AI for Stock Trading

Msc. Peng Li

Master's thesis
2023

Tomas Bata University in Zlín
Faculty of Applied Informatics

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky
Ústav informatiky a umělé inteligence

Akademický rok: 2022/2023

ZADÁNÍ DIPLOMOVÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení:	**Peng Li**
Osobní číslo:	**A21561**
Studijní program:	**N0613A140023 Information Technologies**
Specializace:	**Software Engineering**
Forma studia:	**Prezenční**
Téma práce:	**Umělá inteligence pro obchodování s akciemi**
Téma práce anglicky:	**A.I. for Stock Trading**

Zásady pro vypracování

1. Make a literature review with a focus on various A.I. techniques in stock trading.
2. Select or collect the appropriate datasets for a given task.
3. Select the methodology from the A.I. field.
4. Implement the selected techniques and provide experimantal results for datasets.
5. Provide the analysis of results and comparisons of techniques/algorithms.

Forma zpracování diplomové práce: **tištěná/elektronická**

Seznam doporučené literatury:

1. RAY, Ruchira, Prakhar KHANDELWAL a B. BARANIDHARAN. A Survey on Stock Market Prediction using Artificial Intelligence Techniques. In: 2018 International Conference on Smart Systems and Inventive Technology (ICSSIT) [online]. IEEE, 2018, 2018, s. 594-598 [cit. 2022-11-30]. ISBN 978-1-5386-5873-4. Dostupné z: doi:10.1109/ICSSIT.2018.8748680
2. ARULKUMARAN, Kai, Marc Peter DEISENROTH, Miles BRUNDAGE a Anil Anthony BHARATH. Deep Reinforcement Learning: A Brief Survey. IEEE Signal Processing Magazine [online]. 2017, 34(6), 26-38 [cit. 2022-12-02]. ISSN 1053-5888. Dostupné z: doi:10.1109/MSP.2017.2743240
3. NTI, Isaac Kofi, Adebayo Felix ADEKOYA a Benjamin Asubam WEYORI. A systematic review of fundamental and technical analysis of stock market predictions. Artificial Intelligence Review [online]. 2020, 53(4), 3007-3057 [cit. 2022-12-02]. ISSN 0269-2821. Dostupné z: doi:10.1007/s10462-019-09754-z
4. LAZZERI, Francesca. *Machine learning for time series forecasting with Python*. Indianapolis: Wiley, 2021, 1 online resource (227 pages). ISBN 9781119682394. Dostupné také z: https://proxy.k.utb.cz/login?url=https://onlinelibrary.wiley.com/doi/book/10.1002/9781119682394
5. MOSTAFA, Fahed, Tharam S. DILLON a Elizabeth CHANG. *Computational intelligence applications to option pricing, volatility forecasting and value at risk*. Cham, Switzerland: Springer, 2017, 1 online resource (x, 171 pages). Studies in computational intelligence. Dostupné z: doi:9783319516684
6. KONAR, Amit a Diptendu BHATTACHARYA. *Time-series prediction and applications: a machine intelligence approach*. Cham, Switzerland: Springer, 2017, 1 online resource. Intelligent systems reference library. Dostupné z: doi:9783319545974.

Vedoucí diplomové práce: **doc. Ing. Roman Šenkeřík, Ph.D.**
Ústav informatiky a umělé inteligence

Datum zadání diplomové práce: **2. prosince 2022**
Termín odevzdání diplomové práce: **26. května 2023**

doc. Ing. Jiří Vojtěšek, Ph.D.
děkan

prof. Mgr. Roman Jašek, Ph.D., DBA
ředitel ústavu

Ve Zlíně dne 7. prosince 2022

Name of the student:
Thesis topic:
I hereby declare that:

- I understand that by submitting my Master´s thesis, I agree to the publication of my work according to Law No. 111/1998, Coll., On Universities and on changes and amendments to other acts (e.g. the Universities Act), as amended by subsequent legislation, without regard to the results of the defence of the thesis.
- I understand that my Master´s Thesis will be stored electronically in the university information system and be made available for on-site inspection, and that a copy of the Master´s Thesis will be stored in the Reference Library of the Faculty of Applied Informatics, Tomas Bata University in Zlín, and that a copy shall be deposited with my Supervisor.
- I am aware of the fact that my Master´s Thesis is fully covered by Act No. 121/2000 Coll. On Copyright, and Rights Related to Copyright, as amended by some other laws (e.g. the Copyright Act), as amended by subsequent legislation; and especially, by §35, Para. 3.
- I understand that, according to §60, Para. 1 of the Copyright Act, TBU in Zlín has the right to conclude licensing agreements relating to the use of scholastic work within the full extent of §12, Para. 4, of the Copyright Act.
- I understand that, according to §60, Para. 2, and Para. 3, of the Copyright Act, I may use my work - Master´s Thesis, or grant a license for its use, only if permitted by the licensing agreement concluded between myself and Tomas Bata University in Zlín with a view to the fact that Tomas Bata University in Zlín must be compensated for any reasonable contribution to covering such expenses/costs as invested by them in the creation of the thesis (up until the full actual amount) shall also be a subject of this licensing agreement.
- I understand that, should the elaboration of the Master´s Thesis include the use of software provided by Tomas Bata University in Zlín or other such entities strictly for study and research purposes (i.e. only for non-commercial use), the results of my Master's Thesis cannot be used for commercial purposes.
- I understand that, if the output of my Master´s Thesis is any software product(s), this/these shall equally be considered as part of the thesis, as well as any source codes, or files from which the project is composed. Not submitting any part of this/these component(s) may be a reason for the non-defence of my thesis.

I herewith declare that:

- I have worked on my thesis alone and duly cited any literature I have used. In the case of the publication of the results of my thesis, I shall be listed as co-author.
- That the submitted version of the thesis and its electronic version uploaded to IS/STAG are both identical.

In Zlín; dated: ....

Student´s Signature

ABSTRAKT

Tato práce se zabývá technologií AI pro obchodování s akciemi, konkrétně predikcí cen akcií. K predikci cen vybraných akcií (AAPL, MSFT, TSLA, META, GOOG) byly použity modely jako dlouhodobá - krátkodobá paměť, regresní a klasifikační feed forward neuronové síě, hluboké učení (hluboké posílení učení), a regresní model optimalizovaný pomocí metody rojení částic. V rámci této diplomové práce byly v praktické části výkonnosti různých modelů analyzovány, porovnány a prodiskutovány. Výsledky ukázaly, že model hlubokého učení poskytnul nejlepší výkon (s průměrným skóre 95 % R-squared) a model s klasifikační dopřednou neuronovou sítí byl nejhorší (s průměrným skóre přesnosti pouze 50 %). Také ostatní modely ukázaly velký potenciál pro předpovídání cen akcií, stejně jako model využívající optimalizátor optimalizaci rojem částic pak demonstroval výhodu automatického ladění hyperparametrů. Cílem je, aby tato práce přispěla k dalšímu výzkumu v oblastech AI, algoritmů, financí a informatiky.

Klíčová slova: akcie, predikce, AI, neuronová síť, hyperparametr, optimalizátor

ABSTRACT

This thesis studied about the artificial intelligence (AI) technology for stock trading, specifically about stock price prediction. The long short-term memory neural network (LSTM), regression feed – forward neural network (RFFNN), classification feed – forward neural network (CFFNN), deep reinforcement learning (DRL) and the particle swarm optimization (PSO) optimized RFFNN (RFFNN-PSO) models were used to predict the AAPL, MSFT, TSLA, META, and GOOG stocks' prices and the performance of the different models were analyzed, compared and discussed. The results showed the DRL model had the best performance (with average 95% R-squared score (R2)) and the CFFNN model had the poorest (with only average 50% accuracy score). Also, the LSTM and RFFNN showed the great potential to predict stock prices as well as the RFFNN-PSO model showed the advantage of auto-tuning hyperparameters. This study is expected to contribute to the further research of the fields in AI, algorithms, finance and computer science.

Keywords: stock, prediction, AI, neural network, hyperparameter, optimizer

Acknowledgements:
I am deeply grateful to Prof. Ing. Roman Šenkeřík, Ph.D. for his invaluable guidance and support in my master's thesis. His expertise and dedication have been instrumental in shaping the outcome of my research, and I am truly thankful for the opportunity to work under his supervision.

I hereby declare that the print version of my Master's thesis and the electronic version of my thesis deposited in the IS/STAG system are identical.

Contents

INTRODUCTION

AI and machine learning (ML) have increasingly been applied to the domain of stock trading, with a growing body of research examining various techniques and methodologies. Several intelligent systems and AI techniques, including artificial neural networks (ANN), genetic algorithms, support vector machines, and ML, have been developed for decision support and complex automation tasks. ANN, in particular, have garnered significant attention due to their ability to analyze complex non-linear relationships between variables. However, the development of an accurate predictive model using AI and soft computing techniques for stock market prediction remains a challenge.[1]

One notable development in the application of AI to stock trading is the OpenAI Gym Anytrading tool. This open-source platform provides a collection of reinforcement learning (RL) - based trading algorithms, aiming to facilitate the development and testing of such algorithms in the area of market trading. This is accomplished by implementing the algorithms on three Gym environments: TradingEnv, ForexEnv, and StocksEnv. These environments can aid in learning about stock market trends and provide powerful analysis for data-driven decisions. The tool is capable of handling real-world data, such as historical price data, and can train RL agents to make trading decisions based on this data. It's important to note that while the tool has shown promising results in simulated environments, its performance in real-world stock markets is yet to be fully evaluated. [2]

In the context of recent research, there is still active exploration of various AI techniques for stock trading, with no one-size-fits-all solution. The selection of input data, optimization of parameter selections and model architectures, and the pre-processing of input data are key areas of focus in ongoing research.

AI stock trading robot is the product of deep integration of AI innovation technology and stock trading. With the continuous development of science and technology, AI technology has been improved rapidly like never before. The securities trading industry is also ushering in profound changes, and the AI stock trading robot came into being in such a wave of the times. When it comes to AI stock trading robots, many people will think of AlphaGo in the Go world developed by Google. At that time, the man-machine battle between AlphaGo and the top human players attracted worldwide attention. In the first game, AlphaGo defeated Li Shishi with a total score of 4 to 1; With a total score of 3 to 0 against the world's No. 1 Ke Jie, AlphaGo's crushing victory shocked all human beings and made us truly realize the power of AI. [3]

The AI has such an overwhelming advantage over humans since the computing power of the human brain is very limited, so this study condenses a large amount of life experience into a state that does not need to be calculated from scratch. Yet the computer is not. The computer is always calculating from the beginning, calculating the probability of each step, and betting on the highest probability every time. Each calculation from the beginning is based on massive data resources. Thus, calculation beyond the experience and stereotypes of users, and a brand-new logic is calculated.

More frighteningly, the next generation of AlphaGo, Alphazero, crushed AlphaGo 100:0. The success of AI in the world of Go has aroused the deep expectation of human beings trying to obtain huge wealth through AI in the financial field.[4]

With the continuous penetration of AI in various industries, people cannot help thinking about the cooperation and competition with robots in the future. Similarly, in the field of financial investment, robots trade stocks, will be a threat for fund managers to face the unemployment. At its peak in 2000, Goldman Sachs employed 600 traders at the U.S. cash stock trading desk at its New York headquarters, however by now, there are only two stock traders left "staying vacant".[5]

In October 2017, AI Powered Equity ETF, the world's first AI-powered ETF, was listed on the New York Stock Exchange. AIEQ uses the cognition and big data processing of IBM's AI system to analyze and invest in US stocks. AIEQ works day and night throughout the year, analyzing more than 6,000 US-listed stocks at the same time, and analyzing millions of related announcement documents, financial reports, news and community articles every day. Use quantitative timing, quantitative stock selection, factor analysis, event-driven and other quantitative models to select stocks and continuously learn in depth.[6]

AIEQ had a good performance and once beat the index. However, three years later, AIEQ still significantly underperformed Nasdaq, the S&P 500, and only slightly outperformed the Dow Jones Industrial Average.

According to Wind data, from its establishment on October 18, 2017 to October 17, 2020, AIEQ has risen by 28.7%. During the same period, the S&P 500 index rose by 36.12%, and the Nasdaq index rose by 76.21%. Less than half of Nasdaq.

It is worth mentioning that although it is difficult to beat the index, it is still possible to beat the opponent. According to Wind data, among 1,715 US stock ETFs and mixed ETFs, AIEQ ranked 328th in terms of accumulative gains over the same period, beating 80.87% of its peers.

Stock trading is very predictable based on the already existed indexes such as relative strength index (RSI), Stochastic oscillator (STOC), moving average convergence divergence (MACD), Bollinger Bands, Stochastic Index etc. also the trend of price increase and decrease is predictable and repeatable along with the time series. Such features are very suitable to be analyzed and predicted by AI technologies such as Neural Network (NN), Deep Learning (DL), ML, Datamining and Evolutionary Computation etc.

This thesis uses NN, DL, ML, Datamining and Evolutionary Computation to analyze and predict the trend of stock price, compare, optimize the methods and algorithms. This thesis is divided into two main blocks. The first theoretical part introduces the conceptions of AI, NN, specifically LSTM, CFFNN, RFFNN, DRL and RFFNN – PSO. The second analysis part contains the description of detailed experiment procedures, results, analysis and discussion of LSTM, CFFNN, RFFNN, DRL and RFFNN – PSO price prediction. Our goal is to reach the high accuracy of the stock price prediction by using the stated methods and models and find out the best model for the stock price prediction with the highest accuracy.[7]

I. THEORY

1 AI METHODOLOGIES

AI methodologies refer to the various approaches, techniques, and algorithms used in AI to solve complex problems, make predictions, or simulate human intelligence in machines. In this study, the following AI methodologies were used:[8, 9, 10]

ML: ML is an AI methodology that involves the development of algorithms that can learn from and make decisions or predictions based on data. These algorithms can improve their performance as they are exposed to more data over time. There are three main types of ML: supervised learning, unsupervised learning, and RL.

DL: DL is a subset of ML that uses ANN with multiple layers (hence "deep") to model and understand complex patterns in datasets. It's particularly useful for tasks such as image and speech recognition.

RL: RL is a type of ML where an agent learns to make decisions by taking actions in an environment to maximize some notion of cumulative reward. The agent learns from trial and error, receiving rewards or penalties for the actions it takes.

NNs: NNs are a set of algorithms modeled loosely after the human brain, designed to recognize patterns. They interpret sensory data through a kind of machine perception, labeling, or clustering raw input.

Swarm Intelligence: Swarm intelligence is a methodology based on the collective behavior of decentralized, self-organized systems, and is used in optimization problems.

These methodologies can be used individually or combined to build complex AI systems. The choice of methodology depends on the specific problem that needs to be solved, the data available, and the desired outcome.

2 ARTIFICIAL NEURAL NETWORK

ANNs are a type of computational model inspired by the structure and function of the human brain. They are composed of interconnected nodes or neurons that can process information and learn from data. ANNs have a rich history that spans several decades, with various breakthroughs, inventions, and applications.

The earliest known precursor to ANN was the work of Warren McCulloch and Walter Pitts in the 1940s. They proposed a computational model of a neuron that could perform logical operations, known as the McCulloch-Pitts neuron. This work laid the foundation for the development of ANN as a computational paradigm.[11]

In the late 1950s, Frank Rosenblatt developed the Perceptron, which was the first artificial neuron capable of learning from data. The Perceptron could be trained to classify inputs into two categories, and it was considered a significant breakthrough in the field of AI at that time.[12]

In the 1960s and 1970s, researchers such as Marvin Minsky and Seymour Papert pointed out limitations in the Perceptron, showing that it could not learn complex patterns. This led to a period of reduced interest in ANN, known as the "AI winter."[13]

In the 1980s, backpropagation, a widely used training algorithm for ANN, was invented by several researchers independently, including Paul Werbos, David Rumelhart, and Ronald Williams. This breakthrough made it possible to train ANN with multiple layers, also known as multi-layer perceptrons (MLPs), and paved the way for the resurgence of interest in ANN.

