature contributes equally to the model's training process, preventing any potential dominance of features with larger scales.

**Train-Test Split**

To assess model performance and generalizability to new, unseen data, we divided the dataset into training, validation, and testing subsets as 70-20-10 ratios. This split maintains the integrity of our evaluation metrics and enables us to measure the model's ability to generalize to unseen scenarios.

After the data is appropriately prepared and preprocessed, we obtained and curated a robust dataset ready for advanced analytical processing and subsequent predictive modeling using generative AI techniques. The next stage is model training and its validation.

**Model Training**

The selected models are trained on the preprocessed data using advanced techniques like adversarial training, variational inference, or self-attention mechanisms, depending on the model architecture. A discussion of appropriate models is presented in section 3.2.

**Forecasting**

Once trained, the models can generate synthetic future data samples conditioned on the historical time series, and the aim is to forecast future values of the multiple interdependent variables.

**Evaluation**

The forecasted values are evaluated against held-out test data using appropriate metrics such as mean squared error (MSE), mean absolute percentage error (MAP), and directional accuracy.

A diagram of a system architecture

Description automatically generated

Figure 2: System architecture

### Systems Security

All three models, noted in the above section, encode data as part of the unsupervised machine learning process. In this process, data is converted into numerical vectors. This conversion can be made using three different raw data: relational, textual, and images [30].

### Systems Data Flows

The system data flow for multivariate time series forecasting is depicted in Figure 3. Initially, the process commences with the aggregation of stock market data as the input data, which is subjected to multiple data preprocessing steps to ensure consistency and to mitigate any discrepancies, such as missing values. After preprocessing, an optimal model is selected based on the specific objectives and inherent patterns within the data. This model is then trained, leveraging historical data to discern underlying trends and dependencies. Post-training, the model undergoes a thorough evaluation phase, wherein its predictive performance is scrutinized against a set of unseen data to ascertain its forecasting accuracy and generalizability. The culmination of this pipeline is the forecasting phase, where the model is deployed to generate predictions about future stock market trends, providing valuable insights for decision-making processes. Model exploration and implementation during development is needed at that stage, the model and overall forecasting pipeline are still being explored, tested, and refined. Evaluating the model's effectiveness by performing evaluation and validation on the entire pipeline is highly during this development phase.

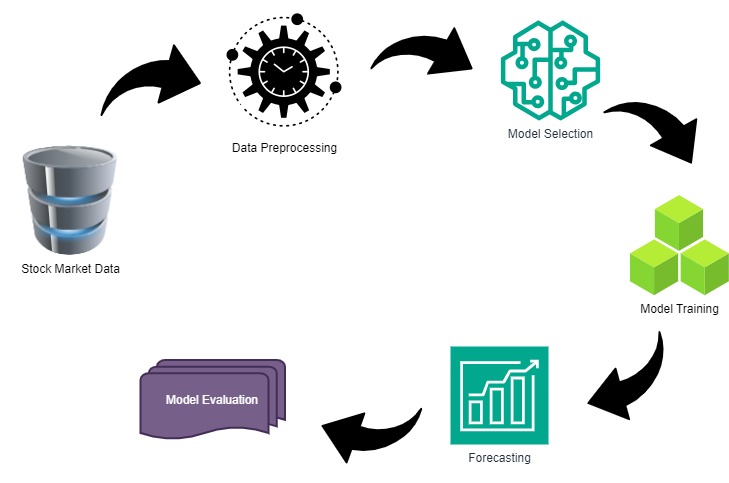


Figure 3: A schematic system of data flows, from the input to the output.

### Algorithms & Analysis

Our research efforts have focused on the implementation of generative AI models capable of handling multiple variables, answering the needs of predicting time series. The algorithms employed are aimed to capture complex temporal dependencies and patterns inherent in the dataset.

**Transformer model**

Given the robust self-attention mechanism of transformers, it can effectively capture both long-range spatial information and local patterns. Hence, the transformers can be used to specifically focus on relevant parts of the input sequence and potentially figure out the underlying patterns.

**Integrated VAE-GAN**

The investigation of VAE and GAN networks, as well as their combined variations, showed an enormous potential in using these algorithms because the underlying aspects of both is to encode data into a compressed yet representative format and this means that the network is trying to extract the most important aspects of the dataset. It is important to note that the decoding aspect or generating synthesized undetectable samples based on the training requires successfully recognizing those underlying patterns. Hence, this would potentially match the expectations of this study.

Both approaches can be adopted with certain considerations. First, it is important to note that the default GAN or VAE version, are not capable of working with time series data. Conversely, the transformers' default behavior, since it is based on the sequence of data, can be utilized for time series data. This aspect potentially leads to more complicated implementations of VAE-GAN in comparison to transformers. Transformers can analyze entire sequences at once, allowing for quicker training periods and the capacity to manage sequences significantly longer than those handled by RNNs. Also, the use of a self-attention mechanism in transformers means the model can consider the full sequence of data all at the same time.

## Machine Learning

### **Model Exploration**

During the first phase of our research, we investigated various machine learning models designed to address the complexity inherent in time series data processing, notably in forecasting and prediction. Our research focused on two key criteria: the models' ability to handle temporal data and their compatibility with multivariate time series analysis. These factors were critical since we wanted to attain high accuracy in forecasting future values while also understanding the various temporal connections inherent in the dataset.

As we looked deeper into the literature, narrowing our focus to finance and stock markets as a demonstration dataset, we observed that it is required to tackle domain-related issues. Notably, it was found that it is important to deal with the volatility, non-linear patterns, and seasonal variations found in financial time series datasets.

Through a thorough examination of a variety of machine learning algorithms, spanning from traditional statistical methods to cutting-edge deep learning architectures, we were able to make educated conclusions about which models were best suited to the complexities of financial time series research as presented below.

**Transformer Models**

The goal of the project is to develop a time series forecasting system that can accurately forecast future values using a transformer model. To address this challenge, we have chosen to employ Transformer neural network architectures. Although transformers are primarily designed for natural language processing tasks and are optimized to understand and generate human language, with some adaptation, they can be used to model numerical time series data and ultimately forecasting multivariate time series in the domain of stock market data.

Transformers have self-attention mechanisms that allow them to effectively model both long-term dependencies and local patterns in the multivariate input features, such as historical OHLCV (Open High Low Close Volume) and technical indicators. The self-attention mechanism enables the model to selectively focus on relevant parts of the input sequence, capturing intricate relationships within the data.

Furthermore, Transformers can handle the inherent complexities of stock market data, including the presence of missing values, the need for feature normalization, and the engineering of derived indicators. By preprocessing the data appropriately and feeding it into optimized Transformer models, we can leverage their powerful sequence-to-sequence modeling capabilities to generate multi-step ahead forecasts for key stock market variables.

Concerning ready-to-use platforms and libraries, HuggingFace is an outstanding platform that offers a vast array of pre-trained Transformer models, each with unique characteristics and suited for various tasks. While these models are primarily designed for natural language processing tasks, their inherent ability to model sequential data makes them potentially suitable for time series forecasting as well. However, adapting these pre-trained models for time series forecasting would require significant customization, such as reformatting the time series data into a suitable input structure and potentially modifying the model architecture or training process to better accommodate the unique characteristics of time series data.

