BABEŞ-BOLYAI UNIVERSITY CLUJ-NAPOCA
FACULTY OF MATHEMATICS AND COMPUTER SCIENCE
SPECIALIZATION Computer Science, English

DIPLOMA THESIS

Sentiment and Credibility-Adjusted Stock Price Prediction

Supervisor
Assist. Bogdan-Eduard-Mădălin Mursa

Author
Sergiu-Rareș Goian

2025

ABSTRACT

In the last years, there was a significant change in people's opinions about stock markets. Some decades ago, the majority of people were skeptical about investing their money and stock markets were usually seen as being accessible only for the richest. However, the number of investors has grown a lot, and today, the percentage of people who invest globally is in range 15% - 20%, according to statistics. Despite this growth, there is room for improvement, especially among younger generations who, unfortunately, don't have the opportunity to learn the basics of economics. Many believe that a large amount of money is needed to start investing, it is too risky or just expect to see results in a very short amount of time.

Nowadays, lots of tools are providing assistance for investment-related activities like stock prediction and financial news sentiment analysis. Through this app, I want to provide a risk-free environment where users can learn and become familiar with these markets without losing their own money. They will also get a long-term overview on the changes of prices based on reliable news sentiment analysis using text classification and machine learning for time series forecasting.

The main feature of the app is the prediction of prices for some stocks within a 12-month span. Time series forecasting for the prices will be performed using ARIMA - auto regressive integrated moving average, a technique based on statistical time series analysis which predicts based on changes in prices of the available historical data for each specific stock. The predicted prices will then be adjusted based on macroeconomic and technical indicators, as well as the reliability and sentiment of related news. Additionally, the duration of the news' impact on the prices (short-term or long-term) will be taken into consideration when adjusting price predictions. If some news event has a long-term sentiment impact, the adjustment will extend over several months and if the sentiment impact is short-term, the adjustment will only apply for a month or two. The impact of the sentiment over the prices will be weighted using a model which outputs the credibility percentage for some news article.

All of these models will be integrated into a mobile app that has educational purpose in the finance field and it also provides users with an overview of the future prices for various stocks. In addition to this section, learning and quiz sections will be included, for allowing users to earn rewards in the form of virtual money, which can be used in a demo investing platform to build their own portfolios. These features create together a well organized step-by-step plan that allows users to learn financial basics from scratch, getting hands-on experience in a safe environment with real listed stock prices and use AI-powered tools to help them make the best choices.

Contents

Chapter 1

Introduction

1.1 Stock Markets and Technology in the Context of Financial Literacy

Stock markets are an important factor in the global economy, they drive economic growth and are often seen as an indicator of the economic situation in their specific region. Through stock markets, an individual can invest almost any amount of money. If they don't have enough funds to purchase a specific stock, they can use alternatives such as mutual funds or exchange-traded funds. This creates great opportunities for both businesses and investors to grow together and help each other indirectly.

Unfortunately, there are large categories of people who are financially illiterate. Annamaria Lusardi and Jialu L. Streeter approached this subject in their study in 2023 to get evidence from US[1] by creating a survey that included basic financial questions together with a question which asked the respondents to rate their self-assessed financial literacy. According to the results, despite the low actual level of financial literacy, most respondents gave themselves relatively high ratings. The statistics gained in the above study shows data about US only, which is considered a highly developed country with highly educated people, especially comparing to other regions of the world. This means that financial literacy is significantly lower among the majority of people in many other countries, which represents a sad truth and it implies that these people shouldn't start learning investing and stock markets' basics until they master the basics of financial domain.

Luckily, there are many solutions to help these people and technology is one of them. Technology plays a crucial role and contributed to the development of the financial sector over the years. If back in the days investors needed to go to physical exchange places and get a handwritten/printed certificate, talk to their brokers, and rely on newspapers for financial information, today, on the other hand, everything is done using digital platforms. Technology makes easier the information gathering and learning process too, since users can search any information on search engines really quick. This also means that the learning curve for beginners in this domain or for individuals desiring to acquire knowledge is much smoother compared to previous decades. However, there is still a large number of people who seem to fear getting started due to insecurities, lack of confidence, or simply limited financial education. While technology has already helped this sector evolve, there is still room for improvement, especially when it comes to AI-powered solutions. According to [2], technology such as artificial intelligence may change the way investors look at business decisions, risk, and credit analysis. AI might represent a learning tool for people who want to improve their knowledge about economy and stock markets. Large Language Models like Chat GPT already help in various fields. Of course, the information must also be verified in trusted sources, but the learning process is clearly faster.

1.2 Problem Definition and Motivation

Whether new in financial field or just looking to have a fresh start and a transition from theory to practice, many beginners still face challenges. Despite spending 2 or 3 months reading economy books, articles and acquiring knowledge from various sources, the major difficulty often stands in applying that knowledge to actual investment, which means getting actually started in a practical manner. The idea of putting real money into an account and have the fear of losing a part or all of it can be really discouraging. This is completely understandable, especially due to the lack of practical experience, which can only be gained by learning the hard way through real-world involvement. Traditional educational resources may not offer practical learning experiences, leaving novice investors unprepared to make good decisions in the future.

