A red circle with a white silhouette of a person holding a spear and shield

Description automatically generated

Saint Petersburg State University

Graduate School of Management

Master in Corporate Finance Program

“ARE MARKETS SO DIFFERENT? FORECASTING MARKETS' RETURN WITH LSTM MODEL AND OPTIMIZATIONS”

Master’s Thesis by the 2nd year student

Concentration – Master in Corporate Finance

Ensa Haskasa

Research Advisor:

Dr. Darko B. Vukovic

Department of Finance and Accounting

St. Petersburg

2024

**DECLARATION OF ORIGINALITY**

I, Ensa Haskasa, declare that this master’s thesis is my original work and has not been submitted in whole or partially, for any degree or other purposes at any other university.

To the best of my knowledge, this paper does not include any quotation or publication of any other person’s material unless cited explicitly within the paper. Additionally, I declare that while writing this paper, I acted according to the ethical rules of scientific and academic conduct of St. Petersburg State University.

**ЗАЯВЛЕНИЕ ОРИГИНАЛЬНОСТИ**

Я, Энса Хаскаса, заявляю, что эта магистерская диссертация является моей оригинальной работой и не была представлена полностью или частично, для какой-либо степени или других целей в любом другом университете.

Насколько мне известно, эта статья не включает в себя какое-либо цитирование или публикацию материала любого другого лица, если только это прямо не указано в статье. Кроме того, я заявляю, что при написании этой статьи я действовала в соответствии с этическими правилами научного и академического поведения Санкт-Петербургского государственного университета.

**ABSTRACT**

This study analyzes how the Long Short-Term Memory model (LSTM) coupled with parameter optimization techniques, can help predict the financial markets returns. Due to the various non-linearities, and non-stationarities, as well as other controversial dynamics it is not obvious at all how to accurately forecast financial markets. For example, market responses to quantitative easing going forward may not be reflected in historical correlations since the underlying reasons are different. By default, many traditional forecasting methods presume data is constant and linear. However, modern technology is rapidly reshaping how we understand the world. LSTM, a type of recurrent neural network (RNN), has proven to be the best model for analyzing sequential data, especially when it has a long-term memory.

This study's main goal is testing the LSTM model's capability in forecasting market re-turns for different financial markets. Furthermore, the study aims to explore parameter optimization techniques which affect the model's functioning behavior. The study process entails gathering historical market data from a variety of financial markets. Several LSTM designs and parameter optimization strategies are constructed and tested utilizing back testing and statistical analysis. The literature review covers a brief analysis of the currently available studies related to the prediction of returns in financial markets, with LSTM model and parameter optimization methods. These are all explored clearly and with the advantages and disadvantages included. Moreover, the topic is placed within the broader context of literature and investigates which gaps are uncovered by preceding studies. Results from empirical research revealed that LSTM models were indeed capable of tracing the underlying patterns embedded in the market data to generate accurate return forecasts in various markets. Parameter optimization techniques, including feature engineering, hyperparameter tuning and ensemble methods, could improve the predictive power of LSTM model. The study's results have impactful consequences for finance, in terms of academic research and real-world applications. The utilization of LSTM models and parameter optimization methods enables investors, and financial analysts to obtain essential insights to enhance portfolio management and trade execution, as well as protect investments in fluctuating market conditions. Subsequent research avenues may entail the examination of other neural network designs, the expansion of data sources utilized, and the application of market regime shifts to the forecasting process.

**Keywords:** LSTM,financial markets, forecasting, optimization, investors.

**АВТОРЕФЕРАТ**

В этом исследовании анализируется, как модель долговременной краткосрочной памяти (LSTM) в сочетании с методами оптимизации может помочь спрогнозировать доходность финансовых рынков. Из-за различных нелинейностей и нестационарностей, а также других противоречивых динамик, совершенно не очевидно, как точно прогнозировать финансовые рынки. Например, реакция рынка на дальнейшее количественное смягчение может не отражаться в исторических корреляциях, поскольку основные причины иные. По умолчанию многие традиционные методы прогнозирования предполагают, что данные постоянны и линейны. Однако современные технологии быстро меняют наше понимание мира. LSTM, тип рекуррентной нейронной сети (RNN), оказался лучшей моделью для анализа последовательных данных, особенно если она имеет долговременную память.

Основная цель этого исследования — проверить возможности модели LSTM в прогнозировании доходности рынка для различных финансовых рынков. Кроме того, исследование направлено на изучение методов оптимизации, которые влияют на поведение модели. Процесс исследования включает в себя сбор исторических рыночных данных с различных финансовых рынков. Несколько проектов LSTM и стратегий оптимизации разрабатываются и тестируются с использованием бэктестинга и статистического анализа. Обзор литературы охватывает краткий анализ доступных в настоящее время исследований, связанных с прогнозированием доходности на финансовых рынках с использованием модели LSTM и методов оптимизации. Все они подробно рассмотрены с указанием преимуществ и недостатков. Более того, эта тема помещена в более широкий контекст литературы и исследует, какие пробелы были обнаружены в предыдущих исследованиях. Результаты эмпирических исследований показали, что модели LSTM действительно способны отслеживать основные закономерности, заложенные в рыночные данные, для создания точных прогнозов доходности на различных рынках. Методы оптимизации, включая разработку признаков, настройку гиперпараметров и ансамблевые методы, могут улучшить прогностическую способность модели LSTM. Результаты исследования имеют важные последствия для финансов с точки зрения академических исследований и реальных приложений. Использование моделей LSTM и методов оптимизации позволяет инвесторам и финансовым аналитикам получать важную информацию для улучшения управления портфелем и исполнения сделок, а также защиты инвестиций в меняющихся рыночных условиях. Последующие направления исследований могут повлечь за собой изучение других конструкций нейронных сетей, расширение используемых источников данных и применение изменений рыночного режима в процессе прогнозирования.

**Ключевые слова:** LSTM, финансовые рынки, прогнозирование, оптимизация, инвесторы.

**LIST OF ABBREVIATIONS**

NDX – Nasdaq 100

SPX – Standard’s & Poor 500

DJI – Dow Jones Industrial Average

UKX – Financial Times Stock Exchange 100

SXXP – Stoxx Europe 600

NKY – Nikkei 225

LSTM – Long Short-Term Memory Model

EMH – Efficient Market Hypothesis

RWH – Random Walk Hypothesis

RNN – Recurrent Neural Network

ANN – Artificial Neural Networks

ESNs – Echo State Networks

SGP – Symbolic Genetic Programming

GARCH – Generalized Autoregressive Conditional Heteroskedasticity

ARCH – Autoregressive Conditional Heteroscedasticity

PCA – Principal Component Analysis

ECB – European Central Bank

PMI – Purchasing Managers' Index

EU – European Union

ADF – Augmented Dickey-Fuller Test

**LIST OF FIGURES**

Figure 1. Training of the model with validation split

Figure 2. Conceptual Framework

Figure 3. Imported libraries

Figure 4. Data preparation for training

Figure 5. Building the LSTM model

Figure 6. Forecasting of market returns

Figure 7. Real vs. Forecasted data for each index

Figure 8. Augmented Dickey-Fuller test results

Figure 9. Forecasted returns for NDX Index

Figure 10. Real vs. Forecasted data for NDX Index

Figure 11. Forecasted returns for SPX Index

Figure 12. Real vs. Forecasted data for SPX Index

Figure 13. Forecasted returns for DJI Index

Figure 14. Real vs. Forecasted data for DJI Index

Figure 15. Forecasted returns for UKX Index

Figure 16. Real vs. Forecasted data for UKX Index

Figure 17. Forecasted returns for SXXP Index

Figure 18. Real vs. Forecasted data for SXXP Index

Figure 19. Forecasted returns for NKY Index

Figure 20. Real vs. Forecasted data for NKY Index

**TABLE OF CONTENTS**

ABSTRACT……………………………………………………………………………….....3

LIST OF ABBREVIATIONS………………………………………………………………...6

LIST OF FIGURES………………………………………………………………………......7

**1. INTRODUCTION**………………………………………………………………………10

* 1. Background……………………………………………………………………………10
  2. Research Goal…………………………………………………………………………12
  3. Research Questions……………………………………………………………………13

**2. LITERATURE REVIEW**……………………………………………………………....18

* 1. Efficient Market Hypothesis (EMH)………………………………………………….19
  2. Random Walk Hypothesis (RWH)…………………………………………………….21
  3. Traditional Models vs LSTM Model………………………………………………….25
  4. Parameter Optimization Techniques for LSTM Model………………...……………..30

**3. RESEARCH METHODOLOGY**……………………………………………………....32

* 1. Data Collection and Preprocessing…………………………………………………....32
  2. Data Analysis Techniques……………………………………………………………..34
  3. Training of the Model………………………………………………………………....38
  4. Optimization Technique and Mean Square Error……………………………………..39
  5. Research Methodology Conceptual Framework……………………………………...41

**4. RESULTS**………………………………………………………………………………..42

* 1. Data Description……………………………………………………………………....42
  2. Empirical Analysis…………………………………………………………………….43
  3. Results Interpretation………………………………………………………………….49
  4. Comparative Analysis Across Markets………………………………………………..60

**5. DISCUSSION**…………………………………………………………………………....63

* 1. Possible Implications………………………………………………………………….63
  2. Directions for Future Research……………………………………………………….64

**6. CONCLUSION**………………………………………………………………………....65

**7. REFERENCES**…………………………………………………………………………68

**I. INTRODUCTION**

**1.1 Background**

Recent years have seen financial markets becoming more intricate and unstable, induced by complex multi-faceted factors of geopolitics, economics, and technology. The better a market movement can be forecasted, the more informed decisions can be made, and the more competitive each stakeholder stays in the fast-evolving ecosystem of global finance. Market forecasting has traditionally been done using econometric and statistical methods generally based on the Efficient Market Hypothesis. Market returns are movements in the cost of a financial market or an investment portfolio over a given period. They are commonly used to gauge how well a particular market is performing. Market returns have implications for the market efficiency hypothesis. Financial markets are efficient according to the EMH implying that security prices reflect all available information, thus making it difficult for investors to consistently obtain higher than average returns through trading strategies. This means that the average return should only have a risk measure and be proportionate to the return predicted by investors. On this basis, it is necessary to examine how market returns relate with investor behavior under an efficient market environment. Investors who think that markets are efficient and that prices reflect all the available information might choose passive investments like funds indexed on broad equity portfolios that try to replicate aggregate market performance net of costs, transaction fees and taxes. Such investors would expect to earn the average market return over long periods. [[1]](#footnote-1)

Depending on the specific market or investment being considered, market returns can fluctuate significantly. Stock market returns, for instance, generally denote the general performance of a stock market index, namely S&P 500 and Dow Jones Industrial Average. Conversely, bond market returns depict how bonds and fixed-income securities have fared during a certain period. It is important to remember that there are inherent doubts and risks associated with market returns. Various factors such as economic conditions, interest rates, geopolitical events, corporate earnings, and investor sentiment influence financial markets. Therefore, market return might be volatile at times making it hard to predict. The presence of huge volumes of data, the rate at which artificial intelligence and machine learning techniques are advancing and the improved computational ability of this machine, enable generation of complex algorithms for predicting stock prices. Meanwhile, today’s investment markets in stocks are more opaque than ever before due to the proliferation of investing options. One approach that recently became quite popular and shows promising results is applying deep learning methods, particularly Long Short-Term Memory models, for financial forecasting. LSTMs are a type of Recurrent Neural Networks that have proven to be very efficient in capturing temporal dependencies and nonlinear relationships within sequential data. Given these properties, LSTMs are widely used for time series predictions tasks. However, there are still numerous issues in optimizing their work for financial forecasting and ensuring robustness across different types of a market. Firstly, markets can vary significantly by efficiency, volatility, and liquidity, which makes forecasting an interesting but quite challenging task. Secondly, data preprocessing, feature selection and hyperparameters impact the performance and efficiency of LSTMs. By presenting this thesis, the challenges and the success of LSTM model and parameter optimization methodologies are disclosed; as well as the computational and technical processes to uncover the force majeure of LSTM in predicting market returns when trained with actual market data.

Also, it is projected that such research reveals valuable insights in addressing predictability of the market and ensure the most suitable ML model for time-series trading in all market conditions. The LSTM model employs hard and soft optimization strategies to forecast market returns, thereby addressing these challenges. It is against this backdrop that the research seeks to examine the feasibility of using Long Short-Term Memory (LSTM) models as well as optimization strategies in formulating an answer to some of the problems encountered or faced in forecasting market returns. In this regard, a comparative study will be carried out across 6 different indices such as NASDAQ 100 (NDX), Standard’s & Poor 500 (SPX), Dow Jones Industrial Average (DJI), Financial Times Stock Exchange - FTSE 100 (UKX), Stoxx Europe 600 (SXXP) and Nikkei 225 (NKY); and empirical analysis will be conducted to present deeper insight into what triggers market returns and how LSTM models can capture and predict such patterns. This knowledge would aid investors and policy makers in utilizing advanced predicting analytics through describing strengths and limitations associated with LSTM based approaches in financial forecasting. The most essential feature that stems from an LSTM model is its memory cell, capable of storing and accessing information for a long period. In addition, gates in the form of input gate, forget gate, and output gate were attached to it. These gates control how information is stored, forgotten, or output from memory cells of LSTM allowing it to selectively remember or forget available information depending on its relevance. Internally, during the training process internal parameters including weights and biases are adjusted by the LSTM model to reduce discrepancies between predicted outputs and actual responses. For application of LSTM as a tool for forecasting market return historical market data supplemented with relevant features provides inputs into the model such that it learns patterns as well as relationships among data hence can predict future market returns.

**1.2 Research Goal**

*Research goal:* Testing the efficiency of the LSTM model based on its parameters which determine its optimizations for market return forecasting across different financial markets and investigate if there are disparities in predictive performance of the model, in comparison with real data and other conventional prediction methods, such as random walk.

The research’s’ prime purpose is to see if incorporating LSTM model and parameter optimizations can provide a basis for predicting market rates for the different financial markets. The study will establish whether optimization parameters influence the predictive capacity and efficacy of LSTM model across NDX, SPX, DJI, UKX, SXXP and NKY indices, in order to determine the model's usefulness in real scenarios. The research spans a scope of 20 years daily data, in order to distinguish the algorithms efficiency on well-developed and low-capitalized regions or markets. Furthermore, the comprehensive analysis on how specific parameter optimization practices affect the forecasting models owns a major part of this study, by putting a focus on the confirmation of both the holistic nature of the report, and the employment of LSTM to market conditions. This research will make important contributions to understanding the complex interplay between LSTM model, market idiosyncrasies, and parameter optimization methodologies. This objective involves looking into LSTM-based predictive models reviewing their performances in terms of predicting market return across multiple financial markets. Through using parameter optimization technology, like feature selection, hyperparameter tuning, and data preprocessing techniques, learning of LSTM model is enhanced and that enables the model to attain predictable capabilities and become robust across different conditions of the market. Therefore, the research will examine possible volatility in the forecasting abilities of LSTM model if they are utilized in various financial markets. The features that make the exchanges special, such as efficiency, volatility, liquidity, and frameworks on the rules and regulations, can influence the models in various markets differently.

Generally, the research objective consists of providing information that an LSTM model can be helpful for measuring market returns forecasting in varieties of financial markets. It will highlight unique features and faults of LSTM based systems for one or another type of market conditions. This research can serve as a path for improvement of predictive analytics in finance and educate practitioners and policymakers about remedy lines for using LSTM approaches in practical monetary decision-making.