Table 1: Comparative analysis of different models available on HuggingFace

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model name | Type | Application area | Potential for time series forecasting | Estimated Model Size (Parameters) |
| BERT | Language Model | Text classification, named entity recognition, QA | Sequence learning capability, adaptable for numerical sequences | ~110M (BERT Base) |
| GPT-2 | Generative Language Model | Text generation | Generative capabilities, adaptable for sequential output prediction | ~117M to ~1.55B |
| Transformer-XL | Language Model | Long-term dependency understanding | Long-term dependency learning, beneficial for long sequences in time series | ~257M |
| Lent | Language Model | Text classification, robust context understanding | Robust context understanding, useful for complex time series patterns | ~110M (XLNet Base) to ~340M (XLNet Large) |
| T5 | Text-to-Text Transfer Transformer | Text summarization, QA | Flexibility in handling different formats, could be adapted for time series tasks | Varies (e.g., Small to 11B for T5-11b) |

As highlighted in Table 1, each model excels in specific applications and demonstrates the potential for adaptation in time-series forecasting. T5, notable for capturing long-term dependencies, is well suited for complex series due to its architecture that handles variable sequence lengths. However, the adaptation of this model might not be the best approach for time series forecasting due to intricacies associated with modifying time series data to be compatible with a language model. As presented in table 2, there are specifically designed transformers for the time-series datasets which are already modified for that purpose. Among them, Patch TST, PatchTSMixer, Autoformer, and Time Series Transformer can be particularly adapted for such projects.

Table 2: Comparative analysis of different time-series transformers available on HuggingFace

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model name | Type | Application area | Pros/Cons | Model Size |
| Autoformer | Transformer model | Time series forecasting | Efficient handling of time series data with transformer architecture | Various based on d\_model, encoder\_layers etc. |
| Informer | Transformer model | Long sequence time-series forecasting | Utilizes a Probabilistic Attention mechanism for efficient performance, addressing issues like quadratic time complexity and high memory usage. | Various based on d\_model, encoder\_layers etc. |
| PatchTST | Time Series Forecasting Model | Multivariate time series forecasting and self-supervised representation learning | Efficiently handles long-term forecasting with patching design, which reduces computation and memory usage. Performs well in self-supervised pre-training tasks. The complexity of model customization might be a challenge | Various based on d\_model, num\_hidden\_layers etc. |
| PatchTSMixer | Time Series Analysis Model/ MLP-Mixer based model | Multivariate time series forecasting, classification, and regression | Offers lightweight and effective mixing across patches and channels. It can be customized for various attention mechanisms. The model is more efficient than complex Transformer models, with less computing usage. | Various based on d\_model, num\_layers etc. |
| Time Series Transformer | Vanilla encoder-decoder Transformer. | Probabilistic time series forecasting | Designed for diverse forecasting applications with flexibility in prediction horizons and input sizes. Complexity may increase with extensive feature types | Various based on configuration parameters |

Autoformer is a transformer architecture specifically designed for time series forecasting. It features a Decomposition Layer that decomposes the input time series into trend and seasonal components, allowing the model to effectively capture essential patterns. Instead of the standard self-attention mechanism used in vanilla transformers, Auto former employs an Autocorrelation Mechanism that captures frequency-based dependencies in the data, which are particularly relevant for time series analysis. Its Time Delay Aggregation component enhances the temporal dependency recognition, making Autoformer well-suited for multivariate time series forecasting problems. This architecture addresses time series characteristics like seasonality and temporal dependencies more effectively.

Informer addresses Long Sequence Time Series Forecasting challenges. Traditional transformers struggle with computational burdens in processing long sequences due to the quadratic scaling of the attention mechanism. Informer Model addresses this issue with ProbSparse Self-Attention. This replaces the standard attention with a more efficient one that focuses mainly on the important part of the sequence. This also helps in reducing the time and memory complexity. It also incorporates Self-Attention distilling, which also manages long input while preserving the dominant attention patterns.

The Patch Time Series Transformer (PatchTST) introduces a transformative approach in the analysis and forecasting of multivariate time series data, utilizing a unique channel independent processing framework. This approach enables PatchTST to dissect and understand the intricate patterns and relationships within the data with precision. It integrates advanced neural architectures and attention mechanisms, balancing forecasting ability and computational efficiency, making it suitable for applications requiring high-dimensional data analysis and rapid predictions.

Patch Time Series Mixer (PatchTSMixer), for multivariate time series forecasting, segments the data into patches for processing. This approach allows the model to focus on localized patterns, offering an efficient analysis of time dependencies [2]. It employs an inter-channel correlation instead of a channel-independent strategy, which reduces computational overhead and enhances forecasting accuracy.

The time series transformer, a vanilla encoder-decoder transformer model, is designed tailored for time series forecasting. It functions as a probabilistic forecaster, focusing on learning a distribution from which future values can be extracted. Its architecture consists of an encoder, which processes input, and a decoder is used for generating output future values predictions. The model integrates additional components, such as temporal and static categorical features, enhancing its predictive accuracy. The strength of this model is that it can handle diverse forecasting scenarios, accommodating varying lengths of predictions and input dimensions, and multiple feature types.

**Integrated VAE-GAN Architecture**

The integrated Variational Autoencoders (VAE) and Generative Adversarial Networks (GAN) architecture combines the strengths of both models, leveraging VAEs ability to learn efficient data representations and GAN capacity to generate high-quality synthetic data samples. The VAE component of the integrated model can learn a compressed representation of the input data, capturing its essential features and patterns. This learned representation can then be used to generate new samples through the GAN component, which is trained to produce realistic and diverse synthetic data samples that resemble the true data distribution. This approach not only augments training data, potentially enhancing forecasting performance, but also offers interpretable insights into the underlying data structures. Despite its potential, the shift towards transformers for time series forecasting was motivated by their superior performance in various domains, prompted by the challenges in achieving high accuracy with VAE-GANs.

### Model Selection

Based on the characteristics of the data and problem requirements, as well as performance in terms of model accuracy and initial processing, we have opted to work with models based on Transformer architecture. We reached this conclusion because Transformers ability to model sequential data effectively compared to an integrated version of Variational Autoencoders and Generative Adversarial Networks.

# Visualizations / ML Model Training, Evaluation, & Validation

## Overview

Why should we train ML models?

Once we are done with pre-processing dataset and evaluating the model, which is a PatchTST transformer in our case, we need to train it to yield the best and most accurate results when the product is used in the real world. Now that we have our "Time-series dataset" data, we trained our model by matching the input to the predicted output. We have both the input data and the correct/expected output; this dataset is usually carefully created either by engineers as our team developers or by semi-automatic collection of certain data. And we must report the predicted result for each data row here, as supervised learning requires [36].

To evaluate how successfully our model was trained - which depends on the amount of data we have, the value we want to predict, input (and so on) and evaluate the characteristics of the model - means errors, classification errors, accuracy and recall IR models (and so on).

Training Loss: Training loss in deep learning models, including transformers, measures how well the model fits the training data by calculating the error across all examples in the training set. It is computed after processing each batch and often visualized through a curve, reflecting the model's learning progress over time [37].

Validation Loss: Validation loss evaluates a deep learning model's performance on a separate validation set, not seen during training. It calculates the error across all examples in this set to gauge the model's generalization ability, like training loss, but for validating model accuracy on new data [37].

## Visualizations

Below are the visualizations that we created during the data preprocessing stage.

A graph with blue bars and red line

Description automatically generated

Figure 4: Histogram of close price values

From Figure 4 with a fitted density plot showcases the distribution of closing prices of a stock. On the X-axis, the close prices are displayed, while the Y-axis represents the density of the closing prices. The blue bars indicate how many times the closing prices fall within a particular pricing range. We can say that the distribution is skewed towards the right and is not normally distributed. It also indicated the outliers on the higher end of the closing prices. The highest bar, which is in the price range of 2000 to 2500, indicates the most common close price range.

A graph with a line

Description automatically generated

Figure 5: Normal Q-Q visualization

The above Normal Q-Q plot depicts the deviation from the red reference line, which indicates the dataset does not follow normal distribution. We can also see that the points are closely clustered on the left side and dispersed on the right side, indicating right skewness. We followed log transformation and square root transformation to normalize the dataset so that we can use it in our model.

After Performing Normalization on the data, the following figure shows the results.

