Adaptive Model Selection in Stock Market Prediction: A Modular and Scalable Big Data Analytics Approach

MohammadEhsan Akhavanpour* and Saeed Samet

School of Computer Science, University of Windsor, Windsor, ON, Canada
Email: akhavanm@uwindsor.ca (M.A.); saeed.samet@uwindsor.ca (S.S.)
*Corresponding author

Manuscript received January 10, 2024; revised February 6, 2024; accepted April 7, 2024; published May 10, 2024

***Abstract*—This paper introduces an innovative architecture integrating Apache Kafka and microservices to enhance real-time stock market prediction. Our approach dynamically selects the most effective predictive model based on current market conditions, ensuring consistent accuracy. The key research method involves deploying Apache Kafka for real-time data streaming, coupled with a microservices framework to maintain scalability and adaptability. Our methodology includes a thorough evaluation of various machine learning models (specifically focusing on R^2, the coefficient of determination, as the metric) to ascertain their performance across different market scenarios. The results demonstrate the architecture's ability to handle high data volume and velocity, while accurately adapting to market changes. The adaptability is evidenced by the varying performance of models like Convolutional Neural Network (CNN), Gate Recurrent Unit (GRU), and Long Short-Term Memory (LSTM) across different entities such as Royal Bank of Canada, Google, and EUR/USD, with the system successfully identifying the most suitable model in real-time. This architecture not only provides a scalable solution for stock market prediction but also sets the foundation for future exploration in other real-time data-intensive domains.**

***Keywords*—Apache Kafka, microservices architecture, real-time model switching, financial market prediction**

I. Introduction

The stock market plays a pivotal role in global finance, with billions of dollars being traded daily across various exchanges worldwide [1]. These transactions encompass individual investors, hedge funds, and institutional participants who engage in trading with a diverse range of investment strategies. Traditional approaches for predicting stock market movements have predominantly employed either fundamental analysis or technical analysis [2, 3].

Imagine a bustling digital marketplace where the currencies of more than a hundred nations are being exchanged at lightning speeds. On one side, you have traditional stock exchanges with blue-chip companies from the U.S., setting their stock prices based on an intricate dance of supply and demand. On the other side, you have the electrifying world of cryptocurrencies like Bitcoin, where prices fluctuate wildly based on the speculative maneuvers of "bulls" and "bears". This intricate tapestry is woven from countless threads of buying and selling behaviors, influenced by myriad factors that defy simplistic explanation. It's a dynamic ecosystem where the old and the new collide, underscoring the need for more robust predictive models [4].

Fundamental analysis is geared towards understanding a company's intrinsic value by scrutinizing various factors such as revenue, expenses, growth rates, and market position [5]. When forecasting indices, which comprise multiple company stocks, fundamental analysis extends its scope to include macroeconomic factors such as trade balances, exchange rates, and national productivity. This approach endeavors to uncover whether a stock or index is overvalued or undervalued, providing long-term investment signals. In contrast, technical analysis zeroes in on historical stock price and volume data, aiming to predict future price movements [2]. The underlying principle is that all market information is already reflected in stock prices, and therefore, studying price patterns and trends can yield actionable insights. While both these methods offer valuable perspectives, they are increasingly deemed insufficient for capturing the evolving complexities of modern stock markets, which are influenced by a myriad of interrelated variables [6]. Advances in technology have paved the way for machine learning and computational intelligence techniques to fill this gap [7].

Studies leveraging Support Vector Machines (SVM), Neural Networks, and Deep Learning have emerged, aiming to improve predictive accuracy [3]. High-frequency and algorithmic trading have especially been revolutionized through technological advancements [8]. Companies like Renaissance Technologies, for example, are setting new benchmarks for market performance through the strategic utilization of big data and computational algorithms [9]. However, even these sophisticated machine learning models have shown limitations, particularly in their adaptability to changing market conditions and in their capability to integrate multiple data sources seamlessly [10]. Against this backdrop, the need arises for an advanced, adaptive architecture capable of handling the diverse and complex landscape of stock market prediction. Such an architecture should incorporate the strengths of both fundamental and technical analysis while also leveraging state-of-the-art machine learning algorithms like Convolutional Neural Networks (CNN) and Long Short-Term Memory networks (LSTM) for enhanced performance and adaptability [11].

To seamlessly navigate this intricate landscape, we must also consider the challenges introduced by the era of big data. In the age of big data, stock market prediction faces unique challenges related to the 5Vs: Volume, Velocity, Variety, Veracity, and Value [12, 13]. The sheer Volume of data generated by financial transactions, market feeds, and social media is enormous [12]. The Velocity at which this data arrives and needs to be processed is staggering, requiring real-time analytics [13]. The Variety of data sources adds additional complexity, as data comes in structured, semi-structured, and unstructured formats [14]. Veracity deals with the uncertainty in data available, which could be due to data inconsistency, incompleteness, and available structure [15].

DOI: 10.7763/IJCTE.2024.V16.1353

Moreover, extracting Value from this massive and diverse set of data is a significant challenge [16]. Imagine a bustling digital marketplace where the currencies of more than a hundred nations are being exchanged at lightning speeds. In essence, the primary aim of this study is to propose a novel architecture that marries traditional analytical approaches with modern computational techniques. By doing so, we aspire to develop a more reliable, adaptive, and potentially profitable predictive system for stock market investments. This architecture addresses the noted gaps and limitations in existing models and methods, offering a comprehensive solution designed to adapt to diverse market scenarios.

The current landscape of stock market prediction is fraught with challenges, primarily stemming from the rapidly evolving and complex nature of financial markets. Traditional predictive models often fall short in handling the intricate interplay of global economic factors, leading to limited adaptability and accuracy. Furthermore, the sheer volume, velocity, and variety of financial data in the digital age pose significant challenges for conventional analytical methods. Our research method directly addresses these issues by introducing an innovative architecture that combines the power of machine learning, Apache Kafka, and microservices. This approach not only enhances the adaptability and scalability of predictive models but also ensures real-time processing and analysis of diverse data streams. The integration of these advanced technologies enables our system to dynamically adjust to market changes and offer more accurate predictions. Thus, our method presents a significant advancement over traditional models, offering a more robust, flexible, and efficient solution for stock market prediction in today's data-driven financial world.

As we move forward, we delve deeper into the relevant literature that forms the foundation of our research, detailing existing models and identifying the gaps that our study aims to fill. This is followed by the methodology section, which elucidates the architecture, design principles, and machine learning algorithms employed in our predictive model. The "Implementation and Results" section offers a comprehensive analysis of the model's performance, validated against real-time data from various financial markets. Finally, we conclude by summarizing the key contributions of this study and discussing potential avenues for future research.

II. Literature Review

Numerous academic publications have aimed to enhance the precision of stock market forecasts through the creation of advanced predictive models [3, 7]. Some research has even indicated that these models have the potential to be profitable [17, 18]. Despite its importance, accurately predicting stock market trends remains an extremely challenging endeavor within the domain of financial research [19]. Notably, an investor's ability to consistently outperform the market in terms of risk-adjusted returns may be at odds with the principles of the Efficient Market Hypothesis (EMH).