**1.3 Research Questions**

* Does the LSTM model exhibit varying levels of accuracy when applied to different markets?
* Can the effective forecasting of market returns using LSTM model serve as evidence against Random Walk?
* What's the impact of different parameter optimization techniques on the accuracy of market return forecasts using LSTM models?
* **Does the LSTM model exhibit varying levels of accuracy when applied to different markets?**

This research question delves into the comparative performance of LSTM model across six major market indexes: NDX, SPX, DJI, UKX, SXXP, and NKY studied over the span of 20 years. Python, which is the most known of all for data manipulation, thanks to the Pandas library, is used to refine raw data from an excel file. LSTM model (is known to capture very complex temporal dependencies) is created by using TensorFlow and Keras. Each model learns from its corresponding index, which is driven by the collection of historical data from that specific market. The training process is demonstrated by fine-tuning the hyperparameters for maximal efficacy. Alongside cross-verification of the model by means of Scikit-Learn tools, robustness and generalizability are guaranteed. There is a comprehensive evaluation of the LSTM model performed against the performance metrics including Mean Absolute Error (MAE), Root Mean Squared Error (RMSE) and others. The measure of accuracy is performed along various financial markets' differences.

The financial market consists of two variables, which are the economic conditions and the geopolitical events, which distinguish it from the investor sentiment. In a nutshell, the NDX represents the Nasdaq 100 index, whose 100 stocks are heavily technology-related, while the DJI (Dow Jones Industrial Average) comprises 30 large market-capitalization companies across different sectors. Every index carries a different market sector that comes with unique risk and rate of return; and hence makes the task of the predictive modeling process difficult since one must compare many variables.

Analyzing is a known process which is done by training LSTM models via their historical price data that spans 20 years from the beginning step of the year 2000 to the beginning of year 2021. The data preprocessing step is scaling by the so-called scale feature the MinMaxScaler method and normalize values in the range between 0 and 1 that is important for the correction convergence and performance of the model, so it is widely used. The LSTM architecture comprises input layers; LSTM layers having the number of their units adjustable, and lastly the dense layers for predicting. The LSTM models are to be trained first, and then evaluation of their predictive power for each index is to be done in modeling future market returns. Estimations are calculated using the gliding window manner that makes the model effective in predicting future return rates based on historical closing prices sequence data. Then predicted returns, in comparison with the historical statistics, are the measure of forecasting accuracy of the LSTM model. The evaluation of forecasts accuracy across all the markets informs us about the model's suitability for dealing with the multidimensional nuisance of the function of a market and the phenomena that precede up and down movements of the market. Generous inconsistencies in precision suggest that a few markets can be more stable than most of them but since some models do not need either finetuning or customization then they may be more suited to some market’s behavior prediction.

Finally, affirming the determination of factors that cause deviation in predictive accuracy will warrant better strategies for model improvement and adjustments. Shocks, liquidity, and investment volume may subject modeling to market-related risk. The model’s performance may even be negatively affected by market-specific characteristics, which thus calls for a more detailed and sophisticated experimental design.

LSTMs have become a widely used approach, as evidenced by the literature, in the financial markets analysis, such as stock markets, commodity markets, and crypto markets as well. Each of these markets are made up of a set of characteristics, dynamics, and factors. In research paper by Fischer and Krauss (2018) covering LSTM models, including LSTM networks and Convolutional Neural Networks (CNNs), to predict market returns were investigated. The research required the collection of historical stock market data, what inclusive was pricing and volume data, and then training the LSTM model for future forecasts and comparing the results with traditional forecasting methods.[[2]](#footnote-2) In the research from Xu and Wu (2019), a hybrid ensemble model that combines ARIMA, GARCH, and artificial neural networks (ANN) were used to do stock market prediction. The research conducted revealed that the ensemble model (ECM) was better than the individual models since it contained the short-term and long-term dependencies in the data.

* **Can the effective forecasting of market returns using LSTM model serve as evidence against Random Walk?**

The Random Walk Theory is one of the most important concepts in financial economics that postulates that stock prices follow what in economics is known as the random walk and they are therefore unpredictable. Ultimately, it states that the future prices cannot be clearly predicted by historical data only. This theory underpins the belief that stock market related predictions through technical analysis or some other mean, is similar to gambling without any reliable long-term profits. Jegadeesh & Titman's (1993) paper questions the Random Walk Theory concerning the stock return patterns and how these patterns steer the profitability of trading strategies. They find empirical proof of short-term tendency and long-term reversal effect that does not conform to market efficiency. On the other hand, the appearance of machine learning called Long Short-Term Memory (LSTM) models causes a reassessing of this theory, though. LSTM models, which are a variant of recurrent neural networks (RNN) are especially great choices for sequence analysis problems with data that has long-term dependencies. These machines can do natural language processing, following the sequence of events, and forecast in the financial markets. Liew and Soh's (1998) paper discussed the ability of neural networks – including LSTM models – to predict short-term interest rates. It compares the forecasting accuracy of the neural networks with that of the Random Walk Model and it shows by means of that how the market efficiency and the investment strategy can be influenced.

The research question to be examined is whether LSTM models, constituting as an effective method, can be used to forecast market returns thus disproving the Random Walk theory. To be precise, if the LSTM model has the ability to make accurate future market returns based on past records, it casts a doubt on the theory of stock prices following the random walk. This question is addressed by employing an LSTM model in Python to predict market returns for the next seven days. The model uses historical market data that is subject to preprocessing and scalation before it can be trained. Once the model has been trained and its performance assessed with a part of the dataset, the model's prediction for future market returns is generated using this newly constructed model. If the LSTM model is able to generate predictions which are quite similar to the actual market returns during the forecast horizon period, it will imply that there are some detectable patterns/ dependencies in the data that the model has learnt to use. These movements can express the hidden stock market processes, different types of investor actions, economic statistics or any other elements that can affect stock prices. The importance of this result is significant. This means that market prices are not just completely random and that at least partly they can be predicted using innovative machine learning technology like LSTM models. This contradicts the efficient market hypothesis and implies that there is possibly room for investors to capture those predictable fluctuations of the market. Nevertheless, the results should be treated with reluctance. Although LSTMs approach may predict certain situations of the market or assets accurately, their power varies from case to case and the whole situation depends on certain factors. However, their predictive ability can vary, because of a variety of factors. These factors being the quality or the quantity of data available, the choice of feature and hyperparameters, the model architecture and the recent developments in market condition can influence the successful analysis. Moreover, although LSTM models may be able to forecast market returns to some extent, this does not automatically mean that investors can outperform the market or display superior performance. The study of Bao et al. (2023) presents a new line of work by considering the participants’ predictions on the stock price series as well as on the random walk sequences. The report indicates that investors are likely to excessively react to prices swings with fewer rejections but significant reactions that align with those in related behavioral studies. Therefore, the data which is included in the stock time series also develop our understanding of forecasting decisions in the context of finance. To conclude, the results help us comprehend how investors can forecast unpredictable directions in the markets. The profitable market hypothesis suggests that under efficient market conditions, any predictable patterns or anomalies would be quickly spotted and utilized by market participants, which would subsequently lead to such patterns or anomalies vanishing. Mandelbrot and Hudson's (2004) paper was a turning point in the financial theory with their fractal-based modelling approach that presented strong arguments against the Random Walk model and placed light on the complexity of dynamics of financial market.

Overall, using LSTM models to forecast the accuracy of market returns might be contrary to the Random Walk Theory, though it is only one of many studies pursuing the discussion on whether the markets are efficient and predictable or not.

* **What’s the impact of different parameter optimization techniques on the accuracy of market return forecasts using LSTM model?**

The emphasis that is placed on parameter optimization techniques that are designed to maximize the precision of forecasts of returns to market from LSTM models is very significant for this research question. The utilization of ML parameter optimization methods, namely hyperparameter tuning, regularization, and the early-stopping ones, can highly contribute to improving predictive capabilities of LSTM models due to the fact they allow the best model parameters to be chosen, the overfitting to be avoided, and to achieve faster convergence. In their paper, Zaremba et al. (2014) explore regularization techniques such as dropout and weight tying mechanisms, which help reduce overfitting in LSTM models. Through employment of these regularization techniques the authors attain more generalizability and diminish the complexity of the problem of overfitting in their deep learning architecture.

In hyperparameter tuning, there are two particles that work together to find the best hyperparameter set. The first one describes the architecture & behavior of the LSTM models. It is the hyperparameters like the number of the hidden layer, the number of neurons in each layer, the learning rate, and the number of records in the batch that can greatly affect the performance of the model. Adam optimizer and mean squared error loss are examples of the techniques that can be utilized to perform the search of hyperparameter space so as to find the optimal configuration that gives the highest accuracy. Through regularization techniques including dropout regularization and L2 regularization too overfitting can be prevented, and LSTM models enhanced with better performance in the generalization domain. Dropout Regularization Update Deletes neurons randomly during training prevents neurons co-adaptation and improve model robustness. The reason for the L2 regularization is that it punishes heavily weighted parameters in the model, significantly reducing the susceptibility of the model to noise in the data and improve model stability simultaneously.

The mechanism of arresting early is an important regularization method which applies the signal that the performance of the model on a validation set starts to deteriorate and normally stops the training process. Instead of allowing the network to over-learn by processing the data continuously, the early stopping process stops training the networks based on the monitoring result of the validation loss to ensure that the model generalizes well on the unseen data. One way to implement early stopping is via monitoring of the validation loss and terminating the training when the loss does not improve for the defined number of epochs.

To figure out if different parameter optimization techniques can be applied to the LSTM model to forecast market returns being more precise, empirical analysis and performance assessment is required. A study by Greff et al.'s (2016) presents a step-by-step guide on designing LSTM architecture as well as optimizing it based on different search strategies aimed to arrive at a high performing LSTM configuration. It underlines the significance of hyperparameter tuning while displaying the LSTM space search as an improvement tool for various tasks. LSTM networks with specific parameter optimization strategies are trained and tested with historical price and market data using metrics such as mean squared error (MSE) and accuracy rate to measure the predictive accuracy in their performance. Validation for each parameter optimization technique can be done using comparative analysis techniques to evaluate their effectiveness for bettering the model or reducing prediction errors. With that in mind, we implement a sensitivity analysis to check the LSTM model’s reliability considering the alterations made to hyperparameters and optimization method. This is done by systematically changing hyperparameters while evaluating which ones will be the best fit for the model. In this way, optimal ways of improving the accuracy and stability of predicting market returns are discovered. Pascanu, Mikolov & Bengio's (2013) research explored the issues that occur with training the recurrent neural networks (RNNs), one of them being the vanishing and exploding gradient problems. It suggests solutions such as gradient clipping and adaptive learning methods that help overcome these limitations of RNN, and thereby establishes the foundation of stable and efficient deep RNN training.

Through the investigation of the influence of different types of optimization parameters on the forecasting model's precision by using LSTM model, the research aims to explore which type of parameter optimization techniques should be considered while selecting the optimal parameter optimization strategies for financial forecasting applications. The end results of this study might help practitioners and researchers to know the steps to follow and to advance on matters regarding the improvement of LSTM models and the strength of market return forecasts.

**2. LITERATURE REVIEW**

The literature review section of this thesis is an important part that lays the foundations of this research. It includes analyzing and evaluating previous scientific papers, research papers, books and other relevant materials on market return forecasting, LSTM models and optimization techniques. Predictive analytics is a topic of great significance for researchers throughout the world, especially in forecasting non-stationary chronological data such as stock data.

**2.1 Efficient Market Hypothesis**

The primary contribution of Fama (1970), a study about the EMH, was one of the most prominent papers in financial literature. Fama categorizes markets into three forms of efficiency: weak, semi-strong and strong. Weak form means that the past prices already cover the inherent traits of historical data, semi-strong form refers to the idea that public information is mirrored in stock prices and strong-form efficiency leads to the situation where stock prices encompass all information at hand, including the proprietary information. Fama's paper has come to be a keystone of the market efficiency studies while denoting that asset pricing and information flow are closely related. Malkiel's (2003) book is a classic text that popularized the EMH and its implications for investment strategies. The paper focuses on evaluating the empirical support for the EMH and addresses the detractors’ argument from various points of view, among them being behavioral finance. He claims that chances of successfully beating the market through methods like active management and market timing may not be constantly successful, because stock prices strictly follow the random walk and account for all relevant information. Malkiel’s contribution of book gives a lot of knowledge and considerations for both investors and economists who try to understand market efficiency as well as its consequences.

The book by Campbell, Lo, & MacKinlay (1997) is both comprehensive and in-depth, reviewing financial market data as well as the modeling techniques. In the opening chapter of their book, Campbell, Lo, and MacKinlay cover asset pricing models, market efficiency, and empirical forecasting techniques relevant for stock returns. The subject of the paper is an explanation of using econometric methods in financial data and giving concrete recommendations on how to conduct empirical research and what conclusions can be drawn based on such data. The book has been hailed as a vital resource by researchers and practitioners in finance through which they better understand the steps of financial planning and investing. Granger's (1999) study on empirical modeling elaborates on the tools of forecasting stock prices and the rationale behind testing whether financial markets are well regulated or not. Granger stresses the need for model selection and the implementation of various evaluation techniques in model specification. He goes beyond just the theory by discussing topics like model selection, parameter estimation, and hypothesis testing and gives a practical guide to economics. Granger's work has proved to be a powerful contribution to empirical economics, and it is still strongly underlying the research agenda in this subject area. Engle & Granger (1987) broke through with their innovative theoretical construct of co-integration through a pioneer paper, which is necessary for building the economic relationship between finite time series data. Engle and Granger elaborate the study of co-integration and purport error correction models to be utilized in the investigation of non-stationary time series. The co-integration analysis contributes as forecasting measures in stock prices and an efficient test of financial markets. The paper had a noteworthy effect on economics theory and has given rise to a multitude of financial and economical empirical studies.

Andrew's (2004) work on Adaptive Markets Hypothesis (AMH) challenges the old market efficiency view, by expanding it to include evolutionary biology and psychological insights. He contends that the actions of market participants and their behavioral and learning processes get formed by evolutionary pressures and are not in compliance with perfect efficiency. The AMH serves as a comprehensible framework for studying the mechanisms of market adjustment and adaptability to non-stationary environments which involve obtaining new information. Tsay’s (2010) combines all important time series analysis techniques applied to the financial field in her book. The author analyzes a wide array of topics, including autoregressive models, volatility, and forecasting methods. The book aims to provide the tools necessary for financial analysts who are looking to analyze financial time series data. It also provides a practical exposure to statistical techniques and how to apply them to real-world financial problems. The work of Tsay is widely used as a textbook for finance courses, and it represents a major value for those researchers and practitioners who are interested in financial econometrics. The study by Aminimehr et al. (2022) carefully studied various parts of financial market forecasting. It discusses how research on market prediction has changed over time, moving from the Efficient Market Hypothesis (EMH) towards more modern machine learning techniques. Yang et al. (2020) proposed an ensemble forecasting framework for stock returns that incorporates several machine learning models such as random forests, gradient boosting, and support vector regression. The ensemble strategy improved forecast accuracy while surpassing individual models at capturing the data's complex connections. Bao et al. (2017) used LSTM networks and attention processes to forecast stock market returns. The author demonstrated how the attention-based LSTM model improved prediction accuracy by focusing on key information and collecting crucial data properties. Song et al. (2020) research was also focused on the use of deep learning models to predict cryptocurrency prices. The author emphasized the benefits of LSTM networks in capturing temporal dependencies, dealing with non-linear patterns, and making correct forecasts in cryptocurrency markets, where standard models typically fail owing to excessive volatility and non-linearity. Khan's (2024) research contributes to the knowledge on stock market prediction by showcasing the efficacy of the LSTM model in accurately forecasting stock prices. The research seeks to employ sophisticated machine learning algorithms to accurately estimate fluctuations in stock price, highlighting the significance of parameter tuning and model selection. The study examines the Long Short-Term Memory model, a form of Recurrent Neural Network, the Facebook Prophet algorithm for forecasting time series, and the Random Forest Regressor model. These models are usually applied to datasets that include stocks from both the Dhaka Stock Exchange (DSEbd) and worldwide titans, sourced from the DSEbd website and Yahoo Finance, respectively. The paper’s key findings reveal that the Facebook Prophet algorithm is successful at catching long-term trends and seasonality, such as daily, weekly, and annually patterns, along with holiday effects. The LSTM model is renowned for its high accuracy, obtaining outstanding outcomes in evaluation criteria such as Root Mean Squared Error, Mean Absolute Percentage Error, and Mean Absolute Error. The LSTM model's result is particularly outstanding, since it showcases how successful it is in anticipating upcoming stock prices over a 15-day period. The discoveries of this article demonstrate that the LSTM model has the ability to forecast the closing price of the next day, thus confirming its significance as a useful tool in predicting market returns. This research adds to the expanding field of stock market prediction accuracy and its impact on investment decision-making by showcasing the effectiveness of ML techniques in this particular domain.