A graph with a red line

Description automatically generated

Figure 6: Histogram of closing price values after standard scaling

The above histogram with a fitted density plot showcases the distribution of closing prices of a stock. On the X-axis, the close prices are displayed, while the Y-axis represents the density of the closing prices. The bars indicate how many times the closing prices fall within a particular pricing range. We can say that the distribution is skewed towards the right and is almost normally distributed.

A graph of a normal q-q plot

Description automatically generated

Figure 7: Normal Q-Q plot for the Normalized data

The Normal Q-Q Plot graphically examines the normality of a dataset by plotting sample quantiles versus theoretical quantiles from a normal distribution. While most data points follow the reference red line in the center, indicating an approximation normal distribution, they deviate away at the tails, indicating a deviation from normalcy overall. The points are tightly clustered on the left and spread out on the right, indicating a right-skewed distribution.

## Machine Learning

### Hyper parameter tuning

In our study, hyperparameter tuning was systematically conducted to optimize the model’s performance. This process involved an exploratory analysis of various parameters associated with architecture of the network and other specifics for the time-series forecasting. Significant emphasis was placed on the architecture of the model, where factors such as the number of layers (n\_layers), number of attention heads (n\_heads), model dimension (d\_model), and dimension of feed forward fully connected network (d\_ff) were of interest. These parameters directly affect the model’s performance as depicted in Figure 6. In addition to these parameters, batch size is of great importance, balancing computational complexity and robustness of the training process. Each configuration of these parameters was evaluated based on its impact on the model performance, measured through validation loss as illustrated in Figure 7. Optuna hyperparameter tuning pipeline, offers a systematic method to investigate different hyperparameter tuning. It was utilized to effectively search for the optimized parameters. This minimization task was performed withing defined ranges for each parameter. hyperparameter tuning pipeline, offers a systematic method to investigate different hyperparameter tuning [38].

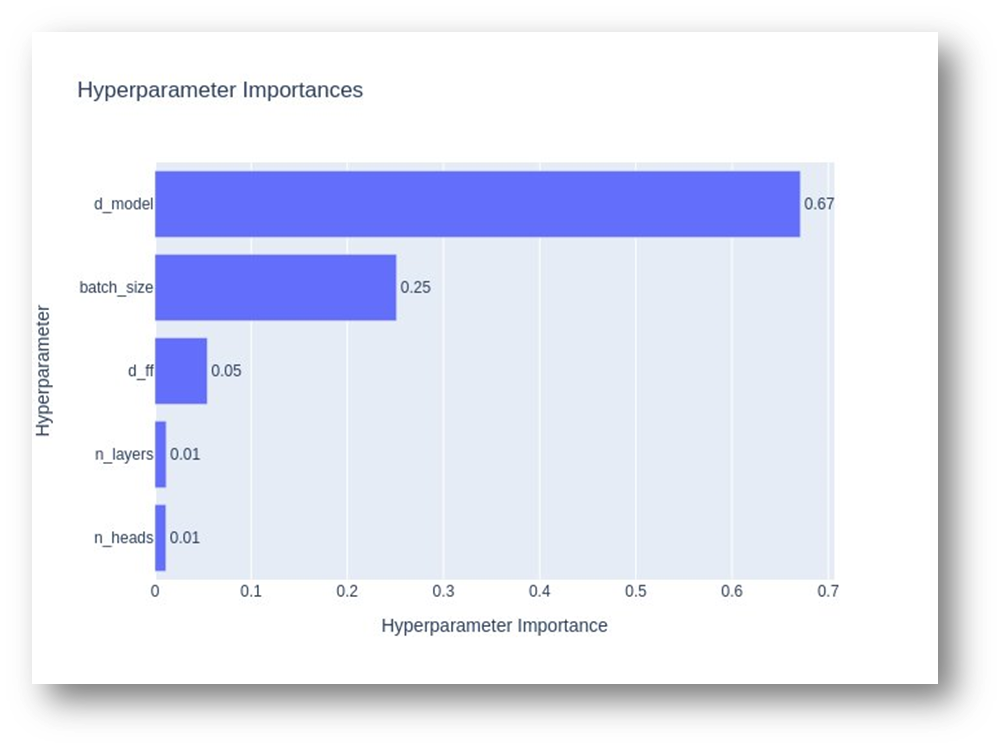


Figure 8: The influence of different hyperparameters on the performance of the model

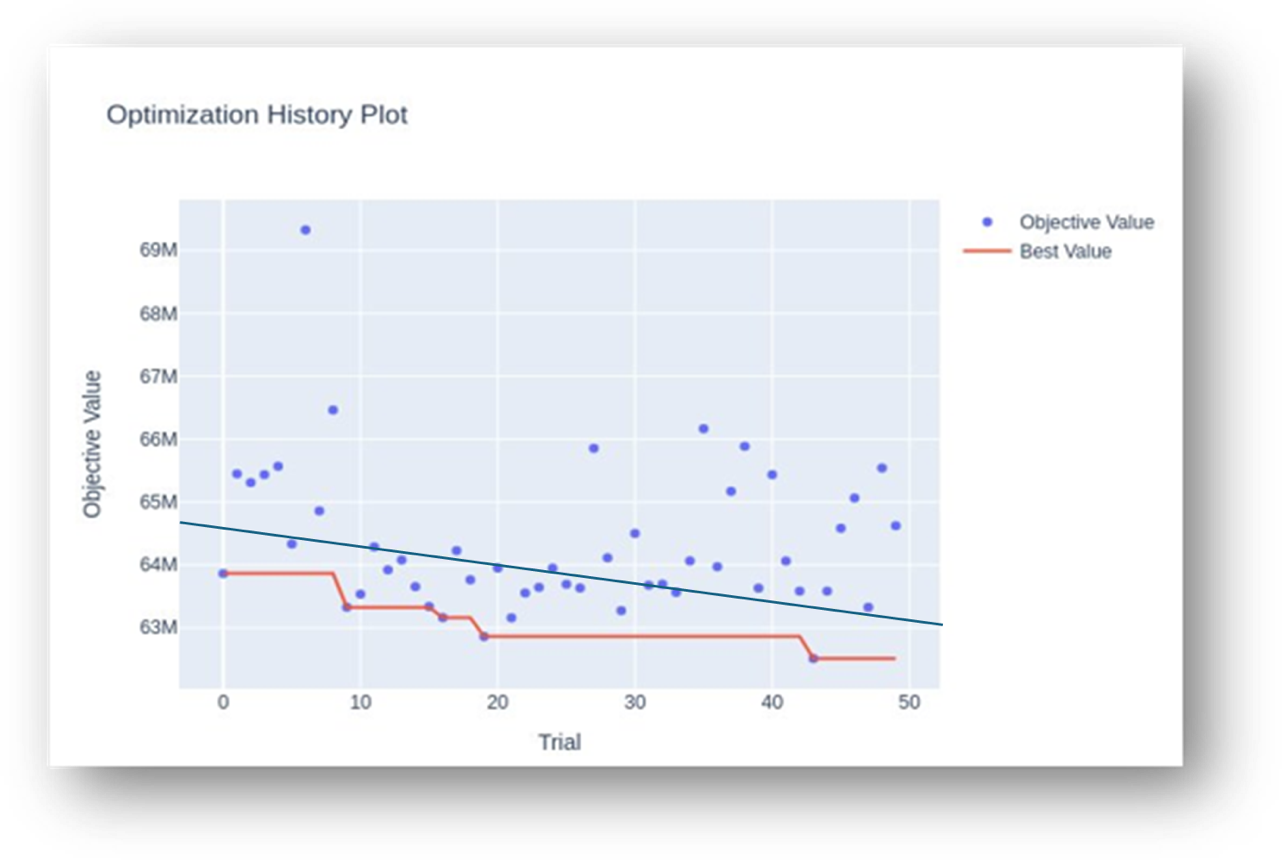


Figure 9: The objective value (validation loss) along different hyper parameter investigations (Trials)

### Model Training

Our code implementation for PatchTST framework for a deep learning project aimed at time series forecasting, incorporating both data manipulation and model training components. It leverages the Hugging Face Transformers library to import PatchTSMixer models—specialized for time series data through a mix of Transformer architectures and patch-based input processing—alongside training utilities. Essential data handling libraries like pandas and numpy are included for preprocessing tasks, as well as sklearn for dataset split and additional preprocessing, such as normalization. PyTorch is used as the deep learning framework, with intrinsically efficient data management and data loading during training. The presence of T5ForConditionalGeneration and T5Tokenizer indicates a potentially broader project scope, possibly involving text generation from time series analysis, demonstrating a multifaceted approach to handling and forecasting time series data within a deep learning context.

