University of Glasgow | School of Computing Science

Stock Market Prediction Using Machine Learning

Ronak Janawa

School of Computing Science
Sir Alwyn Williams Building
University of Glasgow
G12 8QQ

A dissertation presented in part fulfilment of the requirements of the Degree of Master of Science at The University of Glasgow

December 6th, 2024

Abstract

Stock market prediction is a challenging task due to the complex, dynamic, and interdependent nature of financial markets. Traditional models often rely solely on historical price data and technical indicators, limiting their ability to capture the broader relational and contextual factors influencing stock prices. This dissertation explores an enhanced approach to stock market prediction by integrating machine learning models—Linear Regression, Random Forest, and Long Short-Term Memory (LSTM)—with knowledge graph embeddings derived from both generic (Wikidata) and domain-specific (10-K financial reports) sources. The study evaluates the predictive performance of these models using two feature sets: Basic Technical Indicators (TIs) and Advanced TIs enriched with knowledge graph embeddings, and assesses their effectiveness through key metrics such as Root Mean Squared Error (RMSE) and Normalized Discounted Cumulative Gain (nDCG@10).The study also highlights the advantages of integrating event-based knowledge, such as corporate actions, financial disclosures, product launches, and macroeconomic indicators, into the predictive framework.

Education Use Consent

I hereby give my permission for this project to be shown to other University of Glasgow students and to be distributed in an electronic format. **Please note that you are under no obligation to sign this declaration, but doing so would help future students.**

Name: Ronak Janawa Signature: RONAK

Acknowledgements

I would like to express my gratitude to Dr. Richard Mccreadie for his support and guidance throughout the project. Also, I would like to express my gratitude to my family for their support during this time.

Contents

Chapter 1: Introduction

The stock market, a cornerstone of global financial systems, serves as a platform for buying and selling shares of publicly traded companies. Stock prices fluctuate constantly due to many factors, including company performance, macroeconomic conditions, market sentiment, and geopolitical events. These fluctuations make the stock market both an avenue for potential profit and a domain of significant risk. Accurate prediction of stock prices is a long-standing challenge in financial analysis. Stock price prediction is crucial for various stakeholders, including individual investors, financial institutions, and policymakers. The relationships between various factors influencing stock prices (like interest rates, economic growth, earnings reports, etc.) are often non-linear. This means that small changes in any of the above area can cause disproportionate responses in stock prices, which traditional linear models fails to predict accurately. Events such as geopolitical tensions, sudden economic shifts, and global pandemics (such as COVID-19) can drastically affect markets in ways that are difficult to predict with any model. These events can alter investor behavior and market dynamics overnight. The stock market is adaptive in nature, meaning that as strategies and technologies evolve, the market's behavior changes. What worked well in the past may not necessarily work in the future as market participants adjust their behaviors. These challenges make stock market prediction a complex task. However, advancements in machine learning, data analysis, and computational power continue to improve understanding and forecasting market behaviors.

This dissertation focuses on the application of knowledge graphs, which provide a structured way to visualize and analyze the relationships between different stocks and other relevant financial entities. Knowledge graphs excel in representing complex relationships within the data. In stock markets, they can illustrate connections between stocks, industries, and the broader economic factors influencing them. For example, if a particular sector or influential stock experiences a change, the graph can help predict which other stocks or sectors might be affected due to their interconnected nature. The structural data of Knowledge graphs combined with the predictive power of machine learning models can lead to more accurate forecasts then the traditional predication modals that we are using(like Long Short Term Memory (LSTM) networks, Gated Recurrent Units (GRUs), Autoregressive Integrated Moving Average (ARIMA), and Autoregressive Moving Average (ARMA)) [5] . The knowledge graph provides a rich context that helps machine learning algorithms understand not just the historical prices but also the relationships between entities that influence these prices. This contextual awareness can help models account for external factors that affect stock movements that historical price data alone might miss. This approach harnesses both the quantitative and qualitative aspects of market data, potentially leading to superior investment strategies and market understanding. For example, a knowledge graph can represent the impact of a company's acquisition, partnerships, or leadership changes on its stock performance, providing a more holistic view of market dynamics.

The effectiveness of predictions is defined by the quality and structure of the knowledge graph. A well-constructed graph that accurately represents relevant relationships and includes comprehensive and up-to-date information (such as entities [Companies, Products, CEO appointments], relationships [Mergers, acquisitions, partnerships, and divestitures] and trade agreements) can significantly improve model performance. Conversely, a poorly de-

signed graph with missing or noisy data can lead to inaccurate predictions. By combining both quantitative data (e.g., price trends and volatility) and qualitative insights (e.g., inter-company relationships or market sentiment), knowledge graphs empower machine learning algorithms to derive superior investment strategies and a deeper understanding of market behavior.

This dissertation aims to leverage stock market price data in conjunction with knowledge graphs constructed from financial reports to demonstrate their superiority over generic knowledge graphs, such as those derived from Wikidata. By focusing on custom-built domain-specific knowledge graphs that capture detailed and contextually rich financial relationships, this study seeks to improve the stock market price prediction capabilities of machine learning models. The goal is to validate that customized knowledge graphs provide more relevant and actionable insights, leading to improved predictive performance and a deeper understanding of market dynamics compared to generic alternatives.

1.1 Importance and Objectives of the Research

The complexity of financial markets makes accurate stock price prediction a challenging, yet essential task for investors and analysts. This research focuses on evaluating the performance of selected machine learning models, Linear Regression, Random Forest, and LSTM, using well-defined evaluation metrics such as RMSE and nDCG@10. The primary objective is to evaluate the predictive accuracy of these models and to explore how the inclusion of technical indicators derived from knowledge graphs impacts their performance. By incorporating additional contextual information from custom-built knowledge graphs, constructed using financial reports(from 10K reports), the study aims to enrich the feature set used for predictions and provide a more comprehensive evaluation framework.

The importance of this research lies in its systematic approach to understanding how advanced technical indicators derived from knowledge graphs enhance model performance compared to traditional metrics. By re-evaluating the models with these enriched features, the study seeks to demonstrate the added value of incorporating relational insights into financial predictions. This approach not only highlights the comparative strengths of the models under various feature configurations but also underscores the potential of knowledge graphs in revealing interdependencies between financial entities, ultimately contributing to more accurate and actionable stock market predictions.

Chapter 2: Literature Review

Stock price prediction is a fundamental task in financial analysis, driven by investors' desire to forecast future stock prices and adjust investment strategies to maximize profits while mitigating risks. The task involves modeling and anticipating the dynamic behavior of stock prices based on historical data, external factors, and market trends. The prediction of the stock price can be defined as follows: Let Pt T,t=1 represent a time series of stock prices, where Pt denotes the stock price at time t. The goal is to predict the price of the stock in a future time step,Pt+k, where k¿0 is the forecasting horizon. This can be achieved by learning a mapping function: f:X→Y, where X represents the input features derived from historical price data, technical indicators, and other contextual factors, and Y is the target variable, such as Pt+k or the percentage change in stock price.

To address this task, researchers have proposed a variety of predictive models that can be broadly categorized into three groups: traditional methods, machine learning models, and deep learning techniques. Traditional methods for predicting the stock market are models based on time series modals [30], such as the Kalman filter [11] and autoregressive model [2]. Now, the machine learning methods based on Linear regression [28] and random forests [15] are being applied in the field of stock forecasting. However, methods(like Linear Regression and Random Forest) exhibit poor learning effects on complex high-dimensional data and the curse of dimensionality. To solve this problem, deep learning methods [8] have been developed for stock forecasting. With the success of deep neural networks in time series data modeling, long-short-term memory (LSTM) [25], convolutional neural networks (CNNs) [14], and other models have become more effective solutions to predict future stock prices. For predicting the stock market, technical analysis that involves analyzing historical market data, primarily stock prices and trading volumes, to identify patterns, trends, and potential future movements in the stock market is the main approach to analyzing and predicting market trends. The forecast of stock market prices is crucial because market patterns are reliable, although price changes are somewhat difficult to accurately predict. Stock market prediction is covered under the four advancement categories, which are pattern reorganization, sentimental analysis, statistical, and machine learning. In stock market analysis, knowledge graphs explicitly represent the relationships between various entities, such as companies, sectors, and economic indicators, by creating a structured and interconnected graph of these elements. For example, a knowledge graph can capture how a company is linked to its subsidiaries, suppliers, or competitors, as well as how economic indicators such as interest rates or inflation influence specific sectors. By encoding these relationships, knowledge graphs provide machine learning models with contextual insights that go beyond numerical price trends. This enables models to incorporate relational dependencies into their predictions. For example, if a company's supplier is experiencing financial difficulties, this relationship captured in the knowledge graph might signal potential disruptions for the company itself, which could affect its stock price. Similarly, connections between economic indicators, such as rising inflation leading to increased operational costs for specific sectors, can be modeled to anticipate broader market movements. In this way, knowledge graphs help machine learning models understand not only what is happening within an individual entity but also how external relationships and factors contribute to stock market behavior, leading to more accurate and informed predictions.

