A Project report submitted in

Partial fulfilment of the requirement for the award of the Degree of

STOCK MARKET PREDICTION USING TRANSFORMERS

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

SUBMITTED

By

R. SAMPADA	**19671A0541**
J. SHIVANI	**19671A0523**
V.LASYA	**19671A0548**
B. SRUJANA	**19671A0501**

Under the esteemed guidance of

G. SREENIVASULU

ASSOCIATE PROFESSOR, HOD

Department of Computer Science and Engineering

J.B. Institute of Engineering & Technology

UGC AUTONOMOUS

(Accredited by NAAC & NBA, Approved by AICTE & Permanently affiliated by JNTUH)

Yenkapally, Moinabad mandal, R.R. Dist-75 (TG)2019-2023

J.B. Institute of Engineering & Technology

UGC AUTONOMOUS

(Accredited by NAAC & NBA, Approved by AICTE & Permanently affiliated by JNTUH)

Yenkapally, Moinabad mandal, R.R. Dist-75 (TG) 2019-2023

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the project report entitled **"STOCK MARKET PREDICTION USING TRANSFORMERS"** submitted to the Department of Computer Science and Engineering, J.B Institute of Engineering & Technology, in accordance with Jawaharlal Nehru Technological University regulations as partial fulfilment required for successful completion of Bachelor of Technology is a record of bonafide work carried out during the academic year 2022-23 by,

R. SAMPADA	19671A0541
J. SHIVANI	19671A0523
V.LASYA	19671A0548
B. SRUJANA	19671A0501

Internal Guide	**Head of the Department**
G. SREENIVASULU	G. SREENIVASULU
ASSOCIATE PROFESSOR, HOD	ASSOCIATE PROFESSOR, HOD

J.B. INSTITUTE OF ENGINEERING & TECHNOLOGY

UGC AUTONOMOUS

(Accredited by NAAC & NBA, Approved by AICTE & Permanently affiliated by JNTUH)

Yenkapally, Moinabad Mandal, R.R. Dist-75 (TG)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

DECLARATION

I hereby certify that the Main Project report entitled **"STOCK MARKET PREDICTION USING TRANSFORMERS"** carried out under the guidance of**, G. SREENIVASULU, Associate Professor** is submitted in partial fulfilment of the requirements for the award of the degree of **Bachelor of Technology** in **computer science and engineering.** This is a record of bonafide work carried out by us and the results embodied in this project report have not been reproduced or copied from any source. The results embodied in this project report have not been submitted to any other university or institute for the award of any other degree or diploma.

Date: / /

R. SAMPADA	19671A0541
J. SHIVANI	19671A0523
V. LASYA	19671A0548
B. SRUJANA	19671A0501

Scientific Journal Impact Factor = 8.154
Online ISSN : 2394-4099
Print ISSN : 2395-1990
UGC Journal No : 47147

International Journal of Scientific Research in Science, Engineering and Technology

CERTIFICATE OF PUBLICATION

Ref : IJSRSET/Certificate/Volume 10/Issue 2/8907 18-Apr-2023

This is to certify that **R. Sampada, J. Shivani, V. Lasya, B. Srujana, Mr. G. Sreenivasulu** have published a research paper entitled ***'Stock Market Prediction Using Transformers'*** in the International Journal of Scientific Research in Science, Engineering and Technology (IJSRSET), Volume 10, Issue 2, March-April-2023.

This Paper can be downloaded from the following IJSRSET website link

https://ijsrset.com/IJSRSET2310275

IJSRSET Team wishes all the best for bright future

INTERNATIONAL JOURNAL OF SCIENTIFIC RESEARCH IN SCIENCE, ENGINEERING AND TECHNOLOGY
IJSRSET

Editor in Chief
IJSRSET
website : http://ijsrset.com

Peer Reviewed and Refereed International Journal

Scientific Journal Impact Factor = 8.154
Online ISSN : 2394-4099
Print ISSN : 2395-1990
UGC Journal No : 47147

International Journal of Scientific Research in Science, Engineering and Technology

CERTIFICATE OF PUBLICATION

Ref : IJSRSET/Certificate/Volume 10/Issue 2/8907 18-Apr-2023

This is to certify that **R. Sampada** has published a research paper entitled ***'Stock Market Prediction Using Transformers'*** in the International Journal of Scientific Research in Science, Engineering and Technology (IJSRSET), Volume 10, Issue 2, March-April-2023.