We load and initially explore our dataset and we performed preliminary understanding the structure, content, and quality of the data. It involves importing minute-wise trading data along with various technical indicators that are likely essential for our analysis or model training. By executing df.head() (corrected from df.head for functionality), we got the first few rows and columns of our dataset, enabling us to quickly assess its format, the types of indicators included, and to confirm the successful loading of the data. This initial exploration is crucial for understanding subsequent data preprocessing, analysis, and validation steps in the project.

After completing data preprocessing and exploratory data analysis, we proceeded to train and evaluate our time series forecasting model on the stock market data. The model architecture employed a 3-layer encoder with 4 attention heads and a model dimension of 16, along with a fully connected network of 128 dimensions, without attention dropout but 30% dropout on other linear layers. The time series data was segmented into patches of 24-time steps with a stride of 2

### Model Evaluation

For forecasting, the model utilized a historical data window of 300-time steps to predict 30 future steps ahead. The dataset was split into 10% for validation and 20% for testing purposes.

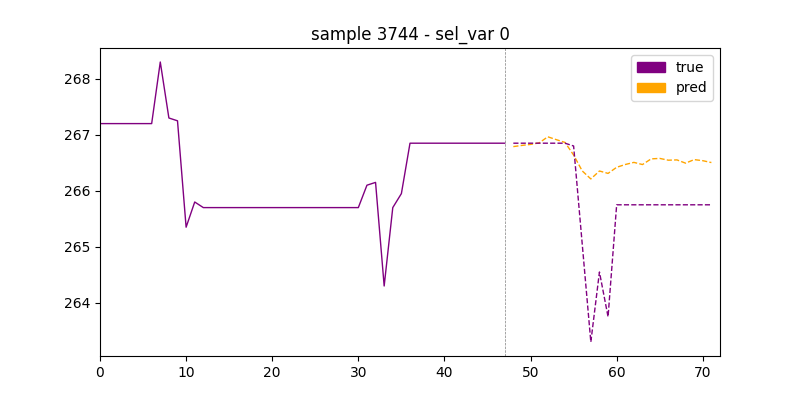
During the training process, we monitored losses on both the validation and training sets. The losses converged reasonably well, with the final validation losses of 0.013886 for mean squared error (MSE) and 0.069274 for mean absolute error (MAE). On the held-out test set, comprising 20% of the data, the model achieved an MSE of 0.101004 and an MAE of 0.160934.

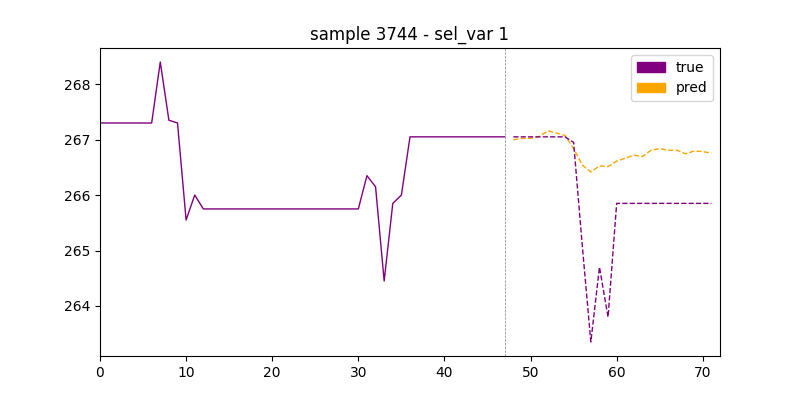
To analyze the model's predictive performance qualitatively, we visualized samples of the actual and predicted time series for various stock market indicators like open price, close price, trading volume, daily high, and daily low. The model effectively captured the overall trends and patterns, though there were some deviations in magnitude for specific local peaks and troughs.

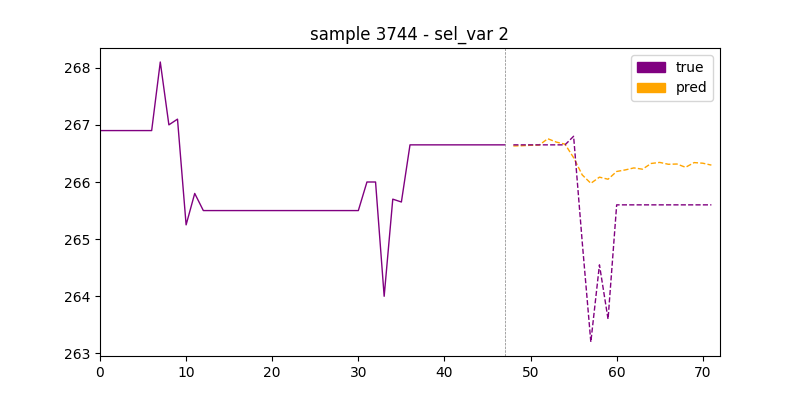
In summary, the quantitative evaluation metrics on the test set demonstrated reasonable forecasting accuracy, with scope for further tuning. The qualitative analysis provided insights into the model's strengths in trend forecasting and potential areas for improvement in precise value prediction during volatile periods. With this evaluation, we gained confidence in deploying the model for generating stock market forecasts to aid decision-making processes.

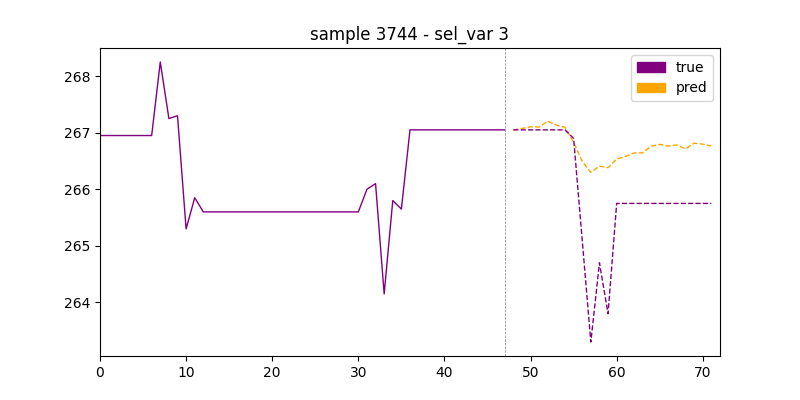
### Model Validation

The investigation of model inference after 30 epochs is depicted in Figure 8 through 12. Each feature is shown separately in each figure. The most important observation is that the number of forecasting horizon, referred to the past and future, plays an important role in the outcome of the prediction. It is very challenging to select a robust value for each of them. However, as shown in the figures, the trend is captured by the model appropriately, although the values are not accurate.