One of the most common ways investors stay informed about companies and the stock market is through news. Financial news articles, reports, and market updates provide important insights that influence investment decisions. Whether it's an earnings report, a major company announcement, or economic trends, investors often rely on news sources to understand the current state of the market and make informed choices. However, the big volume of information available can be overwhelming, especially for beginners who may struggle to differentiate between truly relevant updates and irrelevant information.

Beyond just reading the news, understanding the sentiment behind it plays an important role in interpreting market movements. News can be positive, negative or neutral and investor reactions often depend on how a piece of information is perceived rather than just the facts. For instance, a company announcing a drop in revenue may seem negative, but if the report includes strong future projections, the market might react in a positive manner. Sentiment analysis helps by determining whether the overall trend of financial news is optimistic or pessimistic, allowing for a deeper understanding of how markets might react.

Beyond the sentiment expressed in individual news articles, the overall sentiment of the masses (market sentiment) has a significant impact on stock prices. Financial markets are highly influenced by investor psychology, where optimism or fear can drive prices up or down, sometimes regardless of a company's actual financial performance. When investors collectively believe that the market is strong, they are more likely to buy stocks, pushing prices higher. On the other hand, fear can lead to panic and overselling, even in situations where the fundamentals of companies remain stable. This phenomenon is often amplified by social media, financial influencers and large institutional investors, who can shape public opinion and cause dramatic price changes. By incorporating sentiment analysis at a broader level, we can capture these trends and better predict how public opinion might influence stock prices in the short and long term.

Another key factor is the credibility of news sources and the presence of misinformation. Not all financial news are accurate, and some sources may exaggerate or spread misleading information, intentionally or unintentionally. There were situations when misinformation was spread quickly specifically for making investors panic and sell for dropping the prices and give the opportunity to others to buy at low prices (also known as market manipulation). Unfortunately, this can happen even if it is illegal because it is hard to prove the intent, but this factor doesn't depend on us and we need to be careful at all our sources of information. Investors acting on false or biased reports risk making poor financial decisions, which could lead to unpredictable losses. By integrating credibility analysis, we can decrease the significance of the unreliable sources and ensure that stock price predictions are based on credible and verified information, even from various sources. This is very important in our digital world because social media and other online sources can spread easily information that may not have any evidence.

In addition to news and sentiment, macroeconomic indicators provide essential context for investment decisions and factors such as interest rates, inflation, unemployment rates can impact stock prices significantly. On the other hand, strong economic indicators such as low unemployment can create a favorable market environment. Ignoring these economic factors can result in an incomplete understanding of market conditions and lead to suboptimal investment strategies.

Given the importance of these factors, integrating news sentiment analysis, credibility assessment and macroeconomic indicators into stock price prediction models provides a more complete and realistic approach to market forecasting. By combining these features, investors, especially beginners, can make better informed decisions, reducing uncertainty and improving their confidence in navigating the financial markets.

1.3 Objectives

This app addresses the gap, combining more features. The main feature of the app is predicting stock prices using time series forecasting techniques for next 12 months. News sentiment and credibility analysis are giving scores as output and they are weighted together to increase or decrease the impact of the sentiment depending the credibility of the news article. Longevity sentiment is also considered for deciding if the prices are affected on short-term (1 month) or long-term (more months). The prices resulted from the time series forecasting model are then adjusted based on news sentiment, credibility and longevity analysis for offering the users an overview for the prices of some stocks on long term and finally macroeconomic and technical indicators are used to adjust the prices. The impact of these adjustments over time gets decreased.

The app also includes educational modules with learning materials, quizzes with rewards for motivating the users to learn and follow the steps, a simulated trading platform with virtual amount of money that can be pumped up only by solving quizzes. The path that a user should follow is reading the recommended learning materials, solve many quizzes and then go to the demo investment platform to use the money gained. Even if the user already starts with a virtual amount, he needs to go through the other modules step-by-step and be able to solve quizzes for getting more money, otherwise one will not be able to get his portfolio bigger due to the lack of funds and will get stuck until solves more quizzes. They could also use the main feature, the AI tool for prices prediction for being able to take easier decisions. This prepares the users for the moment when they will enter the stock market with real money and will get exposed to the real risk of losing the initial value of their stocks. With the experience gained by using this app for a period of time before getting started into the real market, they should be able to take more responsible and information-backed decisions, have a better general overview of the stock market situation and be more confident.

1.4 Original Contributions

Time series forecasting models such as statistical models (ARIMA - Auto Regressive Integrated Moving Average) have been used for a long time, whereas recently, models like LSTM (Long Short-Term Memory), based on deep learning, have gained popularity particularly with the rise of AI in the past 2 decades. Text classification models have also emerged, largely due to the recent remarkable progress in large language models (LLMs). LLMs were propelled by the introduction of self-attention layers and transformers, taking the natural language processing (NLP) tasks at another level. These innovations increased the precision of AI tasks for performing sentiment analysis, fake news detection and other NLP tasks.