Existing studies often treat these components, technical indicators, machine learning models, and knowledge graphs, as independent factors, without fully exploring their synergistic effects. Furthermore, while knowledge graphs have been utilized to represent relationships between financial entities, most research relies on generic graphs like Wikidata or DBpedia. These generic graphs lack the domain specificity required to capture the nuanced relationships essential for financial analysis, such as those found in 10-K reports. Additionally, the use of knowledge graph embeddings, such as TransE, to transform these relational insights into machine-readable formats for predictive models is an area that requires further investigation. So firstly, we need to understand more about what and how technical indicators, machine learning models, and knowledge graph work.

2.1 Technical Indicators

Technical indicators are the features that you derive from raw stock market data to identify trends, momentum, volatility, and potential price movements [4]. These indicators play a crucial role in obtaining the optimal result for the prediction of the market price. High, low, open, and closed prices are fundamental components of stock data that provide insight into daily market activity. The open and closed prices reveal the day's starting and ending market conditions, while the high and low prices indicate the day's trading range, these features represent the core dataset from which trends, volatility, and momentum indicators are derived. Moving averages are essential trend-following indicators that smooth out price data to highlight the direction of the market over a specific period [20]. A 7-day moving average represents short-term trends by calculating the average closing price over the last seven days, while a 21-day moving average captures medium-term trends. These indicators help filter out market noise, providing a clearer view of underlying trends. The interaction between short- and medium-term moving averages can signal potential buy or sell opportunities, such as when the short-term average crosses above (bullish signal) or below (bearish signal) the medium-term average [29]. Volatility measures the degree of variation in a stock's price over a specific period, offering insights into market stability and risk. It is typically calculated as the standard deviation of the closing prices. High volatility indicates a turbulent market with significant price fluctuations, while low volatility suggests a more stable environment. Volatility serves as a critical feature in identifying high-risk periods and possible breakout scenarios, improving the model's ability to adapt to changing market conditions [26]. Lagged closing prices represent the closing prices of a stock from previous days (e.g., one or two days ago) and are commonly used in autoregressive models to predict future prices. These features capture the sequential nature of price movements and are essential for identifying short-term trends or momentum, lagged close prices provide a time series perspective, allowing algorithms like LSTM to learn dependencies and patterns from historical price behavior [12]. Return is often used as the target variable or as a feature to model price trends and evaluate the predictive capabilities of algorithms.

2.2 Machine Learning Approaches in Financial Prediction

Linear regression is one of the simplest and most widely used models to predict stock prices. It assumes a linear relationship between dependent (stock price) and independent variables (e.g., trading volume, past prices). However, linear regression struggles with non-linear dependencies, which are common in the stock market where relationships between variables are often complex and influenced by various dynamic factors, such as investor sentiment and external economic events. Although effective for basic trend analysis, the model's inability to capture nonlinear dependencies often results in poorer predictive accuracy in dynamic stock settings, especially compared to more advanced methods [27]. Methods that support deep neural networks (LSTM) are more efficient than simple linear regression.

Random Forest (RF), where multiple decision trees work together to improve predictive accuracy. It is known for its ability to model complex, non-linear relationships, handling high-dimensional data, and prevents overfitting by averaging the outcomes of multiple decision trees. In stock market prediction, this model is effective in situations with many characteristics, such as historical prices and macroeconomic indicators, and performs well even when the correlations among the inputs are low. However, one limitation of RF is that it lacks inherent temporal awareness and can be computationally intensive, which poses challenges in real-time prediction scenarios often required in stock markets [10]. Random Forest is used with other methods and more features(like Knowledge graph embeddings) to improve it's effectiveness. For example, tree bagging shows much better stock price prediction accuracy than logit or stepwise logit [23].

LSTM is a recurrent neural network (RNN) designed to handle sequential data, which makes it ideal for stock price forecasting. LSTMs retain historical information over long sequences, helping the model learn temporal dependencies in stock data. This temporal memory is valuable for financial forecasting, as past trends and prices significantly influence future movements. Studies show that LSTM, particularly when enhanced with bidirectional or convolutional architectures, significantly outperforms simpler methods in predicting volatile stocks such as technology equities (e.g. Tesla and Google) [19]. However, LSTM models are computationally demanding and may overfit if not carefully tuned. Moreover, compared with various economic conditions, such as during the COVID-19 period, LSTM models sometimes show sensitivity to extreme and rare events, making them less robust without further regularization [18]. LSTM can be used with attention [24] and/or artificial rabbits optimization algorithm to optimise performance [7].

2.3 Knowledge Graphs in Stock market prediction

Knowledge graphs (KGs) represent complex relationships between entities in a domain. KG's capture these relationships by connecting entities such as companies, industries, and economic variables, creating a network that reveals patterns otherwise hiddeen in traditional data formats. This interconnected view helps analysts see how events in one area, such as changes in a sector, might influence related stocks or economic outcomes. [9]

KGs capture the contextual information that affects financial performance, providing a multidimensional view of the market. Research has shown that KGs improve interpretability in predictive models by offering context on interconnected events. For example, KGs constructed from financial data sources can illuminate relationships between market trends and company activities, allowing for more nuanced stock predictions that account for both direct and indirect influences (e.g., mergers, product launches, or economic changes). [17]

The use of knowledge graph embeddings further enhances the utility of KGs in machine learning. Embeddings transform entities and relationships in a KG into numerical vectors, which capture the latent structure of the graph; they allow the relationships encoded in the KG to inform predictive models in ways that raw data cannot. For example, embedding techniques have been successfully applied to stock price prediction by mapping companies in a vector space, where the spatial relationships between entities capture similarity.

In the broader field of data-driven industries, such as healthcare and supply chain management, KGs have shown potential to improve model performance by capturing domain-specific knowledge. Financial studies specifically highlight the ability of KGs to enhance data interpretability, with applications ranging from asset recommendation to stock market prediction. For example, in financial asset classification, KG embeddings have proven effec-

tive in identifying the assets most likely to yield favorable returns by revealing interconnected patterns within the financial ecosystem.

2.4 Knowledge Graph Embeddings

Knowledge graph embeddings are a transformative technique in machine learning, enabling the representation of entities and relationships within a knowledge graph as dense numerical vectors in a continuous vector space. These embeddings allow machine learning models to leverage the rich relational information encoded in a knowledge graph, enhancing their ability to uncover latent patterns and complex interdependencies between entities. [13] In the context of stock market prediction, knowledge graph embeddings capture relationships such as mergers, acquisitions, partnerships, and economic indicators, providing contextual insights that complement traditional features like historical prices and technical indicators. Knowledge graph embeddings serve two primary purposes:

Dimensionality Reduction: They transform high-dimensional graph structures into low-dimensional vector representations, making them computationally efficient and suitable for integration with machine learning models.

Relational Representation: By encoding entities and their relationships, embeddings preserve the structure and semantics of the graph, enabling models to understand both direct and indirect connections.

A knowledge graph embedding maps entities (e.g., companies, products) and relationships (e.g., "acquired by," "partnered with") from a graph into a continuous vector space. This process preserves the graph's inherent structure and semantic information, enabling models to use these vectors as input features. [16] Popular KGE methods include:

TransE: Models relationships as vector translations, ensuring that for a triplet (h,r,t), where h is the head entity, r is the relationship, and t is the tail entity, h+r=t. TransE is effective for modeling simple, one-to-one relationships prevalent in financial graphs. [3] Node2Vec: Uses random walks on the graph to generate embeddings that capture both local and global graph structures. It balances breadth-first (BFS) and depth-first (DFS) searches to preserve community and structural similarity. [6] Graph Neural Networks (GNNs): Extend traditional embedding methods by learning higher-order dependencies, allowing for a more nuanced understanding of the relationships between nodes. In this dissertation, knowledge graph embeddings are used to enrich stock price prediction models by integrating relational insights from custom-built financial graphs. The study employs TransE, a translation-based embedding method, to encode relationships from two types of knowledge graphs: a generic graph derived from Wikidata and a domain-specific graph constructed from 10-K financial reports.

2.5 Hypothesis

The stock market is again a complex system influenced by a multitude of factors, including historical prices, technical indicators, economic reports, and relationships between financial entities. Traditional models often rely on base technical indicators derived from simple metrics like historical prices and basic technical indicators. However, these base TIs may not capture the intricate relationships and latent patterns essential for accurate predictions.

Advanced technical indicators, integrating complex relationships and additional features, such as those derived from knowledge graphs, offer a richer representation of financial markets. Knowledge graphs constructed from domain-specific sources such as 10-K financial

reports can provide detailed insights into company relationships, mergers, acquisitions, and other events that affect stock performance.

In addition, the selection of machine learning models plays a crucial role in prediction accuracy. Models like random forest, linear regression, and long-short-term memory networks (LSTM) differ in their ability to handle feature interactions, relationships, and sequential dependencies.

