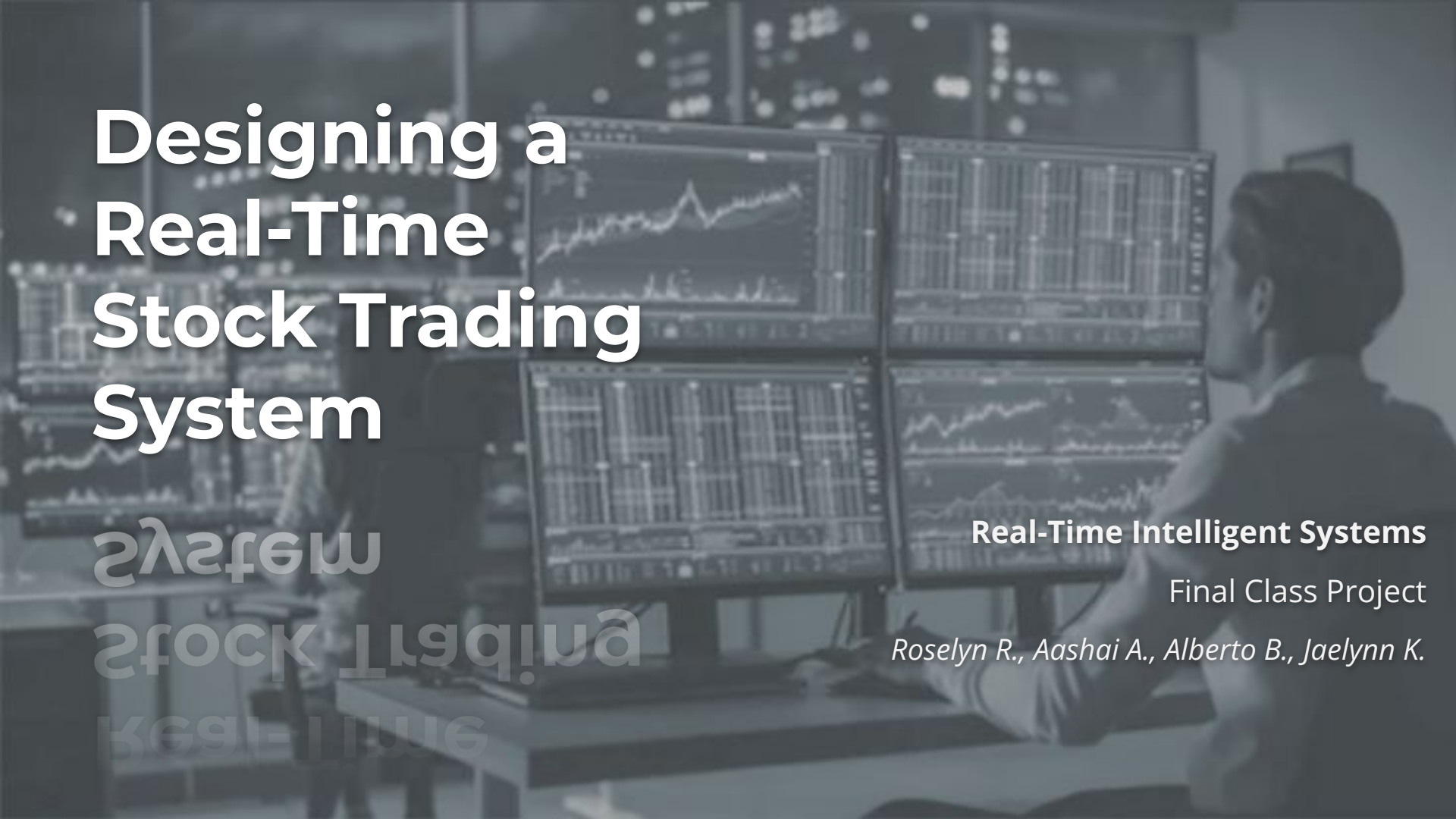


Designing a Real-Time Stock Trading System

Real-Time Intelligent Systems

Final Class Project

Roselyn R., Aashai A., Alberto B., Jaelynn K.

Description of the Problem:
Project Introduction and Goals

Our goal for developing a real-time financial stock trading system is to equip investors with the tools to make well-informed decisions.

By leveraging advanced modeling techniques for precise time-series forecasting and for understanding complex, non-linear patterns in stock price movements, our system can serve as a critical advisor informing on decisions within the dynamically changing tech market.

Description of the Problem:
Problem Definition

- ⚠ High volatility and uncertainty in stock markets
- ⚠ Speed of processing large volumes of real-time data
- ⚠ Lack of tools for integration of multiple trading aspects

[Three Pronged Approach]

Real-Time Forecasting: *Time Series*

Event-Driven Methodology: *Sentiment Analysis*

Risk Management

Solution to this Problem (Methodology): System Architecture and Technologies

Three-Pronged Approach to Informed Decision-Making

Real-Time Forecasting

We experiment with two time-series models, ARIMA and RNN, to predict future stock prices based on historical data. This method helps in identifying potential price movements before they occur, allowing for proactive trading decisions.