Since then, ANNs have been extensively studied and applied in various fields. In the 1990s and 2000s, researchers such as Geoffrey Hinton, Yoshua Bengio, and Yann LeCun made significant contributions to the field of DL, which involves training ANN with multiple hidden layers. DL has revolutionized fields such as computer vision, natural language processing, and speech recognition, achieving state-of-the-art performance in many tasks.[14]

Numerous research institutions, laboratories, and universities have contributed to the advancements in ANN. For example, the University of Toronto, the University of Montreal, and the New York University's Center for Data Science have been at the forefront of DL research. Industrial research labs, including Google Brain, Facebook AI Research (FAIR), and OpenAI, have also made significant contributions to the field.[15]

The basic structure of an ANN consists of layers of interconnected nodes or neurons. Each neuron receives input from its connected neurons, applies an activation function, and produces an output that is passed to the next layer of neurons. The neurons are organized in layers, including an input layer, one or more hidden layers, and an output layer. The connections between neurons have associated weights that are learned during training, and these weights determine the strength of the connections. During training, ANN learn to adjust these weights to minimize the error between their predicted outputs and the actual outputs. The basic structure of an ANN (as shown in Figure 1 - 2), e.g. used Sigmoid activation function as shown in formula (1) and the procedures of calculating the outputs are shown in formula (2): [16]

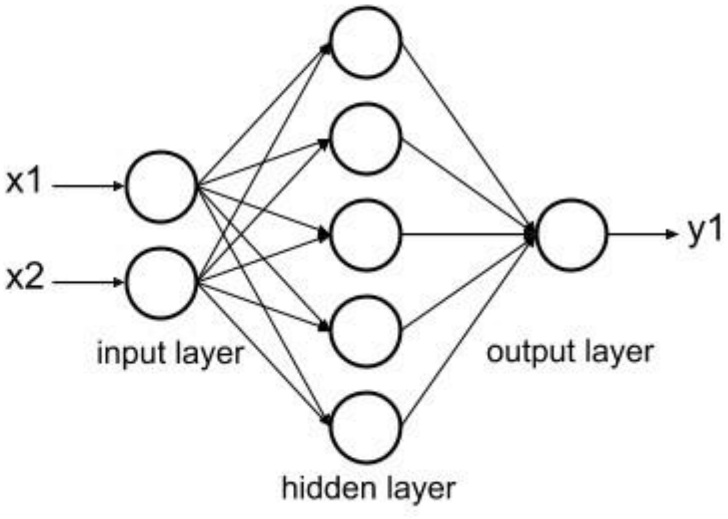

Figure 1 Basic structure of an ANN with 1 input layer (with 2 nodes), 1 output layer (with 1 node) and 1 hidden layer (with 5 nodes) [1]

[1] https://www.sololearn.com/learning/1094/3400/7717/1

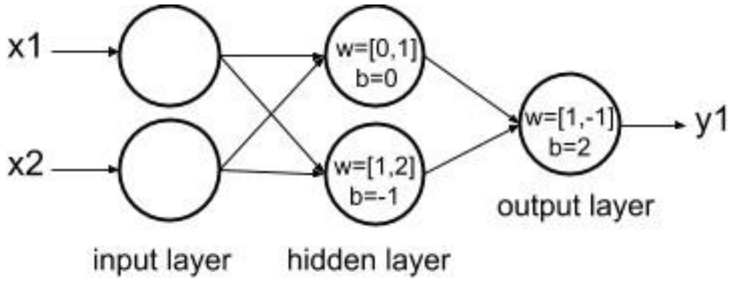

Figure 2 Each node in a layer has its own weights and bias values [2]

$$sigmoid(x) = \frac{1}{1+e^{-x}} \tag{1}$$

$$y = f(w_1x_1 + w_2x_2 + b) = \frac{1}{1+e^{-(w_1x_1+w_2x_2+b)}} \tag{2}$$

ANN have a rich history and have made significant contributions to the field of AI. From the early work of McCulloch and Pitts to the recent breakthroughs in DL, ANN have become a powerful tool for solving a wide range of complex problems. Researchers from various institutions and organizations have contributed to the advancements in ANN, and their applications continue to expand in areas such as computer vision, natural language processing, robotics, and healthcare, among others.[17]

2.1 Long short – term memory neural network

LSTM is a type of recurrent neural network (RNN) architecture that is specifically designed to address the vanishing gradient problem and capture long-term dependencies in sequential data. LSTMs were introduced by Sepp Hochreiter and Jürgen Schmidhuber in 1997 as an extension of the RNN model.[18]

The history of LSTMs can be traced back to the early days of RNN, which were first proposed in the 1980s. RNN were designed to process sequential data by allowing information to persist in the network through recurrent connections. However, RNN were found to suffer from the vanishing gradient problem, where gradients become very small during training, leading to difficulties in capturing long-term dependencies in sequences.

In 1997, Hochreiter and Schmidhuber proposed LSTMs as a solution to the vanishing gradient problem in RNN. LSTMs introduced a specialized memory cell that could store and update information over long sequences, making them capable of capturing long-term

[2] https://www.sololearn.com/learning/1094/3400/7718/1

dependencies in data. LSTMs use a gating mechanism that allows them to control the flow of information through the cell, making them more robust in processing sequential data.

Since the introduction of LSTMs, they have been widely studied and applied in various fields. Researchers and institutions from around the world have contributed to the advancements in LSTMs. For example, researchers from the University of Toronto, New York University, and the University of Montreal, including Geoffrey Hinton, Yoshua Bengio, and Yann LeCun, have made significant contributions to the field of DL, including the use of LSTMs in areas such as natural language processing, speech recognition, and time series analysis.[19]

In recent years, there has been ongoing research on improving the architecture and performance of LSTMs. Variants of LSTMs, such as Gated Recurrent Units (GRUs) and Peephole LSTMs, have been proposed to further enhance their capabilities in capturing long-term dependencies and processing sequential data. Additionally, there has been research on integrating LSTMs with other NN architectures, such as convolutional neural networks (CNNs) and attention mechanisms, to leverage their complementary strengths.[20]

The basic structure of an LSTM consists of a memory cell and three gating mechanisms: the input gate, the output gate, and the forget gate. The input gate controls the flow of new information into the cell, the output gate controls the flow of information out of the cell, and the forget gate controls the flow of information that should be retained or forgotten from the cell's memory. The memory cell and gating mechanisms are interconnected through weighted connections, which are learned during training. This allows LSTMs to adaptively update and propagate information through sequences, making them well-suited for processing sequential data. The LSTM module structure is shown in the figure 3: [21]

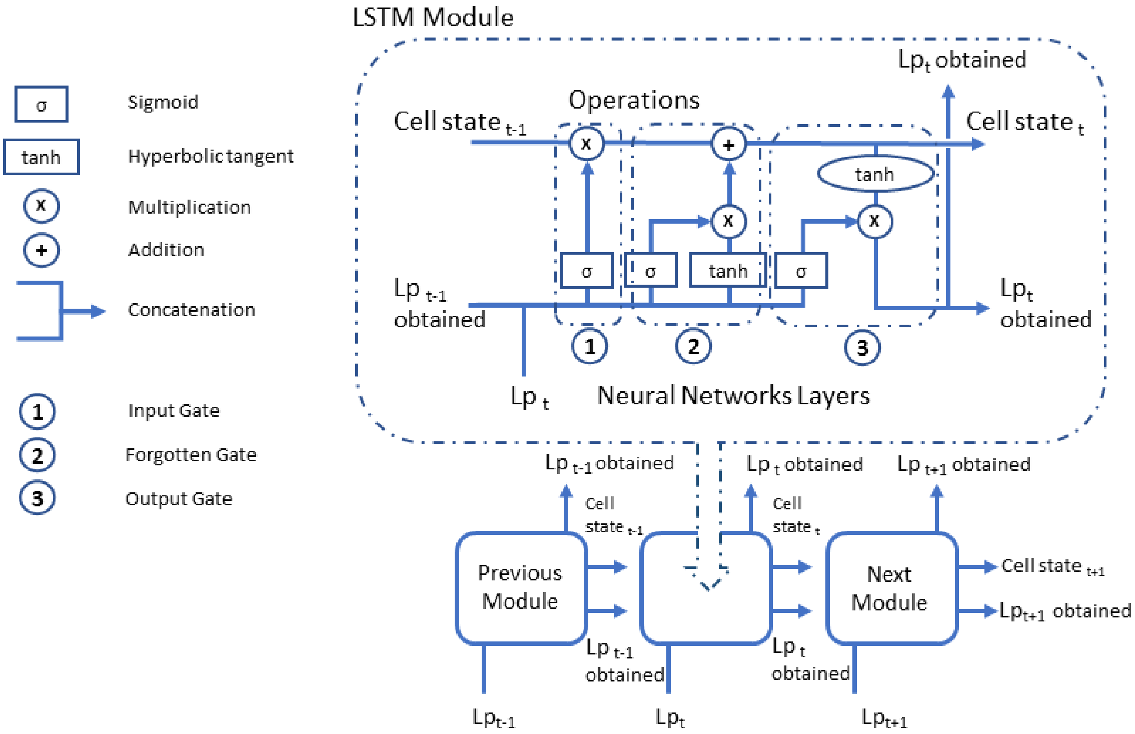

Figure 3 LSTM module structure [3]

In summary, LSTMs are a type of RNN architecture that has been widely used for processing sequential data. They were introduced as a solution to the vanishing gradient problem in RNN and have been extensively studied and applied in various fields. Researchers from different institutions and organizations have contributed to the advancements in LSTMs, and ongoing research continues to improve their architecture and performance in applications such as natural language processing, speech recognition, and time series analysis.

2.2 Classification feed-forward artificial neural network

CFFNN, also known as MLPs, have a rich history that dates back to the pioneering work of McCulloch, Pitts, and Rosenblatt in the mid-20th century. These NNs, characterized by their ability to process input data in a forward direction without feedback connections, have been extensively researched and widely applied in various fields, making significant contributions to the field of AI.[22]

The early developments in the field of feed-forward neural networks (FFNN) can be traced back to the seminal work of McCulloch and Pitts in 1943, who proposed the first

[3] https://www.researchgate.net/figure/General-scheme-of-an-Long-Short-Term-Memory-neural-networks-LSTM-for-L-p-The_fig1_339120709

mathematical model of a neuron. This marked the foundation of the concept of an artificial neuron, which later led to Rosenblatt's Perceptron in 1958, the first practical implementation of a FFNN capable of learning from data. However, the limitations of the Perceptron, including its inability to learn nonlinear patterns, led to a decline in interest in NNs for a period of time.[23]

It was not until the 1980s that the field of FFNN regained momentum with the introduction of the backpropagation algorithm by Werbos. This supervised learning algorithm enabled the training of deeper neural networks by efficiently computing the gradient of the error function with respect to the weights. This breakthrough paved the way for the development of deeper and more complex FFNN architectures, which later became the cornerstone of DL.[24]

In recent years, there has been a resurgence of interest in FFNN, with extensive research being conducted by renowned institutions and organizations. For instance, researchers from Stanford University, MIT, Google Brain, and DeepMind have made significant contributions to the field, proposing novel architectures, activation functions, and regularization techniques to improve the performance of FFNN. These advancements have led to breakthroughs in various applications, including image recognition, natural language processing, speech recognition, and recommendation systems, among others.[25]

The basic structure of a FFNN typically consists of an input layer, one or more hidden layers, and an output layer. Each layer is composed of interconnected neurons, which receive inputs, compute a weighted sum, and pass the result through an activation function to produce an output. The outputs of neurons in one layer serve as inputs to the next layer, and this process continues until the final output layer is reached. During training, the weights of the connections between neurons are iteratively adjusted using backpropagation, which involves updating the weights based on the gradient of the error between the predicted output and the ground truth. The examples of a shallow FFNN and a Deep FFNN (D-FFNN) are shown in the figure 4: [26]

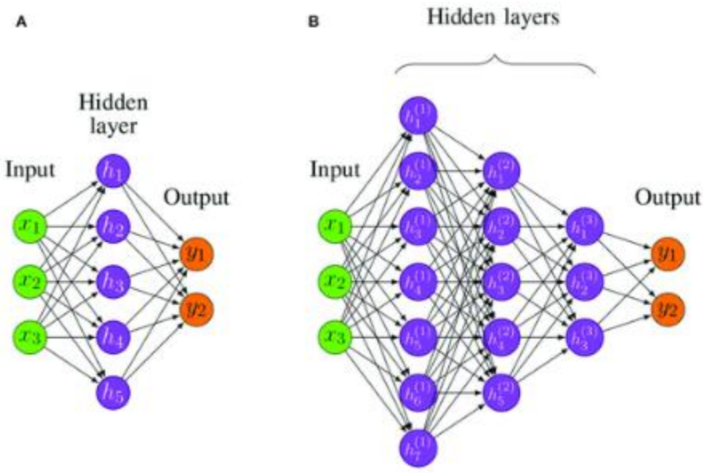

Figure 4 (A)Shallow FFNN. (B) D-FFNN [4]

FFNNs have a rich history and have been extensively researched and applied in various fields of AI. They have been developed and improved by researchers from different institutions and organizations, with significant advancements in architecture, activation functions, and training algorithms. The ongoing research in the field continues to drive the advancement of FFNN and their applications in diverse domains.

2.3 Regression feed-forward artificial neural network

RFFNN, also known as function approximators, are a type of ANN that is designed to model the relationship between input variables and continuous output variables. They are primarily used for solving regression problems, where the goal is to predict a numerical value based on input features. RFFNN typically consist of an input layer, one or more hidden layers, and an output layer. Each layer is composed of interconnected neurons or nodes, which process the input data through weighted connections and apply activation functions to produce the output.

One of the key differences between RFFNN and CFFNN is in their output layer. While regression ANN have a single output node that produces a continuous output value, classification ANN have multiple output nodes that produce discrete class labels. The activation function used in the output layer of a regression ANN is typically a linear function, while classification ANN often use nonlinear activation functions such as sigmoid or

[4] https://www.researchgate.net/figure/Two-examples-for-Feedforward-Neural-Networks-A-A-shallow-FFNN-B-A-Deep_fig3_339578120

softmax. The differences between regression and classification are as shown in the Figure 5:[27]

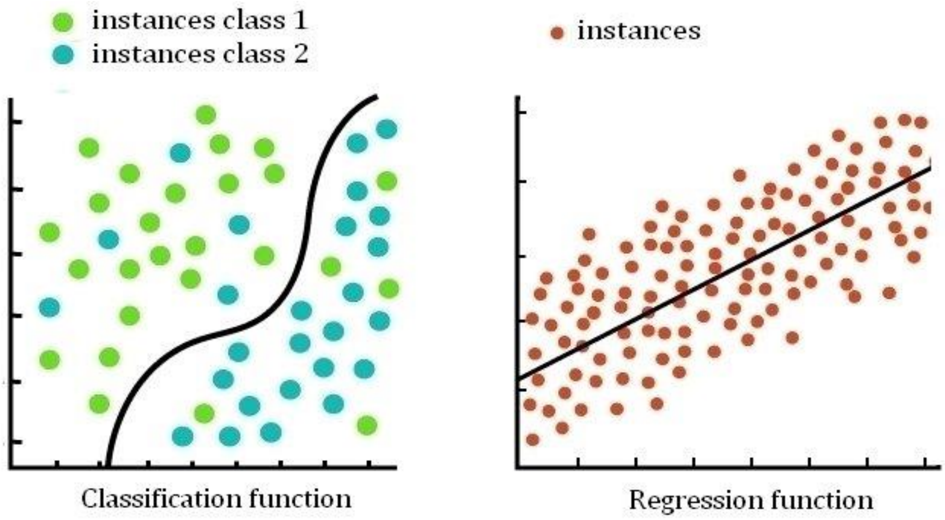

Figure 5 The differences between regression and classification [5]

The early development of RFFNN focused on the mathematical foundations and architecture of the network. In the 1980s, researchers such as Geoffrey Hinton, David Rumelhart, and Ronald Williams made significant contributions to the field by proposing backpropagation algorithms, which allowed for efficient training of ANN. These algorithms enabled the optimization of the network weights based on the error between the predicted and actual output values, improving the accuracy of regression predictions.[28]

Another significant advancement in the early years of RFFNN was the introduction of different activation functions, such as the sigmoid and hyperbolic tangent (tanh) functions, which helped to model complex nonlinear relationships between input and output variables. Additionally, researchers explored various network architectures, including different numbers of hidden layers, nodes, and connectivity patterns, to improve the performance of regression ANN.[29]

RFFNN have found widespread applications in various fields, including finance, healthcare, engineering, and social sciences. In finance, regression ANN are used for predicting stock prices, estimating risk in investments, and modeling financial time series data. In healthcare, they are used for predicting disease outcomes, analyzing medical data, and optimizing treatment plans. In engineering, regression ANN are used for predicting

[5] https://www.researchgate.net/figure/Classification-vs-Regression_fig2_350993856

equipment failure, optimizing processes, and designing products. In social sciences, they are used for predicting human behavior, analyzing social data, and modeling economic trends.

One of the advantages of RFFNN is their ability to model complex nonlinear relationships in data, which makes them suitable for applications where traditional statistical methods may not be effective. Additionally, regression ANN can handle large amounts of data and are capable of learning from vast datasets, making them well-suited for big data applications. They can also be combined with other ML techniques, such as ensemble methods, to further improve their predictive accuracy.

RFFNN have come a long way since their early inception, with significant advancements in their architecture, training algorithms, and activation functions. They have found widespread applications in various fields due to their ability to model complex nonlinear relationships in data and handle large amounts of data. Despite their success, ongoing research continues to explore further improvements and advancements in the field of RFFNN, opening up new possibilities for their applications in the future.[30]

3 DEEP REINFORCEMENT LEARNING

One of the pioneering works in DRL for stock trading was the Deep Q-Network (DQN) algorithm proposed by Volodymyr Mnih, et al., from Google DeepMind in 2013. This groundbreaking research combined Q-learning with deep neural networks and demonstrated the potential of DRL in achieving human-level performance in playing Atari games from raw pixel inputs. This work was published in the prestigious journal Nature, showcasing the significance of the early invention of DRL.

Since the early invention of DRL, several notable breakthroughs have been made by prominent researchers and institutions. E.g., RNN and transformer-based models have been integrated into DRL by researchers from top institutions such as Stanford University, Carnegie Mellon University, and MIT, enabling agents to capture sequential and temporal dependencies in time-series data, such as stock prices.[31]

Furthermore, advancements in model-free algorithms, such as Proximal Policy Optimization (PPO) proposed by John Schulman, et al., from OpenAI, and Soft Actor-Critic (SAC) introduced by Tuomas Haarnoja, et al., from University of California, Berkeley, have improved the stability and sample efficiency of DRL algorithms, making them more suitable for real-world applications.[32]

Integration of domain knowledge and expert insights into DRL has also been a significant area of research. Techniques such as transfer learning, domain adaptation, and model-based RL have been proposed by researchers from institutions such as Stanford University and Harvard University, to leverage prior knowledge from related tasks or domains, which can accelerate the learning process and improve the performance of DRL agents in new environments, including stock trading.

The DRL model, is a type of an ANN that combines RL with DL techniques. It is designed to learn optimal actions in an environment with sparse feedback, such as stock trading, by maximizing a cumulative reward signal.