Natural language processing models are widely used in finance to, assisting with tasks such as sentiment analysis of financial news, detection of misleading information and automation of the decicion-making processes. However, most stock price prediction models focus either on time series forecasting or sentiment-based adjustments without integrating multiple signals to adjust predictions.

This work introduces a hybrid stock price prediction technique that extends traditional ARIMA models that predict based on historical patterns in changes of price, by integrating news sentiment and credibility analysis together with technical and macroeconomic indicators. Unlike existing approaches that rely completely on historical price trends, our model dynamically adjusts predicted prices based on real-time external influences, offering a forecasting system aware of context. The main contributions of this work are adjusting prices on different factors, using news sentiment, credibility and longevity scores. The reliability of the news is evaluated for reducing impact of misinformation on prices. Financial indicators like unemployment rate are also included for a broader perspective of the economic situation and the app has an integrated simulated trading platform for practical experience to fill the gap between theoretical and real-world investment decisions.

By integrating all these components together, this model makes stock price prediction more accurate and provides a support tool for investors' decisions.

1.5 Thesis Structure

This paper was structured in 6 chapters, the current chapter which summarizes the background of stock markets together with the concepts that investors use for making decisions and the importance of financial literacy, motivation of this paper with its structure. The second chapter 2 reviews previous studies and existing solutions in the field, highlighting methodologies and limitations, providing comparisons of other models and techniques used in literature. Third chapter 3 sets the theoretical concepts (natural language processing and text classification using transformers, the statistical foundation for time series models) needed to understand the implementation, the next chapter 4 is representing the details of the implementation, describing data collection, processing and training for the models and the functionalities available on the graphical user interface and backend. The last 2 chapters include drawing a conclusion based on the results of the models 5.3 and the future work, adjustments and ideas to be implemented for a better and more accurate experience 6.0.1.

Chapter 2

Related Work

2.1 State of the Art

There are already many apps that are built as beginner-friendly educational environments in the financial field. In this chapter, we will explore previous methodologies, existing work, comparisons, and limitations—mainly focused on stock price prediction and text classification tools.

When it comes to demo investing sections, platforms like Etoro and Trading212 already integrate this type of feature, giving users a chance to experiment, learn, and make choices using real-time stock data but with virtual money. This creates a risk-free environment for learning how the stock market works.

As for rewarded quizzes, apps like SaverLife or GoHenry offer money missions or gamified lessons to make financial education more fun. These activities motivate users to keep learning and trying new features, especially because of the competitive points-based system.

Today, many platforms also include AI tools. One important use is stock price prediction, where models analyze historical data and sometimes other factors (like news) to predict how prices might change.

Some common statistical models include ARIMA and SARIMAX. The key difference is that SARIMAX can handle outside influences (like oil prices or interest rates), while ARIMA only looks at past stock prices. These models are good at finding patterns, but only if the data follows certain rules (like seasonality or linear trends).

Machine learning models have been used for years in predicting stock prices, mainly because they're good at finding patterns and making decisions based on past data. These models work by learning from examples—like how stock prices reacted to certain news or economic indicators in the past—and then trying to apply that knowledge to new, unseen situations.

Models like **Random Forest** and **XGBoost** are very popular because they're fast, powerful, and easy to use. Random Forest works by building lots of decision trees, each looking at different parts of the data, and then averaging their predictions. This makes it less likely to overfit. XGBoost takes it one step further by building trees one by one, each correcting the errors of the last. The overall prediction is the sum of the smaller trees:

$$\hat{y}_i = \sum_{k=1}^{K} f_k(x_i), \quad f_k \in \mathcal{F} \tag{2.1}$$

These models are great with structured inputs like past prices, volume, technical indicators (like moving averages), or even categorical features like sectors or economic conditions. But they often need careful feature selection and manual tweaking. Also, they don't really "understand" the order of events or context the way deep learning models can.

Traditional machine learning also includes algorithms like **Support Vector Machines (SVM)**, **Naive Bayes**, and **Logistic Regression**. These are usually faster and simpler but may lack the power to deal with complex or noisy financial data. For example, logistic regression tries to fit a line that separates gains from losses:

$$P(y = 1|x) = \frac{1}{1 + e^{-(w^T x + b)}} \tag{2.2}$$

These models are interpretable, which is good when you want to understand why a prediction was made, but they may miss deeper non-linear relationships in the data.

Deep learning takes things further. Instead of manually choosing features, it lets the model learn which patterns are important by itself. It's especially good when you're working with large and complex datasets, like stock prices over time, financial reports, or social media sentiment.

One of the first deep learning models used in finance was the **Recurrent Neural Network (RNN)**. These models are designed to handle sequences—so they work well with time-series data. RNNs process one data point at a time and keep a "memory" of what they've seen before using a hidden state:

$$h_t = \tanh(W \cdot x_t + U \cdot h_{t-1} + b) \tag{2.3}$$

But RNNs forget things over time, especially when the sequence is long. That's because the gradients (used for learning) shrink with each time step. This makes it hard to learn long-term effects, like how a news event today might influence prices a week from now.