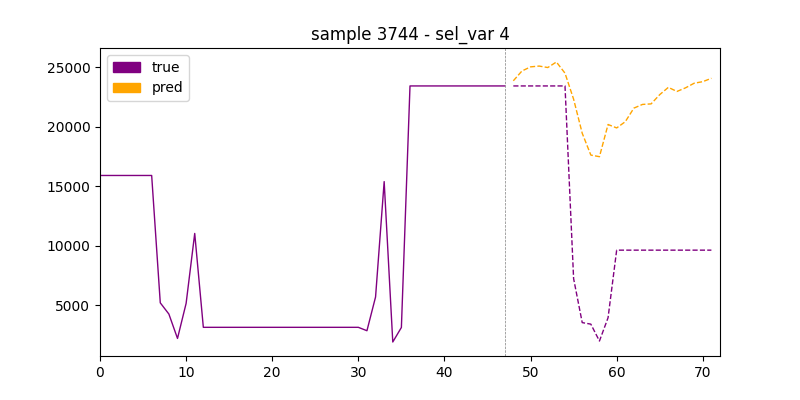
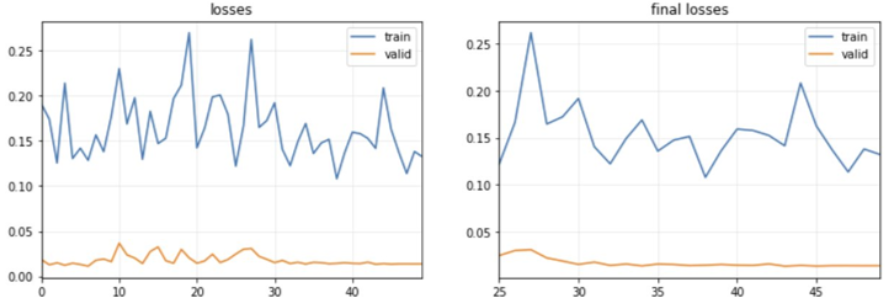


Figure 10: Results of the model prediction after 50 epochs



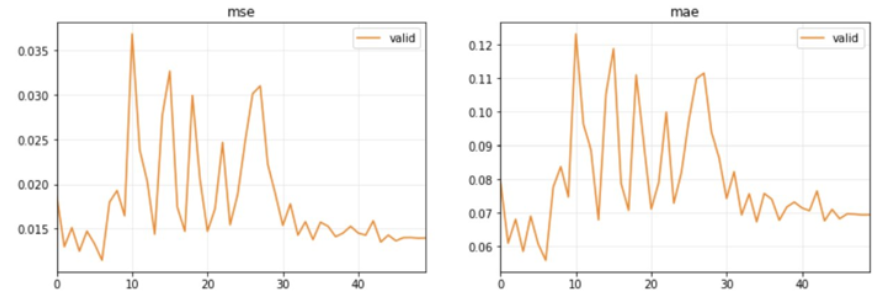


Figure 11: Results and loss graphs of the MSE and MAE calculations after 50 epochs

Stock market forecasting is a complex endeavor, and our project aimed to develop a multivariate time series forecasting model tailored to capture the intricate patterns within stock data. The training and validation graphs provide a comprehensive overview of the model's performance across 50 training epochs. During the validation phase, we observed a Mean Squared Error (MSE) of approximately 0.013886 and a Mean Absolute Error (MAE) of around 0.069274. These metrics suggest that the model exhibited proficiency in identifying the underlying dynamics of the stock market variables utilized for validation, potentially indicating a robust predictive capability within the confines of the validation dataset.

However, the test set evaluation revealed a divergence in model performance when applied to out-of-sample data. The test set exhibited higher error values, with an MSE of 0.101004 and an MAE of 0.160934. This notable increase in error suggests that the model's performance deteriorated when encountered with previously unseen data. Several factors could contribute to this phenomenon, including overfitting to the validation set, a shift in market conditions not represented in the training data, or the inherent unpredictability and noise associated with financial time series data.

These findings underscore the intricate nature of stock data forecasting and the importance of rigorous model evaluation. It is crucial to assess a model's generalization capability through comprehensive testing on out-of-sample data, as this can reveal potential limitations and areas for improvement. Additionally, the discrepancy between validation and test set performance highlights the need for continuous model monitoring and adaptation to account for evolving market dynamics.

# Findings

We have successfully implemented a PatchTST framework using a model architecture with 3-layer encoders, 4 attention heads, and appropriate dimensions for the model and fully connected networks. Systematic hyperparameter tuning played a critical role in optimizing the model's performance. Hyperparameter exploration included the number of layers, attention heads, model dimensions, and the dimensions of the feed-forward network.

The analysis of the healthcare dataset using the PatchTST model architecture has yielded insightful results, reflective of the model’s ability to learn and predict health-related outcomes. Through the training over 50 epochs, the loss metrics have shown a promising downward trend for both training and validation, suggesting the model's good fit. Validation MSE and MAE stand at 0.362872 and 0.382686, respectively, indicating a solid grasp of the dataset's patterns.

However, when assessing the model against the unseen test set, the errors increased, with the MSE rising to 1.439492 and the MAE to 0.765352. This highlights a potential overfitting to the validation set or a difference in the distribution between the training/validation and the test sets. It also underscores the complexities of predictive modeling in the healthcare field, where numerous variables can affect outcomes.

The length of the forecasting horizon was also a critical factor in the model's performance. Finding the optimal balance between past information used for predictions and the future horizon remains a challenge, particularly for health-related data, which is subject to rapid and often unpredictable changes.

Despite the model effectively identifying general trends, it experienced difficulties in pinpointing specific peaks and troughs, a common challenge in time-series forecasting. This emphasizes the need for ongoing model refinement and testing to ensure that it can adapt to new patterns and maintain accuracy over time. The findings from this analysis stress the importance of continuous model evaluation and enhancement to achieve the highest predictive accuracy in healthcare applications.

The model training was conducted over 50 epochs, with close monitoring of losses on both training and validation datasets. The losses converged well, with the final validation losses indicating a good fit of the model on the validation set. The MSE and MAE (as shown in Figure 11) were 0.013886 and 0.069274, respectively, after 50 epochs.

The length of the forecasting horizon was identified as a significant factor in the model’s predictive performance. The choice of how far back to look (past horizon) and how far ahead to predict (future horizon) greatly influenced the accuracy of the model's predictions.

A notable divergence in performance was observed on the held-out test set, with higher error values (MSE of 0.101004 and MAE of 0.160934). This could suggest overfitting to the validation set, shifts in market conditions not represented in the training data, or the unpredictable nature of financial time series data.

Qualitative analysis through visualizations showed that the model effectively captured overall trends and patterns in stock market indicators. However, the model faced challenges in accurately predicting specific local peaks and troughs, indicating potential areas for improvement in precise value prediction, especially during volatile market periods. The study highlights the necessity of continuous model monitoring and adaptation, especially considering the evolving nature of market dynamics that may affect the predictive performance.

The number of past and future time steps used for forecasting were critical to the model’s output. The challenge lay in selecting robust values for these parameters to achieve accurate predictions. The findings stress the importance of evaluating the model's ability to generalize beyond the training data. Comprehensive testing of sample data is crucial to find potential limitations and further model refinements.

# Summary

This project, titled “Multivariate Timeseries Forecasting with Generative AI” focused on advancing forecasting capabilities through the application of Transformer neural network architectures across various domains. These models utilized the self-attention mechanisms inherent in Transformers to effectively model complex dependencies and temporal sequences in multivariate time series data.

The primary objective was to develop and evaluate a multivariate time series forecasting model using different datasets, including an influenza virus dataset and a stock market dataset with technical indicators. The influenza virus dataset contained variables such as age, virus weightage, total occupancy, and other healthcare-related factors. The stock market dataset was composed of features like open, close, high, low, volume, and 55 technical indicators.

In the analysis of the influenza virus dataset, the model achieved demonstrated effective forecasting capabilities for the 6 target variables. The convergence of the training and validation loss curves suggested a robust model fit. The final validation metrics, mean squared error (MSE) of 0.362872 and mean absolute error (MAE) of 0.382886, indicated accurate predictions. Due to the volatile nature of stock data, the model trained on the stock market dataset showed potential for determining short-period forecasting and possible future trends. Comparative qualitative analysis between the performance of these two domains revealed that the forecasting performance was more effective for healthcare dataset rather than for the stock market data. The difference is likely due to the inherent complexity and volatility of stock market data.