The efficacy and feasibility of LSTM models for forecasting market returns are further supported by these studies, where the researchers utilized LSTM models to predict intraday stock price fluctuations. This demonstrates the model's capability to identify short-term trends and execute intelligent trading strategies. LSTM models have been used in a variety of financial disciplines, including stock markets, commodities markets, and high-frequency trading, as well as market returns, energy markets, and cryptocurrency markets. They have showed advantages in dealing with temporal dependencies, integrating several data sources, and improving prediction accuracy.

**2.2 Random Walk Hypothesis (RWH)**

The book written by Sornette (2003) focuses on the science of stock market crashes from a complex systems viewpoint that is discrepant with the conventional view that financial assets follow a random walk. Sornette (2003) claims that financial markets, as many of us know them, are not 100% efficient, and have, on more than one occasion, fallen into crisis, even though some financial experts have always assumed that the financial market is always forward-looking and effective. Physics and network theory's concepts are used here to understand the working mechanisms of the market; besides that, it provides an explanation of the limits of the traditional predictive models. Didier Sornette's research contributes to the understanding that disregard of such complexities may have disastrous consequences on the market forecasts. Shiller (2015) book unveils the role of the irrational side in the financial market and the theory of random walk used in it. According to Shiller, the investor sentiment and speculative bubbles are the driving forces underlying asset prices, making them diverge from their fundamental values and thus from the market efficiency. He explains historical cases of a bubble market (the dot-com bubble and the housing market bubble) and advances the crucial role of investor psychology in predicting future market behavior. Shiller shows that the random walk theory cannot reflect all aspects of market behavior since it is only a simplified version of the process. The issue that Hansen & Sargent (2018) address in their paper is the indisputable importance of uncertainty in economic modeling and its impact on financial markets. The scholars mentioned the fact that the existing models that depend on rational expectations and perfect foresight assumption (as well as other assumptions) could result in deviations from the random walk hypothesis and thus the uncertainty and learning behavior should be considered as a factor in forecasting process. The paper explores the complexities of market activity and the challenges offered in forecasting asset returns.

The study done Chitenderu, Maredza & Sibanda (2014) on the Johannesburg Stock Exchange (JSE) statistical study offers empirical data evidence that the RWH is applicable to stock prices. In their study, Chitenderu et al. (2014) examined the degree to which the ALSI (All Share Index) of the JSE followed the RWH over the period from 2000 to 2011. Stock prices analysis for such financial time series adopts various statistical methods such as unit root tests, autocorrelation tests and ARIMA modeling, among others. Unit root tests, for example the Augmented Dickey-Fuller test, are inspections that are popularly employed to find out if time series is stationary or follows a random walk. The research findings of Chitenduru et al. prove that the local stock market follows patterns of a random walk model. Random walk is autocorrelated and cannot be predicted. Hence, the local stock market is efficient in the weak form. Moreover, the modelers utilized the ARIMA (AutoRegressive Integrated Moving Average) method of modeling to detect if there may have been any recurring patterns or trends. The ARIMA (1, 1, 1) process evolves as the best model for this data set, which also shows the random walk possibility. The positive news is that Chitenderu et al. (2014) also run residual tests on the ARIMA model, and the residuals displayed similar features of a pure random process. The authors support their conclusion that the stock prices within the JSE follow a random walk pattern that features unpredictability, producing forecasting inefficiency. Time series analysis is broadly used in financial forecasting. To catch temporal conditions and anticipate market returns, analysts have utilized methods, for example, ARIMA models (Box et al., 2015), GARCH models for unpredictability gauging (Bollerslev et al., 1992), and state-space models (Durbin and Koopman, 2012). Autoregressive Incorporated Moving Normal (ARIMA) models are mostly used to predict market returns. Papers by Mills and Coutts (1995) and Lahmiri et al. (2015) have shown that ARIMA models operate successfully in catching temporal patterns and forecasting short-term returns. Mills and Coutts (1995) did an analysis on the predicting effectiveness of several time series models, including autoregressive (AR), moving average (MA), and autoregressive coordinated moving average (ARIMA) models, with regards to stock returns. The authors found that ARIMA models beat different models in estimating short-term returns.

In order to test their hypothesis, the authors perform a variance ratio test considering heteroskedasticity which is a measure of market efficiency by quantifying how fast and accurately information gets disseminated and the prices adjust. The findings of this test further reinforce the efficiency of JSE in the weak form since random movement of the prices hypothesis was not opposed. Godfrey, Granger, and Morgenstern, authored the paper "The Random Walk Hypothesis of Stock Market Behavior", published in 2007, and they have contributed to the continuing debate surrounding the validity of random walk hypothesis and its application to stock market price movements. Authors come up with a sample where the price of a share at any time t, denoted by x\_t, is a result of a sequence of independent random variables; these are represented by e\_t. This model is called the random walk. The latter, however, suggests that price changes are driven by mechanisms following a stochastic process, with changes being independent of previous fluctuations. The investigation of the probability distribution function of the random quantity e\_t, which is amongst the major highlights of the research, will be presented in the following sections. Nevertheless, the book offers the contrary to the previous research, distribution function is still described as a normal one. Such a discovery leads to the shape of stock market price random movements being similar to how you would draw a Gaussian distribution. This finding has been very useful for modeling and forecasting.

The work is designed to illustrate the diversification of stock market prices in respect to time intervals- minutes, hours, and days- through daily observations and transaction-level data. According to the scholars, the random walk model is crucial to the fact observed in the student work and it can naturally explicate the variations of the prices for different sized time intervals. This illustrates that stock prices behave in a way that cannot be predicted and consequently represents the basic, random walk model. Additionally, it has been noted that leveraging the price making mechanism is particularly interesting during the times when the market is not trading. Considering trading is not taking place during the inactivity of matching period, the mechanism keeps working and doing its operations at a slower speed. This finding shows that market dynamics still exist outside of trading hours. Kumar and Ravi (2018) have also predicted stock returns using ensemble forecasting, which incorporates forecasting techniques such as ARIMA and ANNs, and support vector regression. According to the researchers, the ensemble approach beat individual models and yielded more precise predictions. Dixon et al. (2019) studied how LSTM models may be used to anticipate market indices in the cryptocurrency sector. The scientists compared LSTM models to standard time series models and discovered that LSTM models performed better at capturing volatility and irregular patterns in bitcoin market indexes. Tsantekidis et al. (2017) employed LSTM models for forecasting electricity market prices. He compared LSTM to various forecasting techniques and discovered that LSTM models outperformed in terms of capturing pricing patterns and generating accurate predictions in the electricity market. Singh and Kaur (2019) studied the utilization of deep learning models, including LSTM networks, to predict stock market returns. They investigated the advantages of LSTM models for capturing temporal dependencies, addressing non-linear interactions, and adding features to improve stock market forecast accuracy.

Agrawal and Biswas (2021) did a study on the employment of LSTM models in predicting stock market indices, particularly focusing on the Nifty 50. The authors analyzed the efficacy of LSTM models and discovered that LSTM models exhibited the ability to capture intricate patterns and non-linear correlations within stock market data, in comparison to other forecasting techniques. Consequently, this resulted in enhanced prediction performance. The model's overall performance improves as its design adjusts to the ARO algorithm's variables. In terms of predicting accuracy, LSTM-ARO outperforms LSTM1D, LSTM2D, LSTM3D, ANN, and LSTM-GA networks. This makes it an essential tool for traders and investors. Alaghi and Yu (2020) used LSTM models to forecast stock market indexes, such as the Dow Jones Industrial Average. Researchers compared LSTM to other deep learning models and discovered that LSTM models were better at capturing long-term dependencies, dealing with noisy data, and improving stock market index projections. Yao et al. (2019) used LSTM models to forecast stock market indices, especially the Shanghai Stock Exchange Composite Index (SSECI). The authors found that LSTM models outperformed standard time series models in terms of accuracy, capturing long-term dependencies, and coping with non-linear patterns in stock market data.

* 1. **Traditional Models vs Long Short-Term Memory Model (LSTM)**

For a long time, the most usual forecasting models have been used in quantitative analysis not only in the field of finance, but also economics and business. These models, which are the results of statistical theory and time-series analysis, have provided valuable insights on historical tendencies, patterns, and relationships within data. The main principle of conventional forecasting is the translation of past records to the future to prepare decision-makers and planners for the future. Traditional forecasting is embedded in a variety of statistical tools, one at a time being another one for the diverse data under analysis. Of these methods ARIMA, exponential smoothing (ETS), and linear regression are the main solutions. These models use historical data to recognize underlying patterns, trends, and seasonal variations, which helps them to forecast impacts of environmental events on the future. Among the main strong points of the conventional forecasting models is the fact that they are quite simple and clear. Using clear approaches and specified assumptions, these models provide the audience with a clear picture about how the data works and moves. Therefore, a wide deployment of tools and resources in various industries has triggered the development of toolset, resources, and best practices which has resulted in the easy engagement of practitioners and analysts. Nevertheless, the traditional forecasting models face some defects too. Development of their models based on linear relationships and stationary assumptions may result in loss of their ability to catch the complexities of actual world data. Besides, the traditional models usually prove to be inept in dealing with nonlinearity, seasonality, and randomness, especially in the domains characterized by high fluctuations and swift variations.

On the positive side, LSTMs that were invented recently to build neural networks to capture temporal dependencies have been used with great success in predicting processes where ordering matters. As a member of the RNN family, LSTM models are fed with sequential information and use a mechanism of storing an internal memory state to capture the dependencies in the temporal data. In contrast to the linear structures, LSTM networks do not presuppose exact mathematical formulations for complex patterns in data. This way, they can match the structure of the learning problems they face. This type of architecture is very powerful because LSTM models can be very effective in capturing long-term memory dependencies, excel in dealing with nonlinear relationships, and are able to adapt to changes in the patterns of sequential data. A notable advantage of LSTM models over traditional ones is their auto-learn property that involves learning features and representations naturally, unlike the manual one. This capability is what makes LSTM models learn the behavior of data, discover hidden patterns and nonlinear relationships, which are sometimes not very clear from traditional methods. These neural networks also have the capacity to manage uncertainties within data like missing values, noise inputs, and non-stationary trends, thus making them strong and applicable in the real-world trial and error.

Fischer and Krauss (2018) investigated how deep learning models, including LSTM networks and Convolutional Neural Networks (CNNs), predicted stock market returns. The study examined historical stock market data, including price and volume, and prepared it for use in deep learning models. To assess their relative usefulness in predicting market returns, the authors compared the performance of these deep learning models with that of traditional forecasting methods such as ARIMA or regression. The authors discovered that LSTM models outperformed traditional models in capturing temporal dependencies and enhancing prediction accuracy. Zhang et al. (2018) established a deep learning system called Echo State Networks for anticipating stock market returns. He compared ESNs to traditional models and discovered that ESNs behaved better in terms of prediction accuracy, particularly when dealing with large-scale financial time series datasets. Huang et al. (2019) proposed a hybrid approach for forecasting stock market returns using wavelet decomposition, stacked autoencoders, and LSTM networks. Traditional models were surpassed by hybrid models, which produced more accurate predictions by accounting for both short-term and long-term associations in the data. Jiang et al. (2020) used a deep learning model named Deep Belief Networks (DBNs) to forecast market returns. The authors compared DBNs with traditional models and noticed that DBNs are more successful in capturing non-linear relationships and boosting forecasting accuracy. In his study, Dang et al. (2020) explored the implementation of a hybrid model that integrated LSTM networks with fuzzy logic for anticipating stock market returns. This hybrid model surpassed classical models by blending unclear rules and detecting the tricky dynamics of financial markets, concluding in more accurate forecasts. Gao et al. (2020) analyzed the use of LSTM networks to anticipate stock market behavior. Researchers drew an analogy between LSTM models and traditional time series models and noticed that LSTM surpassed the classic models in terms of predicting accuracy, long-term dependencies, and non-linear patterns in stock market data. Meanwhile, in their research Kim et al. (2018) employed the LSTM models for forecasting cryptocurrency returns. The author did a comparison between LSTM and traditional forecasting models and noticed that LSTM outperformed in capturing the volatility and non-linearity of cryptocurrency price fluctuations, resulting in more accurate predictions. Singh et al. (2018) recommended an LSTM-based deep learning model for stock prediction. The authors of the study compared the LSTM model to traditional time series models and demonstrated that it can identify long-term dependencies, handle non-linear relationships, and make accurate forecasts in stock market data. Nair et al. (2018) employed LSTM models to forecast market returns for the Indian stock market. He compared LSTM models to traditional time series models and noticed that LSTM models performed much better in terms of predicting accuracy, capturing intricate patterns, and dealing with non-linear correlations in market return data. In their 2020 research, Nair and Priyadarshini utilized LSTM models to predict market returns in the Indian stock market. To increase forecast accuracy, the authors utilized technical indicators and sentiment analysis in their LSTM model. The LSTM-based method outperformed traditional models in predicting market returns. Tang and Luo's (2020) research focused on forecasting stock market returns using LSTM models and attention processes. The authors suggested introducing attention mechanisms into the LSTM architecture to improve the model's ability to concentrate on key features and time intervals. This based on attention mechanisms LSTM model, outperformed traditional models in predicting stock market returns. The paper by Li, Kamaruddin, Yuhaniz et al. adds to the existing literature by introducing an innovative hybrid approach for stock price prediction that blends LSTM neural networks with symbolic genetic programming. The researchers aim to overcome the challenges of feature engineering and data incorporation deep-rooted in traditional LSTM models by blending SGP's evolutionary computing potential with the deep learning capacity of LSTM. The proposed approach has the power to boost prediction accuracy and robustness by combining LSTM networks with SGP algorithms. Moreover, introducing symbolic genetic programming (SGP) into the modeling process makes it easier to find relevant patterns and features in stock market data, improving the interpretability of the forecasting model.

Tsay and Tiao (2001) also examined closely the forecasting ability of stock returns using ARIMA and GARCH models. The research showed the ability to predict stock returns, especially at shorter time periods, and pointed out that including GARCH models may take control of wild up and down prices clustering and further improve the return predictions. Generalized Autoregressive Conditional Heteroskedasticity models are usually used to forecast prices fluctuations and market returns. Bollerslev et al. (1992) and Engle and Kroner (1995) have proved that GARCH models may identify volatility clustering, while also providing reliable market return predictions. Poon and Granger (2003) investigated how regression models predicted volatility. To capture price movement patterns inside the regression model, the authors employed ARCH and GARCH models. They found that incorporating GARCH models improved the accuracy of volatility forecasts. Regression models are commonly employed in market return predictions. To explain cross-sectional differences in stock returns, Fama and French (1988) proposed a three-factor model that considers market risk, size, and value. Chen and Lee (2012) and Abu-Mostafa et al. (2008) employed regression models to estimate returns based on market indices and financial ratios, respectively. The authors have studied the historical data from the S&P 500 index for a time period from September 2000 to February 2021, by doing statistical tests to evaluate data properties such as stationarity and normality. They proposed a method for selecting prediction models, designing features, and preparing data. Various research are done to assess the efficacy of different modeling methodologies, including neural networks, PCA, and Random Forest variable selection. Statistical tests are run to investigate differences in predicting precision, and metrics such as Mean Absolute Error (MAE) are used to assess the models. The results emphasize the pros and cons of each method and provide insightful perspectives for future research. The research enhances our comprehension of market trends and the ongoing progress in predictive modeling techniques for financial market analysis. Econometric models have played a pivotal role in financial forecasting. Lutkepohl (2005) has carefully studied the interdependencies between different financial variables and market returns have been forecasted using VAR models. Xu and Wu (2019) forecasted stock market returns using a hybrid ensemble model that combined ARIMA, GARCH, and artificial neural networks. The ensemble model outperformed individual models in terms of prediction accuracy, taking into consideration both short-term and long-term interactions in the data.