To fix that, **LSTM (Long Short-Term Memory)** networks were introduced. They use special memory units and gates to decide what to remember and what to forget. These gates allow LSTMs to learn patterns over much longer periods.

$$
\begin{aligned}
f_t &= \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) && \text{(Forget gate)} \\
i_t &= \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) && \text{(Input gate)} \\
\tilde{C}_t &= \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) && \text{(Candidate memory)} \\
C_t &= f_t * C_{t-1} + i_t * \tilde{C}_t && \text{(New memory)} \\
o_t &= \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) && \text{(Output gate)} \\
h_t &= o_t * \tanh(C_t) && \text{(Hidden state)}
\end{aligned}
$$

These models can track how major events or slow-building trends influence stock prices, making them useful when the effect of news doesn't happen immediately.

An even more recent development is the use of **transformer models**, like BERT, GPT, or FinBERT. These models are very good at understanding language, especially when it comes to text data like tweets, articles, or reports. Unlike RNNs or LSTMs, transformers can look at all parts of a sequence at once, using a mechanism called self-attention. This allows them to capture relationships between words, even if they are far apart in a sentence.

The attention formula that makes this possible is:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right) V \tag{2.4}$$

Transformers are also easier to train in parallel and can scale better to large datasets. They have achieved state-of-the-art performance in many finance-related tasks like sentiment analysis, fake news detection, and stock prediction.

Hybrid Approaches:

Some of the best-performing systems today actually combine multiple models. For instance:

- Using BERT to extract sentiment from news,
- Then using LSTM to model the time-series impact of sentiment,
- And finally combining all this with XGBoost to make the final price movement prediction.

These hybrid models can get the best of both worlds—rich understanding of text and strong predictive power from historical data.

In the end, the choice of model depends on your dataset, your goals, and how much complexity you're ready to deal with. If you're working with clean historical price data, machine learning might be enough. But if you're including news, social media, or want to model long-term dependencies, deep learning—and especially transformers—are the way to go.

2.2 Existing Work and Comparisons

Many recent models try to include external events like news or social media sentiment to improve predictions. The idea is that news can affect how people feel about a company, and that affects its stock price.

To analyze the sentiment of a text, researchers often use transformer-based models, especially BERT and its variants. One well-known model is CardiffNLP's twitter-roBERTa, which is trained on tweets and performs well with short, informal language.

These models rely on a mechanism called self-attention, which allows the model to focus on the most important parts of the input. The formula used for attention is:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right) V \tag{2.5}$$

where:

- Q: Query vector (what we are looking for)
- K: Key vector (what each word contains)
- V: Value vector (the actual information)
- d_k: Dimension of the key vector.

This allows every word in a sentence to look at every other word, no matter the position, which is useful when the meaning of a sentence depends on context (e.g., sarcasm or negation). The final contextualized representation is what helps the model understand whether a sentence is positive, negative, or neutral.

The result of the sentiment model is a score, which in our approach ranges from -1 (very negative) to +1 (very positive).

Apart from transformer-based models, earlier sentiment analysis approaches include rule-based systems using predefined word lists (like VADER or SentiWordNet). These are fast and easy to understand but often fail to catch sarcasm or complex language. Another common method is using traditional machine learning models like Naive Bayes or SVM with features like bag-of-words or TF-IDF. These work okay for simple text but don't handle word order or meaning very well.

In comparison, transformer models can learn context and meaning automatically. Models like FinBERT or RoBERTa trained on finance-related text perform well on this kind of task and are widely used.

While sentiment is useful, there's a risk if the news itself is fake or misleading. Using bad information can hurt the quality of predictions. That's why our system also includes a credibility analyzer.

Existing approaches for credibility or fake news detection usually combine:

- Source reliability databases (e.g., Media Bias/Fact Check),
- Linguistic features (like excessive punctuation or spelling errors),
- Semantic features (like comparing claims to known facts),
- Deep learning models (such as CNNs, LSTMs, or transformer-based ones like RoBERTa).

Some systems simply label news as *real* or *fake*, while others give a score that reflects confidence. Transformer-based models trained on fake news datasets (such as LIAR or FakeNewsNet) tend to perform the best.

In our system, we combine multiple signals into a single credibility score using the following formula:

$$\text{Credibility Score} = \alpha \cdot S_{\text{source}} + \beta \cdot S_{\text{grammar}} + \gamma \cdot S_{\text{classifier}} \tag{2.6}$$

where:

- $S_{\text{source}} \in [0, 1]$: Score based on domain reputation,
- $S_{\text{grammar}} \in [0, 1]$: Score based on spelling and grammar,
- $S_{\text{classifier}} \in [0, 1]$: Score from a fake news detection model,
- α, β, γ: Weights that add up to 1.

The result is a credibility score between 0 and 1. We then multiply this with the sentiment score:

$$\text{Final Score} = \text{Sentiment Score} \times \text{Credibility Score} \tag{2.7}$$

This way, unreliable news has less influence, while still keeping useful signals from trusted sources.