Hypothesis 1: Using advanced TIs, which integrate complex relationships and additional features derived from knowledge graphs, leads to significantly better stock market predictions compared to base TIs that rely on simpler metrics.Advanced TIs encompass a broader spectrum of information, including the interconnections between companies, sectors, and economic indicators. By capturing these complex relationships, predictive models can identify patterns and trends that base TIs might overlook, resulting in improved accuracy. Effect of Machine Learning Model Selection:

Hypothesis 2: The predictive accuracy of stock market forecasting varies depending on the machine learning model used. Models such as Random Forest, Linear Regression, and LSTM will yield different levels of performance due to their capabilities in handling relationships, sequential dependencies, and feature interactions. Linear regression models may not effectively capture non-linear relationships and complex feature interactions. Random Forest can handle non-linearities and interactions, but may lack temporal sensitivity. LSTM networks are designed to capture sequential dependencies and are expected to perform better with time series data that include advanced TIs, as they can model temporal patterns more effectively.

Hypothesis 3: Integration of knowledge graphs into machine learning models for stock market price prediction enhances their performance compared to models relying solely on traditional technical indicators.

Hypothesis 4: The choice of knowledge graph significantly affects the quality of stock market predictions. Knowledge graphs generated using Gollie from 10-K financial reports are hypothesized to outperform generic knowledge graphs like Wikidata due to their domain-specific insights and customized representation of financial relationships. Gollie-generated knowledge graphs are tailored to extract financial relationships directly from authoritative sources, capturing specific events such as mergers, acquisitions, and partnerships that are crucial for stock performance. In contrast, generic knowledge graphs like Wikidata may lack the depth and specificity needed for accurate financial predictions.

Null Hypothesis (H0):

Integrating advanced KPIs derived from knowledge graphs and using domain-specific knowledge graphs does not significantly improve the accuracy of stock market prediction accuracy compared to models using base KPIs and generic knowledge graphs.

Alternative Hypothesis (H1):

Integrating advanced KPIs derived from domain-specific knowledge graphs (e.g., Gollie-generated from 10-K reports) and employing machine learning models capable of handling complex relationships and sequential data (such as LSTM) significantly improves stock market prediction accuracy. This improvement will be evidenced by lower RMSE values and higher nDCG@10 scores compared to models using base KPIs and generic knowledge graphs.

Chapter 3: Design and Implementation

This section details the design and implementation of the proposed stock market prediction framework, which integrates machine learning models with knowledge graph embeddings to enhance prediction accuracy. The implementation consists of four primary components: data collection and pre-processing, knowledge graph construction, machine learning model integration, and predictive performance evaluation.

3.1 The Framework

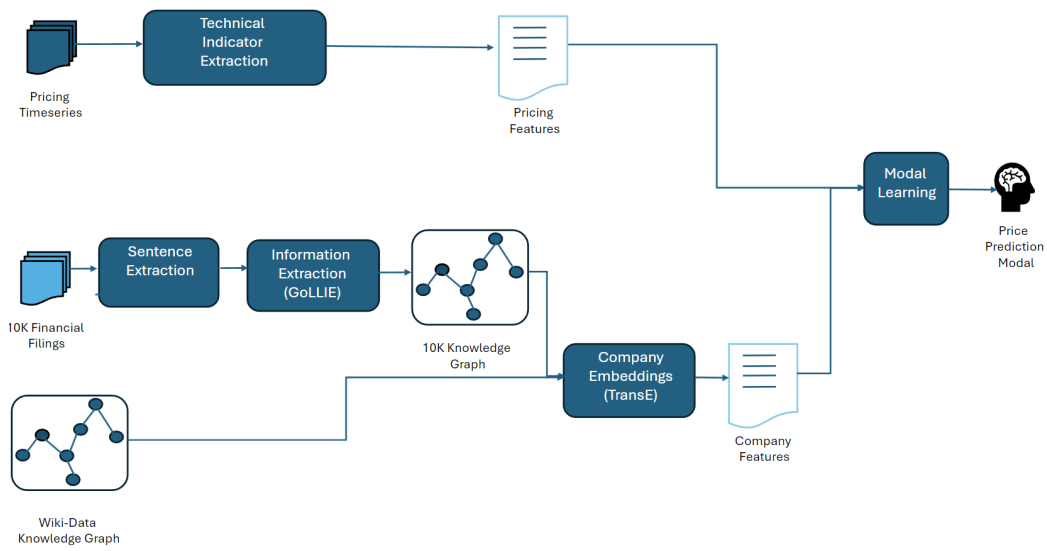

Figure 3.1: Overall Logic of the Project

The experiment combines traditional stock market features (e.g., historical prices, technical indicators) with advanced embeddings derived from domain-specific knowledge graphs. These components are used to train and evaluate machine learning models, including Random Forest, Linear Regression, and Long Short-Term Memory (LSTM) networks. The design aims to address three key objectives:

Enhance Feature Representation: Creating Knowledge graph with the help of 10K reports and extracting relation with the help of GoLLIE then by incorporating knowledge-graph embeddings, the system captures complex relationships between financial entities that are typically ignored in conventional models.

Compare Model Performance: Evaluate the ability of different machine learning models to utilize these enriched features effectively.

Validate Knowledge Graph Effectiveness: Demonstrate the superiority of domain-specific knowledge graphs (constructed from 10-K financial reports) over generic sources like Wikidata.

3.2 Machine Learning Models

Machine learning models serve as the core predictive mechanism in this framework. In designing our stock price prediction framework, we selected three machine learning models: linear regression, random forest, and long-short-term memory (LSTM) networks, each offering distinct advantages in handling financial data:

1. Linear Regression (Baseline Model):

Linear Regression models the relationship between a dependent variable (stock prices) and one or more independent variables (e.g., technical indicators) by fitting a linear equation to observed data. Serving as a baseline, Linear Regression provides a straightforward benchmark to assess the performance gains achieved by more complex models. [1] Its simplicity allows for an easy interpretation of the results, making it a common starting point in predictive modeling. While Linear Regression is effective for datasets exhibiting linear relationships, its performance diminishes with non-linear and complex patterns inherent in stock market data. Studies have shown that Linear Regression often underperforms compared to advanced models in capturing the intricacies of financial time series [21].

2. Random Forest (RF):

Random Forest is an ensemble learning method that constructs multiple decision trees during training and outputs the mode of their predictions for classification tasks or the mean prediction for regression tasks. Random Forest excels at modeling non-linear relationships and interactions between features, making it suitable for the complex nature of stock market data. [10] Its robustness to overfitting and capability to handle high-dimensional datasets are advantageous in financial modeling. Empirical research indicates that Random Forest models often outperform simpler algorithms in stock price prediction due to their ability to capture complex patterns. For instance, a study comparing various models found that Random Forest provided more accurate predictions than Linear Regression and Support Vector Machines.

3. Long Short-Term Memory (LSTM):

LSTM networks are a type of recurrent neural network (RNN) designed to model sequential data by capturing long-term dependencies through memory cells and gating mechanisms. Stock prices are inherently sequential, with current prices influenced by historical trends. LSTMs are adept at learning temporal dependencies, making them well-suited for time-series forecasting in financial markets. LSTM networks have shown superior performance in stock price prediction tasks. Studies have shown that LSTMs outperform traditional models like ARIMA and even other machine learning models by effectively capturing temporal patterns and complex dependencies in stock data. [22]

3.3 Knowledge Graph Construction

The knowledge graph (KG) is a central component of the framework, representing complex relationships between financial entities and enhancing the richness of the set of features. In this work, we construct two financial knowledge graphs from different data sources, Wikidata and 10-K financial reports—to enable comparative analysis and evaluate their impact on stock market prediction. The knowledge graphs encapsulate hyper-relational facts, where each fact comprises entities, relationships, and optional qualifiers to enrich the relational context. Below, we detail the methodology for constructing these knowledge graphs.

1. Wikidata-Based Knowledge Graph; The Wikidata Knowledge Graph was created by extracting financial relationships and metadata from the Wikidata database, a publicly available, general-purpose knowledge base. The graph represents a wide range of entities and relationships, offering a broad but shallow view of the financial domain. Entities serve as seeds for querying Wikidata. A semi-automated process ensures accurate mapping, using the Wikidata Query Service, retrieve identifiers, names, and aliases of entities related to these stock exchanges, direct matches between stock tickers and Wikidata entries are automatically linked.The relationships captured by this graph include corporate links between parent companies, subsidiaries, and competitors, as well as industry and sector associations and limited integration of external macroeconomic indicators like inflation or GDP, depending on their availability in Wikidata. However, its limitations lie in the lack of domain-specific granularity and the potential for missing or outdated information, given its reliance on publicly contributed data.