The findings suggest that the proposed multivariate time series forecasting approach is well-suited for applications in which volatility and unexpected values occur infrequently. For instance, in healthcare settings, accurate predictions of variables like occupancy and virus weightage are crucial for resource planning and disease management. For stock data forecasting, the model can be reliable for short-term trading strategies but may require further enhancements or the incorporation of additional relevant features to improve long-term forecasting accuracy.

# Future Work

There is a potential in developing hybrid models that combine traditional time series forecasting methods with machine learning algorithms. This approach aims to leverage the strengths of both methods. Additionally, extensive hyperparameter tuning and optimization pipeline should be developed to find the optimal model configuration, potentially leading to improved forecasting accuracy.

Engaging domain experts (e.g., healthcare professionals or financial analysts) in the modeling process is recommended. Their domain-specific knowledge and constraints could be invaluable in enhancing interpretability and reliability of the forecasting models.

Investigating the inclusion of additional data types, such as news content, into the forecasting model could also be beneficial. This aims to incorporate other influential attributes that could reveal uncaptured trends in the forecasting pipeline.

Additionally, implementing feature engineering techniques, such as feature interactions or dimensionality reduction methods before the forecasting pipeline will be critical in capturing complex patterns within the data.

Appendix

Appendix A: Glossary

|  |  |
| --- | --- |
| Term | Definition |
| AI (Artificial Intelligence) | Machines mimicking human thinking, including learning and problem-solving. |
| AIOps (Artificial Intelligence for IT Operations) | AI and machine learning applied to automate and improve IT operations. |
| Internet of Things (IoT) | A statistical model for analyzing and forecasting time series, ideal for single-variable data. |
| CNN (Convolutional Neural Network) | Deep neural networks are best for image analysis, recognized for pattern recognition. |
| ECG (Electrocardiogram) | A test to check the heart's electrical activity and health, useful for detecting cardiac issues. |
| Fin GAN | A pipeline focuses on probabilistic forecasting and introduces a novel economics-driven loss function, repositioning GANs for classification tasks and enabling comprehensive conditional probability distributions for price returns. |
| GAN (Generative Adversarial Network) | A machine learning setup where two neural networks compete, commonly used in unsupervised learning. |
| GDP (Gross Domestic Product) | The total value of goods and services produced in a country indicates economic health. |
| Generative AI | AI that creates new data and results, not just categorizing existing information. |
| Generative AI | A network of devices connected via sensors and software, sharing data over the internet. |
| Index GAN | The platform for integrating financial domain knowledge, news context, and a multi-step attentive seq2seq learning network, which operates within a Wasserstein GAN framework, uniquely incorporates market sentiment and news analysis, reflecting the stock market's inherent characteristics and demonstrating enhanced predictive performance in real-world indices. |
| LSTM (Long Short-Term Memory) | A type of RNN effective in learning from long-term data, crucial for time series analysis. |
| MLP (Multi-Layer Perceptron) | A neural network with multiple layers, using backpropagation for training. |
| MTS (Multivariate Time Series) | Data sequences over time involving several related variables, used in complex forecasting. |
| Predictive Modeling | Using statistical and machine learning methods to forecast future events based on past data. |
| RNN (Recurrent Neural Network) | A neural network where data flows through time-sequenced nodes, ideal for language and speech tasks. |
| Predictive Modeling | Analyzing ordered data points to spot trends and predict future values. |
| VAE (Variational Autoencoders) | Autoencoders handle data in a probabilistic way, often in image processing. |

Table : Glossary Table

Appendix B: GitHub Repository

Overview

There is a need to predict future values to pursue critical, cost-effective paths in business operations or in providing safety impacting tools for government agencies. Multivariate time series forecasting faces the challenge of analyzing complex and interrelated data. However, limited datasets create obstacles in this process. Every traditional model from the basic regression to LSTM-RNN has limitations such as needing epoch volumes of data or only identifying linear relationships. There is a need for a system that can find subtle relationships in data and make accurate forecasts from multiple factors over time. The solution is the usage of a transformer algorithm to process smaller volumes of data. A transformer uses multi-headed attention parallel processing with weighted positional encoding to yield more accurate predictive values.

GitHub Repository Link

<https://github.com/ansarisam/multiverse/tree/main>

GitHub Repository Contents

The repository will include the source code, dataset, and instructions on how to use it.

Appendix C: Risks

Sprint 1 Risks

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Risk | Description | Probability | Impact | Mitigation |
| Algorithm Bias | Imbalanced Weights | Medium | High | Hyper-parameter Tuning |
| Ethics | Decision Ambiguity | Medium | High | Policy |
| Inaccuracy | No crystal balls | Medium | Medium | Ensemble Training |

Table : Sprint 1 Risks

In machine learning and data science, algorithm bias, ethics, and inaccuracy are pivotal considerations. Algorithm bias, characterized by a medium probability and risk, underscores the need for meticulous hyperparameter tuning to counteract biases inherent in model predictions. Meanwhile, ethical concerns, with a medium probability but high risk, necessitate the establishment of robust policies to guide responsible AI deployment and mitigate the potential for discriminatory outcomes. Inaccuracy, presenting a moderate probability and risk, highlights the intrinsic uncertainties in machine learning models, prompting the adoption of ensemble training techniques to bolster predictive reliability. Addressing these challenges requires a multifaceted approach that integrates technical rigor with ethical frameworks and policy development, ensuring the integrity, fairness, and transparency of machine learning applications in diverse domains.

The risk persists when we are working with a dataset, and we often see dataset related risks that are specific to the dataset we chose. The below table explains the risks and planned mitigations indicating how much they might impact the project and what is the probability.

Sprint 2 Risks

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Risk | Description | Probability | Impact | Mitigation |
| Algorithm Bias | Imbalanced Weights | Medium | High | Hyper-parameter Tuning |
| False data | Manipulated  data | Medium | High | Policies |
| Overfitting | No diverse  performance | Medium | High | Normalization |
| Volatility and Noise | Stock Markets are  Volatile & noisy | High | Medium | Smoothing Techniques |
| Gradient  explosion | Model can’t  learn | Medium | Medium | Normalization |

Table : Sprint 2 Risks

Sprint 3 Risks

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Risk | Description | Probability | Impact | Mitigation |
| Data Bias | Unfair outcomes | Medium | Medium | Pre-processing method |
| Gradient explosion | Unsuitable training process | High | High | Normalization,  Gradient clipping |
| Volatility & noise | Volatile & noisy data | High | High | Smoothing Techniques |
|  |  |  |  |  |
| Overfitting | No diverse performance | Medium | High | Regularization |

Table : Sprint 3 Risks

The table is a concise overview of potential issues in machine learning models along with their likelihood, consequences, and countermeasures. Data Bias could lead to unfair results, but with a moderate chance and seriousness, can be reduced by using pre-processing methods. Gradient Explosion poses an elevated risk and can significantly affect model performance; it is preventable through techniques, such as normalization and gradient clipping. Volatility and Noise refer to unpredictable data fluctuations and messy data, which are quite likely to occur and have a medium level impact but can be tackled using smoothing techniques to create a more stable model. Lastly, Overfitting happens when a model learns the training data too well and lacks generalization, with medium likelihood and high impact, which is often prevented by regularization to maintain model simplicity and improve its generality.

Sprint 4 Risks

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Risk | Description | Probability | Impact | Mitigation |
| Hopper cluster  availability | HPC wait time | Medium | High | Request early |
|  |  |  |  |  |
| Gradient explosion | Unsuitable training process | High | High | Normalization,  Gradient clipping |
| Volatility & noise | Volatile & noisy data | High | High | Smoothing Techniques |
|  |  |  |  |  |
| Overfitting | No diverse performance | Medium | High | Regularization |

Table : Sprint 4 Risks

As we worked on the project we faced a few big challenges. First, we needed to use Hopper clusters for our work, but sometimes we had to wait too long to get access. We solved this by asking for these resources early, so we didn't have to wait and slow down our work.