The LSTM model, along with other deep learning models, has come to be very important in predicting market returns. Shen et al. (2020) used a hybrid strategy to anticipate stock market returns, combining ARIMA models with deep learning techniques such as long short-term memory (LSTM) networks. The study showed that the hybrid model outperformed standard models and enhanced the accuracy of stock return estimates by considering both short-term and long-term dependencies in the data. This study forecasted stock market returns adopting a hybrid model that incorporated LSTM networks and PCA. Li and Wang (2020) focused their research on the application of LSTM models to forecast stock market indices, namely the Hang Seng Index. The analysts compared LSTM to other machine learning models and discovered that LSTM models caught chronological dependencies, nonlinear correlations, and market sentiment, resulting in more accurate stock market forecasts. Wei et al. (2018) made trading decisions and predicted stock prices using an LSTM-based deep learning model. The authors improved the LSTM model by using technical indicators and sentiment analysis, to capture market mood and boost forecast accuracy. The LSTM-based model accurately forecasted stock prices and delivered efficient trading signals. In their 2019 article, Akita and Yoshizawa implemented an LSTM-based deep learning model to forecast stock prices and make trading decisions. To increase forecast accuracy and trading performance, the authors integrated technical indicators, news sentiment analysis, and data from the markets to the LSTM model. The LSTM-based model generated profitable trading signals. In their study, Gunduz and Karakaya (2020) used LSTM models to predict market returns in the Turkish stock market. The authors studied the way different input variables and model configurations affect the prediction accuracy of LSTM models. The findings showed the importance of LSTM models at capturing intricate patterns and improving the predictions on market return.

Brownlee (2018) examines LSTM models in time series forecasting, including market returns. It investigates the architecture, training method, and several strategies for increasing LSTM performance, including stacked LSTM layers and bidirectional LSTM. The study investigates the benefits and drawbacks of LSTM models for market return forecasting. Meanwhile, Chong and Han (2017) investigated the application of LSTM models to predict the direction of stock market volatility. The authors used technical indicators and news sentiment as input variables and discovered that LSTM models outperformed conventional models in predicting stock market returns. Siami-Namini et al. (2019) predicted future agricultural commodity returns using LSTM models. The study compared LSTM models to other machine learning approaches and discovered that LSTM models outperformed other techniques in detecting the underlying dynamics of agricultural commodity costs and improving return forecast accuracy. Liang et al. (2021) used LSTM models to forecast daily stock returns for several equities. The authors used financial and linguistic data as inputs to determine sentiment and company-specific information. The LSTM model produced promising results in forecasting stock returns, especially when textual data from news items was included. Aldridge and Krawciw (2017) studied the application of LSTM models in high-frequency trading. Liu and Ziji (2023) focus their study on the use of machine learning techniques, particularly on the LSTM model, to predict stock values. The paper underlines the stock market's appealing features to investors looking for potential market returns, while also acknowledging the risks involved. The goal of implementing advanced machine learning techniques is to increase forecast accuracy and assist investors in making smarter decisions. Previous research has examined several machine learning algorithms for stock market prediction, but Liu's study focuses on the LSTM model given its ability to capture long-term dependencies in sequential data. It adds to the literature on stock market prediction by proving the effectiveness of the LSTM model in predicting stock prices.

* 1. **Parameter Optimization Techniques for LSTM Model**

ANNs and Deep Learning Models, conversely, are extensively used to forecast market returns. Deep learning models have shown enormous promise in finance, including computer algorithms for stock market predictions, portfolio optimization, fraud detection, and algorithmic trading. Gülmez's (2023) research proposes LSTM-ARO, a novel approach to stock price prediction that combines deep LSTM networks and the Artificial Rabbits Optimization (ARO) method. LSTM-ARO predicts stock market prices using DJIA index data from 2018 to 2022, with each prediction including 20 prior days' values.

The paper by Öztürk (2023) investigates LSTM (Long Short-Term Memory) Neural Networks' efficiency optimization for time series prediction by presenting the concept of a parallelized classic LSTM (PCLSTM) model and analyzing the derivative-free optimization methods for the tasks of hyperparameter tuning. It starts by showing the importance of sequence data analysis and the role of the LSTM which is now becoming a fundamental tool in the prediction of sequential data. These topics highlight the importance of parameters optimization and parallel approaches to enhance the performance of LSTM models for time series predication tasks. The major contribution of the study lies in the creation of PCLSTM that consists of pre-rating, core-based parallel, and prediction stages. The study examines different types of derivative-free optimization methods to assess their capability to reduce the CPU time as well as increase the accuracy of prediction. Measurements show that the presented PCLSTM approach is superior to other approaches both in terms of performance metrics such as RMSE and CPU time. The author eventually suggests that further research can be extended to include the comparison of various optimization methods, as well as the application of LTSM architectures of different types.

Yadav, Anita, Jha and Sharan’s (2020) research article focuses on predicting the price of stocks in the Indian market using Long Short-Term Memory (LSTM) neural networks. The abstract is an introduction to the LSTM deep learning model, which is considered a popular one and is generally used for time series prediction, which is specially the case of the stock market data, which has a structure of long-term trends, seasonal deviations, and random noise. The last paragraph of the abstract highlights the need for optimization hyperparameters in order to get significant performance from the LSTM models. The introduction of the paper sets the scene that serves two purposes i.e., to provide the context of stock price prediction for investors to make wiser decisions and to increase profits in stock trading. It contrasts the two traditional approaches to prediction: technical analysis, which uses previous data to spot patterns in prices, and fundamental analysis, which considers economic and business factors. The complexity of the stock prediction is discussed throughout the introduction due to its volatile, non-linear nature and external factors. In the literature review, the topic underlies the previous research on time series prediction in financial markets with the emphasis put on the Indian stock market. It covers various techniques that are applied for stock price prediction, and they include classical statistical methods, ML techniques, and deep learning approaches applicable such as LSTM. There are some strengths and weaknesses of available modeling methods for renewable energy integration. This is reflected in the review article while identifying gaps in the existing literature such as the need for more accurate and robust models, adaptation to volatility, and optimization of model parameters. This literature review encompasses previous research on stock price prediction, the specific challenges in stock price prediction which are unique to the Indian market, the application of LSTM for financial time series prediction, and parameter optimization techniques.

The research of Rokhsatyazdi, Rahnamayan, Amirinia, and Ahmed (2020) addresses the issue of stock market prediction by proposing a new model that is based on the Long-Short Term Memory (LSTM) neural networks. The introduction sets the scene for the modern financial market, which is characterized by an oversupply of up to the minute data, as well as underlining the fact that data analytics and prediction are critical in finance. The aim of the paper is to use the Differential Evolution (DE) algorithm which is an efficient evolutionary algorithm to optimize ten network hyperparameters related to the temporal patterns of the dataset for predicting the next day's stock price. The proposed model is compared with three well-known statistical methods: ETS, SARIMA, and NAIVE, that are considered baseline models for measuring the performance of the developed LSTM model. As a result, the authors tune LSTM model hyperparameters via the usage of the differential evolution (DE) algorithm in order to gain a lower value of RMSE for a prediction. The optimized model does better than the statistical methods and, therefore, it adds to the reputation of it as a high-quality stock price forecasting tool. Authors have also discussed the influence of training epochs on model performance, as well as computational efficiency, which reveals the dilemma between the model complexity and accuracy. The study’s conclusion and future work section present the results of the investigation, as well as suggesting future ways to proceed. The LSTM network modified through optimization provides a good example of stock price prediction, outperforming the well-known statistical models. Future research directions include working with parallel computing to boost speed, building hybrid models like CNN-LSTM or ARIMA-LSTM, and improving the DE approach by optimizing the parameters of the algorithm employing different optimization approaches. Bollerslev et al. (1992) studied the capability of GARCH models to anticipate market movements and returns. According to the study GARCH models may successfully recognize price trends and deliver accurate market return predictions. In another study, Lam, and Li (2008) used structural equation models (SEMs) to anticipate market volatility. These SEMs used a variety of financial and economic data to depict the complex linkages that exist within financial markets. Zhu et al. (2009) used Bayesian regression models to forecast stock returns. Their findings emphasized the improved forecasting accuracy and identification of parameter variability achieved by utilizing Bayesian methodologies such as Bayesian model averaging and shrinkage priors. Scott and Varian (2014) employed Bayesian structural time series (BSTS) models to forecast economic and financial variables, such as stock market performance. The BSTS models used both observable data and past information to make accurate predictions.

Khashei et al. (2011) investigated the effectiveness of ARIMA and ANNs in forecasting stock market indices vs support vector machines (SVMs). The findings revealed that ARIMA models performed well, while ANNs and SVMs gave more exact forecasts, emphasizing non-linear models' capacity to describe complex connections. To forecast stock market returns, Chen, and Zheng (2016) used a weighted ensemble model that incorporated different forecasting methods such as ARIMA, GARCH, and regression. The weighting method was based on the individual models' performance, and the ensemble model increased forecast accuracy. Wu et al. (2019) used a combination of wavelet analysis and support vector regression (SVR) in their hybrid ensemble model to predict stock market returns. This study found that not only did the ensemble approach outperform individual models, but it also resulted in more accurate forecasts, especially for short-term forecasting periods.

**3. RESEARCH METHODOLOGY**

* 1. **Data Collection and Preprocessing**

One of the most important aspects of forecasting market returns, or one of the most integral tasks in finance at large, involves painstaking data collection and accurate preprocessing. The outline of this article explores the inner workings of financial markets, deciphering the hidden patterns of market behavior and eventually grasping the mechanism that produces future returns with unparalleled accuracy. The journey starts with a collection and classification of 20 years-old data that the markets of two decades have made, namely from January 2000 to January 2021, at the beginning. This dataset, sourced from Bloomberg, comprises daily closing prices of six prominent market indices: NDX, SPX, DJI, UKX, SXXP, and NK. These indices represent a wide spectrum of global financial spectrums and give a comprehensive and coherent picture of not only movements and specific trends in the markets, but also provide us with a better understanding and visualization of markets. A secure Excel file becomes home to the dataset and the journey of data preprocessing steps is undertaken. These steps are prerequisites for robust predictive modeling. By making use of Python data libraries such as Pandas, NumPy, and Matplotlib, data is readily loaded into a Pandas Data Frame which in turn is a tabular data structure that supports ease of manipulation and analysis. Pandas puts much focus on the data manipulation and analytics steps like loading of the data and pre-processing of the dataset. Since 'date' is a column among the dataset, it is parsed to a 'datetime' format in order to support chronological sorting and temporal analysis, respectively. In the code block, the Pandas library is used to load data from an Excel file (‘pd.read\_excel’), transform the date column using the function ‘pd.to\_datetime,’ set the date as the index (‘df.set\_index’), and manage missing values (‘df.dropna’). Apart from that, Pandas constructs another DataFrame named forecast\_df to store the forecasted returns, which is further visualized using Matplotlib. NumPy is an underlying module for scientific computing in Python, which offers the support for large multi-dimensional arrays and matrices, along with a set of mathematical functions to manipulate these arrays and matrices quickly and effectively. In the code, matrix programming of NumPy is widely applied in dealing with arrays of manipulation, fitting of data to the model attributes and performing the mathematical operations. To be precise, the data is standardized using MinMaxScaler by executing ‘fit\_transform’ method (‘scaler.fit\_transform’) before training, input sequences are obtained (X) for the training, and the data is subsequently reshaped with ‘np.reshape’ function before loading to the LSTM model . Moreover, NumPy is employed to perform regression, adjustments to returns and also inversion of scaled returns. Matplotlib is a powerful library for creating static, animated, and interactive plots in Python. (“Python programming libraries: Pandas, NumPy, Matplotlib - LinkedIn”) The Python code uses Matplotlib to visualize the forecasted returns. After the creation of the DataFrame of forecasted returns (‘forecast\_df’), Matplotlib is used to create a line plot (‘plt.plot’) by weeks for each index with dates on the x-axis and returns on the y-axis. It results in the visualization of the forecasted returns over the planned time period, and, in this way, trends and fluctuations in the market can be viewed.

When the raw market data enters the system, it goes through a multi-layered process of pre-processing that entails data validation, cleansing of erroneous items, feature engineering, and normalization of all inputs in order to ensure that it is free of any anomalies, inconsistencies and ready for further analysis to enhance its accuracy and reliability. The validation stage requires extensive deep data scanning and integrity verification, resolving data anomalies, revision of missed decision variables, and outliers that may blur the objective of the study. The integrity of the predictive model relies on imputing missing values into the dataset; hence, the Data imputation must be an important consideration before model training. The methodical techniques are applied in addressing the missing values, which is done either by interpolation or deletion. The method employed depends on the nature and spread of missingness. As a result, the data set will always be perfectly clean and without any misplacements or mistakes that could affect the outcome to be predicted. Scaling of features eventually surfaces as a foremost preprocessing technique for dealing with the imbalance in the scales of the features. Then, I use MinMaxScaler from Scikit-Learn library to standardize all features to a uniform range, so that the variables with bigger magnitude will not drown out the rest of the features. This guarantees that the predictive model allocates identical magnitudes of importance to every feature, thus increasing the model's capacity to detect relevant patterns in the data. The LSTM model is capable of replicating previous similar patterns thanks to the time series segmentation predicted, which is one of the most important parts of forecasting returns of the market. For the dataset, I divide it into sequentially structured 31-day windows that capture 30 days of the previous month-end stock market index prices. This indication leads to the widespread use of deep learning networks, which provides the network with rich temporal context, as a result of which it is able to fluidly identify the meaningful patterns and tendencies in the data. Equipped with a thoroughly planned and segmented dataset, the stage is now set up for model training and validation. The LSTM model, one that is best known for capturing long-term dependencies in sequential data, is in this case instantiated and compiled through the Keras framework. The model architecture is composed of several stacked layers for LSTM and Dense, which eventually gives a sufficient model for return forecasting in the market.

* 1. **Data Analysis Techniques**

Time series analysis is the estimation of a time sequence and tracking the change of the phenomenon over the course of time. That is why it is the basis of the whole study which is going to be conducted to forecast the market returns. Within the body of this thesis, time series analysis is rigorously applied to the historical market data involving the NDX, SPX, DJI, UKX, SXXP, and NKY indexes as well. The beginning process of the time series analysis is to load the historical data of the market from an Excel file. This is done using the ‘pd.read\_excel’ from-the function borrowed from the Pandas library. The set of data includes a large number of data points for different time periods, each linked to a specified index. Data loading processes the data into the tabular format, thereby facilitating the manipulation and analysis of the data. It should be noted that the time-series data is time-bound in the sense that temporal sequences play a vital role in analysis. The date column is the basis through which analysis takes place in this context. To make sure that we are using the chronological order of the data properly, we set the date column as the index of the dataset using the ‘df.set\_index’ method. By using it, the dataset is indexed by time, and it will grant easy chronological examination and analysis of the market trends over time.