Initially proposed by Fama [20], the EMH posits that market prices evolve in a random manner, thereby making it impossible to predict future price changes based on existing information. The theory further categorizes market efficiency into three forms: weak-form, semi-strong form, and strong-form [18, 20].

In the weak-form efficiency, the hypothesis asserts that current stock prices already incorporate all historical price information, rendering technical analysis ineffective for anticipating future price shifts [20]. The semi-strong form, on the other hand, claims that all public information, not just past prices, is accounted for in the current stock prices. Therefore, even with access to a broad spectrum of publicly disclosed information such as economic indicators or company news, an investor cannot consistently outperform the market [20].

The most rigorous form, the strong-form efficiency, argues that even insider information is reflected in stock prices. Consequently, no investor can achieve consistently better returns than the market average, even when armed with proprietary information [20]. This strong form presents an extreme viewpoint, with even [20] himself noting that it is unlikely that insider information cannot be exploited for superior returns.

In recent developments, Chen *et al.* [21] have addressed the complexities of real-time news impact prediction on financial markets, a task that traditionally challenges finance experts with limited IT expertise. Their approach recognizes the shortcomings of conventional machine learning models that rely on low-level features extracted from event-based streams, often leading to suboptimal outcomes. To overcome these limitations, they proposed a novel technique leveraging real-time data preprocessing tailored to domain-specific event patterns, resulting in higher-quality datasets. This method is encapsulated within a systematic framework that integrates sentiment analysis, complex event processing, and Automated Machine Learning (AutoML). The framework's Service-Oriented Architecture (SOA) ensures flexible component selection and seamless integration, enabling finance experts to define domain-specific patterns and generate meaningful prediction results with minimal IT intervention. This innovation not only streamlines the predictive process but also empowers domain experts to engage more directly in the analysis, as evidenced by their prototype's successful application in a real-life price movement prediction scenario involving three years of news and financial market data.

In the realm of financial enterprise systems, the significance of real-time event streaming cannot be overstated. Sanjana *et al.* [22] underscore this point by emphasizing the role of Apache Kafka as a leading framework for real-time data streaming. Kafka's capabilities extend beyond mere data transmission; it provides a robust, distributed, and fault-tolerant infrastructure for capturing, storing, and processing event streams. The adaptability of Kafka enables seamless integration with various systems and applications, facilitating immediate data analysis and processing. This attribute of Kafka makes it an invaluable asset in diverse sectors, particularly in finance where rapid decision-making and operational efficiency are paramount. The authors elaborate on how Kafka's real-time event streaming can revolutionize business operations by expediting decision-making processes, enhancing operational efficiency, and refining customer experiences. Their study delves into Kafka's potential as a superior alternative for augmenting event streaming within financial enterprises, marking a significant stride in the evolution of real-time data handling in complex financial systems.

In exploring advancements in predictive modeling, recent studies in diverse domains have demonstrated the versatility and potency of machine learning techniques. Chen *et al*. [21] elucidated a robust framework for real-time news impact prediction on financial markets, highlighting the synergy between domain-specific event patterns and Automated Machine Learning (AutoML) to enhance prediction accuracy and computational efficiency in the context of dengue outbreak prediction. Similarly, Mahdavi and Khademi [23] leveraged neuro-fuzzy systems in conjunction with data mining techniques, illustrating their application in the accurate forecasting of oil production, emphasizing the integration of data cleaning and pre-processing to refine ANFIS algorithm performance. Furthermore, Radhika and Shashi showcased the superiority of Support Vector Machines (SVM) over traditional neural network models like MLP in atmospheric temperature prediction, underscoring the critical role of parameter selection in optimizing SVM performance [24]. These studies collectively underscore the expanding horizon of machine learning applications across various sectors, reaffirming the necessity for continuous innovation and adaptation in predictive modeling methodologies.

As for the next section, we provide a general overview of the data and machine learning techniques used in stock market prediction.

III. Analytical Foundations and Methodologies in Stock Market Prediction

A. Data Sources

In stock market prediction research, a variety of variables have been explored to improve forecasting accuracy. Technical indicators, financial variables, and macro-economic variables are the most influential factors affecting stock prices [25]. However, there is no general consensus about the specific variables that should be used, and studies often incorporate different sets of variables. These variables are categorized into four main categories: technical indicators, macro-economic variables, fundamental indicators, and other variables (Fig. 1) [4].

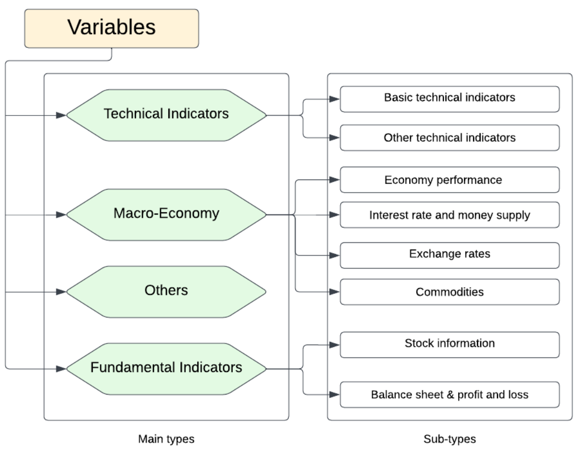

Fig. 1. Variable categories for stock price and return predictions [4].

1) *Technical indicators*

Widely used in most prediction studies, technical indicators are divided into “basic technical indicators” and “other technical indicators” [3, 26, 27]. Some research also explores these indicators in a "momentum space" instead of a continuous variable [28].

2) *Macro-economic variables*

This includes exchange rates, commodities, economic performance, and interest rates and money supply [29, 30].

3) *Fundamental indicators*

This involves stock information variables and balance sheet & profit and loss statement variables [31].

4) *Other variables*

This encompasses diverse data types like price data of other indices, financial news, email data, and social media posts [32, 33].

B. Machine Learning Techniques

1) *Overview of supervised learning*

Supervised learning remains the cornerstone in the realm of stock market prediction. It encompasses a structured approach where the model is trained on labeled data, learning to map input features to the target output [25, 34]. Key algorithms in this category include Artificial Neural Networks (ANNs), Support Vector Machines (SVMs), Random Forests, K-Nearest Neighbors (KNN), and Bayesian Networks. These algorithms have been widely applied due to their ability to handle complex, nonlinear relationships inherent in financial data [25, 35].

2) *Feature extraction techniques*

Effective feature extraction is vital for improving the performance of machine learning models in stock prediction. Techniques such as Convolutional Neural Networks (CNN), Principal Component Analysis (PCA), and Genetic Algorithms are employed to extract meaningful patterns and relationships from financial data. CNNs, in particular, have shown significant promise in identifying intricate patterns in time-series data, making them a popular choice for analyzing stock market trends [1, 33].