For example, a well-written article from an unknown site flagged as fake gets a lower score, while an article from a known, trustworthy source gets more weight. This helps avoid wrong predictions based on questionable content.

Lastly, here's a quick comparison of different models used for both sentiment and credibility analysis:

Model Type	**Accuracy (avg)**	**Pros**	**Cons**
Rule-based (VADER, SentiWordNet)	65–75%	Fast, easy to explain	Poor with complex text
Naive Bayes / SVM	70–78%	Simple, efficient	Needs feature engineering
LSTM / BiLSTM	80–85%	Captures word order	Slower, needs more data
CNN for text	82–86%	Good for patterns	Weak for long context
BERT-based models	88–93%	High accuracy, contextual	Needs more resources
RoBERTa / FinBERT	90–95%	Best accuracy in finance/news	Large, harder to deploy

Figure 2.1: Comparison of model types for sentiment and credibility analysis

Overall, using transformer-based models for both sentiment and fake news detection gives more reliable results. And by combining their outputs, we can make smarter decisions that account for both the tone of the news and how trustworthy it is.

2.3 Limitations

Every model has its weak spots. Even though machine learning and deep learning are powerful, they don't work like magic. Some of the simplest models, like **ARIMA** (used for time-series forecasting), only look at past values and assume that future trends will follow the same rules. This might work for stable environments, but financial markets are full of surprises—like unexpected earnings reports, political news, or viral tweets. ARIMA can't deal with that kind of chaos.

Even models like **LSTM**, which are built to understand patterns over time, don't always keep up. On their own, they often focus just on historical price movements. They don't "read" the news or follow what's trending online. So, if there's a sudden crash because of breaking news, an LSTM-only model might totally miss it. Studies show that when only price data is used, prediction accuracy for short-term movements usually hovers between **70–78%**, but that can drop sharply during volatile periods.

Another issue is with **sentiment-based models**. These use tools like BERT or FinBERT to analyze news articles, headlines, or tweets and figure out if the sentiment is positive, negative, or neutral. That's a good start, but there's a big problem: these models often don't check if the news is even *true*. That means fake or misleading content can trigger strong signals and push the model to make the wrong call.

For example, a tweet from a fake financial account claiming a CEO stepped down could cause a negative sentiment spike. If the model reacts to that without checking the source, it might wrongly predict a stock drop. This is a major issue—about **11% of financial tweets** during earnings season are later flagged as misleading or unverifiable, and around **6%** of sentiment-driven predictions in real-time setups are traced back to such unreliable sources.

Some models try to fight this by filtering out low-credibility sources. That helps—but it's not perfect either. Sometimes, valuable insights or early warnings come from small, unverified accounts or controversial posts. If we filter too hard, we miss those. It's a balancing act between being cautious and staying informed.

Then comes the issue of combining multiple models to solve these problems. A setup that includes:

- LSTM for price trends,
- Transformer (like FinBERT) for reading sentiment,
- Classifier for verifying news credibility,

can definitely improve prediction performance, but it also makes things more complicated. More models means more data, more time to train, and more risk of errors during integration.

Training time can increase by 2x to 5x depending on dataset size. If a basic LSTM trains in 1 hour, a combined setup with sentiment and credibility checks might take 3–6 hours. Also, these models need large, high-quality datasets. For example, FinBERT needs labeled financial text, while LSTM needs clean historical prices. But in real life, data is often noisy, incomplete, or inconsistent.

Another limitation is overfitting. With more layers, more models, and more inputs, it becomes easier for the model to memorize the training data instead of learning general patterns. This can make backtesting results look great—like 90%+ accuracy—but performance in live trading drops back down to 75–80% once exposed to fresh, unpredictable data.

Also, none of these models can perfectly capture human emotion or market manipulation. Some stock moves are driven by fear, hype, or insider activity that no public data source will ever show. Models can detect signals, but they can't explain intent.

So what works best, realistically?

- For **basic stock price prediction**, tree-based models like XGBoost and LSTM hybrids perform well. They're stable, fast, and reach around 75 to even 85% accuracy with the right features.
- For **sentiment analysis**, transformer models like BERT or FinBERT are the clear winners. They understand financial text with much more nuance than older methods like keyword counting. Accuracy on labeled datasets reaches **88–92%**.
- For **credibility filtering**, simple classifiers or trust scoring models work best when combined with news-source databases or fact-checking APIs. They don't catch everything, but they help reduce the number of risky signals.

No single model can handle everything. Stock prices are influenced by time-series data, news, rumors, and even public emotion. The most reliable results come from combining specialized models—each doing one job well. It's a bit like building a team: one model reads the news, another checks if it's legit, and another tracks the price patterns. Working together, they can respond faster, predict better, and avoid dumb mistakes caused by noise or fake data.

While this setup isn't perfect and needs lots of resources to train and maintain, it's the most balanced approach for modern stock prediction systems—especially when you're dealing with high-speed, sentiment-driven markets like crypto, tech stocks, or trending tickers.