2. 10-K Financial Report-Based Knowledge Graph; TThe GoLLIE Knowledge Graph, constructed from 10-K financial reports, provides a domain-specific, semi-open-domain extraction tailored for the financial sector. It represents detailed relationships and events directly relevant to stock performance. pre-processing is crucial to extract meaningful information efficiently. This step involves text segmentation, entity resolution, and preparation for information extraction.While its coverage is limited to explicitly mentioned entities in the reports, the GoLLIE graph excels in providing high-quality, domain-specific data compared to the broader but less granular Wikidata graph. Unlike Wikidata, which relies on public contributions and is more static, GoLLIE offers a dynamic representation of company-level events and superior data quality derived from official financial documents, making it a more effective tool for financial analysis.

GoLLIE operates as a semi-open-domain information extraction tool, meaning it is flexible enough to adapt to different domains but is specifically tuned for financial text. Unlike fully open-domain systems that aim to generalize across all topics, GoLLIE uses predefined templates and classes (e.g., BusinessEvent, FinancialReportEvent) to extract structured relationships from financial reports. This targeted approach ensures high precision in identifying relevant relationships while retaining some flexibility to adapt to various financial documents. GoLLIE's semi-open-domain nature makes it ideal for constructing knowledge graphs in the financial sector. It allows for the extraction of relationships with contextual qualifiers, such as transaction amounts and dates, which enrich the graph's utility for machine learning models.

Sentence Segmentation: Using NLTK's Punkt tokenizer, the raw 10-K reports are divided into sentences. This segmentation ensures that each sentence becomes a manageable unit for entity and relationship extraction.Entity resolution: Many 10-K reports use ambiguous terms such as "we", "the company" or "the corporate" to refer to the reporting organization. These terms are resolved and replaced with the full company name using spaCy's English language model.We employ GoLLIE, a recent LLM-based model, for extracting financial entities and relationships. GoLLIE uses Python class definitions to structure extracted facts, allowing for precise and context-aware extraction. The process includes:

Event Templates:- The chosen templates, Business Event, Financial Report Event, Product Launch Event, and Economic Indicator Event, are carefully selected for their direct relevance to factors influencing stock market performance. These templates are designed to capture critical types of information that significantly impact stock prices, ensuring that the extracted knowledge graph effectively models the relationships and events that drive market movements.These templates are selected because they represent distinct, complementary

dimensions of financial data corporate strategy, financial outcomes, product dynamics, and macroeconomic context. Together, they provide a comprehensive representation of the factors that influence stock prices, making the resulting knowledge graph rich, actionable, and highly relevant for predictive modeling in the financial domain.

1.Business Event: Represents corporate actions such as acquisitions, mergers, or partnerships. Example "Microsoft acquired LinkedIn on June 13, 2016." Extracted Fact:

```
BusinessEvent(
    mention="acquired",
    subjectorganization="Microsoft",
    pointintime="2016-06-13",
)
```

2. Financial Report Event: Captures earnings announcements, profit warnings, or other financial disclosures. Example "Apple reported quarterly revenue of 123.9 billion for Q1 2022."

```
FinancialReportEvent(
    mention="quarterly earnings",
    organization="Apple",
    value=123.9,
    pointintime="2022-01-31",
)
```

3.Product Launch Event: Represents the introduction or discontinuation of products.Example "Tesla launched the Model Y on March 14, 2019."

```
ProductLaunchEvent(
    mention="launched",
    organization="Tesla",
    productname="Model Y",
    pointintime="2019-03-14",
)
```

4.Economic Indicator Event: Captures macroeconomic metrics and their variations. Example"The US unemployment rate dropped to 3.5 percent in July 2022."

```
EconomicIndicatorEvent(
    indicator="Unemployment Rate",
    value=3.5,
    region="USA",
)
```

3.4 Embedding Generation

To transform the structured knowledge graph (KG) into a machine-readable format, TransE (Translation Embedding) is employed. TransE generates numerical embeddings for the nodes (entities) and edges (relationships) in the graph, encoding the latent structure and interdependencies of the financial domain. These embeddings allow machine learning models to leverage relational data for stock price prediction effectively.

TransE: Translation Embedding TransE is a widely-used embedding technique designed to represent relationships as translations in a vector space. It operates under the assumption that relationships between entities can be modeled by translating one entity vector (head) to another (tail) through a relation vector.

For a triplet (h,r,t), where: h: Head entity r: Relation t: Tail entity

TransE ensures that h+r = t; This means that the embedding of the head entity (h) added to the embedding of the relation (r) should be close to the embedding of the tail entity (t) in the vector space.

Example: For the triplet(”Microsoft”,”acquired”,”LinkedIn”), TransE computes the embeddings such that: Embedding(”Microsoft”)+Embedding(”acquired”)=Embedding(”LinkedIn”)

This translation mechanism makes TransE particularly effective for modeling one-to-one relationships like acquisitions, partnerships, or product launches.

3.5 Feature Integration

The proposed framework integrates traditional technical indicators(Open,High,Low,MA7, MA21,Return,Volatality etc.) with knowledge graph embeddings to construct a comprehensive feature set that enables machine learning models to predict stock prices effectively. This integration leverages both temporal patterns from historical price data and relational insights captured by the knowledge graph. Below, we detail the process of feature integration and its role in generating predictions and evaluating performance metrics such as RMSE and nDCG@10.

At each time step t, technical indicators are calculated from historical stock price data. These indicators provide insights into market trends and volatility. The technical indicators are numerical features that represent time-series trends, making them crucial for models like Long Short-Term Memory (LSTM) networks, which specialize in sequential data.Simultaneously, embeddings are generated for financial entities using the TransE model. These embeddings encode the latent relationships and dependencies among entities in the knowledge graph. The framework combines technical indicators and knowledge graph embeddings to create a unified feature vector, known as an asset vector. Asset Vector = [Technical indicators] + [Knowledge graph embeddings]

The enriched asset vectors are fed into the machine learning models (e.g., Linear Regression, Random Forest, LSTM). Then the models predict the price of the stock or the trend for time t+1. The evaluation metrics, RMSE and nDCG@10, further validate the effectiveness of this feature integration in improving the framework performance.

Chapter 4: Research Methodology/ Experimental Setup

This chapter outlines the experimental setup for implementing and evaluating the proposed stock market prediction framework. The experimental design integrates comprehensive data collection, preprocessing, model selection, and training methodologies to leverage both traditional technical indicators and knowledge graph embeddings for stock price prediction. Evaluation metrics like RMSE and nDCG@10 are employed to measure predictive accuracy and ranking quality, ensuring a robust assessment of the framework's performance.

4.1 Data collection and Pre-processing

The proposed stock market prediction framework begins with a comprehensive data collection and pre-processing to ensure the availability of high quality inputs for model training and evaluation. This phase involves the integration of historical stock price data, technical indicators, and domain-specific knowledge graph data.

The data set used in this study includes historical stock price data, technical indicators, and domain-specific knowledge graph data. Historical daily pricing data, including open, close, high, low prices, and trading volume, are collected from Yahoo Finance. Spanning 2020 to 2023, this extensive dataset ensures that the training data captures diverse market conditions, including economic recessions, booms, and anomalies such as the 2020 COVID-19 pandemic.

Annual filings sourced from the SEC EDGAR database provide granular insights into corporate activities such as mergers, acquisitions, product launches, and financial metrics. These reports are processed to build a fine-grained, domain-specific knowledge graph. Two knowledge graphs are constructed to represent financial relationships: the Wikidata Graph and the 10-K Graph. The Wikidata graph is extracted from Wikidata using SPARQL queries, representing over 100,000 entities and 450,000 relationships. This graph offers broad coverage, but limited granularity. In contrast, the 10-K graph is derived from 10-K reports using NLP-based information extraction techniques, focusing on high-granularity relationships such as acquisitions, partnerships, and financial disclosures.

To ensure that the data set is suitable for machine learning, extensive pre-processing is performed. Technical indicators are derived to capture essential market trends and patterns. The 7-day and 21-day moving averages capture short- and medium-term trends, filtering out noise, and highlighting directional changes in stock prices. Volatility is calculated as the standard deviation of closing prices, quantifying the risk associated with stock movements over a specified period. Lagged features, such as the closing prices for the previous day (t-1, t-2), are included to capture sequential dependencies, which are essential for time-series models like LSTM.

The dataset is divided into training and test sets to facilitate model evaluation. Training data includes all time points up to December 31, 2019, while testing data span from July 1, 2020, to December 31, 2023. A six-month gap between training and testing sets prevents data leakage and ensures that testing data represent unseen market conditions. The time points

are spaced one week apart, and Mondays serve as a reference point, resulting in 73 training time points and 25 testing time points.

Data cleaning is performed to ensure consistency and reliability. Assets appearing in the pricing data but missing from the knowledge graph are excluded. Outliers, such as stocks with extreme profits, such as meme stocks during the 2021 trading frenzy, are removed to prevent biased evaluations that could compromise the validity of the results.