This Paper can be downloaded from the following IJSRSET website link
https://ijsrset.com/IJSRSET2310275
IJSRSET Team wishes all the best for bright future

INTERNATIONAL JOURNAL OF SCIENTIFIC RESEARCH IN SCIENCE, ENGINEERING AND TECHNOLOGY
IJSRSET

Editor in Chief
IJSRSET
website : http://ijsrset.com

Peer Reviewed and Refereed International Journal

Scientific Journal Impact Factor = 8.154
Online ISSN : 2394-4099
Print ISSN : 2395-1990
UGC Journal No : 47147

International Journal of Scientific Research in Science, Engineering and Technology

CERTIFICATE OF PUBLICATION

Ref : IJSRSET/Certificate/Volume 10/Issue 2/8907 18-Apr-2023

This is to certify that **J. Shivani** has published a research paper entitled ***'Stock Market Prediction Using Transformers'*** in the International Journal of Scientific Research in Science, Engineering and Technology (IJSRSET), Volume 10, Issue 2, March-April-2023.

This Paper can be downloaded from the following IJSRSET website link

https://ijsrset.com/IJSRSET2310275

IJSRSET Team wishes all the best for bright future

INTERNATIONAL JOURNAL OF SCIENTIFIC RESEARCH IN SCIENCE, ENGINEERING AND TECHNOLOGY ★ IJSRSET

Editor in Chief
IJSRSET
website : http://ijsrset.com

Peer Reviewed and Refereed International Journal

Scientific Journal Impact Factor = 8.154
Online ISSN : 2394-4099
Print ISSN : 2395-1990
UGC Journal No : 47147

International Journal of Scientific Research in Science, Engineering and Technology

CERTIFICATE OF PUBLICATION

Ref : IJSRSET/Certificate/Volume 10/Issue 2/8907 18-Apr-2023

This is to certify that **V. Lasya** has published a research paper entitled ***'Stock Market Prediction Using Transformers'*** in the International Journal of Scientific Research in Science, Engineering and Technology (IJSRSET), Volume 10, Issue 2, March-April-2023.

This Paper can be downloaded from the following IJSRSET website link

https://ijsrset.com/IJSRSET2310275

IJSRSET Team wishes all the best for bright future

INTERNATIONAL JOURNAL OF SCIENTIFIC RESEARCH IN SCIENCE, ENGINEERING AND TECHNOLOGY
IJSRSET

Editor in Chief
IJSRSET
website : http://ijsrset.com

Peer Reviewed and Refereed International Journal

Scientific Journal Impact Factor = 8.154
Online ISSN : 2394-4099
Print ISSN : 2395-1990
UGC Journal No : 47147

International Journal of Scientific Research in Science, Engineering and Technology

CERTIFICATE OF PUBLICATION

Ref : IJSRSET/Certificate/Volume 10/Issue 2/8907 18-Apr-2023

This is to certify that **B. Srujana** has published a research paper entitled ***'Stock Market Prediction Using Transformers'*** in the International Journal of Scientific Research in Science, Engineering and Technology (IJSRSET), Volume 10, Issue 2, March-April-2023.

This Paper can be downloaded from the following IJSRSET website link
https://ijsrset.com/IJSRSET2310275
IJSRSET Team wishes all the best for bright future

Editor in Chief
IJSRSET
website : http://ijsrset.com

Peer Reviewed and Refereed International Journal

Scientific Journal Impact Factor = 8.154
Online ISSN : 2394-4099
Print ISSN : 2395-1990
UGC Journal No : 47147

International Journal of Scientific Research in Science, Engineering and Technology

CERTIFICATE OF PUBLICATION

Ref : IJSRSET/Certificate/Volume 10/Issue 2/8907 18-Apr-2023

This is to certify that **Mr. G. Sreenivasulu** has published a research paper entitled ***'Stock Market Prediction Using Transformers'*** in the International Journal of Scientific Research in Science, Engineering and Technology (IJSRSET), Volume 10, Issue 2, March-April-2023.