However, one of the key features of time series analysis is the integrity of data. Final predictions and forecasts are made based on data used and their appropriateness, and having insufficient data or missing data will skew the analysis. Hence, a primary data preprocessing step comprises the dealing with the missing data values in the dataset. The ‘dtype.dropna’ function is used to erase any missing value in a row and make the dataset free from inconsistency and incompleteness. This thorough data cleaning phase is the foundation for the subsequent analysis which will be more grounded and reliable. The next step is performing the Augmented Dickey-Fuller test for each index, which will ascertain if a unit root is present in the time series dataset. The loop 'for column in ‘df.columns[0:]:' is used to iterate over each column excluding the first column supposed to contain dates. This test is tremendously useful since it allows for distinguishing between whether a time series is stationary or possesses a trend. The ADF test for each column produces some output statistics, which include ADF statistic, p-value, as well as the critical values at the different significant levels (1%, 5%, and 10%). A low ADF statistic signifies the high degree of evidence against the null hypothesis that explains the hypothesis of stationarity. The p-value is the significance degree of the ADF statistics. If the observed p-value is less than a threshold (usually 0.05), we accept the alternative hypothesis that the unit root is absent in the data series and, as a result, the series is stationary. On the contrary, a greater p-value means weaker statistical evidence to reject the null hypothesis than the alternative hypothesis, which indicates non-stationarity. LSTM networks are a particular type of RNNs (recurrent neural networks), which are constructed with this aim to deal with data sequences and capture and analyze data patterns lasting over an extended period. However, the implementations of LSTM networks have demonstrated to be priceless for financial market data, which shows a great degree of time dependencies and non-linear relationships, and as a result, provides a very compelling tool for predictive modeling and research. Early portion of script of LSTM code creates the model using the Keras library, defining two LSTM layers followed by the dense layers for regression. The Sequent model is composed of LSTM model architecture which will be created within the Sequential model container. We specify in the input layer of the model the use of the 'Input' function that taking as an argument the shape of the input data that is a sequence of historical market data, represented as a multidimensional array with the dimensions (‘batch\_size’, ‘n\_steps’, ‘n\_features’) where ‘n\_steps’ is the number of time pairs and ‘n\_features’ is the number of input features. A recurrent neural network layer with Long Short-Term Memory (LSTM) processes the input sequence step-by-step. The network state and its memory cells are updated through each time step. LSTM layer numbering is set to 50 that is not lost to them, meaning the output space dimensionality.

The 'return\_sequences' parameter is set to True to the first LSTM layer to return the full sequence of outputs whereas it is set to False to the second LSTM layer only to return the last output. The final layer LSTM is where the output of this takes place as the process goes through a series of dense layers which do regression task by transforming the learned representation space to the actual output space. This training process implies establishing the weights of the LSTM model parameters in a way that minimizes a certain predefined loss function, typically the mean squared error (MSE) in trend forecasting tasks. This utilizes an optimizer, like Adam optimizer which is able to change the learning rate dynamically depending on the gradient of the loss function with respect to the model parameters. The 'compile' method sets the model for training, accomplishing that by tuning the optimizer that will be used during training sessions and the loss function to be used. The model is after that trained by the use of the 'fit' function which is executed iteratively and processes batches of input sequences and updates the model parameters to minimize the loss function, calculating and comparing predictions to the ground truth labels. The optimizer next optimizes the parameters of the model by employing backpropagation which, in general, results in having the model make better predictions at each epoch. Training the LSTM networks requires parameter optimization of the number of LSTM layers and dense layers, aiming to improve the model's accuracy. As a result of training, the LSTM model is subsequently used for generating returns forecasts for future markets by processing information which the model is unaware of. In the code, the model provides predictions for the next time step after the last observed data that have already been seen. Then, those predictions are inverse transformed using a scaler to obtain interpretable market return values. The expected market returns are represented using matplotlib, which in prospects makes the analysis process very clear and provides an in-depth examination of the model performance and predicted market trends. One of the measures to be taken to avoid overfitting is the inclusion of early stopping- technique. The early stopping is provided through the 'EarlyStopping' call back that looks at the validation loss over the course of epochs and breaks the training if for a specified number of epochs, it has failed to improve. Through the implementation of these methods, the LSTM system gets to generalize well to unexplored data, which brings about more precise predictions of future market returns in addition to minimizing the risk of overfitting to the training data.

The last step in LSTM model training is considered the most critical since that is where the model makes the most profound conclusions on forecasting the market returns of the future based on the data patterns it has learned in the past. During this phase the model is fed with the most recent historical data which is denoted with the 'last\_n\_steps' variable and then, based on this information, it generates the forecasting predictions for the next time step. By virtue of the sequential capabilities of LSTM networks, the model can identify the repetitiveness of the data while at the same time extrapolating trends and patterns embedded in the data which it projects into the future. The code is designed carefully to give finesse to the prediction creating process, by the usage of the 'model.predict' function which invokes the predictive powers of the trained model. Historical data that is most recent is obtained, and then fed into the model in a form that is structured so much so that it can easily fit into the designated input format. The model employs a set of sophisticated calculations and procedures that run within the LSTM layers to achieve the goal of decoding the intricate paths which make up the market dynamics and then the forecasting of the market returns. Once the model is done with the predictions, the resulting scaled returns are converted back to their original scale using the 'scaler.inverse\_transform' function. This significant stage ensures that the forecasted returns are in alignment with the original units of the market data that further simplifies and enhance the interpretation and evaluation. Reversion of the scaled forecasts back to their original values provides an opportunity to model builders to inspect the absolute dosage of market performance, which based on their predictions is expected by the model. However, visualization also becomes a pivotal point whereby data information fills in the gap between raw data and actionable knowledge. In this particular write-up, graphing has a sort-of the ultimate role in making the future market returns for every single index clearer and more transparent. Among the variety of excellent data visualization libraries, Matplotlib- a powerful data visualization library would be used to render the forecasted returns into a visual narrative. Code effectively utilizes Matplotlib to generate charts depicting forecasted market returns, using line plot as a means of symbolizing the movement of index returns over a given time frame. Each line plot sketches the projected returns which are on the y-axis and the time represented by dates is on the x-axis. The use of the careful selection of details and the visual appeal of the code are able to articulate a very concise impression of the returns hence the easy interpretation by the users.

* 1. **Training of the Model**

A math equation with black text

Description automatically generated with medium confidence

Figure 1. Training of the model with validation split.

The 5th part of the code covers the training processes of the LSTM model. In that regard, the next step is significant as it is a process which involves multi-iteration of the model parameters to minimize the loss function and thereby increase its predictive performance. X stands for the input data that will be utilized to train the LSTM model. In time series forecasting context, is represented by temporal sequences of historical data points. In each such sequence, also called a “window,” there is a fixed number of data points designated as a model’s input parameter for further predictions. For example, if the time series data is the daily stock prices, then the window will have the prices of the previous 31 days and the target label will be the price of the next day. These input-output pairs are useful for training on patterns from the historical data which might then be used for prediction of future values. y stands for the target labels attached to the input data X. In particular, in time series forecasting, the target label for each input window is usually the value of the next data point in the time series. Therefore, if the input specifies the prices of the previous 31 days, the target label would be the price of the subsequent day. This model, therefore, aims to predict the specified target label depending on the windows of input presented. That's why an epoch means one complete pass of the entire data through the neural network during the training process. In each epoch, the model goes through the complete set of the input data (X) and updates the parameters, which have an influence on the loss function. The loss function is computed on the difference between the predicted output and an actual target label (y). Completing for the multiple epochs enables the model to tune its parameters in such a way as to improve the model’s performance over time. Increasing the epochs allows the model to converge better and have a higher accuracy level, but there are chances that the model can overfit itself to the training data if too many epochs are used. Training is coded to run 100 epochs denoting the model should go through the whole training data set 100 times during the training process. Then, the batch size determines how many components of the data set are used for processing in each training era that comes before an update of the model’s parameters in every training cycle. In place of the parameters being updated after every sample processing (batch size of 1, known as stochastic gradient descent), by batching together multiple samples, however, the training process is stabilized, and the computation speed is improved. The number of samples per batch is 236, meaning that the model takes 236 samples at a time before updating the parameters. At batch size like this, there's a balance between computational efficiency and accuracy of gradients.

The next step that follows is the 'validation\_split' parameter, which enables the formation of the validation dataset from the training data. Validation is a tool which allows validating the process of training due to the fact that a fixed amount of the training data ('validation\_split') is withdrawn for validation at the given moment. The size of the validation split is set to 0.2, which means that 20% of the training data will be used for validation in this case and the remaining 80% will be used to carry out the training process. After the validation split parameter comes the ‘callbacks’ function. These functions can be applied constrainedly throughout the whole training process at different stages, such as at the end of each epoch or upon reaching a specific condition. They enable additional functionality, like the ability to set the early stopping criterion, model checkpointing and logging during training process. Hence, the goal of this line of code is to avoid overfitting through monitoring with the help of validation loss callback which is known as 'EarlyStopping'. At this point, the training will stop if the validation loss (which is determined by using the 'patience parameter') does not improve after a specified number of epochs. The model with the best performance on the validation set will be restored then. Throughout training, the LSTM network is trying to reduce the loss function value (mean square error), by means of adjusting its internal parameters (weights and biases) using the backpropagation concept as a mechanism. The optimizer (the Adam optimizer) determines the specific update rule which defines the directions and magnitudes of the parameters update step by calculating the gradients (the derivatives) of the loss function with respect to the parameters.

* 1. **Optimization Technique and Mean Square Error**

The line model.compile (optimizer='adam', loss='mean\_squared\_error') in the written code is one of the most important steps in the process of configuring a neural network model prior to training. The 'compile ()' function serves as a fundamental step to complete the preparation of the neural network model for the ahead training phase. Here is the start to architect the parameter optimization of the algorithm and the loss function that it will be using for training. Such decisions are of great significance as they greatly impact how the model learns from the training data and its improvement in predictive capabilities over time. The optimizer used in this code is ‘adam.’ Adam stands for Adaptive Moment Estimation which is an advanced technique of optimizing that is widely used during neural network training. Known for its efficiency in handling large datasets and complex models, it is widely used by machine learning specialists. The Adam optimizer accomplishes this by keeping separate learning rates for every parameter in the model and keeping track of the moving averages of these rates and both the first-order moment (mean) and the second-order moment (uncentered variance) of the gradients. Hence this flexible mechanism of access to accurate data facilitates faster and more stable outcomes compared to the conventional approaches such as the stochastic gradient descent (SGD). Together with the 'adam' optimizer, there is the loss function: MSE or 'mean\_squared\_error' is another widely used metric in classical machine learning algorithms. The MSE (mean squared error) loss function is the key application for regression tasks, where the purpose is to estimate continuous values. It measures the mean value of the squared residuals, that is, the differences between predicted values from the model and the actual targets in the training data. Squaring the differences between each predicted and actual value and further dividing the sum by the overall number of samples gives mathematical MSE. Minimizing the MSE while training model signifies minimizing the difference between the models’ predictions based on given training data and the ground truth, which consequently obtain the more accuracies in predicting. By adopting the neural network model and using Adam optimizer and MSE loss function to compile the model, the ‘compile ()’ function creates a perfect condition for the future training process. In the process of training, the model sequentially fine-tunes its parameters (weights and biases) by employing calculated gradients of a loss function relative to these parameters. This iterative parameter optimization process aims at decreasing the MSE, hence, appending the models' repertoire of making accurate predictions even on previously unseen data. The selection of ’mean\_squared\_error’ as a loss function directly specifies the loss metric with respect to the condition for early stopping. Here in this piece of code, the training loss ('val\_loss') is being inspected through the training procedure which is directly derived from the MSE loss function that was specified at the beginning of the model compilation. Employing MSE as a loss function and inspection of the model performance on the validation set demonstrates that the early stopping prevents the model from continuing the training that would result in degradation of its generalization abilities and thus avoid the overfitting situation. This, in turn, is the determinant of the subsequent steps such as early stop which assists in regulating the training process and avoids the model from overfitting and improving its generalization performance.

* 1. **Research Methodology Conceptual Framework**

|  |  |
| --- | --- |
| A diagram of a flowchart  Description automatically generated | Subchapter 3.1  Subchapter 3.2  Subchapter 3.3   Subchapter 3. 4  Getting results |

Figure 2. Conceptual Framework

**4. RESULTS**

* 1. **Data Description**

The dataset used in this study comprises daily closing prices of six prominent market indices: E.g., NDX (Nasdaq 100), SPX (S&P 500), DJI (Dow Jones Industrial Average), UKX (FTSE 100), SXXP (STOXX Europe 600), and NKY (Nikkei 225). The data is spread over a considerable time frame, starting from January 13, 2000, and going up to January 12, 2021. These indices are well-known indexes that symbolize the performance of the American, British, European, and Japanese financial markets, respectively. The data provided is extracted from Bloomberg, a reliable financial data vendor marked out for extensive coverage of worldwide financial industries. Bloomberg's data is scrupulously collected, tampered with, and facilitated, thus being a dependable source for financial analysis and researching. Every row of the database has got a date column and and further to it, six more columns listing the daily closing quotes of the indices concerned. In simple terms, the closing price of an index at the end of each trading day can be described as the prices at which these stocks were sold by the end of the day. It is a significant measure that is used by investors, analysts, and traders to determine general market performance and direction thereof. The dataset is preserved in an Excel file format that is not only simple to access but can also be compatible with a wide array of data analysis tools and platforms. The usage of an Excel file makes it possible to unify the Excel file with common data manipulation and analysis libraries on Python, Pandas, Numpy and Scikit-learn being some of them. Before any analysis or modeling the data is intended for, it goes through the preprocessing steps to make sure it is of good quality and well prepared for the particular analysis. These data preparation steps start with converting the date column to datetime format, setting it as index, and giving attention to any missing or incorrect data points.

On the other hand, the data is also scaled, using the ‘MinMaxScaler’ type of the function to normalize the values and bring them within a consistent range that is necessary for machine learning model training. The chosen method of forecasting market returns is based on the implementation of LSTM (long short-term memory) neural network, which is a type of recurrent neural network (RNN) developed for the analysis of sequential data especially those that are time series data. LSTMs models have shown satisfactory results in lots of functions about forecasting in financial area which is the reason they have come to be the most favorite predict techniques for discovering of market trends and other kinds of movements. The LSTM model architecture is strategically planned and developed to capture the intricate temporal patterns and the interdependences of market data trends. LSTM is one of the layers in the structure with a relatively high size of units. Dropout layers are also included to help to prevent the network from overfitting and densely connected layers enable the network to go through both non-linear transformations and feature extraction. Throughout the training phase, the performance is evaluated with the use of a validation split that serves the purpose of evaluating the generalization and preventing the process called the overfitting of the model to the training data. The training of the model is stopped after the validation loss fails to improve after a certain number of epochs has been reached and thus the model does not learn noise or irrelevant patterns. The parameter optimization is done by adjusting the parameters with the aim of finding the right number of LSTM units, dropout rates, learning rate for the optimizer, batch size, and number of epochs among the possible values. These hyperparameters are tweaked iteratively so as to bring about better performance and to build in robustness across different market conditions. When the model is adequately trained and validated, it shall be utilized to predict market returns for the next seven days. A set of return forecasts is created by performing iterations through predicted future values. During each iteration, the model uses patterns and dynamics learned from historical data. The forecasting results undergo forward transformation so as to acquire the values of actual returns. These are then expressed in the form of a table and presented for further assessment and understanding. These forecasted returns are also graphically presented through line graphs that link the predicted pattern and fluctuations in the market indices over the forecast period. This graphical representation helps to grasp how the price and amplitude of market fluctuations might progress; therefore, it can serve as the basis for appropriate financial decisions and risk management techniques.

* 1. **Empirical Analysis**

The first step is to import all necessary libraries that serve as tools for data manipulation, machine learning modeling, parameter optimization, and visualization. These essential libraries are Pandas, Numpy, Matplotlib, Scikit-learn and Keras with TensorFlow as its backend. Pandas is used for its powerful data manipulation functions. It performs very important tasks that include the reading of the dataset from an Excel file and the initial preprocessing steps of the data. Particularly, the function read\_excel of Pandas is applied for the purpose of loading the dataset into a DataFrame.

A screen shot of a computer

Description automatically generated

Figure 3. Imported libraries.

In addition, date-time conversions are performed by Pandas functions like 'to\_datetime, set\_index, and dropna that are further used to set the index to Date column and to erase any missing values into the data frame, ensuring data integrity and consistency. NumPy deals with numerical data manipulation through different arrays at a tremendous speed. It provides tools for operation with multi-dimensional arrays as well as matrices which are needed to include dimensionality of inputs from dataset and model. In the code, scaled data along with the inputs and outputs of the model are stored in the form of numpy arrays, which enables their efficient usage within Keras machine learning framework. Matplotlib is used for data visualization that focuses on plotting time data series as well as model performance metrics. The plotting function from the Matplotlib module allows for the visualization of forecasted market returns for the next seven days through its line plot. Further, the use of specific Matplotlib functions for personalizing plot appearances, including figure, xlabel, ylabel and legend, is incorporated to help the audience to read and understand the plot.