4.2 Model Selection

Three machine learning models are selected to evaluate the impact of technical indicators and knowledge graph embeddings on the prediction of stock prices. Each model is chosen for its unique strengths in the handling of financial data.

Linear Regression the dependent variable y represents the stock price (e.g., closing price) at a specific time step t. The independent variables X=[x1,x2,x3....xn] are the features used for the prediction. These features include moving averages, volatility, lagged prices and embeddings.The model learns the values of B0 and Bi (cofficients of the learning function) during training by minimizing the error between the predicted stock prices and the actual stock prices in the historical dataset. Once the coefficients are learned, the model can predict future stock prices based on new feature values. For example, if we want to predict the closing price of a stock using the x1: 7-day moving average and the x2: 21-day moving average and x3: Volatility,the model learns the coefficients (B0,B1,B2,B3) using historical data. For a new day, if:

x1=150, x2=155, x3=2.5 and the learned coefficients are B0 = 10, B1 = 0.8, B2 = 0.6, B3 = -5, then predicted price

(y) = 10+(0.8×150)+(0.6×155)+(-5×2.5) = 192.5

Linear Regression offers a transparent and computationally efficient approach, making it as a starting point in predictive modeling for stock prices.

Random Forest takes a set of features derived from historical stock price data, technical indicators (e.g., moving averages, volatility), and additional features knowledge graph embeddings. The price of the stock at the future time step Pt+k(or percentage change in price) is the target variable the model aims to predict. Each tree in the forest independently predicts the stock price for the given input features. For example, Tree 1 might predict 100 doller, Tree 2 might predict 105 doller and Tree 3 might predict 102 doller for the same set of input features.

Long Short-Term Memory (LSTM) networks are a type of Recurrent Neural Network (RNN) designed to handle sequential data, input sequences consist of past stock prices and technical indicators (e.g., moving averages, volatility).Features are normalized to ensure numerical stability during training. This involves scaling stock prices and indicators between 0 and 1 using techniques like Min-Max normalization.Historical data is divided into overlapping windows. For example, a window size of 10 days means that data from days (t - 10) to (t - 1) are used to predict the price on day t. The input sequence (e.g., past 10 days of stock prices) is fed into the LSTM layer, the LSTM layer processes the data step by step, using its memory cell to retain relevant patterns (e.g., upward or downward trends).The final hidden state (hT) after processing the entire sequence represents the learned temporal features. After processing, the LSTM layer processes the sequence, and the output from the final LSTM cell is passed to a fully connected (dense) layer, A linear activation function(ReLU) is used for

stock price prediction, ensuring that the output can take any continuous value. LSTM uses a specialized version of backpropagation to adjust weights. Gradients are calculated and updated for all time steps to minimize the loss.LSTM predicts stock prices with higher accuracy than traditional models, especially when combined with enriched features like technical indicators and knowledge graph embeddings.

The models are trained using two distinct feature sets to evaluate the contribution of different data types to predictive performance. The Basic KPIs consist of straightforward metrics derived solely from stock price data, including 7-day and 21-day moving averages, as well as open, close, high, low prices, and trading volume. These features provide a foundational understanding of stock behavior based on historical prices and trading activity, offering a simple effective baseline for model evaluation. By focusing on these fundamental indicators, the basic technical indicators allow the models to capture short- and medium-term market trends, as well as overall price fluctuations.

In contrast, the Advanced technical indicators extend the Basic technical indicators by incorporating additional features designed to capture more complex patterns in the data. These include volatility, which measures the standard deviation of closing prices and provides insight into market risk and stability, and return, which reflects the percentage change in stock prices over a specific period. Advanced technical indicators also include lagged features, such as close lag 1 (previous day's closing price) and close lag 2 (closing price from two days prior), which are critical to capture sequential dependencies and momentum in the movements of the stock price. These enhancements enable the models to analyze a richer set of temporal dynamics compared to the Basic technical indicators.

Once Advanced technical indicators are calculated, knowledge graph embeddings are integrated into the feature set. These embeddings, generated using the TransE model, capture the latent relationships and interdependencies among financial entities such as companies, industries, and macroeconomic factors. By representing these complex relationships as numerical vectors, the embeddings enrich the feature set with relational insights that are not captured by traditional technical indicators. This final feature set, which combines advanced KPIs and knowledge graph embeddings, allows the models to leverage both temporal trends from price data and contextual relationships from the knowledge graph.

4.3 Model Training and Evaluation

The models are trained and evaluated on the enriched feature sets, and their performance is assessed using key metrics to ensure a comprehensive evaluation. During the training process, the integration of features is achieved by combining traditional technical indicators, such as moving averages and volatility, with embeddings generated using the TransE model for each asset. This enriched feature vector incorporates both temporal patterns from the technical indicators and relational insights from the knowledge graph embeddings, providing a comprehensive input to the predictive models. The enriched feature vector enables the models to capture both sequential dependencies and complex interdependencies between financial entities, enhancing their predictive capabilities.

The training and testing data are split based on specific time periods to simulate real-world market scenarios and prevent data leakage. The training dataset includes stock market data from January 1, 2019 (train start date) to May 31, 2022 (train end date), ensuring that the models are exposed to a diverse set of market conditions over this period. The testing dataset covers the period from June 1, 2022 (test start date) to December 31, 2023 (test end date), allowing the models to be evaluated on unseen data that reflects more recent and potentially

volatile market behavior. This temporal division ensures a clear separation between training and testing phases, maintaining the integrity of the evaluation process. Hyperparameter tuning is performed to optimize model performance for each algorithm. For Linear Regression, regularization parameters are optimized to balance bias and variance, preventing overfitting. For Random Forest, the number of trees and maximum depth are tuned to enhance predictive performance and manage computational complexity. For LSTM, the sequence length, the hidden layer size, the learning rate and the dropout rates are adjusted to prevent overfitting while maintaining temporal accuracy.

Models are optimized using appropriate loss functions tailored to their specific predictive tasks. The mean squared error (MSE) is used for regression-based predictions to minimize the difference between predicted and actual stock prices. Cross-entropy loss is employed for ranking-related tasks to enhance the ordering of predicted stock returns.

Evaluation metrics are used to evaluate the models' performance comprehensively. The root mean square error (RMSE) [31] quantifies the prediction error by measuring the average square difference between the predicted and actual stock prices. A lower RMSE value indicates higher accuracy and better model performance in predicting stock prices. The Normalized Discounted Cumulative Gain (nDCG@10) evaluates the ranking quality of the top 10 stocks, measuring how well the model's ranking aligns with actual stock performance. Higher nDCG@10 scores indicate better alignment between predicted and actual rankings, which is crucial for investment decision making. [32]

The experimental setup faced several challenges and limitations that affected the data preparation, knowledge graph construction, and model training processes. Addressing these challenges was critical to ensuring the reliability and accuracy of the framework, but some limitations remain. 10-K financial reports, while rich in information, often contained noise in the form of ambiguous language or redundant disclosures, making the extraction of precise entities and relationships challenging. Additionally, inconsistencies in Wikidata entries, such as outdated information or conflicting attributes, required extensive manual verification and filtering to ensure accuracy. The construction of knowledge graphs from various sources presented technical and conceptual challenges. The extraction of entities and relationships from unstructured financial text, such as 10-K reports, often encountered ambiguities due to the inherent vagueness of corporate language. For example, phrases like "the company acquired another firm" needed to be contextually resolved to identify specific entities. The limitations of NLP tools, including spaCy and Gollie, further restricted the extraction process, particularly in the handling of complex or nested relationships. In the Wikidata graph, the broad coverage led to irrelevant connections, which required additional filtering to retain domain-specific information. Training machine learning models introduced computational and algorithmic challenges. The LSTM models, while effective for time-series predictions, were prone to overfitting due to the complexity of the feature set and the sequential nature of the data.Random Forest, on the other hand, faced scalability issues when handling large feature sets, especially after incorporating advanced TIs and knowledge graph embeddings. The computational demands of training Random Forest on high-dimensional data required significant memory and processing power, which constrained the experimentation process.

Chapter 5: Results

This chapter presents the results of the experiments conducted to evaluate the proposed stock market prediction framework. The analysis focuses on the predictive performance of the machine learning models (Linear Regression, Random Forest, and LSTM) using two distinct feature sets: traditional technical indicators (Basic TIs) and enriched features incorporating knowledge graph embeddings (Advanced TIs). The performance of models is assessed using key evaluation metrics, including Root Mean Squared Error (RMSE) and Normalized Discounted Cumulative Gain (nDCG@10). Below is the result matrix, which holds the values of RMSE and nDCG@10 and tells us how the modal performs in actual terms.