Chapter 3

Theoretical Concepts

3.1 Theoretical Overview

This section will explore in depth the main theoretical concepts behind the development of the educational finance app. The app integrates deep learning models for stock price prediction, sentiment analysis, and fake news detection. These models will be discussed in detail, including their architectures, methodologies, and how they work together to enhance predictions, provide educational value, and offer insights into future stock market trends. The first section will cover stock price prediction, comparing ARIMA and LSTM in time series forecasting context and exploring their architectures. The next one will examine sentiment analysis, focusing on well-known models such as BERT and transformers, explaining their architectures and the fine-tuning process. The last section related to the AI models in this chapter will cover fact-checking techniques and models used for news credibility assessment, including source credibility and grammars.

3.2 Time Series Forecasting for Stock Prices

3.2.1 Statistical Approaches: Auto Regression, Linear Pattern Recognition and Differencing (ARIMA)

3.2.2 Machine Learning Approaches: Gated Recurrent Unit (GRU) - !!! de verificat daca e ml sau dl

3.2.3 Deep-Learning Approaches: Recurrent Neural Networks

3.2.4 Long Short-Term Memory Arhitecture

3.2.5 Macroeconomic and Technical Indicators

3.3 Natural Language Processing

3.3.1 Text Classification and Sentiment Analysis

3.3.2 Transformers Arhitecture

Embedding Layer and Positional Encoding

Self-Attention Mechanism

Encoder-Only Arhitectures: Bidirectional Encoder Representations Transformers (BERT)

3.3.3 Fine-Tuning For Financial News Sentiment Analysis

3.3.4 Sentiment Regressor

3.4 News Credibility Analysis

3.4.1 Supervised Learning

3.4.2 Support Vector Machines

3.4.3 Source Reliability / Grammatical Assessment

Chapter 4

Implementation

4.1 Structure of Implementation Chapter

In this chapter, the development and technical details of the project will be explained. It starts with a general overview of the app's main features and details of implementation on both server and client side, explaining the functionalities that the users can interact with. Next, the chapter walks through the various models used to predict stock prices, analyze news sentiment, assess the credibility of news sources and putting them together to find a compact AI-powered functionality. Each of these components is broken down into simple steps, starting with gathering and processing data, followed by building the model, fine-tuning it and finally training and evaluating it by revealing the metrics. Each section is structured to show how the different pieces fit together, from the initial data collection to the final deployment of the models, and provides a cohesive understanding of the app's backend structure and the AI models powering it.

4.2 Overview of the Educational Finance App Functionalities

4.2.1 Client

The app starts with an authentication panel that ensures secure access to user specific features. New users can register by providing a username, email and password. During registration, their password is securely hashed before being stored and each account is initialized with a virtual balance. Once registered, users can log in by submitting their credentials. If verified, the system generates a secure access token (JWT - Json Web Token), which allows the user to remain authenticated for a limited period (3 hours). This token must be included in future requests to access protected resources such as checking account information or updating virtual balances. The app also includes a logout feature that invalidates the current token by adding it to a blacklist, ensuring it can no longer be used. Users can access the functionalities of the app through authenticated routes. This design promotes security and access to data specific to a user across the app. Besides the home page that contains the user details, there are also the 4 main sections. First main section is called "Learning Materials" and contains a description of the app and the importance of financial literacy among individuals. It also contains links to public websites that contain relevant financial materials such as courses. "Quizzes" and "Demo Investing" are the 2 next main sections, first of them is providing the user with the opportunity of gaining rewards represented in an amount of virtual money, depending on the difficulty of the solved quiz. This amount of virtual money can be used in "Demo Investing" section to allow the user build and manage their own portfolios with listed stock companies and indices. The last section contains the AI functionality, represented by a stock price prediction model (its implementation will be discussed in more details in the later chapters). Its predicted price will finally be adjusted depending on the sentiment extracted from the news analyzer. As news analyzer, in our case will output a score in the interval [-1, 1], closer to -1 representing a negative sentiment, closer to 0 being a more neutral one and closer to 1 marking a more positive news article. The article will be given as input by the user and some stock symbol / ticker, for which to predict the prices based on the impact that the news may have. The impact is reduced if the news are less credible, which is found by using a fact-checking model that gives a score to the source of the article (in case it is specified by the user) that also finds a grammar score based on possible grammar mistakes in the article, since most of the less credible news imply an increased possibility of mistakes in the text.

4.2.2 Server

As backend I used FastAPI, a modern high-performance Python web framework designed for building APIs fast and efficiently. FastAPI leverages Python type hints for automatic data validation and serialization and supports asynchronous programming, enabling it to handle high-concurrency workloads effectively.

To highlight FastAPI's performance advantages, the following table compares key metrics across popular Python web frameworks:

Framework	Requests per Second	Average Latency
FastAPI	45,000–50,000	10–15 ms
Flask	30,000–40,000	20–25 ms
Django	35,000–45,000	15–20 ms

Figure 4.1: Performance comparison of Python web frameworks

As shown, FastAPI outperforms Flask and Django in both throughput and latency, primarily due to its asynchronous capabilities and efficient request handling. This performance makes FastAPI an excellent choice for building scalable and responsive backend services.