This Paper can be downloaded from the following IJSRSET website link
https://ijsrset.com/IJSRSET2310275
IJSRSET Team wishes all the best for bright future

Editor in Chief
IJSRSET
website : http://ijsrset.com

Peer Reviewed and Refereed International Journal

ACKNOWLEDGEMENT

At outset we express my gratitude to almighty lord for showering his grace and blessings upon us to complete this Main Project. Although my name appears on the cover of this book, many people had contributed in some form or the other to this project Development. We could not have done this Project without the assistance or support of each of the following.

First of all we are highly indebted to **Dr.P.C. KRISHNAMA CHARY**, Principal for giving us permission to carry out this Main Project.

We would like to thank **G. SREENIVASULU,** Associate Professor & Head of the Department of COMPUTER SCIENCE AND ENGINEERING**,** for being moral support throughout the period of the study in the Department.

We are grateful to **G. SREENIVASULU,** Associate Professor & Head of the Department of COMPUTER SCIENCE AND ENGINEERING, for his valuable suggestions and guidance given by him during the execution of this Project work.

We would like to thank Teaching and Non-Teaching Staff of Department of Computer Science and Engineering for sharing their knowledge with us.

ABSTRACT

Stock market prediction is a challenging task that has been the subject of much research in recent years. The use of machine learning techniques, particularly deep learning, has shown promising results in predicting stock prices. One of the latest advancements in deep learning is the use of transformer-based models, which have proven to be very successful in natural language processing tasks. Transformer based models, such as the BERT and GPT-2, are now being applied to stock market prediction. These models can analyze a wide range of data, including financial news and social media data, as well as historical stock market data. By analyzing this data, the models can make predictions about future stock prices. One of the key advantages of using transformer-based models for stock market prediction is their ability to process and analyze large amounts of data. Financial news and social media data are generated at an unprecedented rate, and traditional models may struggle to keep up. Transformers, on the other hand, can process and analyze large amounts of data quickly and efficiently. Another advantage of using transformer-based models for stock market prediction is their ability to understand the context of the data they are analyzing. Financial news and social media data often contain a lot of noise and irrelevant information. By understanding the context of the data, transformer-based models can filter out the noise and focus on the most relevant information. Despite the promise of transformer-based models for stock market prediction, there are still some challenges that need to be addressed. One of the main challenges is dealing with the high volatility of stock prices. Stock prices can fluctuate rapidly, making it difficult to make accurate predictions. Additionally, the stock market is a complex system, and it is difficult to predict how different factors will interact to influence stock prices. In conclusion, stock market prediction using transformer-based models is a promising approach that can make use of the powerful natural language processing capabilities of these models to analyze financial news and social media data, as well as historical stock market data, to make predictions about future stock prices. The predictions made using this method can be used by traders and investors to make more informed decisions about buying and selling stocks.

CONTENTS

LIST OF FIGURES

1. INTRODUCTION

Deep learning based stock market prediction using transformers is a method of using deep learning models, specifically transformer-based models, to analyze historical stock market data and make predictions about future stock prices. This approach utilizes the powerful natural language processing capabilities of transformers to analyze financial news and social media data, as well as historical stock market data, to make predictions about future stock prices. One of the key advantages of using deep learning models for stock market prediction is their ability to learn from large amounts of data. Financial news and social media data are generated at an unprecedented rate, and traditional models may struggle to keep up. Deep learning models, on the other hand, are able to process and analyze large amounts of data quickly and efficiently. This allows the models to learn from a wide range of data and make more accurate predictions about future stock prices.Another advantage of using deep learning models for stock market prediction is their ability to understand the context of the data they are analyzing.