Another library employed is Scikit-learn. Among the features of Scikit-learn there are multi-dimension machine learning algorithms and tools for preparing the data, selecting the model, and evaluating the performance. In the code, MinMaxScaler from the Scikit-learn library is employed for feature scaling. Feature normalization is a vital first step in feature preprocessing which allows all features to have equivalent scale, so as not to let some features overshadow the others during the model training. Through scaling the features on to a specified range (usually between 0 and 1), both the convergence and the performance of the model are improved. Keras with TensorFlow back end is employed to specify and build the LSTM (Long Short-term Memory) neural network model. With Keras, a high-level neural networks API that can greatly simplify building and training of deep learning models, it is much easier to work on these problems. The code implements the Sequential API of the Keras library to build the architecture of the model sequentially, layer after layer. Layers include LSTM, Dense, Input, and Dropout coming from Keras, and these are later on integrated into the model definition.

Besides that, TensorFlow is applied as a computational backend for efficient parameter optimization and computing operations of the neural network model. The early stopping is a characteristic in Keras which stops the training process when a monitored parameter (which in this particular case is the validation loss metric) is no longer improving. Overfitting is one of the complexities that may affect machine learning methodologies when the model shows good performance on the training data but fails to generalize when applied to collected unseen data. Early stopping is a way to handle overfitting by stopping the training once it is obvious that the performance on the validation set is starting to worsen. Through the EarlyStopping function, the model selects the best weights based on the observed metric, thereafter, ensures that the final model achieves the optimum performance by preventing the occurrence of overfitting. Then, based on the training data, Adam optimizer is employed to update network weights repeatedly. In the code, the Adam optimizer is set up with a specific learning rate (0.0001) to have control over the gradient step size during weight updates. The optimizer's objective is to make the mean squared error loss as small as possible between the preciseness of the model and the actual market returns during training, to increase the model's predictive accuracy and generalization capacity. Another important part of this study is performing the Augmented Dickey-Fuller (ADF) test for each index. The ADF test is a statistical test that will ascertain if a unit root is prevailing in a time series dataset. The loop 'for column in df.columns[0:]:' is used to iterate over each column excluding the first column supposed to contain dates. For all columns (indices), the ADF test is used in order to test whether a unit root exists in a time series data. This test is tremendously useful since it allows for distinguishing between whether a time series is stationary or possesses a trend and this is central to a number of analyses and forecasting tasks in finance, economics, and many other related fields. The ADF test for each column produces some output statistics, which include ADF statistic, p-value, as well as the critical values at the different significant levels (1%, 5%, and 10%). These facts deliver deep knowledge of the stationarity character of time series data. The ADF statistic shows evidence against the null hypothesis that the unit root is present in the model. A low ADF statistic signifies the high degree of evidence against the null hypothesis that explains the hypothesis of stationarity. The p-value is the significance degree of the ADF statistics. If the observed p-value is less than a threshold (usually 0.05), we accept the alternative hypothesis that the unit root is absent in the data series and, as a result, the series is stationary. On the contrary, a greater p-value means weaker statistical evidence to reject the null hypothesis than the alternative hypothesis, which indicates non-stationarity. Through these critical values, we interpret the ADF statistic. The ADF statistic can be compared to the critical values to know whether the time series data shows stationarity or not.

Data cleaning and preparation is also another key step that must be completed before the training of the LSTM model can be done. The integer 'n\_steps' is defined as a parameter in order to indicate the number of time steps to use in each input sequence. It is this parameter that tells how much of the historical data are utilized to predict future values. The 'n\_steps' is specified as 31, meaning that for each sequence of input, data from the past 31-time steps is used. Through a sliding window technique input-output sets are produced from the scaled data.

A computer code with text

Description automatically generated with medium confidence

Figure 4. Data preparation for training

The input sequence is a vector of length n\_steps which is initialized using the scaled data from the time steps 'i' (from 'n\_steps', all the way to the number of time steps in the scaled data) and which starts at the index 'i - n\_steps' and goes till the index 'i'. Meanwhile, the array which contains the target value is the data item with index i. These input-output pair values are contained within the 'X' and 'y' lists.

The next step is taking the input-output pairs and converting those to NumPy arrays using the np.array() function. Therefore, this is a necessary step to guarantee that the data is presented to the LSTM model in a proper format. The form of input data ('X’) set is changed to fit the input requirements of the model. Specifically, it is reshaped to have three dimensions: sample size, number of time steps ('n\_steps'), and number of features included in the DataFrame. This remodeling is done through the 'np.reshape()' function. The shape of the input data shall be specified by the Input method, thus defining the input layer. This is when the input shape (X.shape[1], df.shape[1]) is utilized, where X.shape[1] is the number of time steps in each input sequence, and df.shape[1] is the number of features in the dataset. Once the data is prepared, it is segmented into training and validation sets using the 'train\_test\_split()' function from Scikit-learn. The sets made up of these are used in order to train the LSTM model and to assess the efficiency of the model, respectively.

Two LSTM layers are implemented based on the LSTM function. Every LSTM layer has 128 units that define the number of dimensions in the output space. The first LSTM layer is set to return sequences (return\_sequences=True), therefore, it will produce the whole sequence of the outputs for each input. The second LSTM layer cannot return the sequence (return\_sequences=False), hence, it provides only the last output of the sequence. Dropout layers are incorporated between each LSTM layer using the Dropout function.

A screenshot of a computer code

Description automatically generated

Figure 5. Building the LSTM model

Dropout is a method similar to regularization, employed to avoid the problem of overfitting by eliminating a specific part of the layer randomly during training and is related to the units (neurons) dropping. With this implementation, the number of stops is 0.2, that way, the training process is going to continue randomly with a 20% drop out of the units in each iteration. Next, the model uses the Dense function to add two fully connected (Dense) layers to it. The output of the first dense layer has 64 units and is a ReLU (Rectified Linear Activation) activation function, which presents non-linearity to the structure. The last dense layer has the number of units equivalent to the number of data features in the dataset (df.shape[1]), because it has to predict all these features in order to obtain a model. The model architecture is compiled using the compile method, right after defining it. After that, Adam optimizer is applied to training together with the mean squared error (MSE) loss function, which needs to be minimized during training. The 'fit' function is then called on the 'model' object to proceed with the training process. The '(X\_train, y\_train)' training data are the first two arguments passed to 'fit' function. 'X\_train' stands for the training data, input features, whereas 'y\_train' denotes the values or target labels. After defining the first two arguments, the number of epochs is set. With the 'epochs' value set to 100, the model enters the learning phase. This means that the model is trained for 100 epochs, because it goes over the training dataset 100 times during the training process. After the 'epochs' value, the batch size is adjusted. The 'batch\_size' in this model is set to 236, which implies that the model will work with 236 samples at a time and then update its parameters. Tuning batch size may lead to the change in training time and memory consumption. The 'u' argument is set to learn how well the machine is performing on the validation dataset during the training phase. 'X\_val' and 'y\_val' are tuples which establish the validation set. Validation data would be used to check that the model runs through the training process smoothly by monitoring its performance on this separate set of data during training. This allows capturing overfitting or to estimate generalization accuracy. Then the 'EarlyStopping' callback tracks the validation loss '(monitor='val\_loss')' and causes the training to terminate when the validation loss is not improving anymore '(patience=10)'. On the other hand, the 'restore\_best\_weights=True' argument guarantees that the model's weights are restored to the best they have been during training.

Once the LSTM model is trained, the process of forecasting market returns is conducted. It starts with setting up variables and getting prepared the data for input. For forecasting, the initializing of the input data is done on the last ‘n\_steps’ days by using the original dataset which is then scaled. Next, a loop iterates over 7 days to forecast returns for each day: Within the loop, the LSTM model which has already been trained tries to predict the next days' returns given the input data, as shown in Figure 5.

A screenshot of a computer program

Description automatically generated

Figure 6. Forecasting of market returns

The predicted returns that are obtained from this model are rescaled to the original scale using an inverse transformation method of Min-Max scaler. Future returns are added to the originally generated ‘forecast\_returns’ array. The data considered for the next prediction is changed by having the most recently predicted return and removing the first data point of the sequence. Based on the forecasted returns, the next 7 days are constructed of data range starting from the day after the last date of the original dataset. This makes sure that return forecasts, in the proper manner, comply with the dates. Thereupon, a DataFrame called 'forecast\_df' is built up to arrange the forecasted returns orderly. Forecasted returns from the 'forecast\_returns' list are put together along the columns' axis to produce a NumPy array, which is used to build the DataFrame where rows represent forecasted days, and the columns represent market indexes. The index of the DataFrame is aligned with the date range which has been previously created. The return forecasts are then displayed on the console to be analyzed. This shows the corresponding forecasted returns per market index in a tabular format to allow a deeper analysis. In addition, the returns that are predicted by the model are shown graphically though Matplotlib. The development of multiple subplots, one each to represent the expected returns on different market indices, is done to showcase their movements over the 7-day period. Each subplot comparatively shows the predicted returns along with the designated dates so as to identify the trends and patterns better. Lastly, debugging comments that clarify the columns and index range of the compared DataFrame are added. This also serves as a check of the accuracy of the generated forecast DataFrame and ensures data correctness. In the end, the displayed plot of forecasted returns is enabled by 'plt.show()', where plot titles, labels, and legends are also customized accordingly.

* 1. **Results Interpretation**





Figure 7. Real vs. Forecasted data for each index

In machine learning, training loss and validation loss are two of the most important indicators used to measure the model’s learning process and generalization ability. A model’s prediction errors are called training loss, and these errors are counted during a training process, while validation loss estimates the model performance on unseen data. The training loss indicates the mismatch between the model predictions and the actual labels in the training data. Whereas in training, the parameters of the model were iteratively tweaked to minimize this loss with the aid of Adam optimization algorithm. Generally, the model learns to recognize data patterns discovering which one is the most correct, hence in the early periods of the training process the loss function will go down in a rapid manner. This loss reduction declares the best parameters optimization and manages to get the model to reduce the gap between its predictions and the real output. Likewise, the curve of validation loss across epochs shows the model's capacity to continue beyond the observed data. Validation loss ideally should decrease or at least remain the same as training continues, which indicates that the model is learning interesting features from the data. The model quickly identifies the characters in data which contribute to the overall loss and as a result there is a significant decrease in loss values. The trend is slightly different from the first epochs as the rate of decrease in loss of both training and validation gradually slows down. The parameters constantly fine-tune during the training; hence loss values keep getting smaller with time. The drop is less significant, but it is still proof of the fact that the model is able to use the information coming from the data properly. However, training and validation both showed a smooth decreasing trend throughout the training period.

A white page with black numbers and letters

Description automatically generated

Figure 8. Augmented Dickey-Fuller test results

ADF statistic assesses the stationarity of the time series as well as the presence of a unit root in the data (which shows the reflections of random walk). All indices of (NDX, SPX, DJI, UKX, SXXP, NKY) reveal the measure of significance higher than usual significance levels (1%, 5%, 10%). This implies that we do not reject the null hypothesis of unit root for these data. The discovery of a unit root indicates that the employed time series exhibit a non-stationary and a random walk behavior. A time series whose values change by random walk is inherently unpredictable because its future value will be unrelated to past values. This implication leads to difficulty in application of classical time series forecasting approaches. Nevertheless, LSTM networks are designed to identify even complex patterns and dependencies which might be missed by traditional statistical methods leading to more precise prediction forecasts. LSTM models are effective because they were developed to handle sequential data and can adapt to long-term dependencies within time series data. Although a random walk is revealed in the data, nevertheless, LSTMs are still capable of predicting well by identifying otherwise unseen subtle patterns and temporal structures in a stationary analysis.

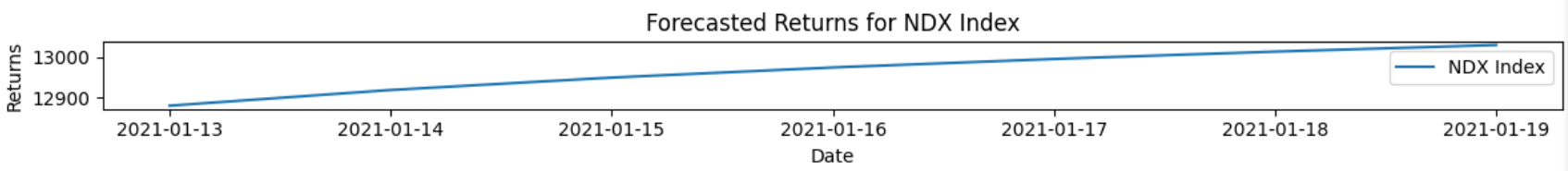


Figure 9. Forecasted returns for NDX Index

The expected returns of the NDX stock market index appear to rise gradually along the forecasting horizon, suggesting a positive movement and growth opportunities within the technology sector. The flat projected number rise from day to day and therefore, the forecasted values progressively increase from 12880.723633 on January 13 to 13030.318359 on January 19. This trending upward motion implies that the Nasdaq-100 Index is foreseen to have moderate to strong success in the coming short run, with each following day’s expected value exceeding the previous day. A rising trend in forecasted returns for the NDX index suggests to analysts, traders, and investors that the technology field and the entire market are perceived with optimism and confidence. The investors can rely on the information they obtain from the enhanced returns forecast for NDX to correct their strategies. From the projections presented above can be seen that technology-related equities or NDX index-based investment instruments may suggest an opportunity for the investors to benefit from market rallies and then enhance their returns by increasing their exposure to technology-related stocks or index-based investment assets. On the other hand, investors should do proper research and evaluate the underlying fundamentals and potential risks of those particular companies which compose the index.

A graph of blue and orange bars

Description automatically generated

Figure 10. Real vs. Forecasted data for NDX Index

The LSTM model’s predicted values for the NDX Index are closely aligned with the actual values. Over the predicted period, the projection values become almost similar to actual values, though with small overestimations and underestimations. For instance, the 14th of January witnessed a prediction of 12,919 in the forecast while the actual value turned out to be 12,898, representing a very small overestimation. The disparities are nevertheless small indicating that the model depicts the overall tendency of the NDX Index fairly good. This shows the LSTM model has been effective and can predict the NDX Index fittingly, in a way that demonstrates consistency and precision.

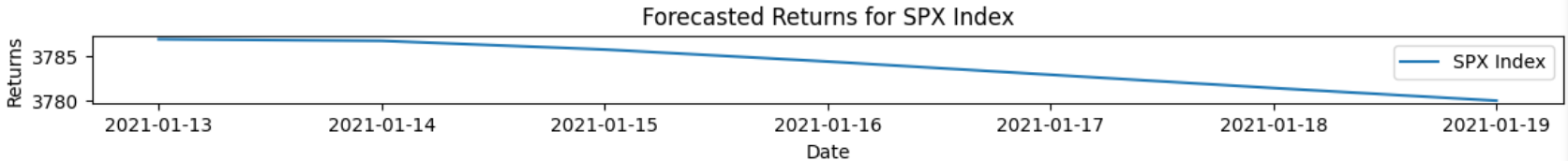


Figure 11. Forecasted returns for SPX Index

The 7-day SPX index predicted returns appear to increase as the trend line moves upward. During the selected period, forecasted values start at 3751.46 on January 13th and reach the peak of 3761.38 on January 18th, before a small decline on January 19th to 3761.06. The starting days are depicted as increases steadily, but the latter part of the forecast snapshots the possibility of a plateau. The small decline on January 19th (from 3761,38 to 3761,06) could be an indicator of a minor stock market correction or of the beginning of a stabilizing phase after the time of a consistent increase. A correction in the market usually happens after a long period of a constant increase because investors need to realize their profit. Nevertheless, the outcome signals a favorable projection for the SPX index over the 7-day forecasted period showing a positive trend which is gradually stabilizing. This shows that we have a positive vibe towards the SPX index with a small amount of volatility and gains projected to happen gradually. The predicted values are displayed with very minimal volatility. The disparities between the forecasted daily values are small and lean towards a period of stable market for the SPX index. The predicted values suggest a positive trend, but investors should also pay attention to the external factors that may significantly impact the market. In real-time situations, factors like monetary policy alterations, inflation, unemployment data can have a huge effect on the performance of the financial markets.