3) *Enhancing prediction accuracy*

Enhancing the prediction quality of models involves optimizing their learning algorithms and fine-tuning hyperparameters. Hybrid methods, combining different machine learning techniques, have emerged as a powerful approach in this domain. For instance, integrating KNN, ANN, and SVM can lead to a model that leverages the strengths of each individual method, resulting in improved accuracy and robustness in financial market predictions [17, 36].

4) *Application of deep learning*

Deep learning, a subset of machine learning, has gained traction in recent years for its effectiveness in handling vast datasets typical in finance. Deep learning architectures like Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks are particularly suited for time-series analysis, such as predicting stock prices or market trends. These models can capture temporal dependencies and patterns in historical data, which are crucial for accurate financial forecasting [37, 38].

5) Challenges and future directions

While machine learning offers promising solutions for financial prediction, challenges such as model overfitting, data quality, and the dynamic nature of financial markets persist. Future research in this area should focus on developing more adaptive, resilient models that can better handle market volatility and data anomalies. Additionally, there is a growing need for models that can interpret and explain their predictions, enhancing transparency and trust in machine learning-based financial decision-making [39, 40]. For a visual representation of the typical workflow involved in a supervised learning approach for stock market prediction, refer to Fig. 2.

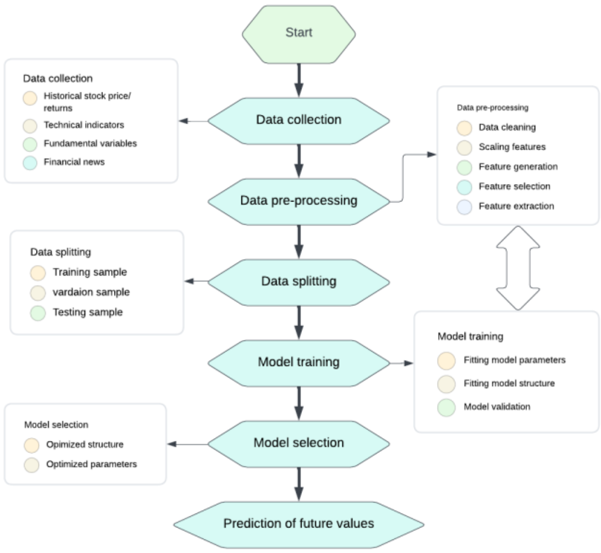

Fig. 2. Workflow of a stock market prediction model with supervised learning [4].

C. Benchmarking & Limitations

The scope and sophistication of machine learning algorithms for stock market prediction have significantly evolved, as evidenced by the various studies and approaches outlined in Table 1. However, it is essential to benchmark these advanced machine learning and deep learning algorithms against other strategies and traditional statistical methods. Data loss, overfitting, and model robustness are still significant concerns that require attention [33].

The comprehensive work summarized in Table 1 presents an array of approaches based on different stock indices across the world, employing multiple machine learning and deep learning models. From ANNs used in MSCI United Kingdom to more complex Deep Belief Networks (DBN) used in the Nikkei 225 index, the variety is evident. Each approach uses different scopes like price data, technical indicators, or a combination of both to improve prediction accuracy. For instance, the NASDAQ incorporates Price data along with technical indicators and utilizes Principal Component Analysis (PCA) with Deep Neural Networks (DNNs) for prediction [41]. Such diversity indicates the ongoing exploration for the optimal blend of variables and computational models. Additionally, the application of machine learning techniques varies significantly across different economic markets, suggesting that geographic and economic factors could also influence the model's performance. For example, simpler machine learning models like Binary mapping are used for the Tehran stock exchange [42], whereas more complex techniques like DNNs are applied for more developed markets like the Australian securities exchange [43].

Table 1. Comprehensive work on prediction algorithm based on feature extractions [6]

Reference	Scope	Prediction Algorithm/s
[10]	Korean stock index	Price data AE, Principal Component Analysis (PCA), Restricted Boltzmann Machine (RBM), ANN, DNN
[41]	NASDAQ	Price data, technical indicators PCA DNN
[42]	Tehran stock exchange	Technical indicators Binary mapping Machine learning and deep learning models
[43]	Australian securities exchange	Price Data Neural network IOWA
[44]	Nikkei 225 index	Price data RBM, Recurrent Neural Network-Deep Belief Network (RNN-DBN)
[45]	MSCI, UK	Price data ANN, LSTM, RF, SVR
[46]	Indian stock market	Price data, Technical indicators Scaled raw data LSTM

However, despite these advances, there are inherent limitations:

- Overfitting: Many machine learning models, especially deep learning models, are prone to overfitting, especially when trained on limited or noisy data.
- Data Loss: The preprocessing steps, including data normalization and dimensionality reduction, can sometimes result in the loss of critical information, affecting the model's predictive accuracy.
- Benchmarking: Most studies focus on the predictive power of individual algorithms, with less emphasis on comparing these newer machine learning models against traditional statistical models or even simpler machine learning models.
- Scalability: An additional concern is the need for a system to be scalable, reliable, and maintainable to effectively predict a wide range of markets and currencies [47]. Scalability remains a challenge, as models that perform well on specific indices or economic conditions may not necessarily generalize well to larger or more complex financial landscapes. It is here that our research intends to focus, aiming to address this crucial issue of scalability.

IV. Proposed Method

This research proposes a novel method for predicting real-time stock market trends by integrating machine learning algorithms with streaming data processing platforms and microservices. The architecture aims to harness the power of Apache Kafka for handling a high volume of real-time data streams emanating from various sources such as markets and currency sensors. We aim to enhance scalability while maintaining prediction accuracy. The architecture of the proposed system is schematically illustrated in Fig. 3.

A. *Overview of Phases*

The entire system is designed to operate in three distinct but interconnected phases. Each phase aims to address a set of challenges and requirements to ensure the system's overall effectiveness, scalability, and accuracy.

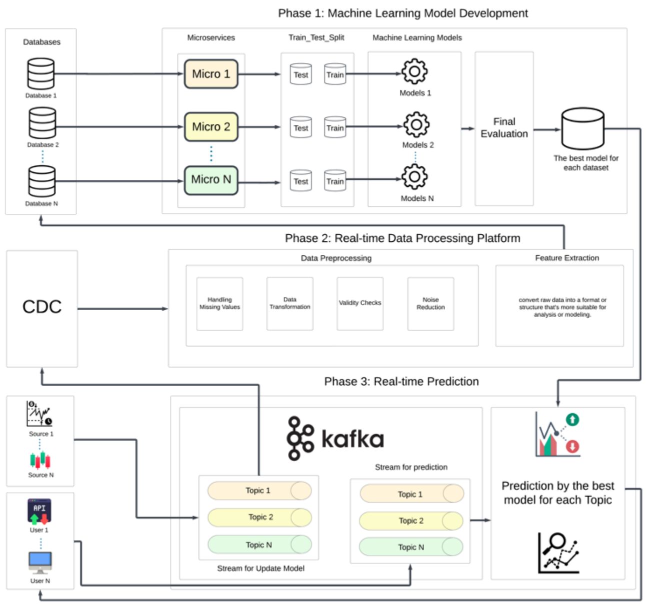

Fig. 3. Workflow of a stock market prediction model with supervised learning [4].