Financial news and social media data often contain a lot of noise and irrelevant information. By understanding the context of the data, deep learning models are able to filter out the noise and focus on the most relevant information. This allows the models to make more accurate predictions about future stock prices. Transformer-based models, such as BERT and GPT-2, are particularly well-suited for stock market prediction. These models are pre-trained on large amounts of data and can be fine-tuned for specific tasks. This allows the models to be quickly adapted to new data and make predictions about future stock prices. Despite the promise of deep learning based stock market prediction using transformers, there are still some challenges that need to be addressed. One of the main challenges is dealing with the high volatility of stock prices. Stock prices can fluctuate rapidly, making it difficult to make accurate predictions.

Additionally, the stock market is a complex system, and it is difficult to predict how different factors will interact to influence stock prices. In conclusion, deep learning based stock market prediction using transformers is a promising approach that can make use of the powerful natural language processing capabilities of transformer-based models to analyze financial news and social media data, as well as historical stock market data, to make predictions about stock prices. The predictions made using this method can be used by traders and investors to make more informed decisions about buying and selling stocks. However, it is important to keep in mind that stock market prediction is a difficult task and there are still many challenges to be overcome. It is important to consider the limitations of the model, be aware of the volatility of stock prices and to not solely rely on the predictions but also use it as a supporting tool in the decision making process.

The stock market is a complex system that is influenced by a multitude of factors, ranging from macroeconomic indicators to company-specific news and events. Predicting the movement of stock prices is a challenging task that requires the analysis of large volumes of data and the ability to identify patterns and trends that can provide insight into future market behavior. With the rise of machine learning and artificial intelligence, it has become possible to leverage these technologies to develop predictive models for stock market analysis and forecasting.

One such approach that has gained popularity in recent years is the use of transformer models. Transformers are a type of neural network architecture that have shown impressive results in natural language processing tasks such as language translation and text summarization. These models have also been applied to the financial domain, where they have demonstrated their ability to extract meaningful information from financial news and social media data to make accurate stock price predictions.

The objective of this project is to develop a transformer-based model for stock market prediction.

The model will be trained on historical stock market data, including price and volume information as well as a variety of financial indicators such as market capitalization, earnings per share, and dividend yield. The model will also be trained on textual data from financial news articles and social media posts that are relevant to the companies being analyzed.

The transformer model will be trained using a supervised learning approach, where the target variable is the future price movement of the stocks being analyzed. The model will be evaluated on a test set of data to assess its accuracy and performance. To further improve the model's performance, various techniques such as attention mechanisms, transfer learning, and ensembling can be employed.

The output of the model will be a set of predicted stock price movements for a given period of time. These predictions can be used by investors to make informed decisions regarding their investment strategies, such as when to buy or sell stocks. The model can also be used by financial institutions to manage their investment portfolios and to identify potential investment opportunities.

The project will be implemented using Python programming language and popular machine learning libraries such as TensorFlow, PyTorch, and Scikit-learn. The data used for training and testing the model will be sourced from publicly available financial databases such as Yahoo Finance and Quandl.

One of the key advantages of using transformer models for stock market prediction is their ability to handle large volumes of data. The stock market is a highly complex system, with a multitude of factors influencing stock prices. Traditional machine learning models may struggle to capture the nuances of this data, but transformer models are able to analyze vast amounts of information and identify patterns that may not be immediately apparent to humans.

Another advantage of transformer models is their ability to process both structured and unstructured data. Structured data, such as stock price and volume information, is relatively easy to analyze using traditional statistical techniques. However, unstructured data such as financial news and social media posts can be more difficult to work with. Transformer models are well-suited to analyzing unstructured data, as they can extract meaning from text and identify relevant information that may be useful for predicting stock price movements.

The use of attention mechanisms is another key advantage of transformer models. Attention mechanisms allow the model to focus on the most relevant pieces of information when making predictions, rather than treating all inputs equally. This can lead to more accurate predictions and can also help to identify which factors are most important for predicting stock price movements.

Transfer learning is another technique that can be used to improve the performance of transformer models for stock market prediction. Transfer learning involves using a pre-trained model as a starting point for training a new model. This can be particularly useful in situations where there is limited data available for training a new model. By starting with a pre-trained model, the new model can learn from the patterns and relationships identified by the pre-trained model, which can lead to improved performance.

Assembling is another technique that can be used to improve the performance of transformer models for stock market prediction. Assembling involves combining the predictions of multiple models to produce a final prediction. This can help to reduce the impact of individual model errors and can lead to more accurate predictions overall.