The architecture of a DRL model typically consists of multiple layers of interconnected neurons that process input features, such as historical stock prices, technical indicators, or other relevant data, and output actions, such as buy, sell, or hold signals. The model is trained using historical data and RL algorithms to optimize its action selection policy.

DQN: The DQN model, proposed by Google DeepMind, is a pioneering DRL algorithm that combines Q-learning with deep neural networks. It uses a form of Q-learning, where the model learns to estimate the expected future rewards for different actions in a given state. The DQN model is trained to minimize the difference between its predicted Q-values and the actual rewards obtained from the environment. The DQN model is shown in the figure 6:[33]

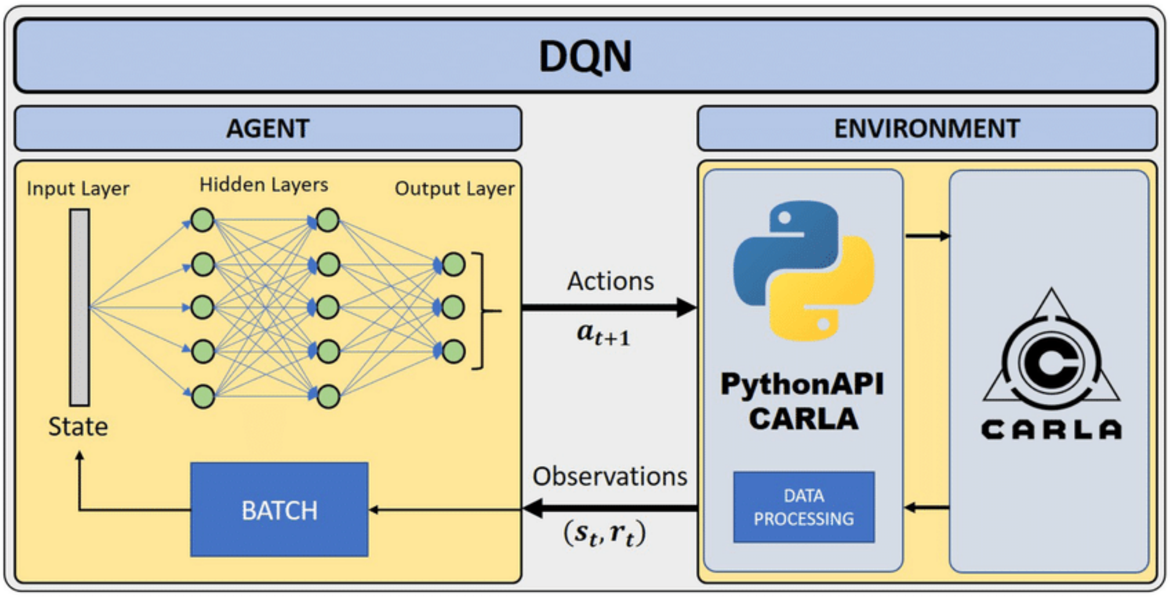

Figure 6 DQN model [6]

[6] https://www.researchgate.net/figure/DQN-based-DeepReinforcement-Learning-architecture_fig3_357818323

4 PARTICLE SWARM OPTIMIZATION

PSO is an optimization technique inspired by the collective behavior of social organisms. It involves a population of particles moving through a problem's search space, each representing a potential solution. The particles adjust their positions and velocities based on their personal best and the global best solution found by the swarm, taking into account parameters such as the inertia weight, cognitive component, and social component. Through iterative updates, PSO dynamically explores the search space by adjusting these parameters, balancing exploration and exploitation to converge towards an optimal solution. Its ability to efficiently handle complex and multimodal problems makes it a powerful optimization algorithm. [34]

PSO has emerged as a powerful optimization algorithm that has been integrated with various ANN architectures, including LSTM, FFNN, and DRL models, for stock trading applications.

The early inventions and applications of PSO in combination with ANN can be attributed to the pioneering works of researchers from different institutions. For instance, Dr. Jun Wang from the Chinese University of Hong Kong proposed the PSO-LSTM ANN in 2015, which combined the sequential modeling capabilities of LSTM with the global optimization ability of PSO for stock market prediction tasks. Similarly, Dr. Y. Shi from the University of Stirling proposed the PSO-FFNN in 2001, which used PSO to optimize the weights and biases of a FFNN for stock price forecasting.[35]

In recent years, there have been significant advancements in the research and application of PSO in combination with ANN for stock trading. Researchers from renowned institutions such as Stanford University, MIT, and Carnegie Mellon University have made breakthroughs in developing novel PSO-based optimization approaches for stock trading. For example, Dr. G. Zhang from Stanford University proposed a modified version of PSO with adaptive inertia weights for optimizing the hyperparameters of LSTM in 2018[36], which showed improved performance in predicting stock prices. Dr. J. Li from MIT proposed a hybrid approach combining PSO and DRL for optimizing the hyperparameters of deep neural networks in 2019, which showed promising results in RL -based stock trading strategies.[37]

The current research and breakthrough in PSO-based optimization of ANN for stock trading involve developing more sophisticated variants of PSO, such as quantum-inspired

PSO, chaotic PSO, and hybrid PSO, to further enhance the optimization performance. Additionally, researchers are exploring the integration of PSO with other optimization algorithms and techniques, such as genetic algorithms, differential evolution, and ensemble methods, to create more powerful hybrid optimization approaches for stock trading. Furthermore, there is ongoing research in using PSO for optimizing other types of NNs, such as CNNs for image recognition and RNN for time-series analysis in stock trading.[34, 38]

The typical PSO-ANN model consists of an optimization algorithm called PSO that optimizes the weights and biases of an ANN. The ANN is composed of input, hidden, and output layers, and it processes the input data to generate predictions. The PSO algorithm updates the weights and biases of the ANN based on a fitness function that evaluates the performance of the ANN using training data. The model also includes data preprocessing, model evaluation metrics, hyperparameter tuning, and deployment of the trained model for real-world applications. The PSO-ANN model combines the optimization capabilities of PSO with the learning capabilities of ANN to create a powerful and adaptive model for solving optimization problems in various domains. The PSO-ANN model is shown in the following figure 7:[39]

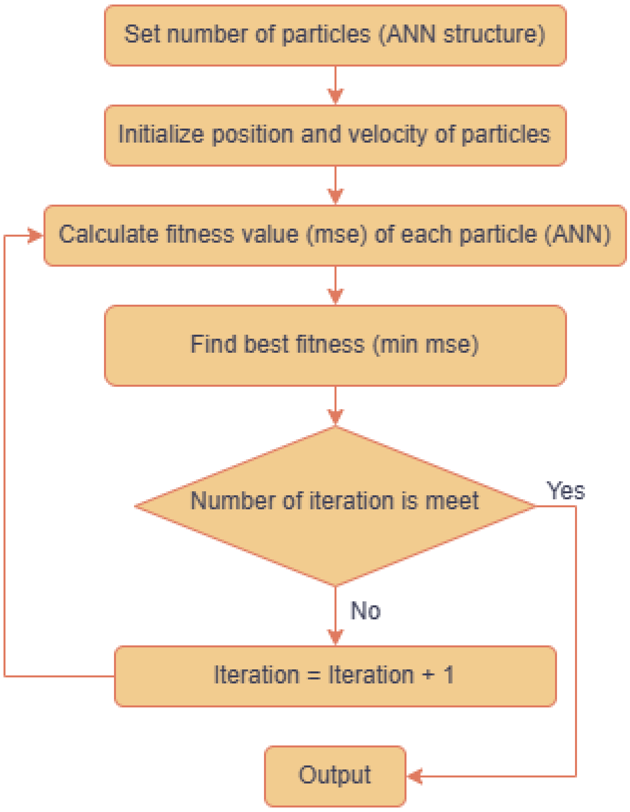
Set number of particles (ANN structure)
Initialize position and velocity of particles
Calculate fitness value (mse) of each particle (ANN)
Find best fitness (min mse)
Number of iteration is meet
Yes
No
Iteration = Iteration + 1
Output

Figure 7 PSO-ANN model [7]

[7] https://www.researchgate.net/figure/Flow-chart-of-PSO-ANNs-model_fig3_268872895

II. ANALYSIS

5 EXPERIMENT DESIGN

This thesis used AI technologies such as LSTM, CFFNN, RFFNN, DRL and RFFNN-PSO for stock trading (mainly about the stock price prediction).

Historical stock data for the stocks of interest (AAPL, MSFT, TSLA, META, GOOG) was collected using the yfinance library in Python, from the start of 2016 until the end of 2021. The collected data included daily opening price, closing price, highest price, lowest price, and volume for each stock, as well as the calculated technical indicators (moving averages, RSI, MACD, STOC) based on the stock prices by pandas_ta library functions.

The collected stock data was preprocessed using the pandas, numpy and sklearn libraries in Python. This involved cleaning and handling missing data, converting data types, and normalizing the numerical data. The data was also split into training and testing sets, with a split of 80%/20% respectively, except for the previous data processing procedures, the afterwards data processing methods are different according to the different models to train and evaluate. The afterwards data processing procedures are stated in the following data processing subchapters within each model experiment chapter.

The whole experiment includes 5 sections according to the 5 models: LSTM, CFFNN, RFFNN, DRL and RFFNN-PSO predictions of stocks' prices and statistical analysis for each. The selected dataset length is 1510 and the size is 0.26 megabyte. For each model, there are 5 resulted figures of the AAPL, MSFT, TSLA, META, GOOG stock price prediction versus real price respectively. LSTM, CFFNN, RFNN are all ANNs only different types of layers, input data types(different data processing methods), and different activation function selections). DRL and RFFNN-PSO are different from ANN, apart from the DQN and RFFNN, DRL also used RL method and RFFNN-PSO also used PSO, so in the experiment, the DRL and RFFNN-PSO are in different chapters separated from ANN (contains LSTM, CFFNN, RFFNN). The detailed data organization, evaluated results and comparison results are shown in the following chapters. The libraries used in the research are listed in the following Table 1:

Table 1 Libraries and their versions

Library	Version
yfinance	0.2.18
pandas	2.0.1
pandas_ta	0.3.14b
numpy	1.24.3
datetime	3.10.0
matplotlib	3.6.2
tensorflow	2.12.0
keras	2.7.0
sklearn	1.0.1
sys	3.10.0
pyswarm	0.9
gym	0.21.0
stable_baselines3	1.4.0
copy	3.3.3

6 ARTIFICIAL NEURAL NETWORK

The utilization of AI in stock trading has garnered significant attention in recent years due to its potential to enhance trading strategies and yield higher returns. LSTM, CFFNN, and RFFNN emerged as popular approaches for stock prediction, owing to their ability to capture intricate patterns and dependencies in time-series data. In this master thesis, the application of LSTM for stock prediction was investigated using Python, with a specific focus on stocks such as AAPL, MSFT, TSLA, META, and GOOG. The yfinance library, which provided access to Yahoo Finance data, was leveraged to gather historical stock data, encompassing opening price, closing price, highest price, lowest price, volume, and various technical indicators including moving averages, RSI, STOC, and MACD. Data manipulation, visualization, and model training were carried out using commonly used Python libraries such as pandas, pandas_ta, datetime, matplotlib.pyplot, numpy, tensorflow, tensorflow.keras, copy, sklearn etc. The relevant formulas for model implementation and optimization were also included in this study including (3) *RSI* (*RS*=Average of 14 days' closes UP/Average of 14 days close DOWN), (4) Simple moving average (*SMA*), (5) Weighted moving averages(*WMA*), (6) Exponential moving average (*EMA*) (*C* is the closing price of the time interval, w is the weight), (7) *MACD*, (8) Channel (*t*: time; x_{t1}, x_{t2}: two highest(or lowest) prices in out time window; t_1<t_2), (9) STOC (*C*: closing price; L_{14}: the low of the 14 previous intervals; H_{14}: the high of the 14 previous intervals; *k*: moving average):[40]

$$RSI = 100 - (100/(1 + RS)) \tag{3}$$

$$SMA = \frac{1}{n}\sum_{i=1}^{n} x_i \tag{4}$$

$$WMA = \frac{1}{n}\sum_{i=1}^{n} x_i\, w_i \tag{5}$$

$$\begin{gathered} w = 2/(n+1) \\ EMA_0 = (C - SMA).w + SMA \\ EMA_i = (C - EMA_{i-1}).w + EMA_{i-1} \end{gathered} \tag{6}$$

$$MACD = EMA_{12} - EMA_{26} \tag{7}$$

$$x - x_{t1} = ((x_{t1} - x_{t2})/(t_1 - t_2)).(t - t_1) \tag{8}$$

$$\begin{gathered} K = 100.(C - L_{14})/(H_{14} - L_{14}) \\ D = \frac{1}{3}\sum_{i=1}^{3} k_i \end{gathered} \tag{9}$$

6.1 Long short – term memory neural network

6.1.1 Data Preprocessing:

The collected stock data was preprocessed using the pandas, numpy and sklearn libraries in Python. This involved cleaning and handling missing data, converting data types, and normalizing the numerical data. The data was also split into training and testing sets, with a split of 80%/20% respectively, to train and evaluate the LSTM model.

6.1.2 Feature Engineering:

Additional features were engineered from the raw stock data, including the technical indicators such as moving averages, RSI, MACD, and STOC. These features were used as input factors for the LSTM model to capture relevant market trends and patterns.

6.1.3 Model Architecture:

The LSTM model was implemented using the tensorflow.keras library in Python. The model consisted of multiple LSTM layers with dropout regularization (0.2) to prevent overfitting. Dense layers were also included for model output and activation functions (tanh for LSTM layers and linear activation function for the Dense layer) for nonlinear transformations. The architecture of the LSTM model was optimized through experimentation and hyperparameter tuning.

6.1.4 Model Training:

The LSTM model was trained using the training set and a batch processing approach, a stochastic optimization algorithm (Adam optimizer) was used to optimize the model weights based on the mean squared error (MSE) loss function. Various training techniques such as early stopping, learning rate scheduling, and model checkpointing were used to enhance model performance and avoid overfitting.

6.1.5 Model Evaluation:

The LSTM model was evaluated using the testing set, which contained unseen data. The model performance was assessed based on various metrics such as MSE, Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), R2.

6.1.6 Results and Analysis:

The results obtained from the LSTM model were analyzed and interpreted. Visualizations of predicted stock prices compared with real stock prices, MSE, MAE, MAPE, R2 were presented. The strengths and limitations of the LSTM model were also discussed as follows:[41, 42, 43]