A graph of blue and orange bars

Description automatically generated

Figure 12. Real vs. Forecasted data for SPX Index

In the case of the SPX Index, the LSTM model also yields good and precise forecasts. On January 13th, the forecast was 3,751, slightly lower but very close to the actual value of 3,809. Even though this tendency of underestimation goes on, it stays within a narrow margin. With predicted values like 3,756 for January 14th and 3,759 for January 15th being below but very close to the actual values respectively of 3,795 and 3,768, the model is performing quite well. The model's predictions state they typically have a slight tendency to underestimate the SPX Index, but the differences are not considerable; therefore, it can accurately reflect the general trend of the SPX Index.

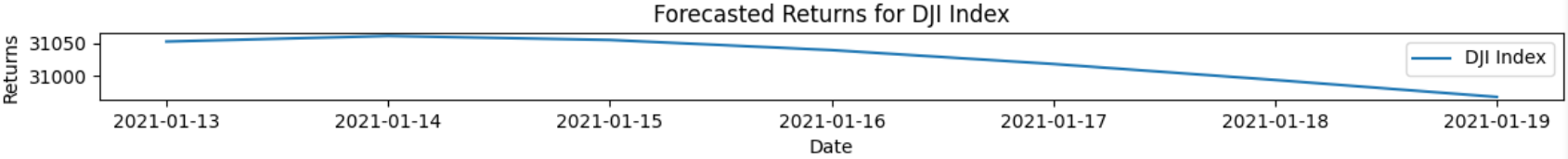


Figure 13. Forecasted returns for DJI Index

With reference to this data, we may observe that the predicted outcomes for the DJI index are coming down continuously for the next 7 days. The descending observed values then approach the trend line until they reach 31052.105469 from January 13 to 30967.957031 on January 19. This trend of decrease indicates that the DJI index is thought to witness a moderately weak decline in the coming days and the level of each day's forecasted value will be lower than that of the previous day. It is quite clear from the sample forecasted returns DJI index that for the next period the market may seem to be slowing its pace or having correctional moment. The decrease in anticipated prices may not represent a major crash but it indicates a transition from the earlier optimistic market. Market forecasts are influenced by a lot of factors including the economic indicators, corporate earnings and geopolitical events among others and thus may explain the observed trend. Investors will need to think about how this trend will influence the returns forecast for the DJI index, when planning the investments. Weakened forecast returns can be the reason for investors to be more attentive and use portfolio management techniques to guard themselves against risks of losses. Furthermore, investors can look at completely different investment opportunities or asset classes that provide superior risk-adjusted returns in the light of the current market setting.

A graph of numbers and numbers

Description automatically generated with medium confidence

Figure 14. Real vs. Forecasted data for DJI Index

The results of the DJI Index forecasts by the LSTM model are almost similar to the current values recorded in the real world. As for January 13th, the forecast of 31,052 almost matched up the value of 31,060 recorded on the actual day. Considering testing of the model, in the forecasting picture the predicted values approximately imitate the actual values, with minor deviations. For example, on 15th January the forecast showed 31,054 and the real figure was 30,814, indicating a slight overestimation. The presence of these patterns shows that small deviations were not enough to distort the model, hence the model proved to be highly accurate and reliable in predicting the DJI Index.

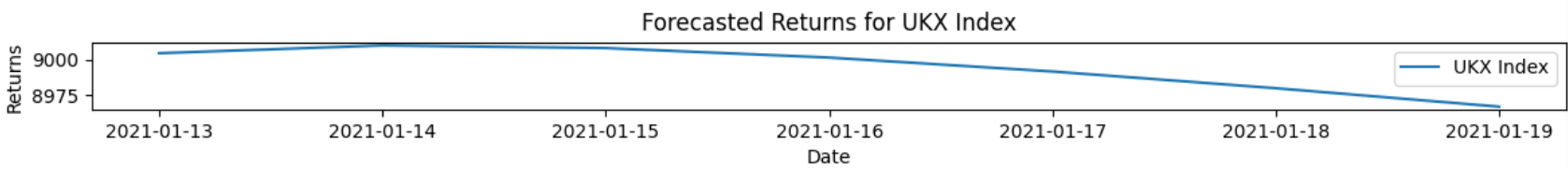


Figure 15. Forecasted returns for UKX Index

The trend of the forecasted returns for the UKX index is a line that descends over the following 7 days based on the forecasted data. On the morning of January 13, the projected value stands at 9004.340820. The value steadily declined to 8967.274414 by the end of January 19. This downward trend can be taken as a sign of a decrease in the overall value of the FTSE 100 during the forecast period. The downward trajectory in forecasted returns for the UKX index could possibly reflect the overall risks and challenges in the UK economy and financial markets at the moment. Factors as Brexit-related worries, an economic slowdown, geopolitical tensions, and uncertainty about the long-term effects of the COVID-19 pandemic may have played a role in the forecasted returns pattern observed. This trend may be seen as a signal of uncertainty by investors who therefore might restrain themselves from investing in British equities. The investors who have been planning to allocate their capital into the UKX index should be very careful about the effect that expected low returns (forecast) will have on their investment portfolios. The downward trajectory of the indices signifies possible obstacles and impediments UK companies could come across, thus negatively influencing investment performance. Fund managers would likely undertake reconsideration of holdings, that should include both sectors and geographical portfolio allocation to help to diversify the risks.

A graph of blue and orange bars

Description automatically generated

Figure 16. Real vs. Forecasted data for UKX Index

The UKX Index has revealed that the forecasts are sometimes a bit underestimated. On the 13th of January, the forecasted value was 9,004 against its actual value which was 9,221. Such a similarity arises again in the following days with the projected value on January 14 being 9,009 while the actual value was 9,283. The models output remains constantly less than the actual values during the forecast period, but still considerably close. This signifies that the model captures the UKX index trends, but it relatively underestimates its values throughout the process showing a scope for improvement concerning the modeling ability of this specific index.

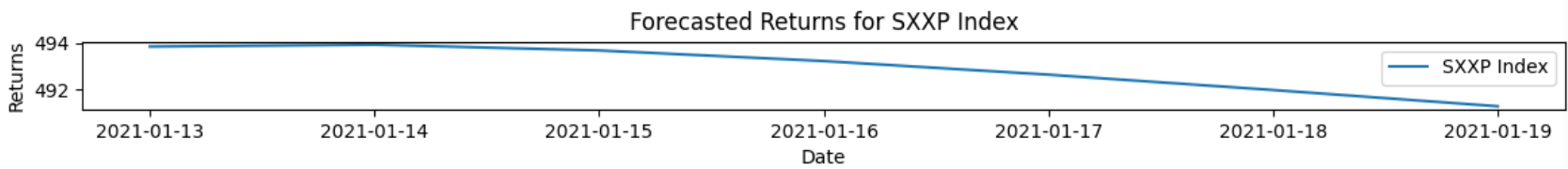


Figure 17. Forecasted returns for SXXP Index

According to the generated data, the expected returns for the SXXP index show a consistently descending trend of predicted market returns for the entire forecasted period. Specifically, the values range from 493.848450 on January 13 to 491.280365 on January 19 implying a decreasing overall index value in the projection period. The forecasted returns could be reflecting the prevailing economic concerns on the Eurozone. Issues of slow economic growth, low consumer expenditure, as well as bad business sentiment may rob European stocks of their positive performance. Political conflicts and trade-related issues may cause investors to be cautious towards investment assets, thus resulting in stock price weakness. Factors such as a slowdown in international exchange of goods, volatility in commodity prices, and unclear picture about the spread of COVID-19 pandemic could give rise to a cautious attitude towards European stocks. The monetary policy decision made by the European Central Bank (ECB) puts investors’ sentiment at stake. Shifts in monetary policy such as tighten, or inflation concerns can cause investor fear causing European market stocks performance to go down. The vulnerabilities of important sectors, which are banking, automotive, and manufacturing industries within the Euro Stoxx 600, might be one of the reasons why forecasted returns are expected to be decreased. Furthermore, poor earnings reports from big European companies give rise to negative sentiment in the markets as well. The unexpected decrease in estimated returns for the SXXP index could be explained by the prevalence of a generalized risk adverse investor sentiment. They may prefer to invest in safer assets. The economic outlook, which is not clear at all, might also create worries about the unsustainability of the equity market rally. This ability will prompt investors to become more careful in Europe equities.

A graph with numbers and lines

Description automatically generated

Figure 18. Real vs. Forecasted data for SXXP Index

The SXXP Index forecasts made by the LSTM model are quite decent. On the 13th of January, the forecast was 493 and the actual measurement was 499 with a slight underestimation. The associated values for the next few days remain very close to the real ones, such as 493 on January 14th versus the actual 501. The deviations were slight, showing that the model was successful in closely tracking the movements of the SXXP Index. This stable performance signifies that the model almost exactly predicts the SXXP Index with a great precision rate.

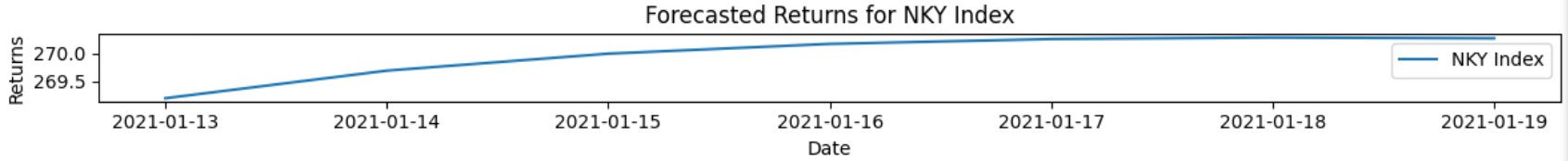


Figure 19. Forecasted returns for NKY Index

The NKY index exhibits a beta trend in forecasted returns with a relatively stable and increasing trend over the 7-day period. The values of the movement observed minor fluctuations, with a gradual rise occurring from 269.188751 on January 13 to 270.273499 on January 19. The unwavering and moderate development of the NKY index is evidenced in the Japanese stock market with a degree of confidence and durability. It can also be interpreted by investors as a sign of resilience and optimism, based on factors like positive economic signs, releasing of earnings statements by corporations, or government measures on stimulating economic process. Also, the gradual increase of the NKY index implies that the market volatility during the forecasted period is likely to be controlled. Such a diversification spread may be helpful for those investors who desire constant returns and less risk in their portfolio. It should be emphasized that even though the anticipated assets growth for the NKY index seems positive and stable, at the same time, external factors, such as geopolitics, global economics, and the mood of investors, could impact the market situation and change trends for the index in the future. Hence, investors should keep a close watch on the latest developments and modify their investment strategies based on them.

A graph of a graph with numbers and a bar

Description automatically generated with medium confidence

Figure 20. Real vs. Forecasted data for NKY Index

The expected NKY index values show that there is a consistent rate of an upward trend, barring minor incremental increases each day. This implies that the LSTM model forecasts the rise in the NKY Index in a balanced manner corresponding to the specific period. A similar upward trend in the real NKY Index values is observed being frequently higher than those that are forecasted. On January 13th, the forecasted value is 269.188751 while the real value was 274.2750968. Even though a bit lower than the actual value, the forecast was very close to indicating good predictive performance. The LSTM model shows the increasing tendency of the index, but the magnitude of change is somewhat underestimated; but exact predictions would be rather a coincidence if predictions are to the Nth decimal. Hence the term prediction. On the following forecasted days, the predicted returns are also marginally less than the actual returned value. Disregarding this, the predicted returns continue to be close to the real data, meaning that the model has a high precision to follow the real market trends. The model proved good at accurately depicting the data by providing forecasts that were pretty close to the actual values, which is an excellent sign of the model’s high level of precision and reliability in predicting the NKY index. The minor variations detected suggest that improvements could be made to fine-tune the model but still confirm that the model managed to accurately replicate the index's trend.

* 1. **Comparative Analysis Across Markets**

Through the analysis of the forecasted returns for different markets which include NDX (NASDAQ), SPX (S&P 500), DJI (Dow Jones Industrial Average), UKX (FTSE 100), SXXP (STOXX Europe 600) and NKY (Nikkei 225), we can spot out common patterns or connections among them. In this study, the forecasted returns for NDX, SPX and DJI show identical processes over this particular time interval and so it is stated that cointegration exists among these indices. This relationship might reflect overall market volatilities due to external factors related not only to the national economy, but also to the global economy as a whole with the help of factors such as economic data publications, monetary policy changes and geopolitical events that have a global impact.

There can be significant differences in the predicted returns of several indices because of numerous underlying factors, such as:

* *Regional Economic Conditions:* Each index is tied to different geographic areas, and may simultaneously show various conditions, depending on the economic situations that are related to the specific territory. Economic factors such as economic growth, inflation rates, unemployment levels and central bank policies of different countries can influence the investment choice of investors from different parts of the world with different prospects. For instance, the forecasted returns for UKX are likely to be influenced by Brexit-related developments and UK economic signals, while the returns for NDX demonstrate a high level of sensitivity to the US tech industry. Or for example, the level of consequences on the Asian stock market indexes , such as the Nikkei (Japanese index) may differ from those of EU or US stocks while China experiences a slowdown in their economy.
* *Sector-Specific Factors:* Driven by discrepant sector compositions, the indices give different prognoses in what follows from the situation of the particular industries. For example, NDX has its primary exposure to tech stocks, thus, it is more likely that the index operates under the influence of the current trends in the tech sector rather than other indices can do under similar circumstances. Also, the changes in energy prices can have tangible implications for those sectors of indices that are primarily composed of energy stocks. One of these could be a sector of the S&P 500.
* *Geopolitical Events:* Political situations are the main factors that consider the trade tensions, political instability, or conflicts. These happenings affect the markets differently based on their geographical exposure and economic ties to the regions. Furthermore, the geopolitical tension in the Middle East may affect oil prices directly and indirectly on indices with considerable exposure to energy market.

This study is actually helpful in determining the accuracy of LSTM models in prediction of market returns across different indices. If the model correctly predicts the conditions for different indices, it proves that it can be considered a robust and reliable tool to capture underlying market dynamics. By analyzing model performance against historical data and comparing it with actual market movements, we can identify its forecasting accuracy and areas where it needs improving or adjusting. The comparative analysis, therefore, displays the interdependence of the global financial markets as well as the unique traits possessed by various markets. Sometimes individual indices show significant links with each other because it could be related to the common economic situation, but others exhibit unique patterns that could be explained by the particular region or by sectoral development. Knowing these relationships and the differences among them is of great importance for investors, policy makers, and others who want to be successful in this undertaking and make a good decision. The direction of movement in the forecasted returns of the NDX and SPX are shown to be consistent, as it is shown that the two indices are exhibiting a strong positive correlation. A high positive correlation is projected here because there is a considerable degree of co-existence of the constituent components across the two indices. The DJI, as a part of the S&P 500, demonstrates a positive correlation with the SPX; however, it may present a somewhat lower return as it is limited to the 30 large-cap stocks of the industrial companies only. The British stocks (UKX) and European stocks (SXXP) have a moderate positive correlation with one another and are rather weakly correlated to the stock markets in the USA (NDX, SPX, DJI), which indicates a certain regional influence of European economic factors and market dynamics. Due to different macroeconomic factors and market conditions in Japan, NKY (Japanese index) has lower correlation with the other indices.

