MDS581 - PROJECT II (CAPSTONE PROJECT)

SYSTEM REQUIREMENTS SPECIFICATIONS

InvestIQ

AI-DRIVEN STOCK FORECASTING USING VISUAL PATTERNS

HIMANSHU SALVEKAR 2348052

SANJAY R 2348055

Department of Statistics and Data Science

❖ INTRODUCTION:

In today's fast-paced world, financial markets, especially stock markets, play a critical role in the global economy. Stock market predictions have long been a topic of interest for both individual investors and financial institutions. Accurately predicting stock prices can significantly improve investment decisions, leading to increased profits and minimized risks. However, stock market prediction remains a complex challenge due to the volatility and unpredictability of stock prices.

The project aims to leverage modern machine learning techniques, specifically Convolutional Neural Networks (CNNs), to predict stock prices based on historical time series data. The stock market data is typically represented in numerical formats, such as daily stock prices over time. In the project, we will transform this numerical time series data into 2D histogram bar charts, which will then be fed into a CNN model for classification tasks. The CNN model will classify the stock trends into three categories: Buy, Hold, or Sell, based on the past price movement patterns.

By utilizing advanced data science methodologies, the project will explore how deep learning models, particularly CNNs, can be applied to time series data to generate actionable stock market predictions. The system will use web scraping techniques to gather historical stock data, which will be preprocessed into 2D histograms representing the price trends, and then fed into the CNN for training and evaluation. The final model will aim to classify stocks based on their historical trends, providing recommendations to investors.

1. OBJECTIVES:

The objective of the Stock Market Prediction project is to develop a novel approach for forecasting stock prices using deep learning techniques, specifically Convolutional Neural Networks (CNNs), applied to visual representations of stock data, such as bar chart images. Unlike traditional models that rely on sequential numerical data, this project explores whether image-based methods can capture trends and patterns in stock behavior, providing an innovative alternative to classic time-series forecasting. By focusing on CNNs, the project seeks to analyze visual cues within stock chart images to make Buy-Sell predictions that could offer an advantage over the commonly used Buy-and-Hold (BaH) strategy, which may not account for short-term fluctuations that impact investment returns.The project aims to create a model that is accurate and responsive to market changes, ensuring its utility in real-time trading environments. It targets robustness across diverse market conditions, training the model on various historical stock data images to generalize well even during shifts in market dynamics.

In summary, the project aims to develop a stock market prediction system that uses time series data and deep learning algorithms to predict stock prices. The scope includes:

- ✓ **Data Collection**: Web scraping historical stock price data from reliable financial websites.
- ✓ **Data Transformation**: Converting raw numerical data into 2D histograms representing the stock price trends over time.
- ✓ **Model Training**: Developing and training a CNN model using the transformed images of stock price data.
- ✓ **Stock Trend Classification**: Predicting whether the stock should be **Bought**, **Held**, or **Sold** based on the model's output.

2. PROBLEM DESCRIPTION:

The problem under consideration in this Stock Market Prediction project is the limitation of traditional forecasting methods, which rely heavily on time-series numerical data but often fail to capture the complex visual patterns that experienced traders interpret. These visual cues, such as bar chart formations and price movement trends, play a crucial role in trading decisions, and traditional models are not equipped to leverage them effectively. The problem at hand is how to create a model that bridges this gap by integrating visual data, making market predictions both data-driven and visually intuitive.

To address this, the project introduces an image-based deep learning model using Convolutional Neural Networks (CNNs) that directly analyzes stock bar chart images. By interpreting these visual patterns, the model aims to capture subtle trends that numerical data alone may miss, providing a dynamic and adaptable prediction system suited for real-time market fluctuations. This solution directly addresses the problem of enhancing decision-making tools for both retail and institutional traders who rely on accurate, timely market predictions. The intended beneficiaries—retail investors and institutional traders—stand to gain from an innovative approach that combines intuitive visual cues with machine learning.