Another big issue was gradient explosion. This can make it very hard to train our models correctly. To fix this, we used two methods: normalization and gradient clipping. These methods helped us keep everything under control and train our models the right way.

We also had to deal with data that was very noisy and unpredictable. This makes it hard for our models to learn properly. To improve this, we used smoothing techniques, which made the data cleaner and easier for the models to understand. This happens when a model is too focused on the training data and does not perform well on new data. We prevented this by using regularization techniques, which helped make our model strong and able to work well with new data.

Sprint 5 Risks

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Risk | Description | Probability | Impact | Mitigation |
| Hopper cluster availability | HPC wait time | Medium | High | Request early |
|  |  |  |  |  |

Table : Sprint 5 Risks

Appendix D: Agile Development

Scrum Methodology

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Close Date | Feb 6 | Feb 27 | Mar 19 | Apr 9 | Apr 30 |
|  | Sprint 1 | Sprint 2 | Sprint 3 | Sprint 4 | Sprint 5 |
| Week | 1 – 3 | 4 - 6 | 7 – 9 | 10 – 12 | 13 - 15 |

Table : Sprint project dates

Like the Agile process, learning it has been its own iteration. At each step in the process of learning, there is a review of what has been learned. Discoveries are made along the way, which may or may not confirm previously held conclusions. This is true for understanding the agile process, machine learning as it relates to the project as well the dynamics of being a team member.

Sprint 1 Analysis

Initial efforts of the team’s assertions were on a micro level, with a specific focus on reviewing the financial markets for both the stories along with the identification of the problem, data set, and solutions. While this is still a viable component of the project, it is not the story, nor the entirety of the problem or solution. By the second interim, based on the client feedback, the team progressed forward. The team gained clarity in their overall view of the general problem of prediction of values using GenAI for a times series with multiple variables. Finding multiple possible solutions was more of a challenge, and it took all of Sprint 1 to accomplish this. There were challenges with the burndown chart in YouTrack as the team added more task's part way through the Sprint. This was one of the lessons learned was to agree on all the tasks at the start of the sprint. Other lessons learned include coming to understand the teams’ differing communication styles along with varying time management skills. It might have been helpful to work with another team from the start to gain a better understanding sooner than later.

Sprint 2 Analysis

During Sprint 2, which lasted three weeks, our team focused only on the exploration, transformation, and cleaning of datasets essential to our model's performance. There were still challenges with the problem statement which required more editing. The sprint's main goal was to completely refine our data, which is essential for maintaining the quality and effectiveness of our predictive models. The team was able to narrow down options for data set to use. Throughout the period, the team maintained a velocity of 11.3 points per member, indicating a high level of production and a focus on improving data quality. This sprint was essential in establishing the groundwork for our project's success. The team took a better understanding of setting up tasks in YouTrack, and the resulting burndown chart reflecting this. The team communicated better as well.

Sprint 3 Analysis

We have done previous research on suitable algorithms to work with. After understanding the strengths and weaknesses of VAE-GAN models, we concluded that it cannot support us in achieving our project goal effectively. We switched to work with transformer models. We were able to implement the code and generate the output for loss coefficient but could not evaluate it because T5 being a LLM model. We tried to work with

Other models of transformers such as PatchTSmixer, PatchTST, require data pipeline to train the model, and we're not able to train using JSON and CSV file. We will work on that in the coming Sprint.

Sprint 4 Analysis

In our recent work, we made significant progress by ensuring our model could accurately predict stock prices. We achieved this by employing various methods to organize and prepare the data, which helped our model make more precise predictions. To test its effectiveness, we used different datasets containing various types of stocks from NIFTY 100, allowing us to assess its performance across different market conditions. we recognized a gap in our understanding of transformers. Therefore, we were committed to improving our communication and learning more about transformers to further enhance the accuracy and reliability of our model in future projects.

Sprint 5 Analysis

During Sprint 5, our team accomplished a significant milestone by finishing the project model and delving into the visualizations, assessments, and comprehensive findings of our work. This sprint's pace was notable at 11.3, indicating our rapid handling of complicated assignments and commitment to detailed analysis. The complete visualizations developed will be an effective tool for both internal review and external presentations, while our model's evaluations confirmed its reliability and accuracy. Collectively, these initiatives have generated useful insights that will guide future project development and optimization.

**This page intentionally left blank**

Reference

[1] “A Time Series is Worth 64 Words: Long-term Forecasting with Transformers,” ar5iv. Accessed: May 02, 2024. [Online]. Available: https://ar5iv.labs.arxiv.org/html/2211.14730

[2] “PatchTSMixer.” Accessed: May 01, 2024. [Online]. Available: https://huggingface.co/docs/transformers/en/model\_doc/patchtsmixer

[3] “Brief Overview Of Traditional Forecasting Methods,” FasterCapital. Accessed: Apr. 29, 2024. [Online]. Available: https://fastercapital.com/keyword/brief-overview-of-traditional-forecasting-methods.html

[4] A. Singh, “Multivariate Time Series Analysis With Python for Forecasting and Modeling (Updated 2024),” Analytics Vidhya. Accessed: Apr. 29, 2024. [Online]. Available: https://www.analyticsvidhya.com/blog/2018/09/multivariate-time-series-guide-forecasting-modeling-python-codes/

[5] J. Harper, “Pentagon testing generative AI in ‘global information dominance’ experiments,” DefenseScoop. Accessed: Apr. 29, 2024. [Online]. Available: https://defensescoop.com/2023/07/14/pentagon-testing-generative-ai-in-global-information-dominance-experiments/

[6] J. L. Coen, M. Cameron, J. Michalakes, E. G. Patton, P. J. Riggan, and K. M. Yedinak, “WRF-Fire: Coupled Weather–Wildland Fire Modeling with the Weather Research and Forecasting Model,” *Journal of Applied Meteorology and Climatology*, vol. 52, no. 1, pp. 16–38, 2013, doi: 10.1175/JAMC-D-12-023.1.

[7] D. Barker *et al.*, “The Weather Research and Forecasting Model’s Community Variational/Ensemble Data Assimilation System: WRFDA,” *Bulletin of the American Meteorological Society*, vol. 93, no. 6, pp. 831–843, Jun. 2012, doi: 10.1175/BAMS-D-11-00167.1.

[8] A. Panesar, “Case Studies,” in *Machine Learning and AI for Healthcare : Big Data for Improved Health Outcomes*, A. Panesar, Ed., Berkeley, CA: Apress, 2021, pp. 293–349. doi: 10.1007/978-1-4842-6537-6\_10.

[9] T. Yan, H. Zhang, T. Zhou, Y. Zhan, and Y. Xia, “ScoreGrad: Multivariate Probabilistic Time Series Forecasting with Continuous Energy-based Generative Models.” arXiv, Jun. 18, 2021. Accessed: Jan. 29, 2024. [Online]. Available: http://arxiv.org/abs/2106.10121

[10] “CDX-NET: Cross-Domain Multi-Feature Fusion Modeling Via Deep Neural Networks for Multivariate Time Series Forecasting in AIOps | IEEE Conference Publication | IEEE Xplore.” Accessed: Jan. 29, 2024. [Online]. Available: https://ieeexplore.ieee.org/document/9746242

[11] J. Xie, Y. Cui, F. Huang, C. Liu, and K. Zheng, “MARINA: An MLP-Attention Model for Multivariate Time-Series Analysis,” in *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, Atlanta GA USA: ACM, Oct. 2022, pp. 2230–2239. doi: 10.1145/3511808.3557386.