Model	TIs	KG	Embedding	RMSE	nDCG@10
Linear Regression	Base	-	-	0.9514	0.0125
Linear Regression	Advance	-	-	0.8450	0.0137
Random Forest	Base	-	-	0.7019	0.5019
Random Forest	Advance	-	-	0.6218	0.5715
LSTM	Base	-	-	0.6253	0.7315
LSTM	Advance	-	-	0.5321	0.7657
Random Forest	Base	Wikidata	TransE	0.4415	0.7515
Random Forest	Advance	Wikidata	TransE	0.4286	0.7715
LSTM	Base	Wikidata	TransE	0.4582	0.7612
LSTM	Advance	Wikidata	TransE	0.4252	0.7689
LSTM	Base	GoLLIE KG	TransE	0.4357	0.7859
LSTM	Advance	GoLLIE KG	TransE	0.4157	0.7759
Random Forest	Base	GoLLIE KG	TransE	0.4351	0.7659
Random Forest	Advance	GoLLIE KG	TransE	0.4317	0.7674

Table 5.1: Final Result Matrix

5.1 RQ0: Which Baseline Model is the Most Effective?

To determine the most effective baseline model for stock price prediction, the performance of Linear Regression, Random Forest, and Long Short-Term Memory (LSTM) models was evaluated using the Basic Technical Indicators (TIs) feature set. The Basic TIs include essential metrics such as moving averages (7-day and 21-day), open, close, high, low prices, and trading volume, and advance TIs (Close lag1, close lag2, volatility, Return which are added. The results were assessed using two metrics: Root Mean Squared Error (RMSE), which measures predictive precision, and Normalized Discounted Cumulative Gain at rank 10 (nDCG@10), which evaluates the ranking quality of the top 10 predicted stocks.

The Base model of Linear Regression using only Basic TIs exhibited the weakest performance among the three models, with an RMSE of 0.9514 and an nDCG@10 score of 0.0125. These results indicate that Linear Regression struggles to capture the complex and non-linear relationships in stock price data, leading to low predictive accuracy and poor ranking performance. The Advanced Linear Regression model, while slightly better, achieved a reduced RMSE of 0.8450 and a marginally improved nDCG@10 score of 0.0137.

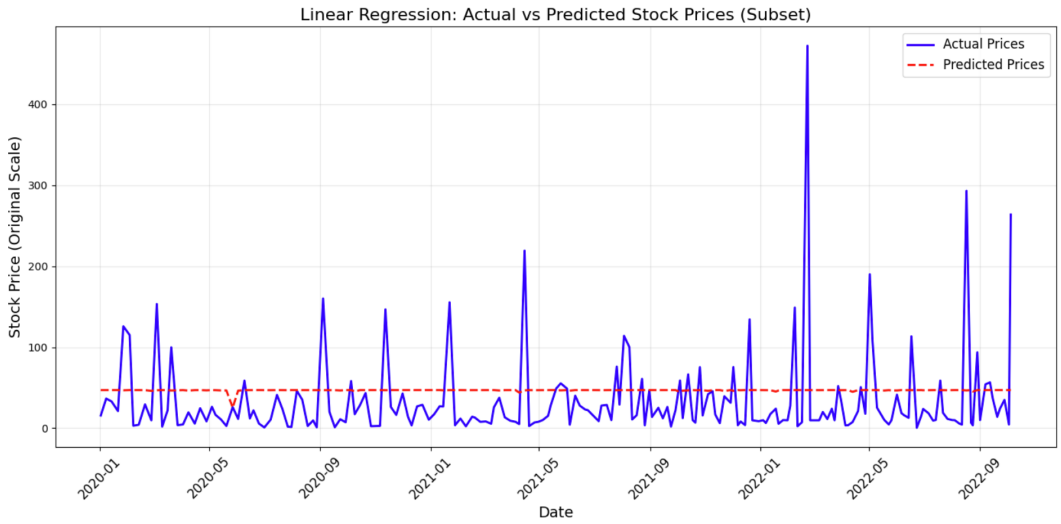

Figure 5.1: Linear Regression Predictions

The Base Random Forest model demonstrated significant improvement over Linear Regression. With an RMSE of 0.7019 and an nDCG@10 score of 0.5019, it effectively captured the underlying interactions between the technical indicators, leading to better predictions and rankings. The Advanced Random Forest model, which incorporates refined features, further reduced the RMSE to 0.6218 and improved the nDCG@10 score to 0.5715.

The Base LSTM model, designed to capture temporal dependencies, outperformed both Linear Regression and Random Forest with an RMSE of 0.6253 and an nDCG@10 score of 0.7315. This highlights LSTM's strength in sequential data to capture trends and patterns in stock price movements. The Advanced LSTM model achieved the best overall performance, with an RMSE of 0.5321 and an nDCG@10 score of 0.7615. The model's ability to incorporate enriched features led to superior predictive accuracy and exceptional ranking quality, making it the most effective baseline model.The results clearly demonstrate that LSTM outperforms Linear Regression and Random Forest across both metrics. Its ability to model sequential data gives it a distinct advantage in capturing the temporal dependencies of stock price movements.

5.2 RQ1: Does the Integration of More Advanced Technical Indicators Enhance Performance?

To evaluate whether the integration of advanced technical indicators (TIs), through the inclusion of Wikidata Knowledge Graph embeddings (generated using the TransE model), improves the performance of stock price prediction models, we tested the Random Forest and LSTM models under both Base and Advanced configurations. The Base configuration includes traditional technical indicators (e.g. moving averages (7-day and 21-day, open, close, high, low prices, and trading volume), while the Advanced configuration these features with Wikidata Knowledge Graph embeddings.

The Base Random Forest model with Wikidata Knowledge Graph embeddings, achieved an RMSE of 0.4415 and an nDCG@10 score of 0.7515, demonstrating a significant improvement over previous configurations that lacked knowledge graph embeddings. In the Advanced Random Forest model, where advanced technical indicators were integrated with the embeddings, the RMSE improved further and the nDCG@10 score also increased. This highlights that the combination of technical indicators and Wikidata embeddings enables

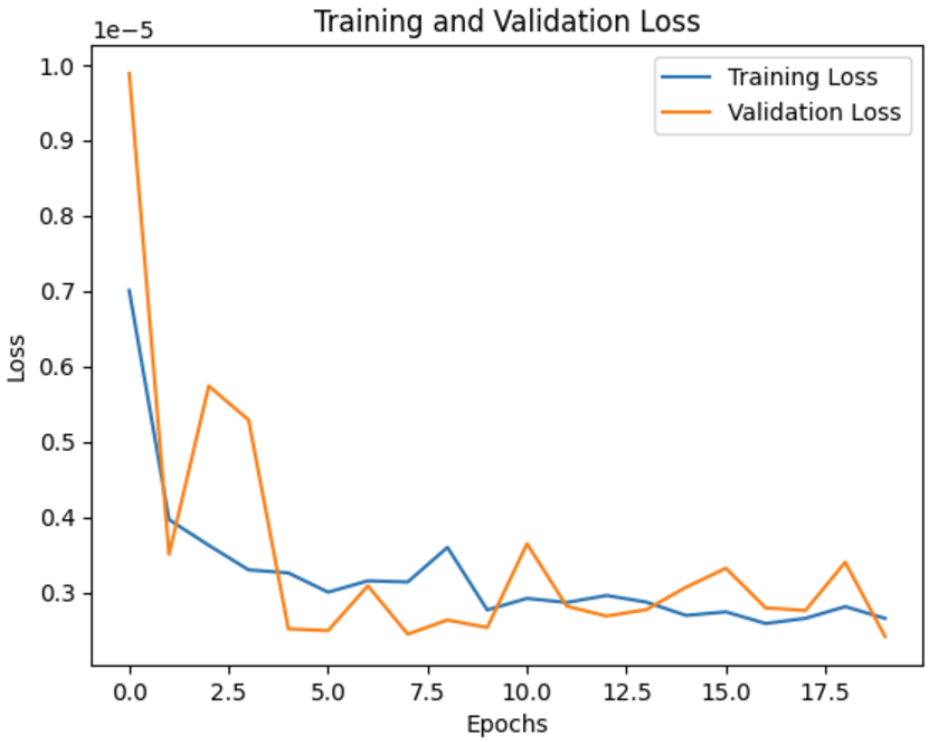

Figure 5.2: LSTM Loss Graph

Random Forest to better capture relational insights, enhancing both predictive accuracy and ranking quality.

The Base LSTM model, utilizing Wikidata embeddings alongside basic technical indicators, achieved an RMSE of 0.4582 and an nDCG@10 score of 0.7612. While its RMSE was slightly higher than Random Forest, its nDCG@10 score indicated better ranking quality, reflecting LSTM's ability to capture sequential dependencies in addition to relational insights.The Advanced LSTM model, with further enriched technical indicators, showed substantial improvement. It achieved the lowest RMSE among all models so far and a competitive nDCG@10 score of 0.7689. These results demonstrate that LSTM, when combined with advanced technical indicators and Wikidata embeddings, effectively gives superior predictions.