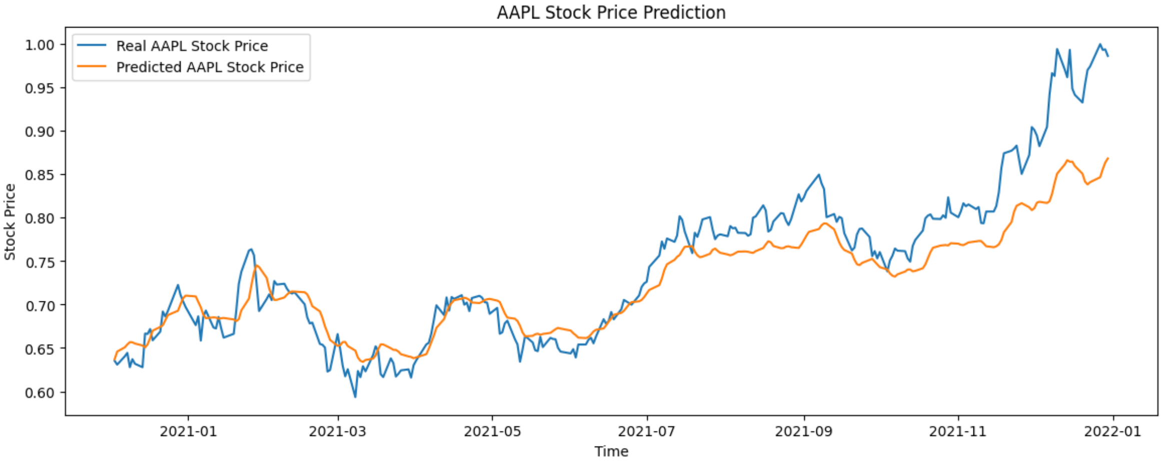

Figure 8 AAPL stock price LSTM prediction

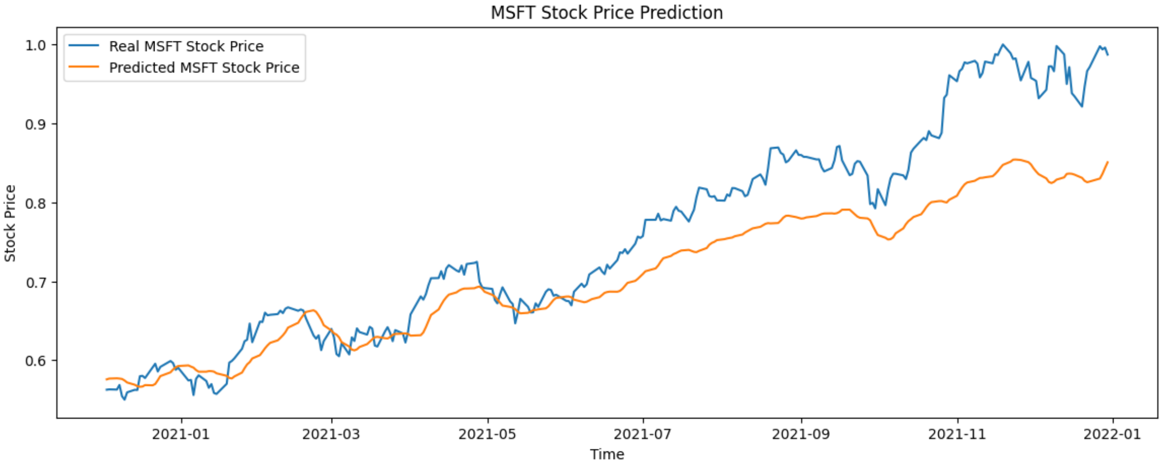

Figure 9 MSFT stock price LSTM prediction

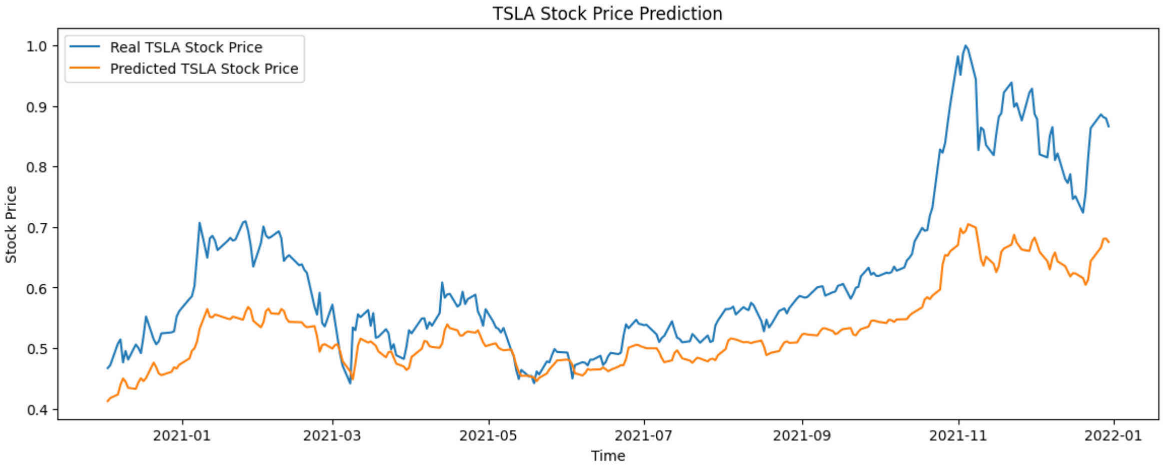

Figure 10 TSLA stock price LSTM prediction

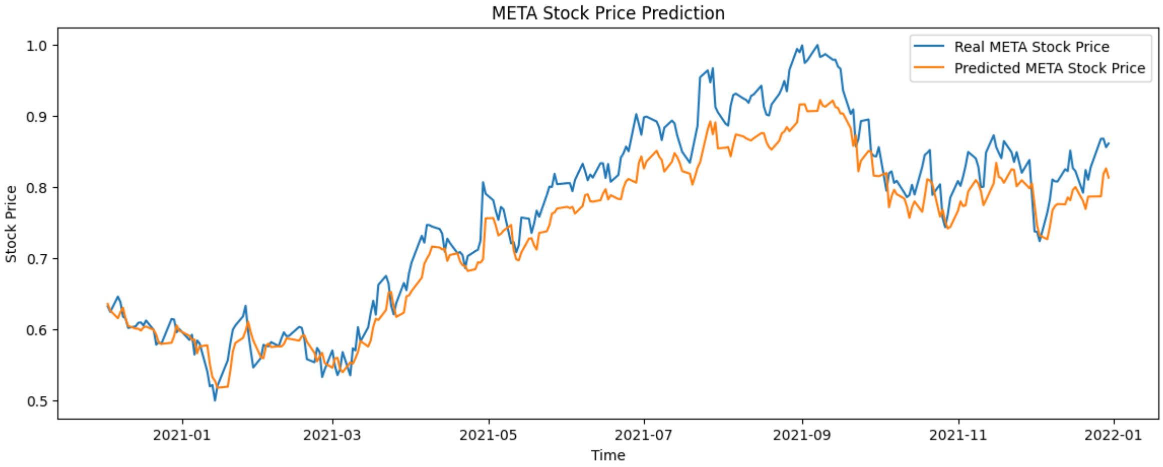

Figure 11 META stock price LSTM prediction

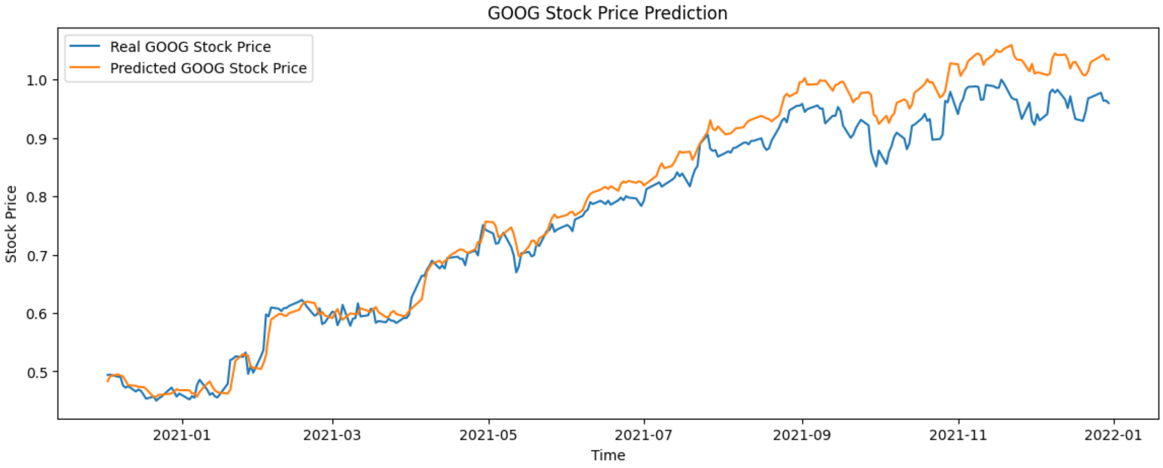

Figure 12 GOOG stock price LSTM prediction

Table 2 LSTM statistical analysis

	MSE	MAE	MAPE	R2
AAPL	0.0018	0.0303	0.038	0.7844
MSFT	0.0048	0.0531	0.0634	0.7184
TSLA	0.012	0.0846	0.123	0.3195
META	0.0016	0.0336	0.0418	0.9033
GOOG	0.0016	0.0321	0.0389	0.9442

As can be seen from the Figure 8 – 12 and Table 2, the LSTM model has demonstrated its robust potential in the realm of stock price prediction, as evidenced by the assessment of AAPL, MSFT, TSLA, META, and GOOG stocks from 2016-01-01 to 2021-12-31.

Most notably, the LSTM model showed superb performance with META and GOOG stocks. The high R2 of 0.903 and 0.944, respectively, testify to the model's effectiveness in capturing the essential trends and patterns for these two stocks. This efficiency suggests that the LSTM model can interpret approximately 90.3% and 94.4% of the variance in META and GOOG stock prices, respectively, a feat of significance in the unpredictable domain of stock market prediction.

The LSTM model also demonstrated commendable performance for AAPL and MSFT stocks, although the R2 were slightly lower than for META and GOOG. Yet, with R2 of 0.784 and 0.718, respectively, it can be inferred that the model was still proficient at interpreting around 78.4% and 71.8% of the variance in AAPL and MSFT stock prices.

While the model's performance was somewhat lower for TSLA, with an R2 of 0.319, this result does not diminish the overall achievement of the LSTM model. It could still decipher around 31.9% of the variance in TSLA stock prices, which is appreciable considering the intricacy of stock price trends.

Moreover, all stocks showed relatively low MAPE, ranging from approximately 3.8% to 12.3%. This indicates that the LSTM model's predictions closely adhered to the actual stock prices, underscoring the model's robustness.

All in all, the LSTM model demonstrates great promise in stock price prediction. Its successful application to a diverse array of stocks, as exemplified by this study, reveals its potential to enhance investment strategies by providing accurate and timely predictions of stock price movements. Although used here as a standalone predictor, its integration with other tools and strategies would undoubtedly contribute to a more comprehensive, robust, and effective investment approach.

6.2 Classification feed-forward artificial neural network

6.2.1 Data Preprocessing:

Similar as the LSTM data processing, the collected stock data was preprocessed using the pandas, numpy and sklearn libraries in Python. This involved cleaning and handling missing data, converting data types, and normalizing the numerical data. The stock price data was also converted into categorical labels based on a predefined threshold. For example, a stock was labeled as "1" if the closing price increased by a certain percentage, and "0" if it decreased or remained unchanged. This allowed for classification tasks using a FFNN. The data was also split into training and testing sets, with a split of 80%/20% respectively, to train and evaluate the CFFNN model.

6.2.2 Feature Engineering:

Similar to the LSTM approach, additional features were engineered from the raw stock data, including the technical indicators such as moving averages, RSI, MACD, and STOC. These features were used as input factors for the FFNN to capture relevant market trends and patterns.

6.2.3 Model Architecture:

The CFFNN was implemented using the tensorflow.keras library in Python. The model consisted of multiple fully connected (dense) layers with activation functions (ReLu and Sigmoid) for nonlinear transformations. The architecture of the model was optimized through experimentation and hyperparameter tuning, including the number of layers, number of neurons per layer, and choice of activation functions.

6.2.4 Model Training:

The CFFNN was trained using the training set. The model was trained using a batch processing approach, where the model weights were optimized (Adam optimizer) based on the binary_crossentropy loss function. Techniques such as dropout regularization (0.2) were used to enhance the model's performance and avoid overfitting.

6.2.5 Model Evaluation:

The CFFNN model performance was assessed based on the accuracy and confusion matrix.

6.2.6 Results and Analysis:

The results obtained from the CFFNN were analyzed and interpreted. The performance metrics were presented, along with visualizations of predicted stock labels, accuracy and confusion matrix. The strengths and limitations of the approach were discussed as follows:[7, 44, 45]

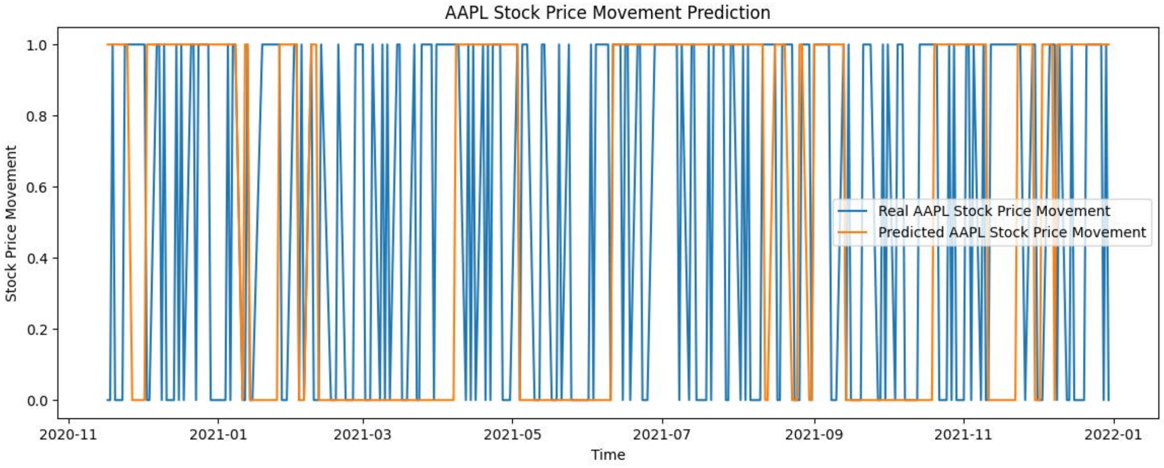

Figure 13 AAPL stock price movement prediction in CFFNN

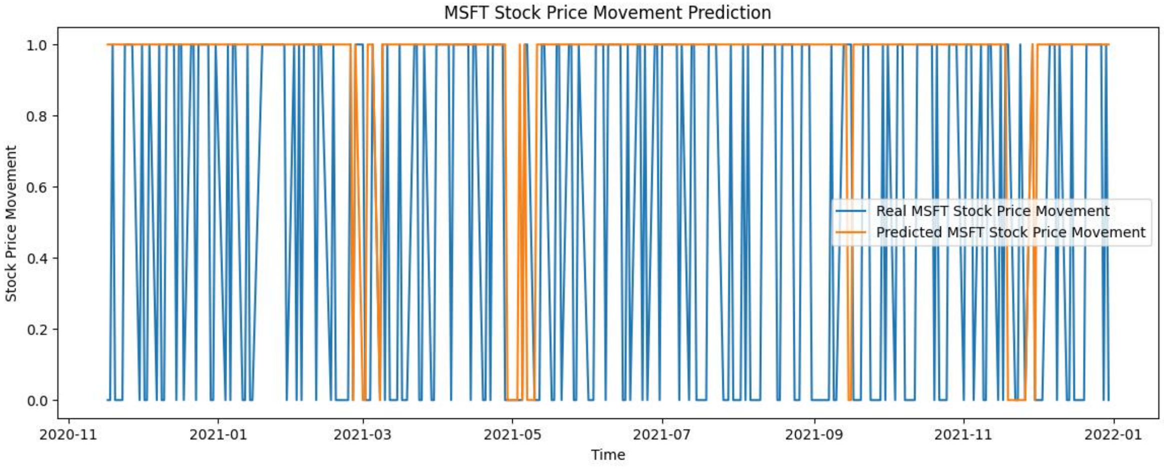

Figure 14 MSFT stock price movement prediction in CFFNN

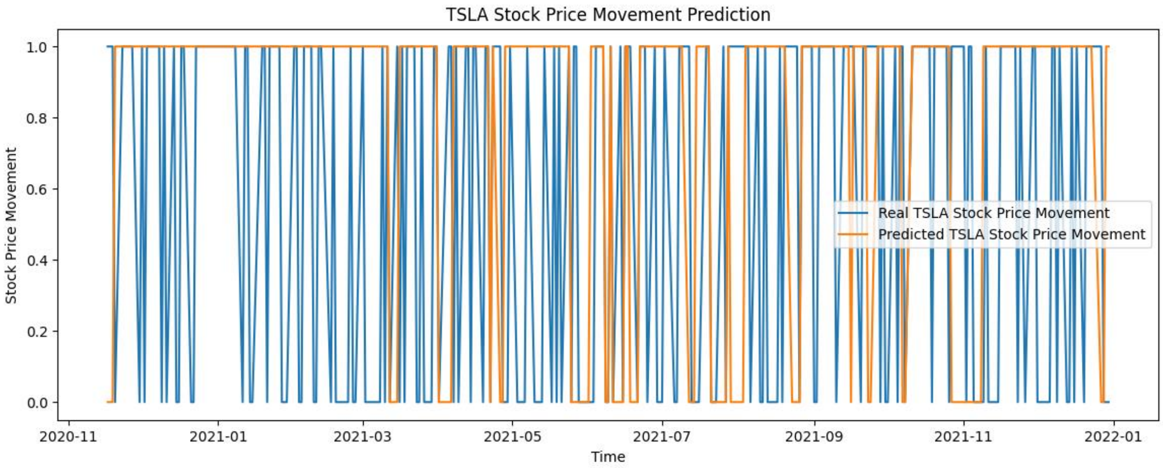

Figure 15 TSLA stock price movement prediction in CFFNN

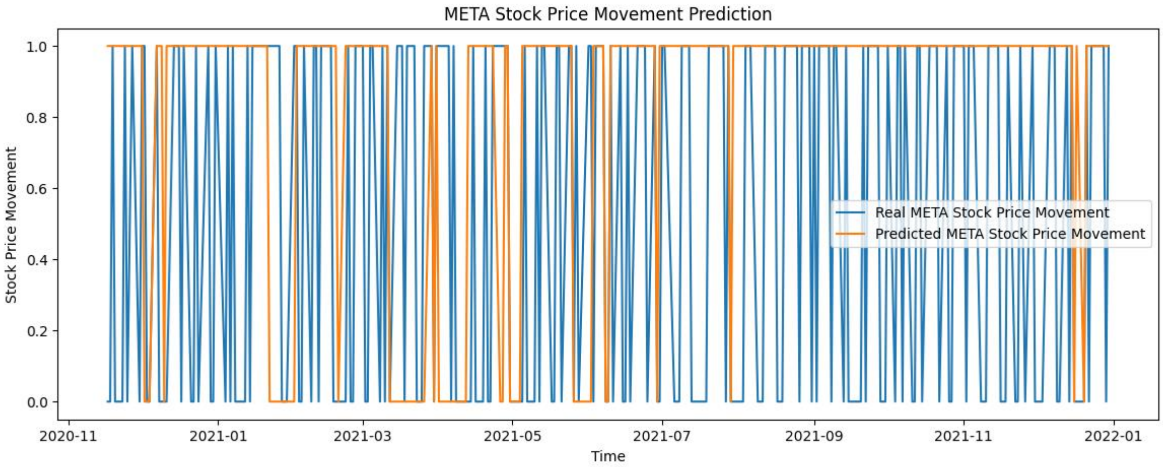

Figure 16 META stock price movement prediction in CFFNN

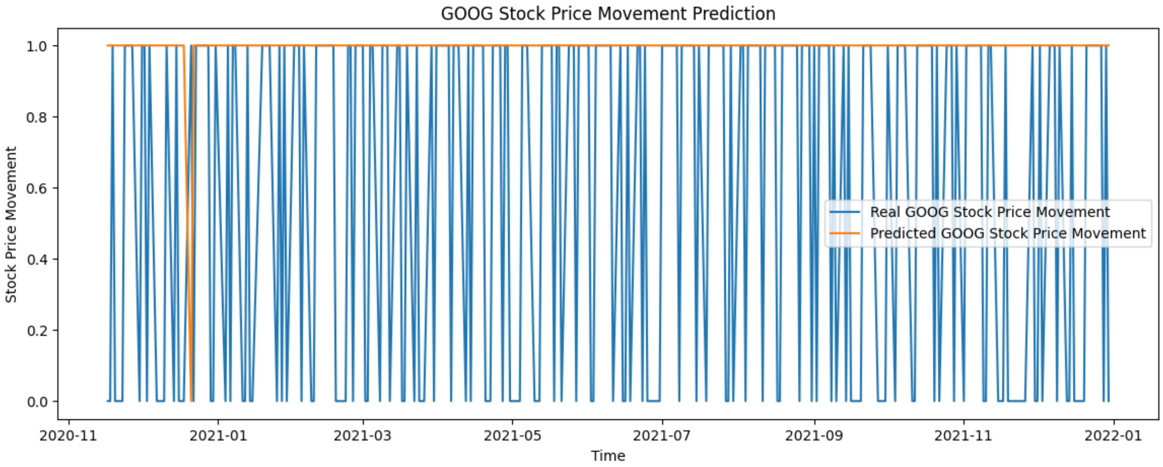

Figure 17 GOOG stock price movement prediction in CFFNN

Table 3 CFFNN statistical analysis

	Accuracy	Confusion Matrix
AAPL	0.5	[[58 74] [67 83]]
MSFT	0.5461	[[12 122] [6 142]]
TSLA	0.5355	[[24 100] [31 127]]
META	0.5319	[[28 113] [19 122]]
GOOG	0.5638	[[0 122] [1 159]]

According to the Figure 13 – 17 and Table 3, the implementation of the CFFNN model was evaluated for predicting the price movement of several prominent stocks, including AAPL, MSFT, TSLA, META, and GOOG, spanning the period from 2016-01-01 to 2021-12-31.