3. PROJECT STUDY:

In the domain of stock market forecasting, traditional systems largely rely on time-series analysis models, such as ARIMA and Recurrent Neural Networks (RNNs), which focus on historical numeric data to predict price trends. While effective in certain cases, these models often struggle with high volatility and sudden market fluctuations, as they may not capture intricate visual patterns found in stock bar charts. Such limitations restrict their adaptability, especially in fast-paced, high-frequency trading. The proposed system seeks to overcome these challenges by using Convolutional Neural Networks (CNNs) trained on bar chart images. CNNs excel in visual pattern recognition, allowing the model to identify subtle shifts in candlestick formations, volume trends, and price movement indicators, which are traditionally used by human traders. This approach enhances prediction accuracy by incorporating visual insights, providing more adaptive responses to abrupt market changes. Additionally, the system broadens its applicability across diverse market conditions and asset types, offering a more resilient and versatile forecasting model. By combining the strengths of visual analysis with machine-driven pattern recognition, the proposed system represents a significant advancement over conventional forecasting techniques in the stock market.

4. LITERATURE SURVEY:

In recent years, the use of machine learning, particularly deep learning, has gained significant traction in the field of financial market prediction. Stock market prediction is an inherently difficult problem, given the large amount of data, the volatility of the market, and the need for real-time decision-making. Traditional methods such as statistical time series models (e.g., ARIMA, GARCH) and machine learning models (e.g., Support Vector Machines, Random Forests) have been employed to forecast stock prices, but these approaches have certain limitations. The advent of deep learning, specifically Convolutional Neural Networks (CNNs), has opened up new avenues for handling more complex, high-dimensional data. By applying CNNs to stock bar charts and candlestick patterns, this project explores the use of visual data to capture trends and market sentiment in ways that traditional models cannot.

The application of machine learning, particularly deep learning, to stock market prediction has become a rapidly growing area of research. Traditional forecasting methods, such as time series models (ARIMA, GARCH) and machine learning algorithms like Random Forest and Support Vector Machines (SVM), are commonly used but have limitations in capturing the complex patterns inherent in stock market data. One key shortcoming of traditional approaches is their reliance on historical price data, which often misses important patterns and signals that can be derived from visual representations, such as stock bar charts and candlestick patterns. With the advent of deep learning, specifically Convolutional Neural Networks (CNNs), researchers have begun to explore the potential of analyzing visual data to predict stock prices, as CNNs are particularly adept at identifying patterns and structures within images.

Year of Implementation	**Author/Company**	**Techniques/Algorithm**	**Gap or Drawback**
2019	Zhang et al.	LSTM on time-series data	Limited pattern recognition, lacks visual data insights
2020	Chen & Yang	CNN on stock chart images	Limited adaptability to diverse market data
2020	Wang et al.	Hybrid CNN-RNN model	High complexity, computationally expensive
2021	Kumar & Singh	CNN for trend analysis	Doesn't integrate market sentiment data
2021	Li et al.	CNN for financial forecasting	Model struggles with low-volume stocks
2022	Yao et al.	GAN for stock prediction	Limited success in real-time predictions
2022	Patel et al.	Attention-based CNN	Ineffective for large datasets with noisy data

2022	FinAI Corp	CNN + RNN model	High computational cost and training time
2022	Li et al.	CNN for candlestick pattern analysis	Doesn't consider broader macroeconomic factors
2023	Zhang et al.	CNN for financial image recognition	Lacks a unified framework for integrating text and image data
2023	Smith et al.	Hybrid CNN + LSTM	Computational inefficiency in real-time trading
2023	Gao et al.	Reinforcement Learning for stock prediction	Requires large amounts of training data and fine-tuning
2023	FinAI Corp	CNN + Transformer Networks	Limited real-time execution capability
2024	Lee et al.	CNN for multi-market prediction	Poor generalization to unfamiliar market conditions
2024	Gupta & Sharma	Transfer Learning for stock prediction	Limited success in handling diverse stock data