For an easier understanding of the overall structure, I start by providing below the architecture diagram for a better overview of the backend structure:

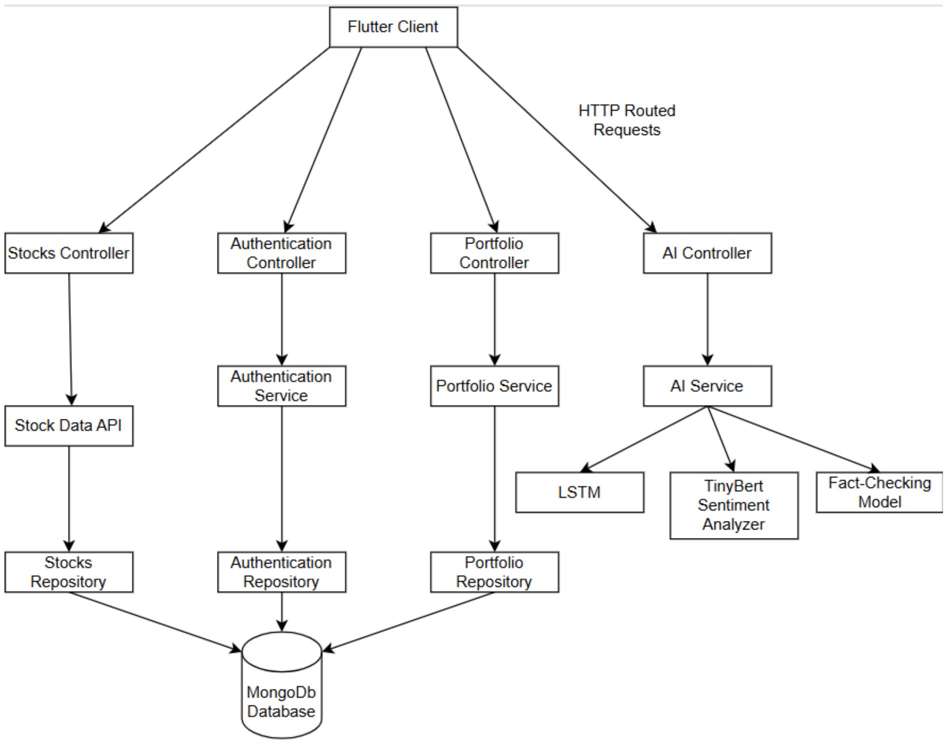

Figure 4.2: Architecture Diagram

Following the architecture diagram, the class diagram provides a more detailed representation of the application's object oriented design. It illustrates how different entities such as users, quizzes and financial data are represented as classes and interact with each other within the system. Each class is defined by its attributes and behaviors and the relationships between these classes are modeled to ensure efficient interactions within the system. In general, the class diagram helps to visualize the interactions between core components like user accounts, portfolio management and the AI price prediction model. This design facilitates seamless data flow and interaction between the client and server components, ensuring that each part of the application is modular and scalable. The following class diagram further details the relationships and design choices made to structure the backend functionality, including user authentication, the handling of virtual balances, and the integration of the AI features. It highlights the key classes and their methods, giving a clearer picture of the application's internal workings:

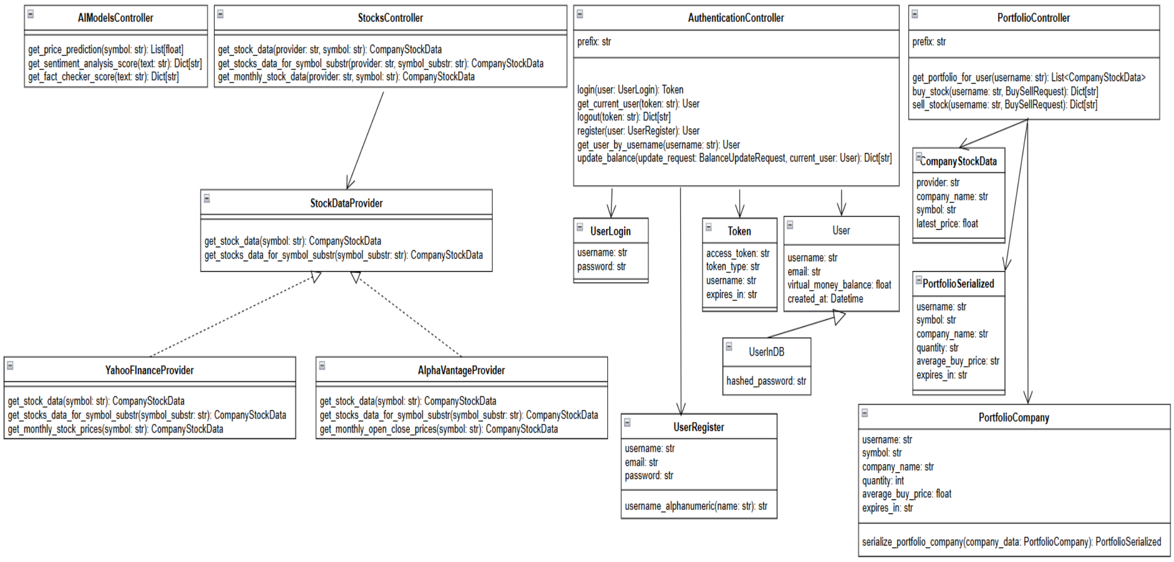

Figure 4.3: Class Diagram

4.2.3 Fetching Stock Data via APIs

Since 2 sections in the app require stock related data such as daily or monthly close prices and company information, the backend integrates two external providers: **Yahoo Finance** and **Alpha Vantage**. These providers are accessed via their respective APIs. Data is fetched either for a single stock symbol or for multiple stocks matching a substring (e.g., "AAPL" or "Tes"). For Yahoo Finance, the `yfinance` and `yahooquery` Python packages are used, allowing retrieval of current prices and monthly close data. Alpha Vantage is queried directly via HTTP requests, using an API key, to access intraday prices and adjusted monthly data. Each provider is wrapped in a class that implements a common abstract interface, making it easy to switch or extend sources, using a function for switching between the 2 APIs depending on the user's needs (provider selected by the user in the client side).

4.2.4 Portfolio Management

Users can either buy / sell stocks in different quantities, this works based on the current prices fetched by the stock data providers / APIs, which help by retrieving the current market prices for the specific ticker / stock symbol desired by the user. If the stock already exists in the portfolio for the connected user, its average purchase price and quantity are updated. Conversely, selling stocks credits the user's virtual money and updates or removes the stock entry depending on whether shares remain. All operations are persisted in MongoDB collections, ensuring a consistent and up-to-date portfolio view per user.

4.2.5 Integrated AI Models

The 4th main section of the app with the 3 AI-powered functionalities has its own page, where the user gives as input the ticker to get the prices for, another textbox for putting the news article regarding the company represented by the ticker, for which will be computed a sentiment score, floating point number in interval [-1, 1]. Another score in the interval [0, 1] is computed for the same text by using a fact-checking model, being used for reducing the significance of the sentiment, depending on the credibility assessed by it. Using the scores, the prices predicted by the price prediction are adjusted in the background and finally retrieved to the user in numerical values and represented in a graph for a better vizualisation. These models are accessed via dedicated backend routes and seamlessly integrated into the client.

4.3 Price Prediction (Time Series Forecasting)

4.3.1 Data Collection and Preprocessing

Data Sources

Feature Selection

Sequence Generation (Sliding Window)

Normalization and Scaling

Train-Test Splitting Strategy

4.3.2 Model Architecture

Why LSTM for Time Series

Two-Layer LSTM Design

Activation Functions and Output Layer

Loss Function and Optimizer

4.3.3 Hyperparameters and Tuning

Window Size and Batch Size

Learning Rate and Optimizer Settings

Dropout and Regularization

Epochs and Early Stopping

4.3.4 Training the Model

Training Strategy

Loss Monitoring and Logging

4.4 Sentiment Regressor For Financial News

4.4.1 Data Collection and Preprocessing

Financial Phrase Bank Dataset

Custom Dataset With Financial Words, Phrases and Scores

Preprocessing and Lemmatization

4.4.2 Model Arhitecture

TinyFinBERT Regressor Design

Attention Mechanism for Weighted Pooling

Chapter 5

Results and Conclusion

5.1 Evaluation Metrics for Each Model

5.1.1 LSTM

5.1.2 Comparison between Fine-Tuned TinyBert and Base TinyBert

5.1.3 Fact Checker

5.1.4 Overall Adjusted Price Prediction

5.2 Educational and Financial Growth, Future Trends Forecasting Using AI Tools. Gamified Learning and Quiz Engagement

5.3 Current Limitations

Model Constraints, data availability (especially for fact checker - checking multiple sources needed)

Chapter 6

Future Work

Enhance AI models, AI-assisted learning features (audio learning materials), enhancing model for fact checking with more fact checked data), fetch data from twitter posts / reddit financial threads to find out overall sentiment among groups of people

6.0.1

Bibliography

[1] Jialu L. Annamaria Lusardi Streeter. "Financial Literacy and Financial Well-Being: Evidence from the US". In: *Journal of Public Policy* (2023). Published online: 05 October 2023. DOI: 10.1017/S0143814X23000238. URL: https://www.cambridge.org/core/journals/journal-of-financial-literacy-and-wellbeing/article/financial-literacy-and-financial-wellbeing-evidence-from-the-us/318307008828D2D7932C13E04B

[2] A. A. P. Pratama. "Investment Opportunity of Financial Technology". In: *The International Journal of Business Management and Technology* 3.5 (2019), pp. 1–10. ISSN: 2581-3889. URL: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3476618.