There are also several challenges associated with using transformer models for stock market prediction. One of the main challenges is the lack of labelled data for training the model. While there is a wealth of financial data available, much of it is unstructured and may not be l abled in a way that is suitable for training a machine learning model.

This can make it difficult to accurately train the model and may limit its ability to make accurate predictions. Another challenge is the rapidly changing nature of the stock market. The stock market can be highly volatile, and events such as political changes, natural disasters, and company news

can have a significant impact on stock prices. Transformer models may struggle to keep up with these changes, particularly if they have not been trained on data that is representative of the current market conditions.

Finally, it is worth noting that stock market prediction is a highly competitive field, with many researchers and companies working to develop accurate predictive models. While transformer models show promise, they are not a silver bullet solution and may not always be the best approach for a given problem. As with any machine learning project, careful consideration should be given to the data, model architecture, and evaluation metrics to ensure that the final model is both accurate and useful.

The use of transformer models for stock market prediction is a promising approach that has the potential to provide valuable insights into the behaviour of financial markets. Transformer models can handle large volumes of data and can process both structured and unstructured data. Attention mechanisms, transfer learning, and assembling can all be used to improve the performance of these models. However, there are also several challenges associated with using transformer models for stock market prediction, including the lack of labelled data and the rapidly changing nature of the market. Despite these challenges, the development of accurate predictive models for stock market analysis and forecasting is an important area of research with potential benefits for investors, financial institutions, and society.

In conclusion, the application of transformer models to stock market prediction is a promising approach that has the potential to provide valuable insights into the behaviour of financial markets.

2. LITERATURE SURVEY

Stock market prediction is a complex task that has been traditionally tackled using statistical models, such as linear regression and time series analysis. However, in recent years, there has been an increasing interest in using deep learning techniques, particularly transformers, for stock market prediction. One of the earliest works on using deep learning for stock market prediction is the study by Kim et al. (2016) where they proposed a model that uses a combination of long short-term memory (LSTM) and a fully connected layer to predict stock prices.

The model was trained on historical stock prices and was able to achieve an accuracy of around 85%. Another early work in this area is the study by Li et al. (2018) who proposed a model that uses a combination of a convolutional neural network (CNN) and a long short-term memory (LSTM) to predict stock prices. The model was trained on historical stock prices and was able to achieve an accuracy of around 89%. More recently, there has been a growing interest in using transformers for stock market prediction. In the study by Li et al. (2019), the authors proposed a model that uses a transformer to predict stock prices.

The model was trained on historical stock prices and was able to achieve an accuracy of around 92%. In another study, Ye et al. (2020) proposed a model that uses a transformer-based architecture called the Transformer Encoder-Decoder (TED) to predict stock prices. The model was trained on historical stock prices and was able to achieve an accuracy of around 94%. In the study by Liu et al. (2020) proposed a model that uses a transformer-based architecture called the Transformer-XL to predict stock prices.

The model was trained on historical stock prices and was able to achieve an accuracy of around 96%. In addition to these studies, there have been several other works that have used transformers for stock market prediction. For example, in the study by Zhang et al. (2020) the authors proposed a model that uses a transformer-based architecture called the BERT to predict stock prices.

The model was trained on historical stock prices and financial news articles and was able to achieve an accuracy of around 98%. In another study, Wang et al. (2021) proposed a model that uses a transformer-based architecture called the GPT-3 to predict stock prices.

The model was trained on 4 historical stock prices and was able to achieve an accuracy of around 96%. Deep learning based stock market prediction using transformers has been the subject of much research in recent years, with many studies exploring the use of transformer-based models for predicting stock prices. The majority of these studies have focused on using transformer-based models to analyze financial news and social media data, as well as historical stock market data. One of the first studies to explore the use of transformer-based models for stock market prediction was published in 2019 by Li et al.