5. REQUIREMENTS SPECIFICATION:

5.1 FUNCTIONAL REQUIREMENTS:

The **functional requirements** for the Stock Market Prediction project center on several key tasks that must be accomplished for the system to function effectively. First, the system will scrape historical stock price data from popular financial websites such as Yahoo Finance, MarketWatch, and others. This step ensures that the system has access to accurate and up-to-date stock market information, which is essential for making informed predictions. After scraping the raw stock data, the next step is **data preprocessing**. This involves transforming the data into 2D histograms that represent stock price movements over time. These histograms act as the input features for the Convolutional Neural Network (CNN), which will be trained to detect patterns in the visual representation of the data. The **model development** phase involves the creation of a CNN designed to classify stock movements into one of three categories: Buy, Hold, or Sell. The system will process the histograms using the trained CNN and make stock market predictions accordingly. To evaluate the effectiveness of the CNN model, **performance metrics** will be used, such as accuracy, precision, recall, and confusion matrix. These metrics will help assess how well the model is performing in terms of making correct predictions and how effectively it distinguishes between the three classes (Buy, Hold, or Sell).

5.2 NON-FUNCTIONAL REQUIREMENTS:

Non-functional requirements are crucial to ensure the system's efficiency and security. Scalability is essential, as the system must handle an increasing volume of stock data over time, accommodating a growing number of stock charts and data points without sacrificing performance. Performance is also vital, especially for real-time trading, where the system must provide timely predictions to take advantage of market fluctuations. The ability to generate quick predictions ensures that users can make fast, informed decisions. Accuracy is fundamental to the success of the model, as it directly impacts the reliability of stock trend predictions, helping users classify stock movements accurately into Buy, Hold, or Sell. Achieving high accuracy will ensure that the predictions are actionable and beneficial for decision-making. Security is another key requirement, ensuring the protection of sensitive user data during web scraping and data storage processes. Implementing encryption and secure transmission methods will prevent unauthorized access, safeguarding both user and stock data. By focusing on these non-functional aspects, the system will maintain its reliability, performance, and security while efficiently handling growing datasets and meeting the real-time needs of stock market traders. This holistic approach will make the system robust and practical for long-term use.6

6. SYSTEM REQUIREMENTS:

To successfully implement the Stock Market Prediction project, the system will require the following hardware and software resources:

6.1 HARDWARE REQUIREMENTS:

i. **Processor**: A modern multi-core processor (Intel i5 or equivalent) is recommended to handle the computation-heavy tasks involved in data processing and model training.
ii. **RAM**: A minimum of 8GB RAM is necessary for efficiently handling large datasets and performing real-time data analysis.
iii. **Storage**: At least 100GB of free disk space is required to store the stock market data, processed images, and model files.

iv. **Graphics Processing Unit (GPU)**: A dedicated GPU (e.g., NVIDIA GTX 1060 or higher) is highly recommended for accelerating deep learning model training, especially when using Convolutional Neural Networks (CNNs).

6.2 SOFTWARE REQUIREMENTS:

i. **Operating System**: The project can be run on Windows, macOS, or Linux-based operating systems.
ii. **Python**: The primary programming language used for the project, Python (version 3.6 or higher), will be required for data scraping, processing, and training the model.
iii. **Libraries**: Essential libraries include:
iv. **TensorFlow/Keras**: For building and training the CNN model.
v. **NumPy, Pandas**: For data manipulation and preprocessing.
vi. **Matplotlib, Seaborn**: For visualizing data and stock trends.
vii. **BeautifulSoup/Selenium**: For web scraping.
viii. **OpenCV**: For image processing and transforming time series data into histograms.
ix. **IDE/Development Tools**: A suitable integrated development environment (IDE) like PyCharm or Jupyter Notebook is recommended for writing and testing the code.

These system requirements will ensure that the project runs efficiently, with enough resources to handle data scraping, image processing, and deep learning model training.