[12] J. Hao and F. Liu, “Improving long-term multivariate time series forecasting with a seasonal-trend decomposition-based 2-dimensional temporal convolution dense network,” *Sci Rep*, vol. 14, no. 1, Art. no. 1, Jan. 2024, doi: 10.1038/s41598-024-52240-y.

[13] H. Liang, L. Song, J. Du, X. Li, and L. Guo, “Consistent Anomaly Detection and Localization of Multivariate Time Series via Cross-Correlation Graph-Based Encoder–Decoder GAN,” *IEEE Transactions on Instrumentation and Measurement*, vol. 71, pp. 1–10, 2022, doi: 10.1109/TIM.2021.3139696.

[14] F. Kong, J. Li, B. Jiang, H. Wang, and H. Song, “Integrated Generative Model for Industrial Anomaly Detection via Bidirectional LSTM and Attention Mechanism,” *IEEE Transactions on Industrial Informatics*, vol. 19, no. 1, pp. 541–550, Jan. 2023, doi: 10.1109/TII.2021.3078192.

[15] Y. Choi, H. Lim, H. Choi, and I.-J. Kim, “GAN-Based Anomaly Detection and Localization of Multivariate Time Series Data for Power Plant,” in *2020 IEEE International Conference on Big Data and Smart Computing (BigComp)*, Feb. 2020, pp. 71–74. doi: 10.1109/BigComp48618.2020.00-97.

[16] W. Li, W. Hu, T. Chen, N. Chen, and C. Feng, “StackVAE-G: An efficient and interpretable model for time series anomaly detection.” arXiv, Sep. 19, 2022. doi: 10.48550/arXiv.2105.08397.

[17] “Multiscale Wavelet Graph AutoEncoder for Multivariate Time-Series Anomaly Detection | IEEE Journals & Magazine | IEEE Xplore.” Accessed: Jan. 29, 2024. [Online]. Available: https://ieeexplore.ieee.org/document/9954430

[18] E. Brophy, M. De Vos, G. Boylan, and T. Ward, “Multivariate Generative Adversarial Networks and Their Loss Functions for Synthesis of Multichannel ECGs,” *IEEE Access*, vol. 9, pp. 158936–158945, 2021, doi: 10.1109/ACCESS.2021.3130421.

[19] F. Viton, M. Elbattah, J.-L. Guérin, and G. Dequen, “Heatmaps for Visual Explainability of CNN-Based Predictions for Multivariate Time Series with Application to Healthcare,” in *2020 IEEE International Conference on Healthcare Informatics (ICHI)*, Nov. 2020, pp. 1–8. doi: 10.1109/ICHI48887.2020.9374393.

[20] A. Arora and A. Arora, “Generative adversarial networks and synthetic patient data: current challenges and future perspectives,” *Future Healthc J*, vol. 9, no. 2, pp. 190–193, Jul. 2022, doi: 10.7861/fhj.2022-0013.

[21] P. Sharma, F. E. Shamout, V. Abrol, and D. A. Clifton, “Data Pre-Processing Using Neural Processes for Modeling Personalized Vital-Sign Time-Series Data,” *IEEE Journal of Biomedical and Health Informatics*, vol. 26, no. 4, pp. 1528–1537, Apr. 2022, doi: 10.1109/JBHI.2021.3107518.

[22] E. Brophy, “Synthesis of Dependent Multichannel ECG using Generative Adversarial Networks,” in *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, in CIKM ’20. New York, NY, USA: Association for Computing Machinery, Oct. 2020, pp. 3229–3232. doi: 10.1145/3340531.3418509.

[23] T. T. Huynh *et al.*, “Efficient Integration of Multi-Order Dynamics and Internal Dynamics in Stock Movement Prediction,” in *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*, in WSDM ’23. New York, NY, USA: Association for Computing Machinery, Feb. 2023, pp. 850–858. doi: 10.1145/3539597.3570427.

[24] M. Dogariu, L.-D. Ştefan, B. A. Boteanu, C. Lamba, B. Kim, and B. Ionescu, “Generation of Realistic Synthetic Financial Time-series,” *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 18, no. 4, p. 96:1-96:27, Mar. 2022, doi: 10.1145/3501305.

[25] S. R. Polamuri, Dr. K. Srinivas, and Dr. A. Krishna Mohan, “Multi-Model Generative Adversarial Network Hybrid Prediction Algorithm (MMGAN-HPA) for stock market prices prediction,” *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 9, pp. 7433–7444, Oct. 2022, doi: 10.1016/j.jksuci.2021.07.001.

[26] A. U. Haq, A. Zeb, Z. Lei, and D. Zhang, “Forecasting daily stock trend using multi-filter feature selection and deep learning,” *Expert Systems with Applications*, vol. 168, p. 114444, Apr. 2021, doi: 10.1016/j.eswa.2020.114444.

[27] X. Li, Y. Liu, C. Huang, and X. Zhou, “Financial Time Series Prediction Based on GGM-GAN,” in *2021 16th International Conference on Intelligent Systems and Knowledge Engineering (ISKE)*, Nov. 2021, pp. 170–174. doi: 10.1109/ISKE54062.2021.9755397.

[28] P. Muthukumar and J. Zhong, “A Stochastic Time Series Model for Predicting Financial Trends using NLP.” arXiv, Feb. 01, 2021. doi: 10.48550/arXiv.2102.01290.

[29] J. Gu, F. P. Deek, and G. Wang, “Stock Broad-Index Trend Patterns Learning via Domain Knowledge Informed Generative Network.” arXiv, Feb. 27, 2023. doi: 10.48550/arXiv.2302.14164.

[30] M. Vuletić, M. Cucuringu, and F. Prenzel, “Fin-GAN: Forecasting and Classifying Financial Time Series via Generative Adversarial Networks.” Rochester, NY, Jan. 18, 2023. doi: 10.2139/ssrn.4328302.

[31] A. Koochali, A. Dengel, and S. Ahmed, “If You Like It, GAN It—Probabilistic Multivariate Times Series Forecast with GAN,” *Engineering Proceedings*, vol. 5, no. 1, Art. no. 1, 2021, doi: 10.3390/engproc2021005040.

[32] “Patch Time Series Transformer in Hugging Face.” Accessed: Apr. 29, 2024. [Online]. Available: https://huggingface.co/blog/patchtst

[33] P. Masset, “Analysis of Financial Time-Series Using Fourier and Wavelet Methods.” Rochester, NY, Oct. 24, 2008. doi: 10.2139/ssrn.1289420.

[34] P. Narayan and S. Popp, “A New Unit Root Test with Two Structural Breaks in Level and Slope at Unknown Time,” *Journal of Applied Statistics*, vol. 37, pp. 1425–1438, Sep. 2010, doi: 10.1080/02664760903039883.

[35] D. Kwiatkowski, P. C. B. Phillips, P. Schmidt, and Y. Shin, “Testing the null hypothesis of stationarity against the alternative of a unit root: How sure are we that economic time series have a unit root?,” *Journal of Econometrics*, vol. 54, no. 1, pp. 159–178, Oct. 1992, doi: 10.1016/0304-4076(92)90104-Y.

[36] “Why do we use a test set in Machine Learning?,” Deepchecks. Accessed: Apr. 29, 2024. [Online]. Available: https://deepchecks.com/question/why-do-we-use-a-test-set-in-machine-learning/

[37] baeldung, “Training and Validation Loss in Deep Learning | Baeldung on Computer Science.” Accessed: Apr. 29, 2024. [Online]. Available: https://www.baeldung.com/cs/training-validation-loss-deep-learning

[38] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, “Optuna: A Next-generation Hyperparameter Optimization Framework,” in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, in KDD ’19. New York, NY, USA: Association for Computing Machinery, Jul. 2019, pp. 2623–2631. doi: 10.1145/3292500.3330701.