1) *Machine learning model development*

In the first phase of our proposed architecture, the emphasis is on developing robust machine learning models tailored to the specific requirements of real-time stock market prediction. This phase encompasses several crucial steps designed to ensure that the models are both accurate and scalable. The following are the sub-tasks involved in this phase:

1) Load the Data from Each Dataset: Data pertaining to various stock market indicators, market trends, and currency sensors will be loaded into the system. These data sources can be quite heterogeneous, demanding different preprocessing steps
2) Run Microservices Independently: Each dataset is hand-led by an independent microservice. The architecture is designed to run these microservices independently so that they can process the data, develop models, and make predictions without affecting each other's performance.
3) Split Data into Train and Test Sets: For each microservice, the loaded data will be divided into training and testing sets. The training set will be used to build and train machine learning models, while the testing set will serve to evaluate these models' performance.
4) Run Probable Models on Datasets: Various machine learning models are run on the training sets. These could range from linear regression models to more complex neural networks, depending on the type of data and the specific prediction needs.
5) Select the Best Model for Each Microservice: After running multiple models, the one yielding the highest prediction accuracy will be selected for each microservice. This model will then be deployed for real-time stock market prediction.
6) Update Best Model in Database as BLOB: Once the best Once the best model is selected, it will be serialized and stored in the database as a Binary Large Object (BLOB). This allows for quick retrieval and deployment, making it easier to update or replace the model as new data becomes available or when the model needs to be refined.
7) By meticulously executing these steps in Phase 1, we aim to build a strong foundation for the following phases. This ensures that the machine learning models are not only accurate but are also seamlessly integrated into the broader architecture designed for real-time data processing and stock market prediction.

Algorithm 1. Training and Model Selection

```
Input: data_topic, db_connection
Output: best_model
procedure TRAINMODELS(data_topic, db_connection)
data ⟵ KafkaConsumer.Consume(data_topic)
train_data, test_data ⟵ SplitData(data)
model_X1 ⟵ InitializeModelX1()
model_X2 ⟵ InitializeModelX2()
model_X3 ⟵ InitializeModelX3()
Train(model_X1, train_data)
Train(model_X2, train_data)
Train(model_X3, train_data)
error_X1 ⟵ Evaluate(model_X1, test_data)
error_X2 ⟵ Evaluate(model_X2, test_data)
error_X3 ⟵ Evaluate(model_X3, test_data)
best_error ⟵ min(error_X1, error_X2, error_X3)
if best_error == error_X1 then
    best_model ⟵ model_X1
else if best_error == error_X2 then
    best_model ⟵ model_X2
else
    best_model ⟵ model_X3
end if
db_connection.SaveModelAsBLOB(best_model)
end procedure
return best_model = 0
```

2) *Real-time data processing platform*

In the second phase of our research, the focus shifts to establishing a robust real-time data processing platform capable of handling streaming data. Apache Kafka serves as the backbone of this phase, enabling high-throughput, real-time data ingestion and processing. The main components of this phase include:

1) Fetching Data from Stream for Model Update: For each topic in Apache Kafka, data streams are fetched in real-time. These data streams can contain new trends, trading volumes, and other market indicators that may influence stock prices.

2) Change Data Capture (CDC): To ensure that the data and models are always up-to-date, two different methods are used to trigger CDC.
- By Count: The system checks the database after every 10 or 20 inserts. If a change is detected, corresponding actions are triggered.
- By Time: Alternatively, the system performs a check at regular intervals, such as every 1 hour or even once a day, to detect any changes in the data.

3) Data Preprocessing: Before the data can be used to make any meaningful predictions, it must first undergo a series of preprocessing steps. Some of the preprocessing actions include:
- Handling Missing Values
- Data Transformation
- Validity Checks
- Noise Reduction

4) Feature Extraction: After preprocessing, feature extraction techniques are applied to convert the raw data into a more suitable format or structure for analysis or modeling. This could involve dimensionality reduction techniques, generating composite features, or selecting only those features that contribute the most to prediction accuracy.
5) Sending Result into Appropriate Database: The processed data, along with any features extracted, are sent to the appropriate database. This is done to ensure that the data is readily accessible for further analysis or for entering the next phase of our architecture.

The completion of Phase 2 ensures that the architecture is capable of handling real-time data efficiently, preprocessing it, and extracting features that will be critical for the machine learning models to make accurate predictions. The use of Apache Kafka facilitates the seamless integration of various data streams, enhancing the system's robustness and scalability.

Algorithm 2. Real-time Data Processing

Input: topic (Kafka topic), db_connection (Database connection)
Output: Updated best_model (Best predictive model based on the latest data)
procedure DATA_PROCESSING(topic, db_connection) {Retrieve data from the specified Kafka topic}
data_stream ⟵ KafkaConsumer.Consume(topic)
{Perform necessary data preprocessing and feature extraction on the consumed data stream}
best_model <- db_connection.FetchBestModel() {Retrieve the current best model from the database}
Update(best_model, data_stream) {Update the best model with the new data}
end procedure
return best_model {Return the updated best predictive model} =0

3) Real-time prediction and feedback loop

The third and final phase of the proposed architecture focuses on leveraging the machine learning models developed in Phase 1 and the real-time data processing capabilities built in Phase 2 to make real-time stock market predictions. This phase operates in a closed-loop manner, continually updating its models and predictions based on incoming data and consumer feedback. The main components of this phase are as follows:

1) Fetching Real-time Data: In this step, real-time data from thousands of diverse sources are fetched and added to the stream for model updating. This data also undergoes the CDC process and the various preprocessing steps as established in Phases 2 and 1, respectively.
2) Handling Prediction Requests: Whenever there is a request for making a stock market prediction, this request is placed in another dedicated Apache Kafka stream called "Stream for Prediction".
3) Fetching the Best Model: The architecture fetches the best-performing machine learning model for the specific stock market in question from the database that was populated in Phase 1.
4) Real-time Prediction and Storage: Utilizing the fetched model and real-time data, a prediction is made. This prediction is then stored to be used in future analyses aimed at improving model precision.
5) Sending Results to Consumers: Finally, the prediction results are sent to consumers who can be individual users, trading platforms, or any other entities interested in stock market predictions.

This final phase ensures that the architecture is not just robust and scalable but also actionable. By connecting real-time data acquisition with machine learning prediction and immediate dissemination of these predictions, Phase 3 transforms the architecture into a complete end-to-end solution for real-time stock market prediction. The closed-loop nature of this phase allows for continual refinement of the models, thereby improving the accuracy and reliability of future predictions.

Algorithm 3. Real-time Data Prediction

Input: topic, db_connection
Output: Sent prediction
procedure DATA_PREDICTION(topic, db_connection)
request ⟵ KafkaConsumer.Consume(topic)
best_model ⟵ db_connection.FetchBestModel()
prediction ⟵ Predict(best_model, request)
db_connection.SavePrediction(prediction)
KafkaProducer.Send(prediction)
end procedure
return Sent prediction =0

B. Architecture Major Components

1) Dataset

The Yahoo! Finance API serves as the primary data source for this research, offering real-time and historical financial information on various markets and securities.