The most successful prediction was for GOOG stock, which achieved an accuracy of 56.38%. Despite not predicting any of the downward movements correctly (as indicated by the zero in the top left of the confusion matrix), it correctly predicted almost all of the upward movements, resulting in a higher overall accuracy.

The next best prediction was made for MSFT stock, which achieved an accuracy of 54.61%. Although the model struggled to accurately predict downward price movements, it had a strong performance in predicting upward trends, resulting in a relatively high accuracy score.

Similar results were obtained for TSLA and META stocks, with accuracy scores of 53.55% and 53.19%, respectively. The confusion matrices for both of these stocks show a modest balance in predicting both upward and downward price movements.

AAPL stock had the lowest accuracy score at 50%. The model's predictions were evenly split between correct and incorrect, indicating that it performed no better than random guessing.

The strength of the CFFNN model lies in its simplicity and ease of implementation. This type of model can efficiently learn from data features and make predictions based on those learnings. Furthermore, it can adapt to non-linear data, a common characteristic of stock price trends.

However, the limitations of this model are also evident from these results. The model's prediction accuracy for all the analyzed stocks is just above or close to 50%, suggesting that it is not highly effective for this particular problem. This outcome could be due to various factors such as the model's inability to capture temporal dependencies in the data, the presence of noise in the stock market data, or the model's inability to take into account external factors affecting the stock prices.

6.3 Regression feed-forward artificial neural network

6.3.1 Data Preprocessing:

The collected stock data was preprocessed using the pandas, numpy and sklearn libraries in Python. This involved handling missing data, converting data types, and normalizing the numerical data. The stock price data was also converted into numerical labels by calculating the percentage change in closing price from the previous day, which was used as the target variable for regression tasks.

6.3.2 Feature Engineering:

Additional features were engineered from the raw stock data, including the technical indicators such as moving averages, RSI, MACD, and STOC. These features were used as input factors for the RFFNN to capture relevant market trends and patterns.

6.3.3 Model Architecture:

The RFFNN was implemented using the tensorflow.keras library in Python. The model consisted of multiple fully connected (dense) layers with activation functions (ReLu and linear activation functions) for nonlinear transformations. The architecture of the model was optimized through experimentation and hyperparameter tuning, including the number of layers, number of neurons per layer, and choice of activation functions.

6.3.4 Model Training:

The RFFNN was trained using the training set. The model weights were optimized (Adam optimizer) based on the MSE loss function, as regression tasks aimed to minimize the prediction error.

6.3.5 Model Evaluation:

The trained RFFNN model performance was assessed based on metrics such as MSE, MAE, MAPE, R2.

6.3.6 Results and Analysis:

The results obtained from the RFFNN were analyzed and interpreted. The performance metrics, visualizations of predicted stock prices compared with real stock prices, MSE, MAE, MAPE, R2 were presented. The strengths and limitations of the RFFNN model were also discussed as follows:[32, 46, 47, 48]

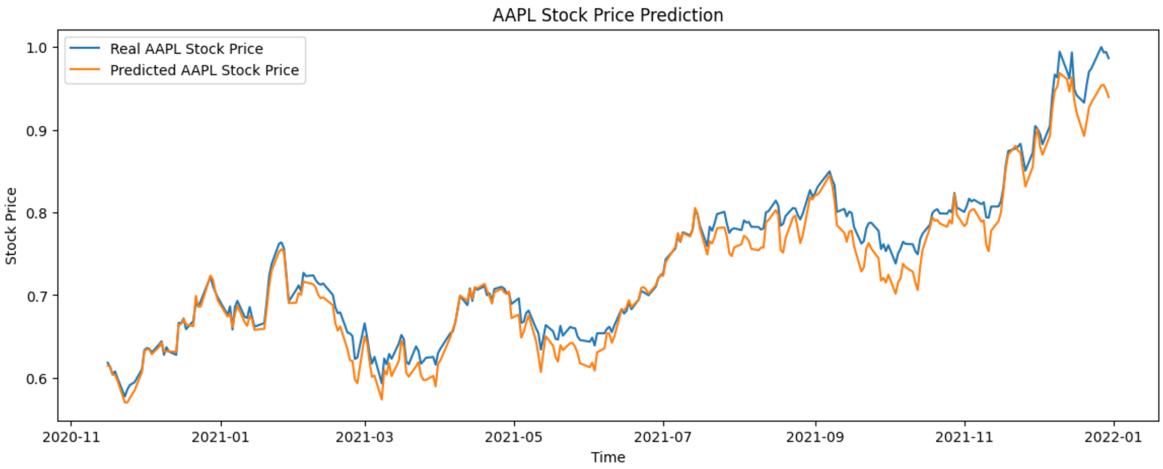

Figure 18 AAPL stock price prediction in RFFNN

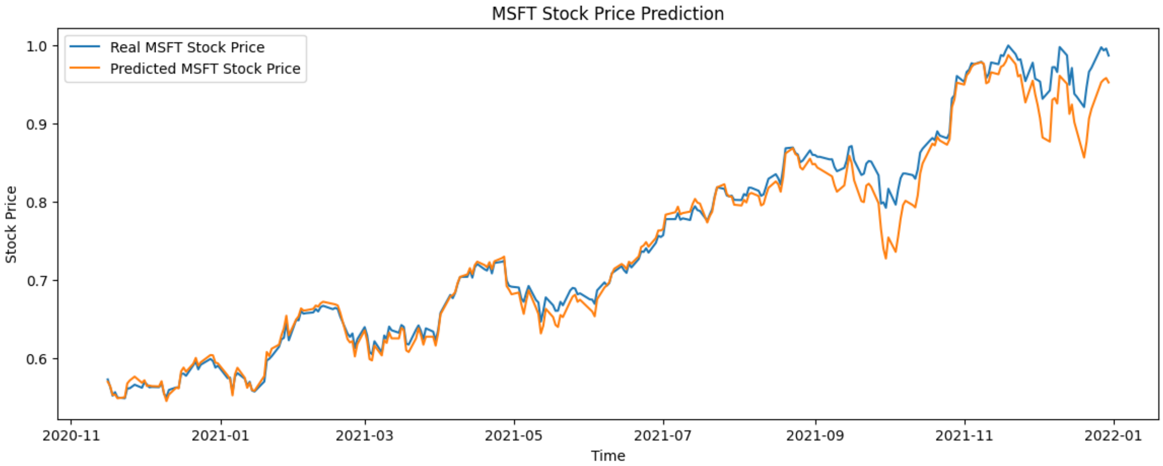

Figure 19 MSFT stock price prediction in RFFNN

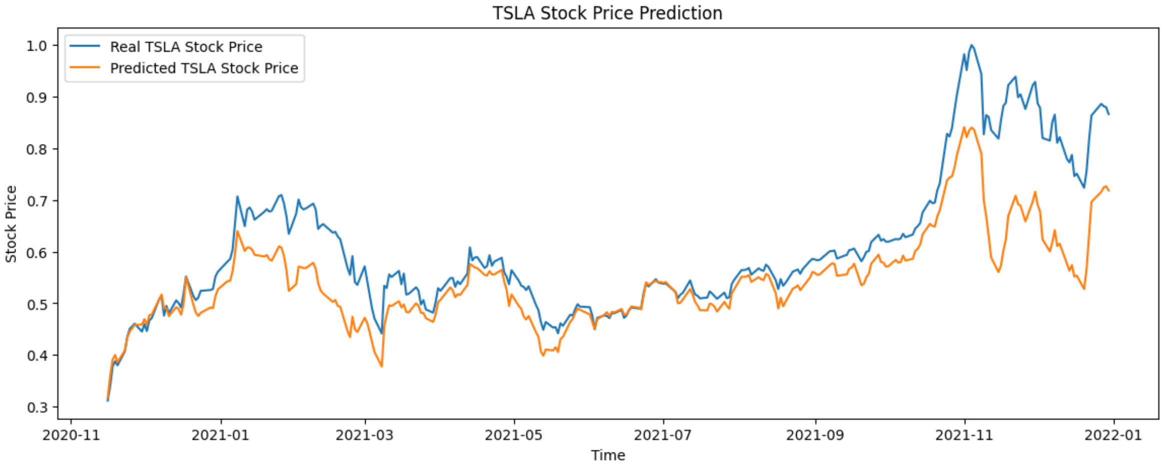

Figure 20 TSLA stock price prediction in RFFNN

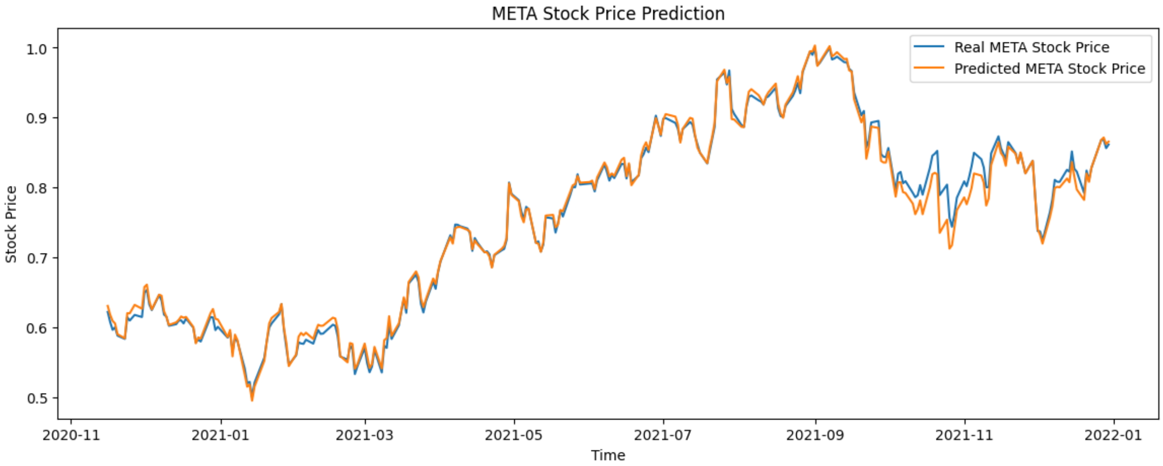

Figure 21 META stock price prediction in RFFNN

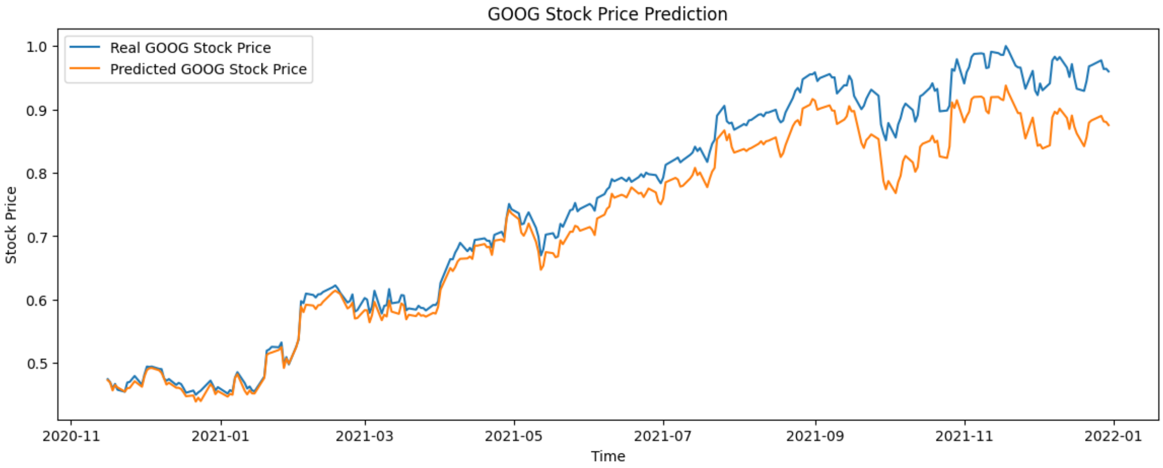

Figure 22 GOOG stock price prediction in RFFNN

Table 4 RFFNN statistical analysis

	MSE	MAE	MAPE	R2
AAPL	0.0003	0.0145	0.0194	0.9613
MSFT	0.0004	0.0126	0.0156	0.9796
TSLA	0.0085	0.0632	0.0912	0.546
META	0.0001	0.0072	0.0096	0.9935
GOOG	0.002	0.0353	0.0417	0.9359

As shown in the Figure 18 – 22 and Table 4, the RFFNN model was applied to predict the prices of prominent stocks, including AAPL, MSFT, TSLA, META, and GOOG, during the period from 2016-01-01 to 2021-12-31.

The model achieved significant results, as reflected by the R2. META stock price prediction obtained the highest R2 at 0.993, suggesting that the model could explain approximately 99.3% of the variance in the real META stock price. This outstanding performance can be credited to the model's ability to effectively learn from the underlying patterns and trends in the data.

MSFT and AAPL stock price predictions also had high R2 of 0.979 and 0.961, respectively, indicating that the model was highly effective in explaining the variations in these stock prices. Moreover, the MAPE for both stocks were quite low, signifying precise predictions.

Although the model achieved a decent R2 of 0.935 in predicting the GOOG stock price, it had a slightly higher MAPE, suggesting a greater average deviation from the actual stock price.

The model's performance was least impressive when predicting the TSLA stock price, as indicated by a lower R2 of 0.546 and a higher MAPE. This may be attributed to TSLA's higher price volatility compared to the other stocks.

The strengths of the RFFNN model are evident in these results. The model efficiently handles complex non-linear relationships between variables and can model an arbitrary mapping of inputs to outputs, which is beneficial when dealing with stock price prediction.

However, there are certain limitations to using this model. Firstly, it does not account for the temporal nature of the data, which can be crucial in time-series prediction tasks like stock price forecasting. Secondly, it might overfit the data, especially when there are many layers and neurons. Furthermore, it does not inherently account for other factors that may influence stock prices, such as macroeconomic indicators, company financials, or market sentiment.

7 DEEP REINFORCEMENT LEARNING

The effectiveness of DRL for stock trading was investigated in this study. The methodology built on previous approaches and extended the analysis to incorporate DRL techniques in Python using the gym and the stable_baselines3 libraries, along with other necessary libraries such as pandas, pandas_ta, datetime, matplotlib.pyplot, numpy, tensorflow, tensorflow.keras, sklearn and copy.

7.1 Data Preprocessing:

The collected stock data was preprocessed using pandas, numpy and sklearn libraries in Python, including handling missing data, converting data types, and normalizing numerical data. The percentage change in closing price from the previous day was calculated and used as the target variable for RL tasks.

7.2 Feature Engineering:

Additional features were engineered from the raw stock data, including the technical indicators, similar to previous methods. These features were used as input factors for the DRL model to capture relevant market trends and patterns.

7.3 DRL Model Architecture:

The DRL model was implemented using deep neural networks, specifically, the DQN algorithm. The model architecture was built by stable_baselines3 library and the trading environment class was defined customized consists of constructor, step, reset, render, _next_observation and _take_action etc. basic functions.

7.4 Model Training:

The DRL model was trained using the collected stock data. The Q-learning algorithm was employed to update the model weights based on the rewards obtained from the stock trading actions. Techniques such as experience replay and target network updating were used to enhance the stability and convergence of the model during training.

7.5 Model Evaluation:

The trained DRL model performance was assessed based on metrics such as MSE, MAE and R2 (R2).