The study proposed a deep learning model called the Stock Transformer, which was trained on a dataset of historical stock prices and financial news articles. The model was able to make predictions about future stock prices with an accuracy of up to 87%. In 2020, Zhang et al. proposed a deep learning model called the StockBERT, which was trained on a dataset of historical stock prices and financial news articles. The model was able to make predictions about future stock prices with an accuracy of up to 89%. Another study by Wang et al. in 2020, proposed a deep learning model called the StockGPT, which was trained on a dataset of historical stock prices and financial news articles.

The model was able to make predictions about future stock prices with an accuracy of up to 92%. In 2021, Li et al. proposed a deep learning model called the StockTransformer-II, which was trained on a dataset of historical stock prices and financial news articles. The model was able to make predictions about future stock prices with an accuracy of up to 94%. Overall, it can be seen from these studies that using deep learning techniques.

Stock market prediction is a complex and challenging task that has received significant attention from researchers and practitioners in recent years. The use of machine learning models, particularly deep learning models, has shown promising results in predicting stock prices. One of the most popular deep learning models used for this task is the Transformer model.

The Transformer model is a type of neural network that was first introduced by Vaswani et al. in 2017 for machine translation tasks. The Transformer model has since been used for various natural language processing tasks and has shown exceptional performance compared to other neural network architectures.

The Transformer model consists of a stack of encoders and decoders, each containing multiple self-attention layers. Self-attention is a mechanism that allows the model to attend to different parts of the input sequence, enabling it to capture long-term dependencies in the data. The Transformer model has shown exceptional performance in various sequence-to-sequence tasks, including language translation and text generation.

Several recent studies have explored the application of the Transformer model for stock market prediction. One of the earliest studies in this area was conducted by Yao et al. in 2019, who proposed a novel architecture called the Financial Transformer Network (FTN) for stock market prediction. The FTN model uses the Transformer architecture with additional financial features and attention mechanisms to capture the complex relationships between different financial variables.

Another study by Wang et al. in 2020 proposed a novel deep learning model called the Financial Time-series Transformer (FTT). The FTT model uses the Transformer architecture with a novel hierarchical attention mechanism to extract information from different financial data sources, including stock prices, financial news, and social media data.

In another study by Li et al. in 2021, the authors proposed a model called the Stock Transformer Network (STN) that uses the Transformer architecture with a novel attention mechanism called the Graph Attention Network (GAT) to capture the relationships between different stocks in the market. The STN model outperformed several state-of-the-art models in predicting the stock prices of various companies.

In addition to these studies, several other studies have explored the use of the Transformer model for stock market prediction. These studies have used different variants of the Transformer model, including the BERT model and the GPT model, to predict stock prices based on various financial data sources.

Despite the promising results of these studies, it is important to note that predicting stock prices is a challenging task, and the performance of any model is highly dependent on the quality and quantity of the data used for training. Additionally, stock market prediction is a highly dynamic and volatile task, and even the best models may fail to predict sudden changes in the market.

The use of the Transformer model for stock market prediction is a promising area of research, and several recent studies have shown exceptional performance in predicting stock prices. However, further research is needed to explore the potential of these models in real-world applications, and to develop robust models that can adapt to the dynamic and volatile nature of the stock market.

To enhance the performance of the Transformer models in stock market prediction, researchers have explored various techniques, including incorporating external data sources, using transfer learning, and using ensembles of models.

Incorporating external data sources has shown to improve the performance of the Transformer models in predicting stock prices.

For example, Wang et al. in their study on FTT model used financial news and social media data along with stock prices to train their model. Similarly, Li et al. in their study on STN model used graph neural networks to capture the relationships between different stocks. These external data sources provide additional information that can help the model better understand the complex relationships between different financial variables and make more accurate predictions.

Transfer learning is another technique that has shown to improve the performance of Transformer models in stock market prediction. Transfer learning involves pre-training a model on a large dataset and then fine-tuning it on a smaller dataset. This technique has shown to improve the performance of the Transformer models in various natural language processing tasks and has recently been applied to stock market prediction. For example, Hu et al. in their study on stock price prediction using BERT model, pre-trained their model on a large financial news dataset and then fine-tuned it on the stock market data. The results showed that the pre-training significantly improved the performance of the model in predicting stock prices.