Features
- Historical Quotes: For trend analysis and model training.
- Real-Time Market Data: Essential for real-time predictions.

In this project, the API is used to obtain data in different time frames, aiming to optimize the machine learning models in Phase 1. This multi-temporal approach aims to capture various market dynamics effectively and feeds into subsequent phases for real-time prediction and analysis [48].

2) Apache Kafka

It is an open-source distributed streaming platform designed to efficiently handle large volumes of real-time data. Its architecture ensures fault tolerance, scalability, and efficient data processing [49].

Features:

- Distributed Streaming: Kafka facilitates data distribution across clusters, enhancing scalability and fault tolerance.
- Data Chunking: By breaking data into smaller pieces, Kafka optimizes the data transfer and processing speed.

In this project, Apache Kafka is pivotal for managing the incoming high-velocity financial data, sourced from the Yahoo! Finance API and other market sensors, in real-time. Kafka will serve as the initial ingestion layer where each stream or "topic" may represent data from a specific market or financial instrument.

The processed data will then be directed to the machine learning algorithms for both training (Phase 1) and real-time prediction (Phase 3). Kafka's role is essential for data preprocessing, ensuring that the real-time data are filtered, transformed, and made ready for subsequent phases of the research.

By integrating Kafka with Microservices and the machine learning models, the architecture aims to create a robust, scalable, and reliable platform for real-time stock market prediction and analysis.

3) Microservices architecture

It is a method of application development wherein each component or service runs independently and performs a specific task. These components communicate with each other through well-defined APIs and protocols [50].

Features:

- Modularity: Microservices break down the application into small, loosely-coupled services that can be developed, deployed, and scaled independently.
- Flexibility: Each microservice can be implemented using different technologies, which allows the use of the best tools for specific tasks.
- Scalability: Microservices can be easily scaled horizontally to handle increased load, making the architecture particularly useful for applications that require high availability.

In the proposed research, microservices architecture is used to encapsulate various machine learning models responsible for different tasks. For instance, each market or financial sensor can have its dedicated microservice to handle data ingestion, preprocessing, and machine learning model training (as described in Phase 1).

During the real-time data processing (Phases 2 and 3), each microservice fetches its relevant data stream from Apache Kafka for model updating or real-time prediction. The modular nature of microservices allows for high throughput and quick updates, crucial for the fast-paced financial market analysis.

The chosen best models for each microservice are stored in the database as BLOBs, enabling rapid and independent updates. By adopting a microservices architecture, the proposed system aims to achieve scalability, fault tolerance, and ease of maintenance, crucial for the robust and real-time analysis of stock markets.

V. Implementation And Results

A. Detailed Implementation Strategy

The practical deployment of our proposed architecture involves leveraging cloud services and modern data processing frameworks to ensure scalability, reliability, and real-time performance. Here, we detail the implementation strategy using AWS services, focusing on the setup, configuration, and operation of the system components.

1) Apache Kafka deployment

Apache Kafka clusters are deployed on AWS to manage high-velocity data streams efficiently. Kafka's distributed nature and fault tolerance make it ideal for handling real-time data feeds from various financial markets. The Kafka clusters are set up to ensure data is ingested, processed, and made available to downstream services without latency.

2) Microservices architecture

We employ AWS EC2 (Elastic Compute Cloud) instances to host independent microservices, each dedicated to specific tasks such as data ingestion, preprocessing, model training, and prediction. These microservices are designed to communicate seamlessly, ensuring a cohesive operation while retaining the flexibility to scale individual components as needed. For instance, one EC2 instance may host a microservice for real-time data ingestion from the Yahoo Finance API, fetching market data at predetermined intervals.

3) Database and model management

A dedicated EC2 instance is used to host the database, responsible for storing processed data, model parameters, and prediction results. This centralized approach ensures data consistency and provides a single source of truth for model performance metrics and historical predictions. The database plays a crucial role in model selection, storing the performance metrics of various models and enabling the system to select the best-performing model dynamically based on current market conditions.

4) User request handling and prediction

The system exposes an API endpoint, allowing users to request stock market predictions for specific entities and timeframes. Upon receiving a request, the system retrieves the most suitable machine learning model for the requested entity from the database. It then processes the latest market data, applies the model, and returns the prediction results in real time. This process ensures that users receive timely and accurate market insights, empowering them to make informed decisions.

5) Security, compliance, and maintenance

Throughout the architecture, security protocols are rigorously enforced to protect sensitive data and ensure compliance with industry standards and regulations. Regular maintenance routines, including system updates, model retraining, and performance monitoring, are established to keep the system optimized and aligned with the latest market dynamics.

By following this detailed implementation strategy, practitioners can realize the full potential of the proposed architecture, deploying a robust, scalable, and real-time stock market prediction system in a real-world environment.

B. Challenges in Real-world Implementation and Scalability

While our proposed architecture demonstrates promising results in an experimental setup, deploying such a system in a real-world environment comes with its unique set of challenges. This subsection outlines the potential obstacles and considerations that practitioners may encounter when implementing and scaling the architecture outside of a controlled experimental framework.

1) Integration with legacy systems

One of the primary challenges in deploying modern data-driven solutions is their integration with existing legacy systems. Financial institutions often rely on outdated, monolithic systems that are not designed to interact seamlessly with modern microservices-based architectures. Ensuring compatibility, data consistency, and minimal downtime during integration requires meticulous planning and a phased implementation approach.

2) Data privacy and security

Financial data is highly sensitive, and any system dealing with such data must adhere to stringent security standards and regulatory requirements. Implementing robust encryption protocols, access controls, and regular security audits are crucial to protect data integrity and confidentiality. Additionally, navigating the complex landscape of financial regulations and ensuring compliance add layers of complexity to the deployment process.

3) Handling high data volume and velocity

The sheer volume and velocity of data in the financial domain can overwhelm systems that are not designed for scalability. Architectures must be capable of scaling horizontally to accommodate spikes in data influx, especially during market volatility. This involves deploying load balancers, auto-scaling groups, and distributed databases, ensuring that the system remains responsive and accurate even under heavy load.

4) Model robustness and continual learning

Financial markets are dynamic, and models that perform well under certain conditions may become obsolete as market dynamics shift. Ensuring model robustness and implementing continual learning mechanisms are essential to maintain prediction accuracy. This requires setting up pipelines for continuous data ingestion, model evaluation, and retraining, coupled with mechanisms to detect and adapt to concept drift in real-time.

5) Operational complexity and maintenance

Operating a complex, distributed system with numerous moving parts introduces operational challenges. Regular maintenance, monitoring, system updates, and troubleshooting are vital to ensure smooth operation. Implementing comprehensive logging, monitoring solutions, and automated alerting systems are critical to promptly identify and address issues, minimizing downtime and service disruptions.