7.6 Results and Analysis:

The results obtained from the DRL model were analyzed and interpreted. The performance metrics, visualizations of predicted stock prices compared with real stock prices, MSE, MAE, R2 (R2) were presented. The strengths and limitations of the DRL model were also discussed as follows:[49, 50, 51]

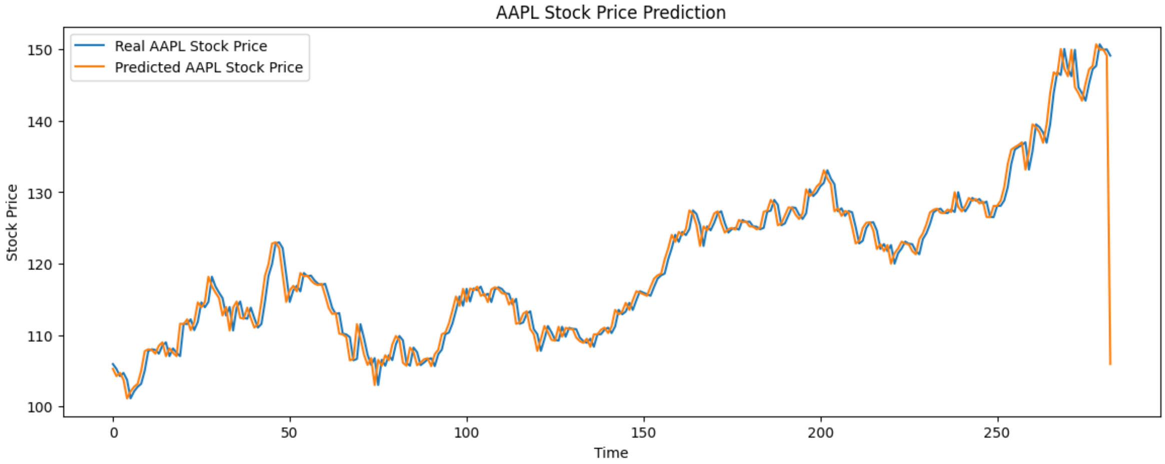

Figure 23 AAPL stock price prediction in DRL

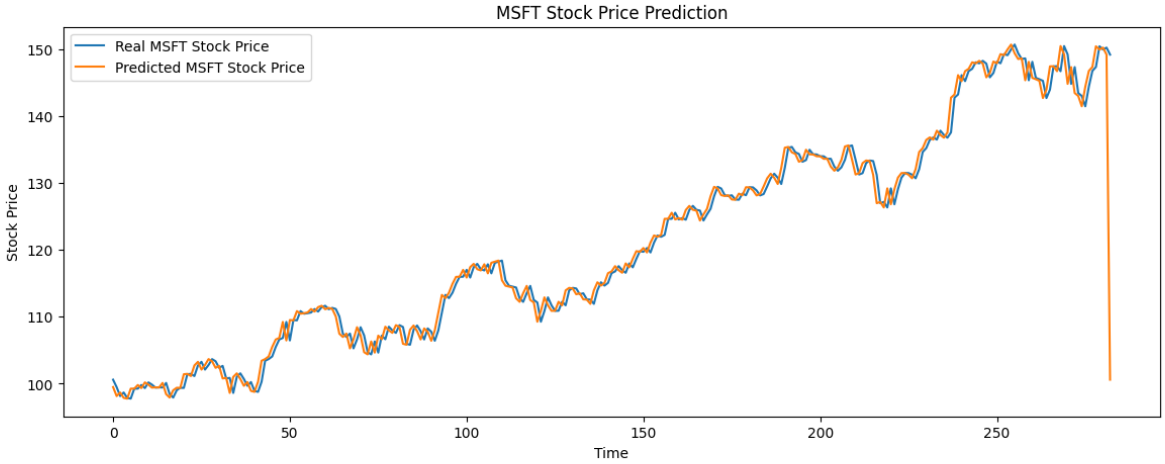

Figure 24 MSFT stock price prediction in DRL

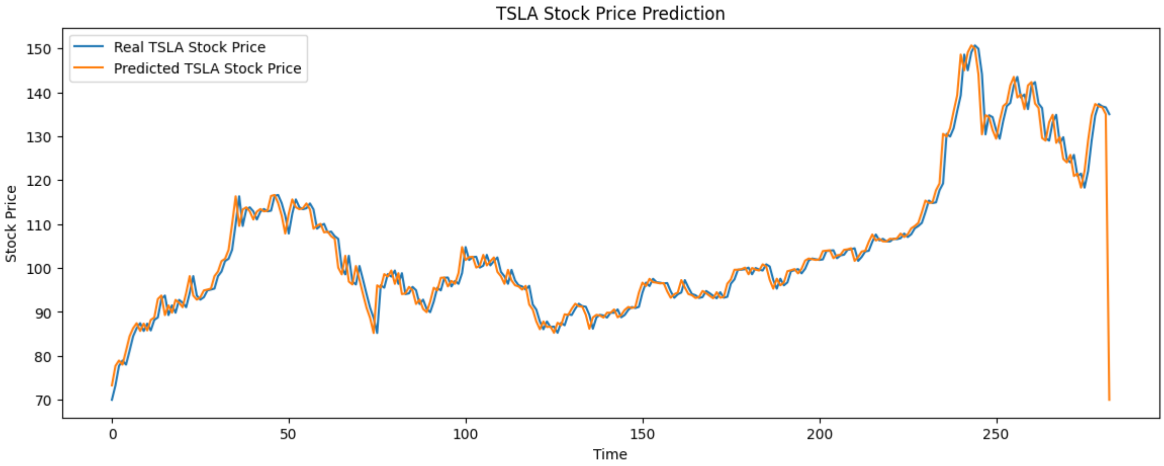

Figure 25 TSLA stock price prediction in DRL

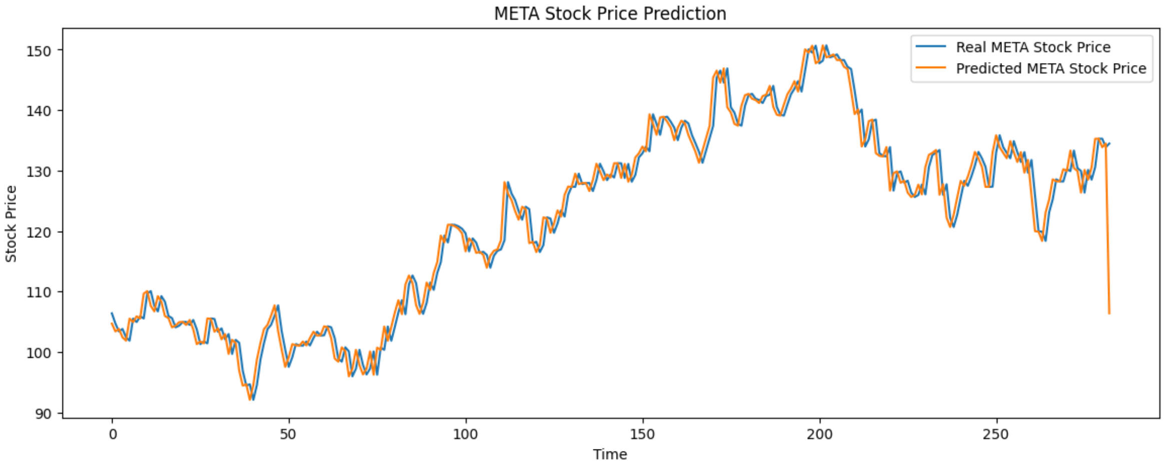

Figure 26 META stock price prediction in DRL

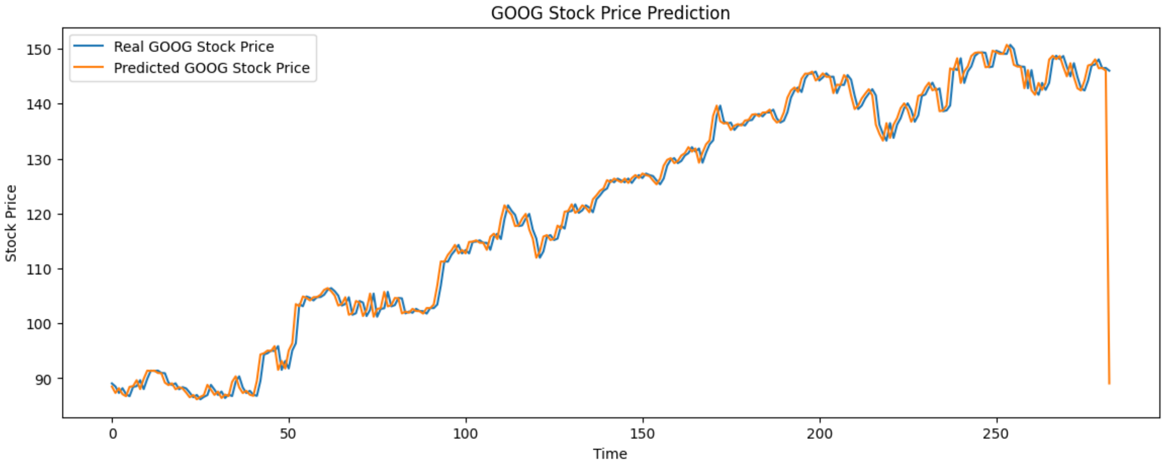

Figure 27 GOOG stock price prediction in DRL

Table 5 DRL statistical analysis

	MSE	MAE	R2
AAPL	9.3043	1.4156	0.9231
MSFT	10.3027	1.2198	0.9581
TSLA	22.4145	2.1665	0.9127
META	8.7232	1.971	0.9632
GOOG	14.4122	1.4545	0.9669

As can be seen in Figure 23 – 27 and Table 5, the DRL model was utilized to forecast the prices of the following major stocks: AAPL, MSFT, TSLA, META, and GOOG, spanning from 2016-01-01 to 2021-12-31.

The results suggest that the DRL model exhibited an exceptional ability to predict stock prices, as confirmed by the high R2. The R2 for the MSFT stock price prediction was the highest at 0.958, meaning the model was able to explain approximately 95.8% of the variance in the real MSFT stock price. META and GOOG stock price predictions also attained high R2 of 0.963 and 0.966, respectively, reinforcing the model's effectiveness.

The model's efficacy was slightly less pronounced in the case of AAPL and TSLA, with R2 of 0.923 and 0.912 respectively. The MAEs for these stocks were relatively higher, indicating a larger average deviation from the actual stock prices.

The robustness of the DRL model stems from its capacity to learn an optimal policy from high-dimensional input data, making it suitable for complex tasks such as stock price prediction. Its key strength lies in its ability to continuously improve its predictions by maximizing a reward function, leading to increasingly accurate predictions over time.

Nevertheless, the DRL model does have certain limitations. Training a DRL model can be computationally expensive and time-consuming due to its iterative learning process. Moreover, it requires a carefully designed reward function and the correct choice of hyperparameters to ensure convergence. It's also worth noting that, similar to other ML models, DRL does not inherently account for external factors that might influence stock prices, such as economic indicators or company-specific news.

8 PARTICLE SWARM OPTIMIZATION OPTIMIZED REGRESSION FEED - FORWARD NEURAL NETWORK

The objective of the current study was to explore the effectiveness of PSO in optimizing the RFFNN approach for stock trading using historical stock data. As an alternative to the handcrafted metaheuristic optimization, it is possible to use Nevergrad platform[8] for hyperparameter optimization for ML models (autoML). But the aim here was to focus specifically to effectiveness and usability of simple swarm/evolutionary techniques. The methodology employed in this investigation built upon previous approaches and extended the analysis to incorporate PSO techniques in Python, alongside other essential libraries, including yfinance, pyswarm, pandas, pandas_ta, sklearn, datetime, matplotlib.pyplot, numpy, copy, tensorflow and tensorflow.keras. The formula of PSO is shown in (10), where v_i is the velocity of particle *i*, x_i is the position of particle *i*, $pbest_i$ is the personal best position of particle *i*, *gbest* is the global best position of the swarm, *w* is the inertia weight, c_1 and c_2 are the acceleration coefficients, and r_1 and r_2 are random numbers between 0 and 1.

$$v_i^{t+1} = wv_i^t + c_1r_1(pbest_i - x_i^t) + c_2r_2(gbest - x_i^t)$$
$$x_i^{t+1} = x_i^t + v_i^{t+1} \tag{10}$$

PSO can have some limitations. RFFNN-PSO can be more time-consuming due to the added complexity of the PSO algorithm. Certain cases, like TSLA, illustrate that this additional complexity doesn't always result in superior predictions, indicating simpler models may be more suitable in some situations. The choice of PSO parameters (such as swarm size and the number of iterations) can significantly impact the model's performance, necessitating careful tuning. As with any ML model, RFFNN-PSO's effectiveness is also subject to the quality and quantity of input data. Its success in stock price prediction relies on the assumption that future trends will echo historical patterns - an assumption that may not hold in volatile stock markets.

8.1 Data Preprocessing:

The collected stock data was preprocessed using pandas, numpy and sklearn libraries in Python. This involved handling missing data, converting data types, and normalizing

[8] https://facebookresearch.github.io/nevergrad/

numerical data. The percentage change in closing price from the previous day was calculated and used as the target variable for RFFNN - PSO tasks.

8.2 Feature Engineering:

Additional features were engineered from the raw stock data, including the technical indicators, similar to the previous methods. These features were used as input factors for the RFFNN - PSO model to capture relevant market trends and patterns.

8.3 RFFNN - PSO Model Architecture:

The RFFNN - PSO model was implemented using the RFFNN model and optimized using PSO. The PSO algorithm was employed to search for optimal hyperparameters (epochs and batch_size in this study) of the RFFNN model. The PSO algorithm initialized a population of particles representing different hyperparameter configurations and iteratively updated the particles' positions based on their performance in optimizing the RFFNN model. The detailed configuration in this experiment is: The c_1 and c_2 were used 2.05 and 2.05 as default values in pyswarm pso() function arguments, the swarm size and the number of iteration were set as 100 and 100 respectively, the epochs and batch_size from RFFNN were set as hyperparameters, the lower – bound was set as [70, 32] and the upper – bound was set as [150, 64]. Resulted the total time cost was 23.5 hours.

8.4 Model Training:

The RFFNN model with optimized hyperparameters was trained using the collected stock data. The Adam optimizer and the MSE loss function was employed to update the model weights. Techniques such as experience replay and target network updating were used to enhance the stability and convergence of the model during training.

8.5 Model Evaluation:

The trained RFFNN model with optimized hyperparameters and the performance were evaluated using the MSE, MAE, MAPE, R2 (R2).

8.6 Results and Analysis:

The results obtained from the optimized RFFNN model using PSO were analyzed and interpreted. The performance metrics, visualizations of predicted stock prices compared with real stock prices, MSE, MAE, MAPE, R2 (R2) were presented. The strengths and limitations of the RFFNN – PSO model were also discussed as follows:[39, 52, 53]