Event-Driven Methodology

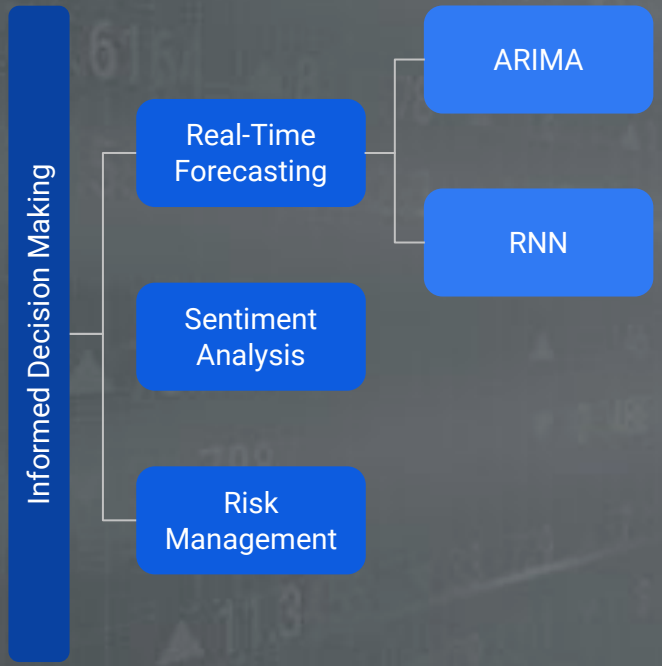
Sentiment analysis is employed on real-time news data to gauge market sentiment towards specific stocks. This approach helps in understanding the impact of current events on stock values, enabling traders to react to news with potential market-moving implications promptly.

Risk Management

Focuses on measuring and managing the risks associated with each stock. Techniques used include calculating the volatility and potential value at risk (VaR) of stocks, which aids in determining the safest and most potentially profitable investment avenues.

Upon evaluation of each of the models, a model was chosen and a form of backtesting was also used for the chosen model to evaluate its performance.

Finally, the model was used within a client/server application to simulate a stock trading system.



Model Chosen

RNN ARIMA

In Strategy 1, it was observed that unlike the Autoregressive Integrated Moving Average (ARIMA) model, which is primarily statistical and struggles with non-linear trends, the Recurrent Neural Network (RNN) model successfully adapted to the dynamic changes in stock prices by learning from the temporal dependencies within the data.

In this way, RNN was chosen as the primary model due to its enhanced predictive capabilities.

Model Chosen

Strategy 2:
Sentiment Analysis

Strategy 3:
Risk Management

Our RNN model leverages deep learning architecture, which is designed for sequential data, unlike the traditional statistical ARIMA model.

While Strategy 2's Sentiment Analysis, based on a limited set of news articles, offers additional market insights, and serves as a supplementary predictive tool further empowering the RNN model.

Strategy 3's risk management models were not selected as the main model due to its tendency to be potentially overly conservative for active stock trading, but was selected as a supplementary toolset that provides context to the integrative model.

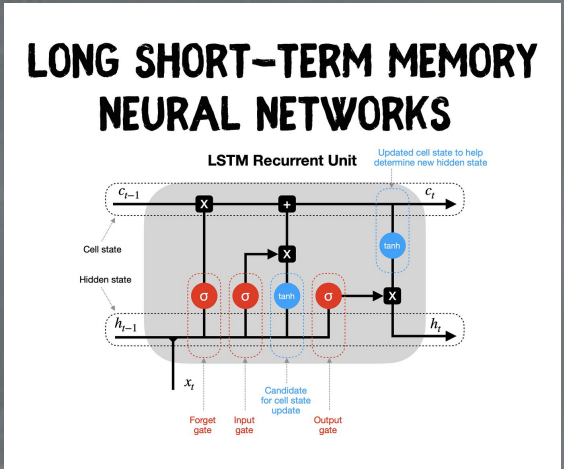
Specifically, these risk management models identified Apple (AAPL) as the stock with the lowest associated risk.

Model Chosen

RNN

Mean
Absolute
Error

30 Days
60 Days
90 Days



RNN can lose the contextual information over a longer amount of time intervals.

This is one of the things that we want to measure!

Implementation

For large scale applications there has to be a client-server architecture in order to stream and inference trades in real time.

- Data was streamed in from a CSV file similar to how real time data would.
- A server was created, which the client connected to.
- Using the model, a trading system is simulated within a client application.

DEMO

THANK YOU FOR A WONDERFUL QUARTER