By addressing these challenges and considering these factors, practitioners can enhance the robustness, scalability, and real-world applicability of the proposed stock market prediction architecture, ensuring that it not only performs well in experimental setups but also delivers reliable results in practical, real-world scenarios.

C. Metric for Evaluation: R^2 Score

In this study, the efficacy of our machine learning models is assessed primarily using the R^2 score, also known as the coefficient of determination. This metric is chosen due to its effectiveness in representing the proportion of the variance in the dependent variable that is predictable from the independent variables. It offers a comprehensive measure of how well the model captures the variability in the dataset.

R^2 Score: The R^2 score is a statistical measure and is mathematically defined as follows:

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

where y_i represents the actual values, $\hat{y}_i$ represents the predicted values by the model, $\bar{y}$ is the mean of the actual values.

The R^2 score is a normalized measure ranging between 0 and 1. An R^2 score of 1 indicates perfect prediction accuracy, signifying that the model's predictions perfectly match the actual data. On the other hand, a score of 0 implies that the model performs no better than a simple mean-based prediction. This metric is particularly useful for comparing the performance of different models on the same dataset and for evaluating the model's ability to capture the variance in the data. It is a standard measure in regression analysis, providing a clear indication of the goodness of fit of the model.

D. Result

1) Machine learning

The results of the performance evaluation for various entities using Convolutional Neural Network (CNN), Gate Recurrent Unit (GRU), and Long Short-Term Memory (LSTM) models are presented in Table 2. This table illustrates the R^2 scores achieved by each model across a diverse range of entities, including different companies and financial pairs. The highest R^2 score for each entity is highlighted in green, indicating the model that best predicts the stock price movement for that particular entity. From the table, it is evident that the performance of each model varies significantly across different entities. For instance, the LSTM model shows superior performance for the Royal Bank of Canada and Microsoft, while the GRU model excels in predicting for Google and Apple. Similarly, the CNN model demonstrates its strength with high scores in predicting for entities like EUR/USD and Amazon.

In addition to the tabulated results, a plot visualizing these performance metrics is provided (see Fig. 4). The plot offers a graphical representation of the R^2 scores, making it easier to compare the performance of the three models across the various entities. Through this visualization, we can observe the trends and patterns in model performance, providing a clear and comprehensive view of the strengths and limitations of each model in different contexts.

Table 2. Performance comparison of CNN, GRU, and LSTM for various entities

Entity	CNN	GRU	CNN
Royal Bank of Canada	0.92	0.61	0.93
Google	0.80	0.96	0.47
EUR/USD	0.97	0.92	0.74
Apple	0.87	0.93	0.83
Microsoft	0.84	0.87	0.94
Amazon	0.90	0.68	0.84
Tesla	0.71	0.80	0.51
Meta	0.91	0.94	0.86
XRP	0.81	0.84	0.89

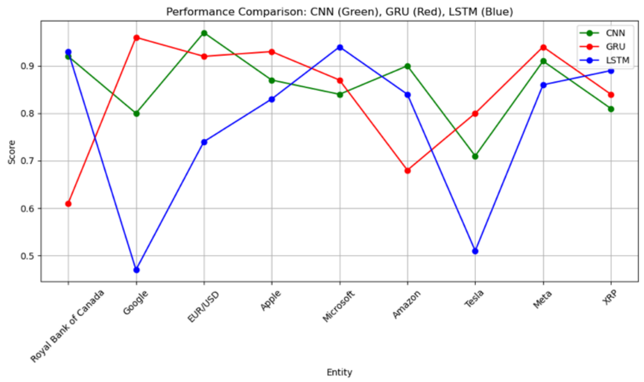

Fig. 4. Comparative Performance of CNN (green), GRU (red), and LSTM (blue) for various entities. The plot showcases score variations across models for key entities, emphasizing their distinct performance metrics.

Overall, these results highlight the importance of model selection in stock market prediction. The variability in performance across different entities underscores the need for a dynamic approach in choosing the appropriate model for each specific market condition, which is a key aspect of the research presented in this thesis.

2) Scalability

The comparative analysis, as summarized in Table 3, showcases distinct improvements in throughput, latency, training time, market analysis capacity, and scalability. For throughput, represented by θ in the baseline model, our proposed model achieves an increased rate, denoted as $\theta + \Delta\theta$. This signifies a higher data processing capability per second, crucial for real-time analytics. Latency, a critical factor in real-time systems, is reduced in the proposed model ($L - \Delta L$), thereby offering quicker response times compared to the baseline (L).

Table 3. Comparative analysis of model performance using symbolic representation

Performance Metric	Baseline Model	Proposed Adaptive Model
Throughput (GB/s)	θ	$\theta + \Delta\theta$
Latency (s)	L	$L - \Delta L$
Training Time (min)	T_{base}	$T_{base} - \Delta T$
Market Analysis Capacity	M markets in T_{base}	M + ΔM markets in $T_{base} - \Delta T$
Scalability	Limited	Enhanced

Furthermore, the training time for the models is indicated by T_{base} for the baseline and $T_{base} - \Delta T$ for the proposed model. This reduction in training time is essential for faster model updates and adaptability to market changes. The market analysis capacity, which reflects the number of markets analyzed within a certain timeframe, is also enhanced in the proposed model (M + ΔM markets in $T_{base} - \Delta T$), compared to the baseline model (M markets in T_{base}). Finally, the scalability of the system is qualitatively assessed, with the proposed model offering enhanced scalability over the baseline, which is crucial for handling large-scale financial data.

This comparative analysis underscores the efficacy of the proposed adaptive model in handling the dynamic and demanding environment of stock market prediction, achieving significant improvements over traditional models.

E. Discussion

The comprehensive analysis of our results reveals the nuanced complexities of employing machine learning for stock market prediction. This study's core insight is the demonstrable efficacy of a dynamic model selection approach, which is essential in navigating the inherently volatile and diverse nature of financial markets.

1) Dynamic model selection in varied market conditions

A pivotal finding of our research is the distinct performance variations of CNN, GRU, and LSTM models across different financial entities. This variation is not random but indicative of the unique characteristics inherent in each market or stock. For example, LSTM's superior performance with the Royal Bank of Canada, in contrast to GRU's effectiveness for Apple and Google, highlights the necessity of a context-sensitive approach in model selection. Similarly, CNN's proficiency in predicting the EUR/USD currency pair underscores its suitability for forex market dynamics.

The dynamic model selection system proposed in this research is thus not merely a technical solution but a strategic approach to stock market prediction. By continuously assessing and selecting the most appropriate model based on real-time market data, our system ensures enhanced accuracy and adaptiveness in predictions. This real-time adaptability is crucial, particularly in the stock market, where patterns and trends can shift rapidly, and the cost of inaccuracy is high.

2) Scalability and real-time performance

Another key aspect of our system is its demonstrated scalability and efficiency in high-throughput environments. As the volume of market data increases, our architecture maintains its performance integrity, handling multiple markets with low latency and high throughput. This capability is vital in the fast-paced realm of financial trading, where delays can lead to missed opportunities or significant financial losses.