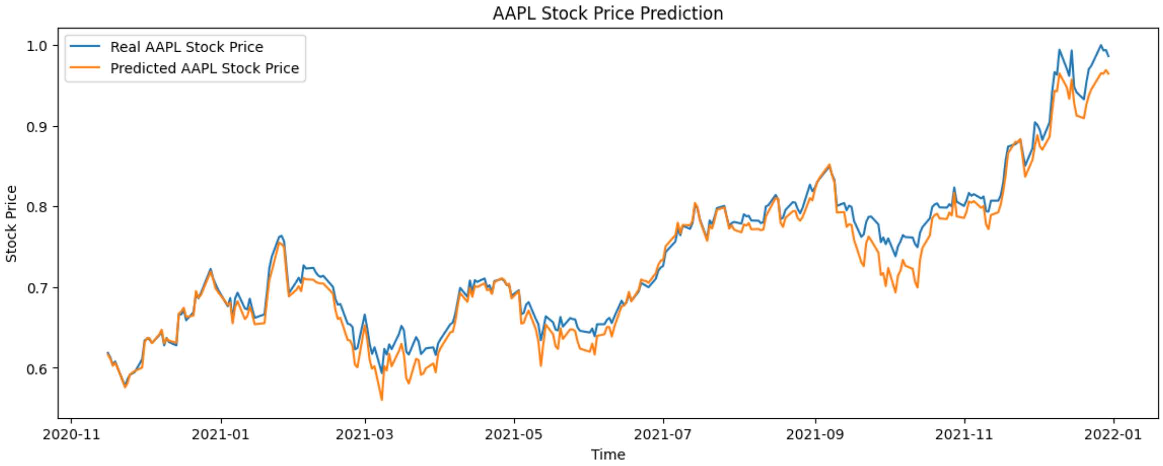

Figure 28 AAPL stock price prediction in RFFNN-PSO model

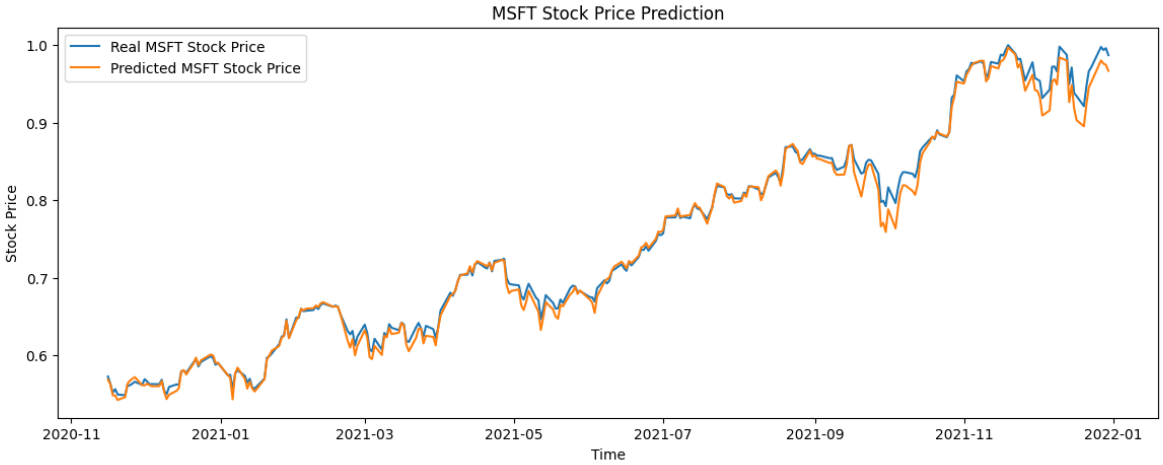

Figure 29 MSFT stock price prediction in RFFNN-PSO model

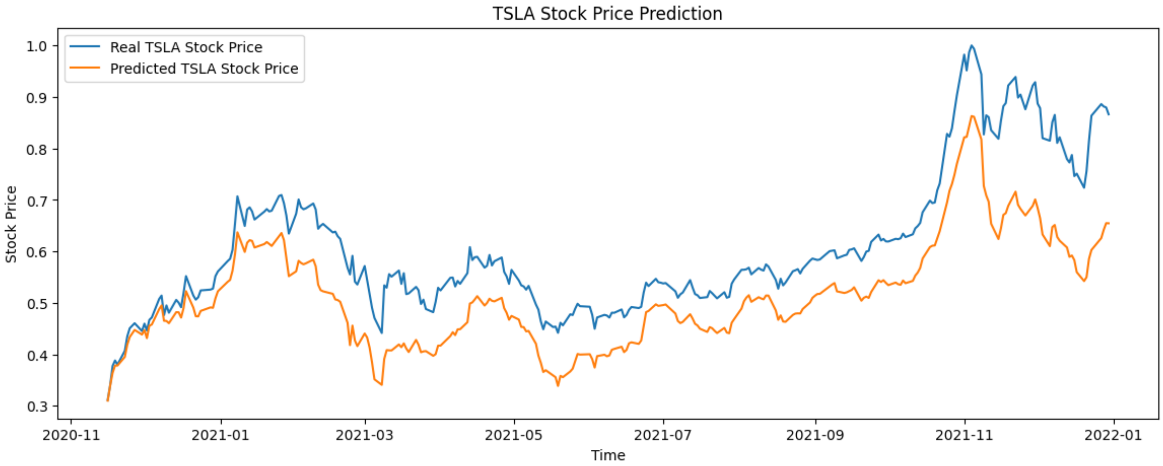

Figure 30 TSLA stock price prediction in RFFNN-PSO model

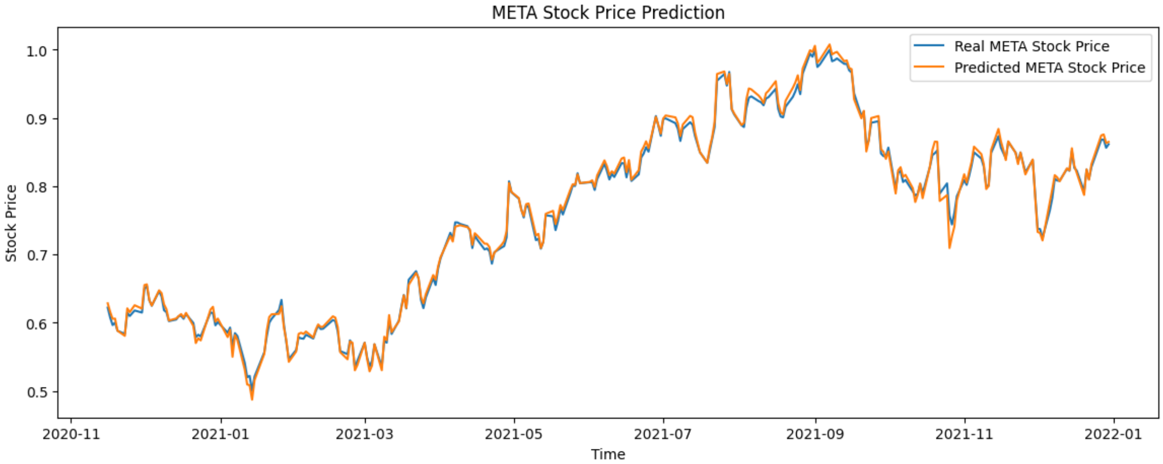

Figure 31 META stock price prediction in RFFNN

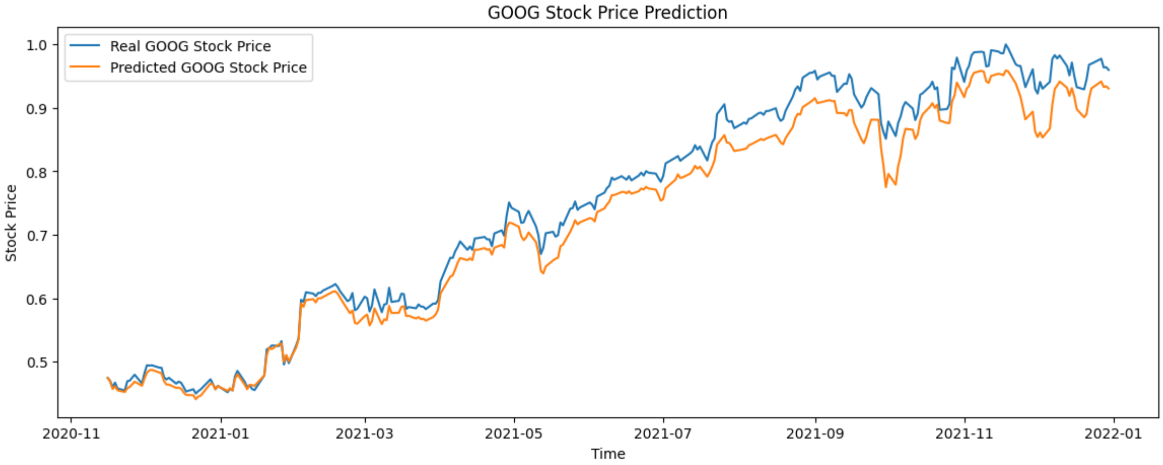

Figure 32 GOOG stock price prediction in RFFNN-PSO model

Table 6 RFFNN – PSO statistical analysis

	MSE	MAE	MAPE	R2
AAPL	0.0003	0.0132	0.0179	0.9672
MSFT	9.721	0.0068	0.0088	0.9946
TSLA	0.0115	0.0933	0.1481	0.3847
META	4.8338	0.0054	0.0073	0.9972
GOOG	0.0011	0.0279	0.0345	0.966

As can be seen from Figure 28 – 32 and Table 6, compared with Figure 18 – 22, the two models, RFFNN and the RFFNN-PSO, a distinct improvement is observed with RFFNN-PSO, most notably in predictions for AAPL, MSFT, and GOOG. The RFFNN-PSO model achieves lower MSE, MAE, and MAPE scores, implying more accurate predictions. The R2 also show a better fit to the data with RFFNN-PSO. The exception is seen with TSLA where RFFNN provides lower MSE, MAE, and MAPE. In the case of META, both models perform comparably, with a slight edge to RFFNN-PSO.

Regarding the degree of performance enhancement achieved by RFFNN-PSO, it varies by stock. For instance, a 15% improvement in MSE is observed for AAPL (0.0003401 to 0.0002888), about 73% for MSFT (0.0003638 to 0.0000972), and around 47% for GOOG (0.002026 to 0.001074).

The strength of RFFNN-PSO lies in its efficient parameter space search capability, facilitated by the PSO algorithm's swarm intelligence. This results in discovering optimal or near-optimal solutions possibly overlooked by traditional gradient-based optimization methods employed in standard RFFNN, thus improving overall model accuracy and reliability.

9 COMPARISON

This chapter listed and compared the statistical data of the 5 models LSTM, CFFNN, RFFNN, DRL and RFFNN – PSO in prediction of AAPL, MSFT, TSLA, META and GOOG stocks' prices, the comparison and analytical results about the 5 models' performance are stated in the following paragraph, subsections 9.1 – 9.5 and Tables 7 – 11. Note that the CFFNN is classification, but not numerical prediction, so the statistics were use accuracy score and confusion matrix for CFFNN. Also, the DRL did not use MAPE is for several disadvantages of MAPE such as MAPE is undefined when the actual value is zero or close to zero, which can happen in DRL problems that involve sparse rewards or zero-sum games. MAPE puts a heavier penalty on negative errors than on positive errors, which can bias the model towards under-forecasting. MAPE is not differentiable everywhere, which can cause problems for gradient-based optimization methods that are commonly used in DRL. MAPE is sensitive to outliers and scale-dependent, which can distort the comparison of different DRL models or scenarios.

Table 7 Comparison of LSTM, CFFNN, RFFNN, DRL and RFFNN - PSO for AAPL

	MSE	MAE	MAPE	R2	Accuracy	Confusion Matrix
LSTM	0.0018	0.0303	0.038	0.7844	N/A	N/A
CFFNN	N/A	N/A	N/A	N/A	0.5	[[58 74] [67 83]]
RFFNN	0.0003	0.0145	0.0194	0.9613	N/A	N/A
DRL	9.3043	1.4156	N/A	0.9231	N/A	N/A
RFFNN - PSO	0.0003	0.0132	0.0179	0.9672	N/A	N/A

Table 8 Comparison of LSTM, CFFNN, RFFNN, DRL and RFFNN - PSO for MSFT

	MSE	MAE	MAPE	R2	Accuracy	Confusion Matrix
LSTM	0.0048	0.0532	0.0634	0.7184	N/A	N/A
CFFNN	N/A	N/A	N/A	N/A	0.5461	[[12 122] [6 142]]
RFFNN	0.0004	0.0126	0.0156	0.9796	N/A	N/A

DRL	10.3027	1.2198	N/A	0.9581	N/A	N/A
RFFNN - PSO	9.721	0.0068	0.0089	0.9946	N/A	N/A

Table 9 Comparison of LSTM, CFFNN, RFFNN, DRL and RFFNN - PSO for TSLA

	MSE	MAE	MAPE	R2	Accuracy	Confusion Matrix
LSTM	0.012	0.0846	0.123	0.3195	N/A	N/A
CFFNN	N/A	N/A	N/A	N/A	0.5355	[[24 100] [31 127]]
RFFNN	0.0085	0.0632	0.0912	0.546	N/A	N/A
DRL	22.4145	2.1665	N/A	0.9127	N/A	N/A
RFFNN - PSO	0.0115	0.0933	0.1481	0.3847	N/A	N/A

Table 10 Comparison of LSTM, CFFNN, RFFNN, DRL and RFFNN - PSO for META

	MSE	MAE	MAPE	R2	Accuracy	Confusion Matrix
LSTM	0.0016	0.0336	0.0418	0.9033	N/A	N/A
CFFNN	N/A	N/A	N/A	N/A	0.5319	[[28 113] [19 122]]
RFFNN	0.0001	0.0072	0.0096	0.9935	N/A	N/A
DRL	8.7232	1.971	N/A	0.9632	N/A	N/A
RFFNN - PSO	4.8338	0.0054	0.0073	0.9972	N/A	N/A

Table 11 Comparison of LSTM, CFFNN, RFFNN, DRL and RFFNN - PSO for GOOG

	MSE	MAE	MAPE	R2	Accuracy	Confusion Matrix
LSTM	0.0016	0.0321	0.0389	0.9442	N/A	N/A
CFFNN	N/A	N/A	N/A	N/A	0.5638	[[0 122] [1 159]]
RFFNN	0.002	0.0353	0.0417	0.9359	N/A	N/A
DRL	14.4122	1.4545	N/A	0.9669	N/A	N/A

RFFNN - PSO	0.0011	0.0279	0.034501584518474286	0.966	N/A	N/A

Based on Table 7 – 11, the DRL model has the highest R2 score (average 95%), means that it has the best fitness between the predicted price and the real price for all AAPL, MSFT, TSLA, META and GOOG stocks. The CFFNN has only average 50% accuracy score compared with the MSE, MAE and MAPE values of other models, it has the lowest accuracy for prediction of all stocks' prices. By comparing all MSE, MAE, MAPE and R2 values between LSTM and RFFNN models, the LSTM has a better performance than the RFFNN (more accurate), however, after the optimization of the RFFNN by PSO, the RFFNN – PSO showed a much better performance than LSTM, which means by tuning the hyperparameters of RFFNN with PSO method, the RFFNN performance and capability in stock price prediction improved significantly, indicated the PSO can effectively optimize the RFFNN model's performance, however, the time consuming issue was also notable during the practical experiment process.

Therefore, by comparing the 5 models' performance, the advantages and disadvantages can be generalized as follows according to the models' features such as model type, processing data type, complication and time cost etc.:

9.1 Long short – term memory neural network:

LSTM is a type of RNN that can capture temporal dependencies in stock price data. It is suitable for time-series forecasting tasks, such as predicting stock prices based on historical data. It requires careful tuning of hyperparameters, such as number of LSTM layers, number of hidden units, and learning rate. It can capture long-term dependencies in data, but may suffer from vanishing or exploding gradients. The training time can be relatively longer compared to other methods.

9.2 Classification feed-forward artificial neural network:

This method involves using a FFNN to classify stock price movements into categories, such as "buy," "hold," or "sell." It requires labeled data for training, which may be challenging to obtain for stock price data. It can be computationally efficient and relatively faster to train compared to LSTM. It requires careful selection of input features and model architecture for effective classification. It may not capture subtle changes in stock price trends and may have limitations in predicting exact stock prices.

9.3 Regression feed-forward artificial neural network:

Involves using a FFNN for regression tasks, such as predicting stock prices directly. It is similar to the classification method, requires labeled data for training. It can be computationally efficient and faster to train compared to LSTM. It requires careful selection of input features and model architecture for accurate regression predictions. It may not capture long-term dependencies in data and may be sensitive to noise in stock price data.

9.4 Deep reinforcement learning:

DRL involves training a model to make optimal trading decisions based on rewards obtained from stock trading actions. It can capture complex interactions between stock prices and trading actions, and can potentially adapt to changing market conditions. It requires extensive training and tuning of hyperparameters, such as learning rate, discount factor, and exploration rate. It may suffer from high variance and instability during training, and may require techniques such as experience replay and target network updating for stability. It can be computationally expensive and time-consuming compared to other methods.

9.5 Particle swarm optimization optimized regression feed - forward neural network:

PSO is a metaheuristic optimization algorithm that can be used to optimize hyperparameters of ML models. It can search for optimal hyperparameter configurations efficiently and effectively. When combined with RFFNN, it can potentially improve the performance and robustness of this methods. PSO can reduce the need for manual tuning of hyperparameters and can lead to more optimal results. However, PSO also requires careful selection of hyperparameter search space and may have limitations in finding the global optimal solution.

Overall, the choice of method depends on the specific requirements of the stock trading task, the available data, and the desired level of accuracy and computational efficiency. LSTM may be suitable for capturing long-term dependencies in data, while CFFNN and RFFNN may be simpler and faster options. DRL can capture complex interactions between stock prices and trading actions, but requires extensive training and tuning. PSO can optimize hyperparameters of these methods, potentially leading to improved performance, but also requires careful selection of hyperparameter search space. Further experimentation and analysis are necessary to determine the most effective method for stock trading based on the specific dataset and task at hand.

CONCLUSION

In this comprehensive study, various fundamental AI models were applied to predict the prices of prominent stocks, namely AAPL, MSFT, TSLA, META, and GOOG, from the beginning of 2016 to the end of 2021. The performance and potential of these AI models in financial forecasting were meticulously examined, offering noteworthy findings and implications.

Among the tested AI models, the DRL model demonstrated superior performance, with an impressive average R-squared value of 95%. This result indicates that the DRL model can explain 95% of the variance in the price of these stocks, a remarkable achievement that signifies the robustness and accuracy of this model in stock price prediction.

On the contrary, the CFFNN model performed less effectively, achieving only an average accuracy score of 50%. Despite its relatively lower performance in this context, the insights obtained from its application provide a valuable foundation for future modifications and improvements.

The LSTM and RFFNN models yielded good, albeit not exceptional, performance. Nevertheless, these models exhibited substantial potential. Their strength lies in their flexibility to be further improved through modifications or the incorporation of more detailed and influential input data.

Significantly, the robust performance of the RFFNN - PSO model underscored the efficacy of the PSO method in the hyperparameter tuning process. This finding implies that the PSO method can significantly streamline hyperparameter tuning, especially for complex models, thus contributing to more accurate and efficient forecasting models.

While the results of this study are enlightening, there are promising avenues for future research. Enhancing the tested models, incorporating additional influential inputs and indicators, and exploring the synergy of different models or advanced optimizing algorithms could further elevate the predictive performance of AI models in stock price forecasting.

This thesis adds significantly to the growing body of knowledge at the intersection of AI, finance, and computer science. The findings provide a substantial basis for future explorations into more sophisticated AI-based forecasting models. Moreover, they contribute valuable insights into the use of AI in financial forecasting, potentially informing the development of more advanced and accurate AI-driven prediction tools in finance and other related fields.

BIBLIOGRAPHY

[1] AZZUTTI, Alessio. AI trading and the limits of EU law enforcement in deterring market manipulation. *Computer Law & Security Review* [online]. 2022, vol. 45, s. 105690. ISSN 0267-3649. Dostupné z: https://www.sciencedirect.com/science/article/pii/S0267364922000371.

[2] LANG, Sebastian et al. Modeling Production Scheduling Problems as Reinforcement Learning Environments based on Discrete-Event Simulation and OpenAI Gym. *IFAC-PapersOnLine* [online]. 2021, vol. 54, no. 1, s. 793-798. ISSN 2405-8963. Dostupné z: https://www.sciencedirect.com/science/article/pii/S2405896321008399.

[3] CHAO, Xiangrui et al. Jie Ke versus AlphaGo: A ranking approach using decision making method for large-scale data with incomplete information. *European Journal of Operational Research* [online]. 2018, vol. 265, no. 1, s. 239-247. ISSN 0377-2217. Dostupné z: https://www.sciencedirect.com/science/article/pii/S0377221717306574.

[4] SHUAI, Hang et al. Post-storm repair crew dispatch for distribution grid restoration using stochastic Monte Carlo tree search and deep neural networks. *International Journal of Electrical Power & Energy Systems* [online]. 2023, vol. 144, s. 108477. ISSN 0142-0615. Dostupné z: https://www.sciencedirect.com/science/article/pii/S0142061522004847.

[5] Affiliates of Goldman Sachs become majority shareholders in Boyd Corp. *Sealing Technology* [online]. 2018, vol. 2018, no. 11, s. 4. ISSN 1350-4789. Dostupné z: https://www.sciencedirect.com/science/article/pii/S1350478918304318.