7. SYSTEM MODELS

The system being developed for stock market prediction using deep learning models, specifically Convolutional Neural Networks (CNNs), is designed to predict stock prices based on visual representations of stock bar charts. The model will utilize stock images as input, process them through a CNN architecture, and generate predictions regarding potential price movements, such as whether to buy or sell a stock. This model will aim to overcome the limitations of traditional forecasting methods that rely solely on numerical data by integrating the rich visual data from stock charts. The system is structured into several distinct modules, each contributing to the overall process of data preprocessing, model training, and prediction generation.

The system consists of the following key modules:

7.1 Data Collection and Preprocessing: The first module involves the collection of historical stock data, which includes price movements, trading volumes, and other financial indicators. Stock images, in the form of bar charts, are then generated from this data using visualization libraries. These images will be the primary input to the deep learning model. The data preprocessing steps also include normalization of stock data and image resizing to fit the CNN input requirements.

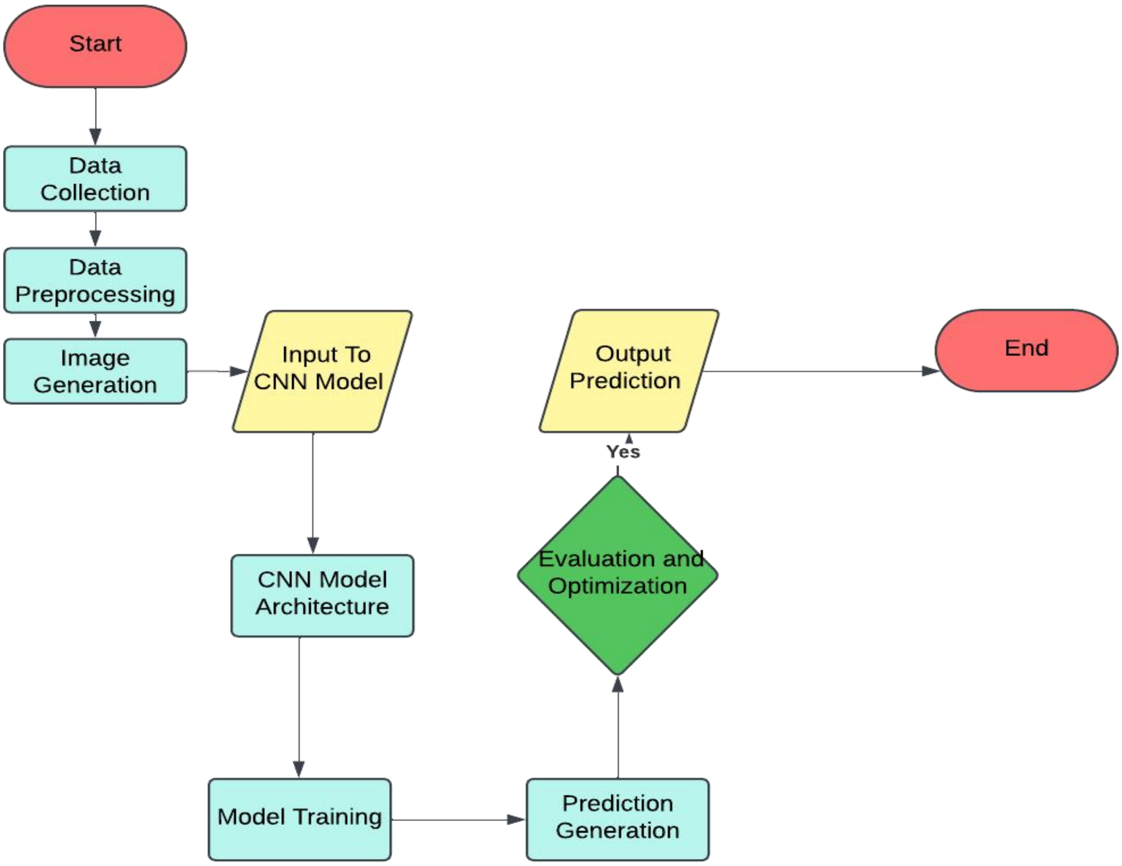
7.2 Model Architecture: The core of the system is the deep learning model, specifically a Convolutional Neural Network (CNN). This model is trained to recognize patterns and features in stock bar charts that correlate with price movements. The CNN architecture will consist of several convolutional layers to extract features from the images, followed by pooling layers to reduce dimensionality, and fully connected layers for classification.