The system's ability to scale and perform under increased workload is not just a technical merit but a critical feature for real-time financial applications. This scalability ensures that our system can adapt to growing data volumes and evolving market conditions without compromising on speed or accuracy.

VI. Conclusion

The work detailed in this thesis successfully demonstrates the integration of machine learning algorithms, Apache Kafka, and microservices architecture to establish a dynamic and efficient framework for real-time stock market prediction. This approach is characterized by its adaptability and effectiveness across different markets and timeframes, achieved through a strategic three-phase implementation.

The initial phase focuses on developing and evaluating machine learning models, where their performance is assessed rigorously using the R^2 score as the primary evaluation metric. This singular focus on the R^2 score, a measure of the variance in the dependent variable that is predictable from the independent variables, ensures a consistent and straightforward metric for model performance evaluation. Phases 2 and 3 integrate these models into a real-time data processing and prediction system, effectively handling high-velocity and diverse data streams.

This architecture stands out by addressing the limitations of previous models, which are typically constrained to specific markets or timeframes. The system's continuous evaluation and updates of its models ensure the selection of the most accurate model for each market condition at any given time. The high R^2 scores achieved in the evaluation phase validate the robustness and accuracy of our approach. The adaptability and efficiency of this architecture hold significant promise for extension to other domains beyond stock market prediction, addressing a notable gap in the existing literature on financial analytics.

In conclusion, this research marks a pivotal step in the field of real-time financial analytics. It opens new avenues for future research to extend and refine real-time data analysis methodologies. The scalable and adaptable framework presented in this thesis offers a novel paradigm in real-time analytics, with the potential to revolutionize not just the domain of stock market prediction but also various other sectors that require sophisticated real-time data analysis.

While the architecture presented in this paper shows promising results in the realm of stock market prediction, it's worth noting that its applicability could extend far beyond this specific domain. One avenue for future work is to explore how the architecture can be adapted and deployed in other real-time data-generating systems. The model switching feature that we proposed in this architecture could make it a versatile tool for solving complex, data-intensive problems in these domains. Furthermore, as the architecture is designed to be domain-agnostic, it has the potential to tackle challenges that are common across different fields requiring real-time analytics. Expanding the architecture to different domains would not only broaden its applicability but could also offer new insights into optimizing performance and reducing errors across various types of real-time data streams. Thus, future work will aim to test the generalizability and adaptability of the architecture in a broader range of applications, thereby contributing to the field of real-time data analytics as a whole.

Conflict of Interest

The authors declare that there is no conflict of interest.

Author Contributions

MohammadEhsan Akhavanpour conducted the research and experiments and was primarily responsible for writing the manuscript. Dr. Saeed Samet contributed by revising the manuscript critically and provided substantial updates in various sections as the supervising author. Both authors have reviewed and approved the final version of the manuscript.

Funding

This work is partial funded by the School of Computer Science at the University of Windsor and the Natural Sciences and Engineering Research Council of Canada (NSERC). Their support has been instrumental in the progress of our research.

References

[1] E. Hoseinzade and S. Haratizadeh, "CNNPred: CNN-based stock market prediction using a diverse set of variables," *Expert Systems with Applications*, vol. 129, pp. 273–285, 2019.

[2] C. Lohrmann and P. Luukka, "Classification of intraday S&P500 returns with a random forest," *International Journal of Forecasting*, vol. 35, no. 1, pp. 390–407, 2019.

[3] M. Sedighi, H. Jahangirnia, M. Gharakhani, and S. F. Fard, "A novel hybrid model for stock price forecasting based on metaheuristics and support vector machine," *Data*, vol. 4, no. 2, pp. 1–28, 2019.

[4] M. M. Kumbure, C. Lohrmann, P. Luukka, and J. Porras, "Machine learning techniques and data for stock market forecasting: A literature review," *Expert Systems with Applications*, vol. 197, 116659, 2022.

[5] J. J. Murphy, *Technical Analysis of the Financial Markets: A Comprehensive Guide to Trading Methods and Applications*, Prentice Hall Press, 1999, vol. 2.

[6] T. Beck and R. Levine, "Stock markets, banks, and growth: Panel evidence," *Journal of Banking & Finance*, vol. 28, no. 3, pp. 423–442, 2004.

[7] Y. Song, J. W. Lee, and J. Lee, "A study on novel filtering and relationship between input-features and target-vectors in a deep learning model for stock price prediction," *Applied Intelligence: The International Journal of Artificial Intelligence, Neural Networks, and Complex Problem-Solving Technologies*, vol. 49, no. 3, pp. 897–911, 2019.

[8] K. Chen, K. Franko, and R. Sang, "Structured model pruning of convolutional networks on tensor processing units," *CoRR*, abs/2107.04191, 2021.

[9] F. Chollet, *Deep Learning with Python*, Simon and Schuster, 2017.

[10] E. Chong, C. Han, and F. C. Park, "Deep learning networks for stock market analysis and prediction: Methodology, data representations, and case studies," *Expert Systems with Applications*, vol. 83, pp. 187–205, 2017. doi: 10.1016/j.eswa.2017.04.030

[11] R. Choudhry and K. Garg, "A hybrid machine learning system for stock market forecasting," in *Proc. the 2008 July International Conference on Information and Knowledge Engineering*, 2008, pp. 315–318.

[12] M. Chen, S. Mao, and Y. Liu, "Big data: A survey," *Mobile Networks and Applications*, vol. 19, no. 2, pp. 171–209, 2014.

[13] P. Zikopoulos and C. Eaton, *Understanding Big Data: Analytics for Enterprise Class Hadoop and Streaming Data*, McGraw-Hill Osborne Media, 2011.

[14] J. Manyika *et al.*, "Big data: The next frontier for innovation, competition, and productivity," *McKinsey Digital*, McKinsey Global Institute, 2011.

[15] Y. Wang *et al*., "A survey of big data," *Journal of Computer Research and Development*, vol. 49, no. 8, pp. 1462–1483, 2012.

[16] D. Laney, "3D data management: Controlling data volume, velocity, and variety," *META Group Research Note*, no. 6, 2001.

[17] B. Weng, M. A. Ahmed, and F. M. Megahed, "Stock market one-day ahead movement prediction using disparate data sources," *Expert Systems with Applications*, vol. 79, pp. 153–163, 2017. doi: 10.1016/j.eswa.2017.02.041

[18] G. S. Atsalakis and K. P. Valavanis, "Forecasting stock market short-term trends using a neuro-fuzzy based methodology," *Expert Systems with Applications*, vol. 36, no. 7, pp. 10696–10707, 2009. doi: 10.1016/j.eswa.2009.02.043

[19] Y. Chen and Y. Hao, "A feature weighted support vector machine and K-nearest neighbor algorithm for stock market indices prediction," *Expert Systems with Applications*, vol. 80, pp. 340–355, 2017.

[20] E. Fama, "Efficient capital markets: A review of theory and empirical work," *The Journal of Finance*, vol. 25, pp. 383–417, 1970.