Both Random Forest and LSTM benefited from the integration of Wikidata embeddings, with consistent improvements in the RMSE and nDCG@10 scores for the Advanced configuration. The results suggest that advanced technical indicators derived from Wikidata Knowledge Graphs provide valuable relational insights, enhancing the models' understanding of interdependencies between financial entities.While Random Forest demonstrated slightly better RMSE scores overall, LSTM consistently delivered higher nDCG@10 scores, highlighting its superior ability to rank top performing stocks effectively. This indicates that while Random Forest excels in predictive accuracy, LSTM offers a better balance between accuracy and ranking quality, particularly in stock selection tasks.

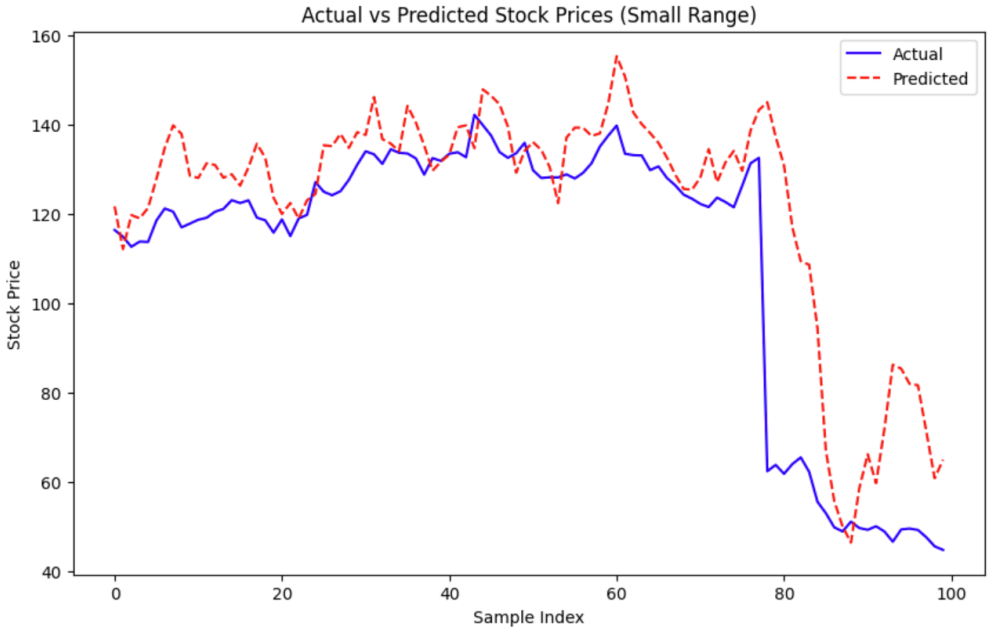

Figure 5.3: LSTM's Best Price Prediction

5.3 RQ2: Does Company Evidence from Knowledge Graphs Result in Better Predictions?

This section investigates whether integrating GoLLIE Knowledge Graph embeddings (generated from 10-K financial reports) into the prediction models results in better performance compared to using embeddings from the generic Wikidata Knowledge Graph. The performance of the Random Forest and LSTM models was evaluated using both base technical indicators (TI) and advanced technical indicators (TI) combined with the embeddings of each knowledge graph.

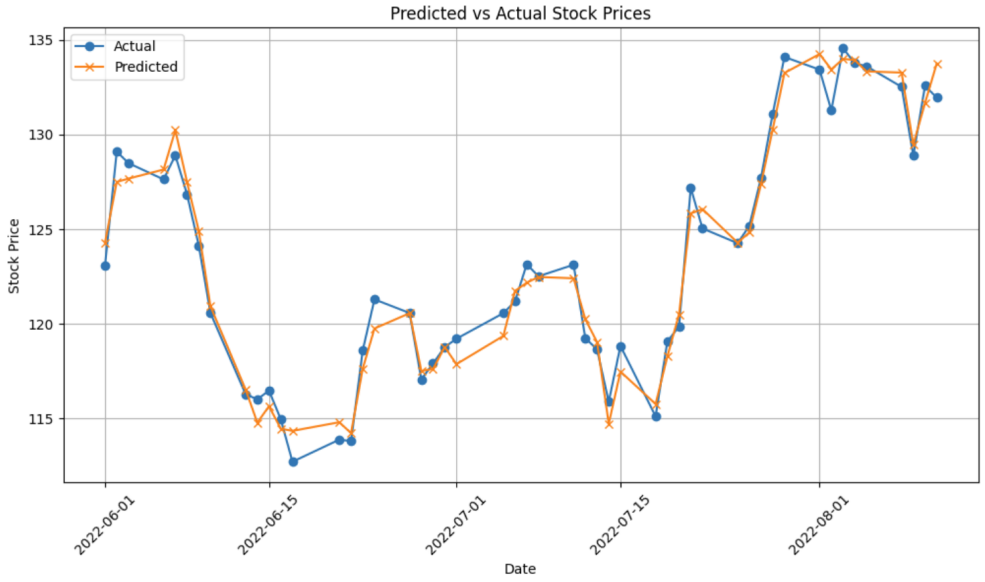

Figure 5.4: Random Forest Best Prediction

When using Base TIs, the GoLLIE Knowledge Graph embeddings improved RMSE to

0.4351 and nDCG@10 to 0.7659, compared to the Wikidata embeddings, which yielded less effectiveness. This demonstrates a slight improvement in both predictive accuracy and ranking quality when using domain-specific GoLLIE knowledge graphs. In the Advanced Random Forest model, GoLLIE KG embeddings also outperformed the Wikidata embeddings, achieving an RMSE of 0.4317 and an nDCG@10 of 0.7674. While the RMSE improvement was marginal, the nDCG@10 scores showed that GoLLIE KG embeddings were slightly less effective for ranking compared to the Wikidata embeddings.

The Base LSTM model saw a significant improvement when using GoLLIE KG embeddings, achieving an RMSE of 0.4357 and an nDCG@10 of 0.7859, compared to the Wikidata. These results indicate that GoLLIE embeddings provide more actionable insights for sequential models such as LSTM, improving both predictive accuracy and ranking quality.For the Advanced LSTM model, the use of GoLLIE KG embeddings further reduced the RMSE to 0.4157, the best among all configurations, and increased the nDCG@10 to 0.7759, outperforming the Advanced LSTM model with Wikidata embeddings. This highlights the value of GoLLIE embeddings in capturing domain-specific relational data critical for stock price prediction. Across all models and configurations, the GoLLIE Knowledge Graph embeddings

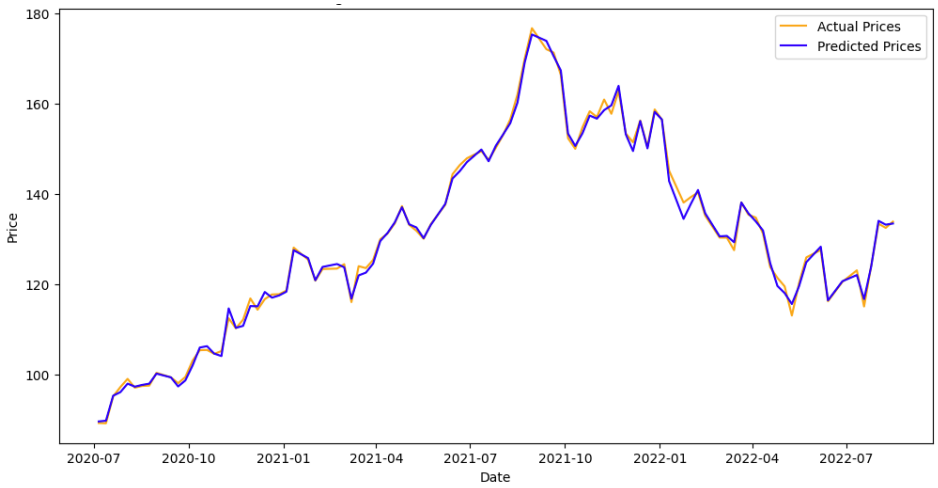

Figure 5.5: LSTM's Custom KG Prediction

consistently outperformed Wikidata embeddings in RMSE, indicating better predictive accuracy. This improvement comes from the domain-specific nature of GoLLIE KG, which captures granular financial relationships, such as mergers, acquisitions, and partnerships, directly relevant to stock market behavior.In terms of ranking quality (nDCG@10), GoLLIE embeddings generally performed better or were on par with Wikidata embeddings, with LSTM models showing the most pronounced improvements. This suggests that GoLLIE KG provides more nuanced relational insights for sequential models. While GoLLIE embeddings improved RMSE for Random Forest, the improvements in nDCG@10 were less pronounced compared to Wikidata embeddings. This may indicate that Random Forest struggles to fully utilize the rich relational insights provided by GoLLIE KG embeddings. The LSTM models benefited the most from GoLLIE embeddings, achieving the lowest RMSE and the highest nDCG@10 scores in both Base and Advanced configurations. This demonstrates LSTM's ability to effectively integrate sequential and relational data for superior predictions and rankings.