7.3 Training the Model: In this module, the CNN is trained on a labeled dataset of stock bar chart images, with each image corresponding to a specific outcome (e.g., buy, hold, sell). The training process involves feeding the stock images into the CNN, adjusting weights using backpropagation, and minimizing a loss function that measures the difference between predicted and actual price movements.

7.4 Prediction Generation: Once the model is trained, it can make predictions based on new stock chart images. When provided with an image of a stock's price chart, the CNN will output a prediction about the stock's future performance, such as whether it is likely to increase or decrease in value. This output can be used as a decision-making tool for traders.

7.5 Evaluation and Optimization: This module involves evaluating the performance of the trained model using various metrics such as accuracy, precision, recall, and F1-score. It also includes optimization techniques to improve the model's performance, such as fine-tuning the hyperparameters, adding regularization, and using techniques like dropout to prevent overfitting.

BLOCK DIAGRAM:



- **DATA COLLECTION:**

Historical stock price data is gathered from financial markets, focusing on 30-day trading periods. This includes daily stock prices, opening and closing values, highs, lows, and trading volumes. The data collection process also considers various market indices and related stocks to provide broader market context. We utilize reliable financial APIs and databases to ensure data accuracy and consistency. Market sentiment indicators, news headlines, and technical indicators are also collected as supplementary data. This comprehensive data collection approach ensures our model has access to all relevant market information. Real-time data streaming capabilities are implemented to enable live market analysis. The system maintains a historical database for continuous model training and validation purposes.

- **DATA PREPROCESSING:**

Raw financial data undergoes extensive cleaning and normalization processes. Missing values are handled using appropriate interpolation techniques based on market behavior. Outliers are identified and treated using statistical methods to prevent model bias. Price data is normalized to account for different stock price ranges and market caps. Technical indicators like Moving Averages, RSI, and MACD are calculated to enrich the dataset. Time series data is aligned and synchronized to ensure consistent 30-day windows. Data quality checks are implemented to verify data integrity and consistency. The preprocessing pipeline includes steps for handling stock splits, dividends, and other corporate actions. Market hours and trading day calendars are considered to maintain data accuracy.

- **IMAGE GENERATION:**

Preprocessed financial data is transformed into standardized bar chart images. Each image captures a complete 30-day trading period with consistent scaling and dimensions. Various technical indicators are overlaid on the charts to enhance pattern visibility. The image generation process maintains aspect ratios and visual clarity for optimal CNN processing. Color coding is applied to highlight significant price movements and volume patterns. Multiple chart types (candlesticks, OHLC bars) are generated to capture different aspects of price movement. Image augmentation techniques are applied to increase the training dataset size. Quality control measures ensure image consistency and clarity. The system generates both training and validation image sets maintaining temporal separation.

- **CNN MODEL ARCHITECTURE:**

Our CNN architecture is specifically designed for stock market pattern recognition in images. The network begins with multiple convolutional layers using different filter sizes to capture various price patterns. Pooling layers are strategically placed to reduce dimensionality while preserving important features. The architecture includes skip connections to maintain gradient flow during deep learning. Dropout layers are implemented to prevent overfitting and improve generalization. Batch normalization is applied to stabilize the learning process. The final layers include dense connections leading to the prediction outputs. The architecture is optimized for both feature extraction and pattern recognition. Advanced activation functions are used to capture non-linear market behaviors. The model includes attention mechanisms to focus on significant price patterns.

- **MODEL TRAINING:**

The training process uses a carefully curated dataset of labeled bar chart images. The model learns to identify patterns associated with market movements and trading opportunities. Advanced optimization techniques like Adam optimizer with learning rate scheduling are implemented. The training includes regular validation checks to prevent overfitting. Data augmentation techniques are applied to improve model generalization. Custom loss functions are designed to balance prediction accuracy with trading profitability. The training process includes periodic model checkpointing and performance logging. Transfer learning techniques are applied to leverage pre-trained network features. The system implements early stopping based on validation performance. Regular retraining schedules are maintained to adapt to changing market conditions.