[21] W. Chen, A. El Majzoub, I. Al-Qudah, and F. A. Rabhi, "A CEP-driven framework for real-time news impact prediction on financial markets," *Service Oriented Computing and Applications*, vol. 17, pp. 129–144, Mar. 2023.

[22] N. Sanjana, S. Raj, H. V. Prabhu, and S. Sandhya, "Real-time event streaming for financial enterprise system with Kafka," in *Proc. the 2023 3rd Asian Conference on Innovation in Technology (ASIANCON)*, Aug. 2023. doi: 10.1109/ASIANCON58793.2023.10270532
[23] Z. Mahdavi and M. Khademi, "Prediction of oil production with: Data mining, neuro-fuzzy and linear regression," *International Journal of Computer Theory and Engineering*, vol. 4, no. 3, pp. 446–447, 2012.
[24] Y. Radhika and M. Shashi, "Atmospheric temperature prediction using support vector machines," *International Journal of Computer Theory and Engineering*, vol. 1, no. 1, pp. 55–58, 2009.
[25] C. F. Tsai and Y. C. Hsiao, "Combining multiple feature selection methods for stock prediction: Union, intersection, and multi-intersection approaches," *Decision Support Systems*, vol. 50, no. 1, pp. 258–269, 2010. doi: 10.1016/j.dss.2010.08.028
[26] E. Hadavandi, H. Shavandi, and A. Ghanbari, "Integration of genetic fuzzy systems and artificial neural networks for stock price forecasting," *Knowledge-Based Systems*, vol. 23, no. 8, pp. 800–808, 2010. doi: 10.1016/j.knosys.2010.05.004
[27] M. R. Hassan, B. Nath, and M. Kirley, "A fusion model of HMM, ANN and GA for stock market forecasting," *Expert Systems with Applications*, vol. 33, no. 1, pp. 171–180, 2007.
[28] J. Patel, S. Shah, P. Thakkar, and K. Kotecha, "Predicting stock and stock price index movement using trend deterministic data preparation and machine learning techniques," *Expert Systems with Applications*, vol. 42, no. 1, 2015, pp. 259–268. doi: 10.1016/j.eswa.2014.07.040
[29] D. Enke, M. Grauer, and N. Mehdiyev, "Stock market prediction with multiple regression, fuzzy type-2 clustering and neural networks," *Procedia Computer Science*, vol. 6, pp. 201–206, 2011. doi: 10.1016/j.procs.2011.08.038
[30] C. F. Tsai, Y. C. Lin, D. C. Yen, and Y. M. Chen, "Predicting stock returns by classifier ensembles," *Applied Soft Computing*, vol. 11, no. 2, pp. 2452–2459, 2011.
[31] S. Barak, A. Arjmand, and S. Ortobelli, "Fusion of multiple diverse predictors in stock market," *Information Fusion*, vol. 36, pp. 90–102, 2017. doi: 10.1016/j.inffus.2016.11.006
[32] S. T. A. Niaki and S. Hoseinzade, "Forecasting S&P 500 index using artificial neural networks and design of experiments," *Journal of Industrial Engineering International*, vol. 9, no. 1, 2013, pp. 1–9. doi: 10.1186/2251-712X-9-1
[33] X. Zhong and D. Enke, "Forecasting daily stock market return using dimensionality reduction," *Expert Systems with Applications*, vol. 67, pp. 126–139, 2017. doi: 10.1016/j.eswa.2016.09.027
[34] M. Kubat, *An Introduction to Machine Learning*, 2nd ed. Springer Publishing Company, 2017.
[35] F. E. Tay and L. J. Cao, "Application of support vector machines in financial time series forecasting," *Omega*, vol. 29, 2001, pp. 309–317.
[36] J. Cao and J. Wang, "Stock price forecasting model based on modified convolution neural network and financial time series analysis," *International Journal of Communication Systems*, vol. 32, pp. 1–13, 2019.
[37] T. Fischer and C. Krauss, "Deep learning with long short-term memory networks for financial market predictions," *European Journal of Operational Research*, vol. 270, no. 2, pp. 654–669, 2018. doi: 10.1016/j.ejor.2017.11.054
[38] M. Dixon, D. Klabjan, and J. H. Bang, "Sequence classification of the limit order book using recurrent neural networks," *Journal of Computational Finance*, vol. 24, no. 1, pp. 113–150, 2020. doi: 10.48550/arXiv.1707.05642
[39] A. B. Arrieta, N. Díaz-Rodríguez, J. del Ser, A. Bennetot, S. Tabik, A. Barbado *et al.*, "Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI," *Information Fusion*, vol. 58, pp. 82–115, 2020. doi: 10.1016/j.inffus.2019.12.012
[40] S. Makridakis, E. Spiliotis, and V. Assimakopoulos, "Statistical and machine learning forecasting methods: Concerns and ways forward," *PloS One*, vol. 13, no. 3, e0194889, 2018. doi: 10.1371/journal.pone.0194889
[41] R. Singh and S. Srivastava, "Stock prediction using deep learning," *Multimedia Tools And Applications*, vol. 76, no. 18, pp. 18569–18584, 2017.
[42] M. Nabipour, P. Nayyeri, H. Jabani, S. Shahab, and A. Mosavi, "Predicting stock market trends using machine learning and deep learning algorithms via continuous and binary data: A comparative analysis," *IEEE Access*, vol. 8, pp. 150199–150212, 2020.
[43] W. Hussain, J. Merigó, and M. Raza, "Predictive intelligence using ANFIS-induced OWAWA for complex stock market prediction," *International Journal of Intelligent Systems*, 2021. doi: 10.1002/int.22732
[44] A. Yoshihara, K. Fujikawa, K. Seki, and K. Uehara, "Predicting stock market trends by recurrent deep neural networks," *Lecture Notes in Computer Science*, vol. 8862, pp. 759–769, 2014.
[45] M. Nikou, G. Mansourfar, and J. Bagherzadeh, "Stock price prediction using DEEP learning algorithm and its comparison with machine learning algorithms," *Intelligent Systems in Accounting, Finance and Management*, vol. 26, no. 4, pp. 164–174, 2019.
[46] K. Khare, O. Darekar, P. Gupta, and V. Z. Attar, "Short term stock price prediction using deep learning" in *Proc. RTEICT 2017, the 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology*, 2017, pp. 482–486.
[47] M. Kleppmann, *Designing Data-Intensive Applications: The Big Ideas Behind Reliable, Scalable, and Maintainable Systems*, O'Reilly Media, 2016.
[48] Apidojo. (2021). Yahoo Finance by Apidojo. [Online]. Available: https://pipedream.com/apps/yahoo-finance-by-apidojo
[49] T. Palino, R. Sivaram, G. Shapira, and K. Petty, *Kafka: The Definitive Guide: Real-Time Data and Stream Processing at Scale*, O'Reilly Media, 2017.
[50] I. Nadareishvili, R. Mitra, M. McLarty, and M. Amundsen, *Microservice Architecture*, O'Reilly Media, Inc., 2016.