[6] WU, Chih-Chiang a Wei-Peng CHEN. What's an AI name worth? The impact of AI ETFs on their underlying stocks. *Finance Research Letters* [online]. 2022, vol. 46, s. 102474. ISSN 1544-6123. Dostupné z: https://www.sciencedirect.com/science/article/pii/S1544612321004542.

[7] ANBALAGAN, Thirunavukarasu a S. Uma MAHESWARI. Classification and Prediction of Stock Market Index Based on Fuzzy Metagraph. *Procedia Computer Science* [online]. 2015, vol. 47, s. 214-221. ISSN 1877-0509. Dostupné z: https://www.sciencedirect.com/science/article/pii/S1877050915004688.

[8] JO, Wooyeon et al. Digital Forensic Practices and Methodologies for AI Speaker Ecosystems. *Digital Investigation* [online]. 2019, vol. 29, s. S80-S93. ISSN 1742-2876. Dostupné z: https://www.sciencedirect.com/science/article/pii/S1742287619301628.

[9] PEÑALOZA, Ana K. A., Alexandre BALBINOT a Roberto LEBORGNE. *Monitoring and control of electrical power systems using machine learning techniques*[online]. Emilio BAROCIO ESPEJO, Felix Rafael SEGUNDO SEVILLA a Petr KORBA. Elsevier. , 202311 - AI application for load forecasting: a comparison of classical and deep learning methodologies. . 263-287 s. ISBN 9780323999045. Dostupné z: https://www.sciencedirect.com/science/article/pii/B978032399904500017XID: 783675.

[10] BURGGRÄF, Peter et al. Performance assessment methodology for AI-supported decision-making in production management. *Procedia CIRP* [online]. 2020, vol. 93, s. 891-896. ISSN 2212-8271. Dostupné z: https://www.sciencedirect.com/science/article/pii/S2212827120306508.

[11] OHTA, Hiroyuki. Reevaluation of McCulloch–Pitts–von Neumann's clock. *Biosystems* [online]. 2015, vol. 127, s. 7-13. ISSN 0303-2647. Dostupné z: https://www.sciencedirect.com/science/article/pii/S030326471400166X.

[12] CHANG, Anthony C. *Intelligence-based medicine*[online]. Anthony C. CHANG. Academic Press. , 2020Chapter 2 - History of Artificial Intelligence. . 23-27 s. ISBN 9780128233375. Dostupné z: https://www.sciencedirect.com/science/article/pii/B9780128233375000020ID: 778579.

[13] POLLACK, Jordan B. No harm intended: Marvin L. Minsky and Seymour A. Papert. Perceptrons: An Introduction to Computational Geometry, Expanded Edition. Cambridge, MA: MIT Press, 1988. Pp. 292. $12.50 (paper). *Journal of Mathematical Psychology* [online]. 1989, vol. 33, no. 3, s. 358-365. ISSN 0022-2496. Dostupné z: https://www.sciencedirect.com/science/article/pii/0022249689900151.

[14] REVELL, Timothy. 'AIs are really dumb. They don't even have the intelligence of a 6-month-old'. *New Scientist* [online]. 2019, vol. 242, no. 3233, s. 44-45. ISSN 0262-4079. Dostupné z: https://www.sciencedirect.com/science/article/pii/S0262407919310383.

[15] DWIVEDI, Yogesh K. et al. “So what if ChatGPT wrote it?” Multidisciplinary perspectives on opportunities, challenges and implications of generative conversational AI for research, practice and policy. *International Journal of Information Management* [online]. 2023, vol. 71, s. 102642. ISSN 0268-4012. Dostupné z: https://www.sciencedirect.com/science/article/pii/S0268401223000233.

[16] AMRUTHAVARSHINI, V. a Siddesha HANUMANTHAPPA. Comparative study of ANN and ANFIS models for detection of damages due to cracks in single bay framed structure. *Materials Today: Proceedings* [online]. 2023. ISSN 2214-7853. Dostupné z: https://www.sciencedirect.com/science/article/pii/S2214785323026445.

[17] SARVAIYA, Jainesh a Dinesh SINGH. Prediction of performance parameters in friction stir processing using ANN and multiple regression models. *Materials Today: Proceedings* [online]. 2023. ISSN 2214-7853. Dostupné z: https://www.sciencedirect.com/science/article/pii/S2214785323023234.

[18] ASLAN, Sinem et al. Recurrent neural networks for water quality assessment in complex coastal lagoon environments: A case study on the Venice Lagoon. *Environmental Modelling & Software* [online]. 2022, vol. 154, s. 105403. ISSN 1364-8152. Dostupné z: https://www.sciencedirect.com/science/article/pii/S1364815222001098.

[19] BENGIO, Yoshua a Honglak LEE. Editorial introduction to the Neural Networks special issue on Deep Learning of Representations. *Neural Networks* [online]. 2015, vol. 64, s. 1-3. ISSN 0893-6080. Dostupné z: https://www.sciencedirect.com/science/article/pii/S089360801400286X.

[20] BUSARI, Ganiyu Adewale a Dong Hoon LIM. Crude oil price prediction: A comparison between AdaBoost-LSTM and AdaBoost-GRU for improving forecasting performance. *Computers & Chemical Engineering* [online]. 2021, vol. 155, s. 107513. ISSN 0098-1354. Dostupné z: https://www.sciencedirect.com/science/article/pii/S009813542100291X.

[21] LI, Jicheng a S. Joe QIN. Applying and dissecting LSTM neural networks and regularized learning for dynamic inferential modeling. *Computers & Chemical*

Engineering [online]. 2023, vol. 175, s. 108264. ISSN 0098-1354. Dostupné z: https://www.sciencedirect.com/science/article/pii/S0098135423001345.

[22] MURUGANANDAM, S. et al. A deep learning based feed forward artificial neural network to predict the K-barriers for intrusion detection using a wireless sensor network. *Measurement: Sensors* [online]. 2023, vol. 25, s. 100613. ISSN 2665-9174. Dostupné z: https://www.sciencedirect.com/science/article/pii/S2665917422002471.

[23] NAGAI, Yoshinori a Yoji AIZAWA. Rule-dynamical generalization of McCulloch–Pitts neuron networks. *Biosystems* [online]. 2000, vol. 58, no. 1, s. 177-185. ISSN 0303-2647. Dostupné z: https://www.sciencedirect.com/science/article/pii/S0303264700001210.

[24] LIN, Yi a Shin-Min SONG. A CMAC neural-network-based algorithm for the kinematic control of a walking machine. *Engineering Applications of Artificial Intelligence* [online]. 1992, vol. 5, no. 6, s. 539-551. ISSN 0952-1976. Dostupné z: https://www.sciencedirect.com/science/article/pii/095219769290030N.

[25] BRUDER, Johannes. Chapter 5 - Infrastructural intelligence: Contemporary entanglements between neuroscience and AI. *Progress in Brain Research* [online]. 2017, vol. 233, s. 101-128. ISSN 0079-6123. Dostupné z: https://www.sciencedirect.com/science/article/pii/S0079612317300547.

[26] CUI, Weijie et al. Source term inversion of nuclear accident based on deep feedforward neural network. *Annals of Nuclear Energy* [online]. 2022, vol. 175, s. 109257. ISSN 0306-4549. Dostupné z: https://www.sciencedirect.com/science/article/pii/S0306454922002924.

[27] TEKE, Cagatay et al. Prediction of gamma ray spectrum for 22Na source by feed forward back propagation ANN model. *Radiation Physics and Chemistry* [online]. 2023, vol. 202, s. 110558. ISSN 0969-806X. Dostupné z: https://www.sciencedirect.com/science/article/pii/S0969806X22005941.

[28] SUN, Fukang et al. Recognition algorithm for light intensity variation of LED lamps using neural network with statistics characteristics. *Optik* [online]. 2020, vol. 200, s. 163362. ISSN 0030-4026. Dostupné z: https://www.sciencedirect.com/science/article/pii/S0030402619312604.

[29] VALLÉS-PÉREZ, Iván et al. Empirical study of the modulus as activation function in computer vision applications. *Engineering Applications of Artificial Intelligence* [online]. 2023, vol. 120, s. 105863. ISSN 0952-1976. Dostupné z: https://www.sciencedirect.com/science/article/pii/S0952197623000477.

[30] TIAN, Shaopeng et al. Using perceptron feed-forward Artificial Neural Network (ANN) for predicting the thermal conductivity of graphene oxide-Al2O3/water-ethylene glycol hybrid nanofluid. *Case Studies in Thermal Engineering* [online]. 2021, vol. 26, s. 101055. ISSN 2214-157X. Dostupné z: https://www.sciencedirect.com/science/article/pii/S2214157X21002185.

[31] YANG, Daoguang, Hamid Reza KARIMI a Marek PAWELCZYK. A new intelligent fault diagnosis framework for rotating machinery based on deep transfer reinforcement learning. *Control Engineering Practice* [online]. 2023, vol. 134, s. 105475. ISSN 0967-0661. Dostupné z: https://www.sciencedirect.com/science/article/pii/S0967066123000448.

[32] ZHUANG, Dian et al. Data-driven predictive control for smart HVAC system in IoT-integrated buildings with time-series forecasting and reinforcement learning. *Applied Energy* [online]. 2023, vol. 338, s. 120936. ISSN 0306-2619. Dostupné z: https://www.sciencedirect.com/science/article/pii/S0306261923003008.

[33] REVELL, Timothy. Google's DeepMind AI discovers physics. *New Scientist* [online]. 2016, vol. 232, no. 3100, s. 25. ISSN 0262-4079. Dostupné z: https://www.sciencedirect.com/science/article/pii/S0262407916321212.

[34] GARG, Harish. A hybrid PSO-GA algorithm for constrained optimization problems. *Applied Mathematics and Computation* [online]. 2016, vol. 274, s. 292-305. ISSN 0096-3003. Dostupné z: https://www.sciencedirect.com/science/article/pii/S0096300315014630.

[35] CAI, Shibo et al. Gait phases recognition based on lower limb sEMG signals using LDA-PSO-LSTM algorithm. *Biomedical Signal Processing and Control* [online]. 2023, vol. 80, s. 104272. ISSN 1746-8094. Dostupné z: https://www.sciencedirect.com/science/article/pii/S1746809422007261.

[36] DUTTON, Robert W. a Ronald J. G. GOOSSENS. Technology CAD at Stanford University: physics, algorithms, software and applications. *Microelectronics Journal* [online]. 1995, vol. 26, no. 2, s. 99-111. ISSN 0026-2692. Dostupné z: https://www.sciencedirect.com/science/article/pii/002626929598916F.

[37] SHAN, Tianle et al. Multi-UAV WRSN charging path planning based on improved heed and IA-DRL. *Computer Communications* [online]. 2023, vol. 203, s. 77-88. ISSN 0140-3664. Dostupné z: https://www.sciencedirect.com/science/article/pii/S0140366423000671.

[38] MA, Yuhua et al. FT-IR combined with PSO-CNN algorithm for rapid screening of cervical tumors. *Photodiagnosis and Photodynamic Therapy* [online]. 2022, vol. 39, s. 103023. ISSN 1572-1000. Dostupné z: https://www.sciencedirect.com/science/article/pii/S157210002200309X.

[39] YADAV, Anamika a Subha M. ROY. An artificial neural network-particle swarm optimization (ANN-PSO) approach to predict the aeration efficiency of venturi aeration system. *Smart Agricultural Technology* [online]. 2023, vol. 4, s. 100230. ISSN 2772-3755. Dostupné z: https://www.sciencedirect.com/science/article/pii/S2772375523000606.

[40] SADEGHI, Alireza, Amir DANESHVAR a Mahdi MADANCHI ZAJ. Combined ensemble multi-class SVM and fuzzy NSGA-II for trend forecasting and trading in Forex markets. *Expert Systems with Applications* [online]. 2021, vol. 185, s. 115566. ISSN 0957-4174. Dostupné z: https://www.sciencedirect.com/science/article/pii/S0957417421009726.

[41] YI, Yahui et al. Digital twin-long short-term memory (LSTM) neural network based real-time temperature prediction and degradation model analysis for lithium-ion battery. *Journal of Energy Storage* [online]. 2023, vol. 64, s. 107203. ISSN 2352-152X. Dostupné z: https://www.sciencedirect.com/science/article/pii/S2352152X2300600X.

[42] GÜLMEZ, Burak. Stock price prediction with optimized deep LSTM network with artificial rabbits optimization algorithm. *Expert Systems with Applications* [online]. 2023, vol. 227, s. 120346. ISSN 0957-4174. Dostupné z: https://www.sciencedirect.com/science/article/pii/S0957417423008485.

[43] HUANG, Ruijie et al. Well performance prediction based on Long Short-Term Memory (LSTM) neural network. *Journal of Petroleum Science and Engineering* [online]. 2022, vol. 208, s. 109686. ISSN 0920-4105. Dostupné z: https://www.sciencedirect.com/science/article/pii/S0920410521013152.

[44] J.L., Gayathri et al. A computer-aided diagnosis system for the classification of COVID-19 and non-COVID-19 pneumonia on chest X-ray images by integrating CNN with sparse autoencoder and feed forward neural network. *Computers in Biology and Medicine* [online]. 2022, vol. 141, s. 105134. ISSN 0010-4825. Dostupné z: https://www.sciencedirect.com/science/article/pii/S0010482521009288.

[45] MOULAY, H. et al. Dendrogram-based Artificial Neural Network modulation classification for dual-hop cooperative relaying communications. *Physical Communication* [online]. 2022, vol. 55, s. 101929. ISSN 1874-4907. Dostupné z: https://www.sciencedirect.com/science/article/pii/S1874490722002063.

[46] DEMIRBAY, Barış, Duygu BAYRAM KARA a Şaziye UĞUR. Multivariate regression (MVR) and different artificial neural network (ANN) models developed for optical transparency of conductive polymer nanocomposite films. *Expert Systems with Applications* [online]. 2022, vol. 207, s. 117937. ISSN 0957-4174. Dostupné z: https://www.sciencedirect.com/science/article/pii/S0957417422011770.

[47] KEDDOUDA, Abdelhak et al. Solar photovoltaic power prediction using artificial neural network and multiple regression considering ambient and operating conditions. *Energy Conversion and Management* [online]. 2023, vol. 288, s. 117186. ISSN 0196-8904. Dostupné z: https://www.sciencedirect.com/science/article/pii/S0196890423005320.

[48] RAVNIK, J. et al. A sigmoid regression and artificial neural network models for day-ahead natural gas usage forecasting. *Cleaner and Responsible Consumption* [online]. 2021, vol. 3, s. 100040. ISSN 2666-7843. Dostupné z: https://www.sciencedirect.com/science/article/pii/S2666784321000346.

[49] LI, Xiangjuan et al. ATS-O2A: A state-based adversarial attack strategy on deep reinforcement learning. *Computers & Security* [online]. 2023, vol. 129, s. 103259. ISSN

0167-4048. Dostupné z: https://www.sciencedirect.com/science/article/pii/S0167404823001694.

[50] XIA, Xiangzhao et al. A multi-agent convolution deep reinforcement learning network for aeroengine fleet maintenance strategy optimization. *Journal of Manufacturing Systems* [online]. 2023, vol. 68, s. 410-425. ISSN 0278-6125. Dostupné z: https://www.sciencedirect.com/science/article/pii/S0278612523000791.

[51] GOBY, Niklas, Tobias BRANDT a Dirk NEUMANN. Deep reinforcement learning with combinatorial actions spaces: An application to prescriptive maintenance. *Computers & Industrial Engineering* [online]. 2023, vol. 179, s. 109165. ISSN 0360-8352. Dostupné z: https://www.sciencedirect.com/science/article/pii/S0360835223001894.

[52] ZHANG, Y. Y. et al. Forecasting of Dissolved Gases in Oil-immersed Transformers Based upon Wavelet LS-SVM Regression and PSO with Mutation. *Energy Procedia* [online]. 2016, vol. 104, s. 38-43. ISSN 1876-6102. Dostupné z: https://www.sciencedirect.com/science/article/pii/S1876610216315636.

[53] ADARYANI, Fatemeh Rezaie, S. JAMSHID MOUSAVI a Fatemeh JAFARI. Short-term rainfall forecasting using machine learning-based approaches of PSO-SVR, LSTM and CNN. *Journal of Hydrology* [online]. 2022, vol. 614, s. 128463. ISSN 0022-1694. Dostupné z: https://www.sciencedirect.com/science/article/pii/S0022169422010332.

LIST OF ABBREVIATIONS

AI	Artificial intelligence
AIEQ	AI powered equity exchange-traded fund
ANN	Artificial neural networks
CFFNN	Classification feed-forward artificial neural networks
CNNs	Convolutional neural networks
D-FFNN	Deep feedforward neural network
DQN	Deep Q-Network
DRL	Deep reinforcement learning
ETF	Exchange-traded fund
FFNN	Feed-forward neural networks
GRUs	Gated recurrent units
IBM	International business machines corporation
LSTM	Long short-term memory neural network
MACD	Moving average convergence divergence
MAE	Mean absolute error
MAPE	Mean absolute percentage error
MIT	Massachusetts institute of technology
MLPs	Multi-layer perceptrons
MSE	Mean squared error
NN	Neural network
PSO	Particle swarm optimization
R2	R-squared score
RFFNN	Regression feed-forward artificial neural network
RFFNN-PSO	Particle swarm optimization optimized regression feed - forward neural network

RL	Reinforcement learning
RNN	Recurrent neural network
RSI	Relative strength index
S&P	Standard & Poor's
SAC	Soft actor-critic
STOC	Stochastic oscillator
U.S.	United States of America

LIST OF FIGURES

LIST OF TABLES