Chapter 6: Conclusion

This study has successfully demonstrated the value of integrating knowledge graph embeddings with machine learning models for stock market prediction. The results emphasize that domain-specific knowledge graphs, such as those derived from 10-K financial reports using GoLLIE, offer significant advantages over generic knowledge graphs such as Wikidata. These graphs capture financial relationships, such as mergers, acquisitions, and partnerships, which are critical for modeling stock price behavior. Among the evaluated models, LSTM consistently outperformed Linear Regression and Random Forest. The integration of advanced technical indicators enriched with embeddings further improved predictive accuracy and ranking quality, underscoring the importance of feature enrichment. Although Random Forest showed strong capabilities in handling nonlinear relationships, its lack of temporal awareness limited its effectiveness compared to LSTM. These findings validate the proposed framework as a robust approach to addressing the complexities of stock market prediction.

6.1 Future Works

Despite the promising results, there remain several opportunities for future research to further enhance this framework. One key direction is the development of dynamic knowledge graphs that evolve with real-time financial events, such as mergers, partnerships, or economic announcements, allowing models to incorporate the latest relational insights. Furthermore, integrating multiple types of graph, such as supply chain networks or sentiment graphs on social media, could provide a broader and richer relational context for financial predictions. Future studies could also explore more advanced embedding techniques, such as graph neural networks (GNNs) or relationship graph convolutional networks (R-GCNs), to capture higher-order dependencies and more complex graph structures. Incorporating macroeconomic indicators, like GDP growth or inflation rates, as additional features could further improve the models' understanding of broader market dynamics.

Real-time prediction frameworks represent another important area for exploration, where streaming data from live financial news or stock tickers could be integrated into the predictive pipeline. Scalability and computational efficiency are also critical areas for improvement, particularly for handling large-scale and real-time datasets. Furthermore, enhancing model interpretability through techniques such as SHAP or LIME would provide actionable insights into how specific features and relationships drive predictions, making the framework more useful to investors and analysts. These advances would not only refine the predictive accuracy but also improve the robustness, scalability, and applicability of stock market prediction models in an increasingly volatile and interconnected financial ecosystem.

Bibliography

[1] Yahya Eru Cakra and Bayu Distiawan Trisedya. Stock price prediction using linear regression based on sentiment analysis. In *2015 international conference on advanced computer science and information systems (ICACSIS)*, pages 147–154. IEEE, 2015.

[2] Rong Chen, Han Xiao, and Dan Yang. Autoregressive models for matrix-valued time series. *Journal of Econometrics*, 222(1):539–560, 2021.

[3] Miao Fan, Qiang Zhou, Emily Chang, and Thomas Fang Zheng. Transition-based knowledge graph embedding with relational mapping properties. In *Proceedings of the 28th Pacific Asia conference on language, information and computation*, pages 328–337. Waseda University, 2014.

[4] Jiali Fang, Yafeng Qin, and Ben Jacobsen. Technical market indicators: An overview. *Journal of behavioral and experimental finance*, 4:25–56, 2014.

[5] Dattatray P Gandhmal and Kannan Kumar. Systematic analysis and review of stock market prediction techniques. *Computer Science Review*, 34:100190, 2019.

[6] Martin Grohe. word2vec, node2vec, graph2vec, x2vec: Towards a theory of vector embeddings of structured data. In *Proceedings of the 39th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems*, pages 1–16, 2020.

[7] Burak Gülmez. Stock price prediction with optimized deep lstm network with artificial rabbits optimization algorithm. *Expert Systems with Applications*, 227:120346, 2023.

[8] MEAG Hiransha, E Ab Gopalakrishnan, Vijay Krishna Menon, and KP Soman. Nse stock market prediction using deep-learning models. *Procedia computer science*, 132:1351–1362, 2018.

[9] Aidan Hogan, Eva Blomqvist, Michael Cochez, Claudia d'Amato, Gerard De Melo, Claudio Gutierrez, Sabrina Kirrane, José Emilio Labra Gayo, Roberto Navigli, Sebastian Neumaier, et al. Knowledge graphs. *ACM Computing Surveys (Csur)*, 54(4):1–37, 2021.

[10] Pavan Kumar Illa, Balakesavareddy Parvathala, and Anand Kumar Sharma. Stock price prediction methodology using random forest algorithm and support vector machine. *Materials Today: Proceedings*, 56:1776–1782, 2022.

[11] João Tovar Jalles. Structural time series models and the kalman filter: a concise review. 2009.

[12] Gang Ji, Jingmin Yu, Kai Hu, Jie Xie, and Xunsheng Ji. An adaptive feature selection schema using improved technical indicators for predicting stock price movements. *Expert Systems with Applications*, 200:116941, 2022.

[13] Shaoxiong Ji, Shirui Pan, Erik Cambria, Pekka Marttinen, and S Yu Philip. A survey on knowledge graphs: Representation, acquisition, and applications. *IEEE transactions on neural networks and learning systems*, 33(2):494–514, 2021.

[14] Teja Kattenborn, Jens Leitloff, Felix Schiefer, and Stefan Hinz. Review on convolutional neural networks (cnn) in vegetation remote sensing. *ISPRS journal of photogrammetry and remote sensing*, 173:24–49, 2021.

[15] Luckyson Khaidem, Snehanshu Saha, and Sudeepa Roy Dey. Predicting the direction of stock market prices using random forest. *arXiv preprint arXiv:1605.00003*, 2016.

[16] Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. Learning entity and relation embeddings for knowledge graph completion. In *Proceedings of the AAAI conference on artificial intelligence*, volume 29, 2015.

[17] Yang Liu, Qingguo Zeng, Huanrui Yang, and Adrian Carrio. Stock price movement prediction from financial news with deep learning and knowledge graph embedding. In *Knowledge Management and Acquisition for Intelligent Systems: 15th Pacific Rim Knowledge Acquisition Workshop, PKAW 2018, Nanjing, China, August 28-29, 2018, Proceedings 15*, pages 102–113. Springer, 2018.

[18] Adil Moghar and Mhamed Hamiche. Stock market prediction using lstm recurrent neural network. *Procedia computer science*, 170:1168–1173, 2020.

[19] Abdullah Bin Omar, Shuai Huang, Anas A Salameh, Haris Khurram, and Muhammad Fareed. Stock market forecasting using the random forest and deep neural network models before and during the covid-19 period. *Frontiers in Environmental Science*, 10:917047, 2022.

[20] Felipe Barboza Oriani and Guilherme P Coelho. Evaluating the impact of technical indicators on stock forecasting. In *2016 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 1–8. IEEE, 2016.

[21] Bhawna Panwar, Gaurav Dhuriya, Prashant Johri, Sudeept Singh Yadav, and Nitin Gaur. Stock market prediction using linear regression and svm. In *2021 International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE)*, pages 629–631. IEEE, 2021.

[22] BS Pramod and Mallikarjuna Shastry Pm. Stock price prediction using lstm. *Test Engineering and Management*, 83:5246–5251, 2020.

[23] Perry Sadorsky. A random forests approach to predicting clean energy stock prices. *Journal of Risk and Financial Management*, 14(2):48, 2021.

[24] Shuai Sang and Lu Li. A novel variant of lstm stock prediction method incorporating attention mechanism. *Mathematics*, 12(7):945, 2024.

[25] Alex Sherstinsky. Fundamentals of recurrent neural network (rnn) and long short-term memory (lstm) network. *Physica D: Nonlinear Phenomena*, 404:132306, 2020.

[26] Robert J Shiller. *Market volatility*. MIT press, 1992.

[27] Gaurang Sonkavde, Deepak Sudhakar Dharrao, Anupkumar M Bongale, Sarika T Deokate, Deepak Doreswamy, and Subraya Krishna Bhat. Forecasting stock market prices using machine learning and deep learning models: A systematic review, performance analysis and discussion of implications. *International Journal of Financial Studies*, 11(3):94, 2023.

[28] Xiaogang Su, Xin Yan, and Chih-Ling Tsai. Linear regression. *Wiley Interdisciplinary Reviews: Computational Statistics*, 4(3):275–294, 2012.

[29] Manuel R Vargas, Carlos EM Dos Anjos, Gustavo LG Bichara, and Alexandre G Evsukoff. Deep leaming for stock market prediction using technical indicators and financial news articles. In *2018 international joint conference on neural networks (IJCNN)*, pages 1–8. IEEE, 2018.

[30] Wayne F Velicer and Peter C Molenaar. Time series analysis for psychological research. *Handbook of Psychology, Second Edition*, 2, 2012.

[31] Weijie Wang and Yanmin Lu. Analysis of the mean absolute error (mae) and the root mean square error (rmse) in assessing rounding model. In *IOP conference series: materials science and engineering*, volume 324, page 012049. IOP Publishing, 2018.

[32] Yining Wang, Liwei Wang, Yuanzhi Li, Di He, and Tie-Yan Liu. A theoretical analysis of ndcg type ranking measures. In *Conference on learning theory*, pages 25–54. PMLR, 2013.