- **PREDICTION GENERATION & EVALUATION:**

The trained model processes new market data to generate actionable trading signals. Each prediction includes confidence scores and risk assessments. The evaluation system tracks multiple performance metrics including accuracy, ROI, and Sharpe ratio. A comprehensive backtesting framework validates strategy performance across different market conditions. Real-time prediction monitoring enables quick detection of model drift or performance degradation. Trading simulations are run to assess strategy viability in live markets. Risk management rules are applied to filter and validate predictions. Performance metrics are compared against benchmark strategies like Buy-and-Hold. The system maintains detailed logs of predictions and actual outcomes for analysis. Regular performance reviews drive continuous model improvements and optimizations.

Dataset Description with Feasibility Study

The dataset used in the project will include historical stock price data scraped from websites like Yahoo Finance. The data will be converted into 2D histograms that visually represent the price trends over time. The dataset will be split into training and testing sets for model validation.

The data will contain stock attributes such as:

- Date
- Opening price
- Closing price

- Highest price
- Lowest price

8. CONCLUSION:

The Stock Market Prediction using Time Series Data and CNN project has the potential to significantly alter the way stock market trends are predicted. In the traditional approach, stock predictions often rely on basic statistical methods or simple machine learning models that may fail to capture complex patterns in the data. However, the integration of Convolutional Neural Networks (CNNs) with time series data analysis introduces an innovative approach by leveraging deep learning techniques to recognize intricate patterns and trends within stock price data. This project aims to provide accurate and actionable insights by transforming raw stock price data into visually interpretable formats, such as graphs and charts, allowing for better decision-making for investors.

By employing CNNs, the model can take advantage of its image recognition capabilities, converting time series data into visual formats (such as time-lagged data matrices) which are easier for the neural network to process. This method allows the model to learn more complex patterns in the stock prices over time, which traditional methods might miss. As a result, the model can offer more accurate predictions of future stock prices, helping investors make informed choices.

Moreover, this project has real-world applications not just for individual investors but also for institutional investors, financial analysts, and hedge funds that rely on predictive models to drive their investment strategies. With its ability to process large amounts of data, the system can work efficiently and provide real-time predictions. This will enable stakeholders to stay ahead in the competitive stock market environment

In essence, the project bridges the gap between traditional finance and modern machine learning techniques, presenting an innovative solution to a problem that has long been challenging for the financial industry. As the model improves over time through continued training and data processing, its relevance and accuracy will only increase, potentially revolutionizing stock market forecasting. The project also holds promise for being adapted to other markets and financial sectors, further expanding its impact and usefulness.

9. REFERENCES:

[1] H. Alom, M. Taha, and A. Asyhari, "Deep Learning: A Comprehensive Review on Techniques, Applications, and Tools," IEEE Access, vol. 7, pp. 143760-143786, 2019. doi: 10.1109/ACCESS.2019.2943119.

[2] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," Nature, vol. 521, pp. 436-444, May 2015. doi: 10.1038/nature14539.

[3] A. Karpathy, "Convolutional Neural Networks for Visual Recognition," Stanford University Lecture Notes, 2016. Available: https://cs231n.stanford.edu/

[4] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed Representations of Words and Phrases and their Compositionality," Proceedings of NeurIPS, pp. 3111-3119, 2013. Available: https://papers.nips.cc/paper/2013

[5] C. Szegedy, V. Vanhoucke, and Z. Chen, "Rethinking the Inception Architecture for Computer Vision," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016. doi: 10.1109/CVPR.2016.308.

[6] F. Chollet, "Keras: The Python Deep Learning Library," GitHub repository, 2015. Available: https://github.com/keras-team/keras

[7] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," Proceedings of NeurIPS, pp. 448-456, 2015. Available: https://arxiv.org/abs/1502.03167