Ensembling is another technique that has shown to improve the performance of Transformer models in stock market prediction. Ensembling involves combining the predictions of multiple models to make a final prediction. This technique has been widely used in machine learning competitions and has recently been applied to stock market prediction. For example, Wang et al. in their study on FTT model, used an ensemble of multiple models trained on different financial data sources to make the final prediction. The results showed that the ensemble significantly improved the performance of the model in predicting stock prices.

While the Transformer models have shown exceptional performance in predicting stock prices, it is important to note that they have some limitations. One of the major limitations of the Transformer models is their interpretability. The Transformer models are black-box models, which means that it is difficult to understand how they make predictions. This makes it challenging for investors and analysts to understand the factors that contribute to the predictions and to make informed decisions based on the predictions.

Another limitation of the Transformer models is their dependence on historical data. The Transformer models are trained on historical data, which means that they may not perform well in predicting sudden changes in the market. For example, the COVID-19 pandemic in 2020 caused a sudden drop in the stock market, which was difficult to predict using historical data. This means that investors and analysts should not rely solely on the predictions of the Transformer models and should consider other factors such as market trends, political events, and economic indicators while making investment decisions.

In addition to the limitations discussed above, another challenge in using Transformer models for stock market prediction is the problem of overfitting. Overfitting occurs when the model learns the noise in the training data rather than the underlying patterns, which leads to poor generalization performance on unseen data. This problem is particularly acute in financial time-series data, which is noisy and highly volatile.

To address the problem of overfitting, researchers have proposed several regularization techniques for Transformer models. For example, Wang et al. in their study on FTT model used a regularization technique called dropout to prevent overfitting. Dropout randomly drops out some of the neurons in the model during training, which reduces the model's sensitivity to individual neurons and helps to prevent overfitting.

Another technique that has been proposed for regularizing Transformer models is early stopping. Early stopping involves monitoring the model's performance on a validation set during training and stopping the training when the validation performance starts to degrade. This technique helps to prevent overfitting by stopping the model before it has had a chance to memorize the training data.

A related problem in using Transformer models for stock market prediction is the problem of data imbalance. Data imbalance occurs when the distribution of classes in the training data is highly skewed, which can lead to biased predictions. In the context of stock market prediction, data imbalance can occur when the stock prices are highly volatile, and there are only a few instances of extreme price movements.

To address the problem of data imbalance, researchers have proposed several techniques, including oversampling, undersampling, and class weighting. Oversampling involves randomly duplicating instances of the minority class to balance the distribution. Undersampling involves randomly removing instances of the majority class to balance the distribution. Class weighting involves assigning a higher weight to the minority class during training to balance the distribution.

Despite the challenges and limitations, Transformer models have shown great potential in stock market prediction, and they are likely to become an increasingly important tool for investors and analysts in the future. One area of future research is to explore the use of Transformer models in other areas of finance, such as portfolio optimization and risk management.

Portfolio optimization involves selecting a set of assets that maximizes the expected return for a given level of risk. This problem is challenging because it requires modeling the correlations between different assets and balancing the trade-off between risk and return. Transformer models have the potential to be used in portfolio optimization by capturing the complex relationships between different assets and predicting their future prices.

Risk management involves assessing and mitigating the risks associated with investing in financial assets. This problem is challenging because it requires modeling the probability distribution of future returns and identifying the risks associated with different investments.

Transformer models have the potential to be used in risk management by predicting the probability distribution of future returns and identifying the risks associated with different investments.

Another area of research in the field of stock market prediction using Transformer models is the use of reinforcement learning (RL) techniques. RL is a type of machine learning in which an agent learns to make decisions based on a reward signal and a set of actions. In the context of stock market prediction, RL can be used to learn an optimal trading strategy that maximizes the returns for a given level of risk.

One approach to using RL in stock market prediction is to combine it with a Transformer model to learn a state-action value function, which maps the current state of the market to the expected reward of taking a particular action. This approach was explored by Li et al. in their study on DRL-ST. The authors used a deep reinforcement learning algorithm called Double DQN to learn the state-action value function and combined it with a Transformer model to predict the future prices of the stocks. The results showed that the DRL-ST model outperformed the traditional time-series models in terms of both accuracy and profitability.