The NDX and SPX often show a greater level of price fluctuations because they primarily comprise the tech-oriented and companies with considerable growth potential, which might be more prone to price fluctuation. UKX and SXXP may have much lower fluctuations of prices compared to U.S. indices due to the difference that exists between these markets regarding the market structure, regulatory standards, and sentiment of investors. The volatility of NKY may be affected by domestic policies and currency fluctuations, and furthermore, it may also be influenced by global trade dynamics. The NDX gets its signals from the technology industry, which dictates how it performs; hence companies such as Apple, Microsoft, and Amazon are the strong drivers of its performance. Events within the geopolitical realm, including but not limited to technological supply chain disruptions or regulatory regime adjustments, may substantially affect the NDX's expected returns. The SPX is by-default broader, as it spans sectors, with tech, healthcare and finance being the biggest representatives. Investment returns can easily be affected by new policy changes or interest rates, healthcare policy reform, or changes in banking regulations. The DJI is industrial and consumer goods sectors-sensitive, therefore it is open to changes in the output of manufactured goods, consumer spending and trade policies. SXXP shows the eurozone areas’ performance and is driven by the European Central Bank’s monetary policy choices, EU laws’ changes, and events in the area which might influence European economies. The automotive, electronics, and manufacturing sectors' performance plays a significant role in that of the NKY. The factors like fluctuations of Japanese yen, export demand and domestic policies decisions of the government can influence index's forecasted returns. The positive economic data from the UK and eurozone like the PMI figures, inflation rates, and retail sales may help the annual returns for the UKX and SXXP funds. Regardless of all the indices, geopolitical difficulties, intra-national conflicts, and global macroeconomic factors may make certain indices vulnerable to different extents. For instance, trade disputes between the United States and China may have amplified impact on the NDX which is heavily weighted towards technology companies that run majorly their operations in China.

**5. DISCUSSION**

* 1. **Possible Implications**

This study supports the proposition that LSTM models can be used to predict market returns across different economic sectors by providing evidence that it works for multiple markets. As a result, this approach contradicts the classical view that the markets individually are unrelated. This way of thinking implies that the markets all over the world share common motif patterns which could be the key drivers of global market dynamics. From an investor and policy maker perspective this has repercussions in the sense when there is any movement in the global markets the interconnections are strongly felt. Knowing these linkages can help the investors to diversify their portfolio more wisely and the policymakers to design more coherent reactions to the economic problems. The application of LSTM (Long Short-Term Memory) model across diverse industries reveal that this model shows robustness and generalizability. Therefore, information that has been acquired from one market can be used as a benchmark for another market, which can help in making better forecasts and minimizing risks. Researchers and practitioners can have more confidence in LSTM models predictive ability with assurance that the models are not too sensitive to market conditions or data characteristics. Such convenience paves the way for the integration of LSTM solutions in both academic research and daily practice. Parameter optimization techniques have been incorporated in the thesis project which indicates the aim to advance the LSTM performance to be higher than its basic capabilities. These techniques may, for example, include hyperparameter tuning, feature selection, or ensemble methods. Thanks to LSTM model fine-tuning, scientists and professionals may achieve better forecast accuracy, reduce overfitting, and increase their model’s recession to noise and data imprecisions. This, in turn, gives rise to more credible forecasts and smarter investments. Precise performance projections of market returns are the key for successful risk management and optimal asset allocation. In case LSTM models are capable of making market returns forecasts, investors and fund managers may benefit from using the predictions to maximize the chances of financial success. Utilizing LSTM-based forecasts as a part of the risk management system will help traders guard against market risks, rebalance their portfolios dynamically, and improve their return on investment.

The research result could be a dismissal of the assumptions underpinning the research of the EMH (Efficient Market Hypothesis), which states that asset prices are already reflecting all the available information. If LSTM models are able to systematically produce forecasts that are sufficiently accurate, this may mean that there are predictable patterns or inefficiencies in the market that one can use to gain financial advantages. Most often, this implication will lead to other research, for example in respect to market anomalies and investors' behavioural biases as well as to the limits of market efficiency. It also emphasizes the point that scientists must be open to formulating a newer approach to market behaviors once new data emerges. With LSTM models tested on different markets, we can gather deep knowledge about the macroeconomic and financial factors responsible for the market's performance. This analysis can provide insights into the global market dynamics, the interconnectedness of our economic systems and the shocks which transfer. Such insights can influence policy makers' decision-making, enable investors to navigate uncertain capital markets, and provide more effective policies and tools directed at reforming the financial system and promoting economic growth. The practical implications of this research go beyond the field of finance for asset managers, financial institutions, and researchers of market forecasting as well as trading. Market returns based on LSTM models building up might become a part of real-life trading strategies and investment decisions. This could result in more efficient performance for investors and asset managers, among others, and better risk management implementation for financial institutions. It also paves the way for the publication of new innovation paradigms such as algorithmic trading, quantitative investing, and risk modeling.

* 1. **Directions for Future Research**

Future research in the field of forecasting asset returns with applications of LSTM models and parameter optimizations promises to provide substantial contribution towards discovering the patterns in financial markets and boosting predictive accuracy of the models. There are several opportunities that can be utilized in order to strengthen the ability of LSTM technique to forecast. Developing the LSTM models further by fine-tuning and adjusting would be a ripe area of research in future for higher accuracy of forecasting. This may involve a trial of different kinds of model architectures, activation functions, and regularization methods so as to determine the best combination for effective modeling of intricate patterns in financial time series. Moreover, blending ensembles and model stacking are other avenues that may provide chances to improve the forecast accuracy through combination of strengths possessed by several models. Further productive areas for investigation are in the fields of feature engineering and data preparation since these are in demand these days. Through an addition of more economic indicators, sentiment analysis data, or alternative data sources, researchers have the ability to obtain more precious information out of financial time series data, which supposedly could improve LSTM models' predictive potential. In addition to that, advanced feature engineering techniques may also reveal some hidden patterns and relations, which can lead to the formation of more accurate forecasting. Not only this but looking into the interrelationships and the spillover effects from different financial markets could be a very good field for future research. Having a phenomenological approach of the mechanisms of shocks and disruptions in the different financial markets through models may bring new concepts on the interconnectedness of the international financial system, which can be used to upgrade forecasting models. On top of that, pursuing ways for dynamic model adaptation and parameter tuning could enhance the temporal efficiency of the LSTM, so that the models can adapt based on changes of market condition. Besides short-term forecasting, the future research needs to expand the forecasting horizon to longer term predictions and scenario analysis. Economic scenario analysis can be used by researchers to evaluate how the performance of LSTM forecasts changes under different economic scenarios, and they can provide useful insights for investors and policymakers in the decision-making process.

**6. CONCLUSION**

The process of researching included building and implementing an LSTM model to predict market prices for the indices. During the process of data preprocessing, modelling building, training, and evaluation, these steps were iterated to achieve the best forecast accuracy possible. The findings derived from the LSTM model's forecasts for different market indices are not only interesting but have also provided answers to the questions of the research in a meaningful way. These findings are based on the synthesis and testing of the model's operations, accuracy, and forecasting impact.

**In regard to the 1st research question, LSTM performance demonstrates different accuracy levels in different markets, based on their trends, which is normal.** The results of the model on several indices including NDX, SPX, DJI, UKX, SXXP, and NKY shows that indeed the model is able to capture the movements and direction of the trend for each index, however, these occurrences may slightly differ and are negligible. For example, the LSTM model precisely tracked the NDX and SPX indices with small over/under estimations indicating that it was highly reliable and would deliver the results with high precision. In contrast, the model for the UKX and SXXP indices tends to give underestimating more consistently, suggesting that the model's parameters could be further adjusted to consider this specific part. Again, as stated above these underestimations are of really small values between predictions and real data.

In response to the 2nd research questions, the p-values for all indices are significantly greater than 0,05 which is evidence that we cannot reject the null hypothesis that the series are characterized as unit roots. Hence the data series are non-stationary which shows the random walk pattern. The successful market returns forecasting with LSTM can prove results against the Random Walk Theory. In the Random Walk theory, stock prices change in a random manner and hence, they become impossible to predict based on their prior prices. However, the LSTM model’s ability to predict future market returns with a high level of accuracy implies that there are discoverable patterns in the historical data the model can learn and exploit to create forecasts.

The ADF test outputs indicate a mixed picture on the stationarity of the indices. While some indices such as the UKX and SXXP show a certain stationarity (ADF statistic closer to the critical value), other indices like the NDX and SPX do not. In spite of this, the LSTM model proved with its predictions that it could identify non-random trend and patterns that drive the process. The conformity of the LSTM model's forecasts to the actual values of all considered indices in this research work demonstrates that there is indeed some predictive power in the past movements of price. **The credible forecasts, in which the model explains the general trend and the price changes slightly around it, speak against the belief in completely random price variations.** LSTMs are capable of identifying complex, non-linear relationships within data which are otherwise not apparent through traditional methods of analysis. **Whereas the ADF test results confirm the random walk expectation (i.e., non-stationarity and unpredictability), the accuracy with which the LSTM models forecast market returns is an indication against the random walk hypothesis.** This means that the LSTM models have the capability to learn and use those underlying structures or patterns in market data to be used in prediction.

Answering the 3rd research question, the adjustment of the learning rate, early stopping with patience, and the addition of dropout layers were used to boost the model performance. These methods assist in the elimination of overfitting and good generalization, and the result is the optimization of model parameters. **Consequently, we end up with more precise forecasts.** Through the application of the parameter optimization methods mentioned above and the choice of the optimal hyperparameters such as the number of time steps, epochs, batch size, number of units, number of layers, dropout rate, learning rate, optimizer, loss function, regularization, early stopping; the researchers can determine the most powerful strategies in order to enhance forecasting accuracy of LSTM model for stock market returns. The current research adds new information to the finance field by proving with the example of LSTM models that the neural networks can be successfully applied to the task of market returns forecasting for different indices. It emphasizes the role played by parameter optimization processes for the purpose of improving forecast precision. Furthermore, the research gives rise to improved simulation methodologies in finance and supplies essential insights for both the researchers and the investors who are looking to use machine learning for prediction.

Concluding this thesis, the research involved developing and deploying an LSTM model to anticipate market indices through a series of steps including data processing, model building, training, and validation with the aim of attaining market indices accuracy parameters. It is supported by the fact that the LSTM model is able to faithfully reflect the tendencies of indices of different markets, and only minor but normal differences in accuracy are observed. The LSTM output showed reliability for all indices without a doubt. The model, however, is highly accurate despite non-stationary data series and patterns in historical data that contradict the Random Walk Theory. Parameters optimizations proved to be effective in improving the model’s response and forecast accuracy by highlighting the usefulness of neural networks in market forecast as well as parameter tuning. Generally, this paper delivers critical knowledge to the world of finance and forecasting models, and it improves our understanding of market prediction with those advanced ML approaches.

**REFERENCES**

Aminimehr, A., Raoofi, A., Aminimehr, A. et al. A Comprehensive Study of Market Prediction from Efficient Market Hypothesis up to Late Intelligent Market Prediction Approaches. Comput Econ 60, 781–815 (2022).

Bao, W., Yue, J., & Rao, Y. (2017). A deep learning framework for financial time series using stacked autoencoders and long-short term memory. PloS one, 12(7), e0180944.

Benoit B. Mandelbrot, Richard L. Hudson, The (mis) Behavior of Markets: A Fractal View of Risk, Ruin and Reward, Basic Books, New York, 2004 (ISBN 0-465-04355-0). European Journal of Political Economy. 21. 797-799.

Burak Gülmez. Stock price prediction with optimized deep LSTM network with artificial rabbit’s optimization algorithm. Expert Systems with Applications, Volume 227, (2023), 120346, ISSN 0957-4174.

Campbell, J. Y., Lo, A. W., & MacKinlay, A. C. (1997). The Econometrics of Financial Markets. Princeton University Press.

Chen, S. H., & Huang, C. J. (2011). "A hybrid ARIMA and support vector machines model in stock price forecasting." Omega, 39(3), 378-385.

Chitenderu, Tafadzwa & Maredza, Andrew & Sibanda, Kin. (2014). "The Random Walk Theory and Stock Prices: Evidence from Johannesburg Stock Exchange." International Business & Economics Research Journal (IBER).

E. Rokhsatyazdi, S. Rahnamayan, H. Amirinia and S. Ahmed, "Optimizing LSTM Based Network for Forecasting Stock Market," 2020 IEEE Congress on Evolutionary Computation (CEC), Glasgow, UK, 2020, pp. 1-7.

Fama, E. F. (1970). "Efficient Capital Markets: A Review of Theory and Empirical Work." (“Efficient Capital Markets: A Review of Theory and Empirical Work”) The Journal of Finance, 25(2), 383–417.

Fischer, T., & Krauss, C. (2018). "Deep learning with long short-term memory networks for financial market predictions." European Journal of Operational Research, 270(2), 654-669.

Gao P., Zhang R., Yang X., “The application of stock index price prediction with neural network,” Mathematical and Computational Applications, 25 (3) (2020), p. 53, 2020.

Granger, C. W. J. (1999). Empirical Modeling in Economics: Specification and Evaluation. Cambridge: Cambridge University Press.

Godfrey, Michael & Granger, Clive & Morgenstern, Oskar. (2007). The Random Walk Hypothesis of Stock Market Behavior. Kyklos. 17. 1 - 30. 10.1111/j.1467-6435.

Hansen, Lars & Sargent, Thomas. (2014). Uncertainty within economic models (to appear).

Jabed, Mahfuz Islam Khan. (2024). Stock Market Price Prediction using Machine Learning Techniques. American International Journal of Sciences and Engineering Research. 7. 1-6.

Jegadeesh, N., & Titman, S. (1993). Returns to Buying Winners and Selling Losers: Implications for Stock Market Efficiency. The Journal of Finance, 48(1), 65–91.

K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink and J. Schmidhuber, "LSTM: A Search Space Odyssey," in IEEE Transactions on Neural Networks and Learning Systems, vol. 28, no. 10, pp. 2222-2232, Oct. 2017.

Li, Q. Kamaruddin, N., Yuhaniz, S.S. et al. Forecasting stock prices changes using long-short term memory neural network with symbolic genetic programming. Sci Rep 14, 422 (2024).

Liu, Ziji. (2023). Stock Price Prediction based on Long Short-Term Memory Model. Highlights in Science, Engineering and Technology. 39. 651-656.

Lo, Andrew. (2004). "The Adaptive Markets Hypothesis: Market Efficiency from an Evolutionary Perspective."

Malkiel, Burton, G. 2003. "The Efficient Market Hypothesis and Its Critics." Journal of Economic Perspectives, 17 (1): 59-82.

Öztürk, M. M. (2023). Hyperparameter optimization of a parallelized LSTM for time series prediction. Vietnam Journal of Computer Science, 10(03), 303-328.

Pascanu, Razvan & Mikolov, Tomas & Bengio, Y.. (2012). On the difficulty of training Recurrent Neural Networks. 30th International Conference on Machine Learning, ICML 2013.

Pushpendu Ghosh, Ariel Neufeld, Jajati Keshari Sahoo. Forecasting directional movements of stock prices for intraday trading using LSTM and random forests. Finance Research Letters, Volume 46, Part A, (2022), 102280, ISSN 1544-6123.

Sornette, D. (2003). "Why Stock Markets Crash: Critical Events in Complex Financial Systems." Princeton University Press.

Shiller, Robert J.: Irrational Exuberance. Revised and Expanded Third Edition. Princeton, NJ: Princeton University Press, 2015.

Te Bao, Brice Corgnet, Nobuyuki Hanaki, Yohanes E. Riyanto, Jiahua Zhu. “Predicting the unpredictable: New experimental evidence on forecasting random walks.” Journal of Economic Dynamics and Control, Volume 146, 2023, 104571.

Tsantekidis, A., Passalis, N., Tefas, A., & Kanniainen, J. (2017). "Forecasting stock prices from the limit order book using convolutional neural networks." IEEE/ACM.

Tsay, S., Ruey. (2010). “Analysis of Financial Time Series, Third Edition.” Wiley-Interscience 2010, Probability and Statistics.

Yadav, Anita & Jha, C & Sharan, Aditi. (2020). “Optimizing LSTM for time series prediction in Indian stock market. Procedia Computer Science.”

Zaremba, W., Sutskever, I., & Vinyals, O. (2014). "Recurrent Neural Network Regularization." arXiv preprint arXiv:1409.2329.

1. Malkiel, Burton, G. 2003. "The Efficient Market Hypothesis and Its Critics." Journal of Economic Perspectives, 17 (1): 59-82. [↑](#footnote-ref-1)
2. Fischer, T., & Krauss, C. (2018). Deep learning with long short-term memory networks for financial market predictions. European Journal of Operational Research [↑](#footnote-ref-2)