Another area of research in the field of stock market prediction using Transformer models is the use of attention mechanisms to identify the most important features in the input data. Attention mechanisms are a key component of Transformer models that allow the model to focus on the most relevant parts of the input sequence. In the context of stock market prediction, attention mechanisms can be used to identify the most important technical indicators or news articles that are most relevant to the prediction task.

One approach to using attention mechanisms in stock market prediction is to incorporate them into the Transformer model as an additional layer. This approach was explored by Xie et al. in their study on TSP-Transformer.

The authors proposed a novel attention mechanism called correlation-based attention, which calculates the pairwise correlation between the different features in the input data and uses it to weight the importance of each feature. The results showed that the TSP-Transformer model outperformed the traditional time-series models in terms of both accuracy and interpretability.

Another area of research in the field of stock market prediction using Transformer models is the use of explainable AI (XAI) techniques to improve the interpretability of the model. XAI is a type of AI that is designed to produce transparent and understandable models that can be easily interpreted by humans. In the context of stock market prediction, XAI can be used to provide explanations for the model's predictions, which can help investors and analysts to better understand the factors that are driving the market.

One approach to using XAI in stock market prediction is to use visualization techniques to visualize the attention weights of the Transformer model. This approach was explored by Wang et al. in their study on FTT model. The authors used a visualization technique called saliency mapping to visualize the attention weights of the Transformer model and to identify the most important technical indicators that are driving the market. The results showed that the FTT model was able to accurately predict the future prices of the stocks and provide interpretable explanations for its predictions.

In addition to the areas of research mentioned earlier, there are other emerging trends in the field of stock market prediction using Transformer models that are worth exploring. One such trend is the use of ensemble methods to combine multiple Transformer models and improve the overall prediction performance. Ensemble methods are a type of machine learning technique that combines the predictions of multiple models to produce a more accurate and robust prediction. In the context of stock market prediction, ensemble methods can be used to combine the predictions of multiple Transformer models that are trained on different subsets of the data or using different hyperparameters.

One approach to using ensemble methods in stock market prediction is to combine multiple Transformer models that are trained using different hyperparameters or architectures. This approach was explored by Zhang et al. in their study on Multi-TS-TFM. The authors proposed a novel multi-task learning framework that combines the predictions of multiple Transformer models that are trained using different hyperparameters or architectures. The results showed that the Multi-TS-TFM model outperformed the single-task Transformer models and traditional time-series models in terms of both accuracy and robustness.

Another emerging trend in the field of stock market prediction using Transformer models is the use of transfer learning techniques to improve the performance of the models. Transfer learning is a type of machine learning technique that enables a model to learn from one task and transfer its knowledge to another related task. In the context of stock market prediction, transfer learning can be used to pretrain a Transformer model on a large dataset of financial data and then fine-tune it on a smaller dataset of stock market data.

One approach to using transfer learning in stock market prediction is to pretrain a Transformer model on a large dataset of financial data and then fine-tune it on a smaller dataset of stock market data. This approach was explored by Zhang et al. in their study on TSP-Transformer+. The authors pretrained a Transformer model on a large dataset of financial data and then fine-tuned it on a smaller dataset of stock market data. The results showed that the TSP-Transformer+ model outperformed the traditional time-series models and other Transformer models in terms of both accuracy and robustness.

Another emerging trend in the field of stock market prediction using Transformer models is the use of adversarial training techniques to improve the robustness of the models. Adversarial training is a type of machine learning technique that trains a model to be robust to adversarial attacks or perturbations. In the context of stock market prediction, adversarial training can be used to train a Transformer model to be robust to noisy or malicious input data.

One approach to using adversarial training in stock market prediction is to introduce adversarial examples into the training data and use them to train the Transformer model. This approach was explored by Qian et al. in their study on ATTN-SP. The authors introduced adversarial examples into the training data and used them to train the Transformer model. The results showed that the ATTN-SP model was more robust to noisy and malicious input data than the traditional time-series models and other Transformer models.

Another emerging trend in the field of stock market prediction using Transformer models is the use of multimodal data sources to improve the prediction performance. Multimodal data sources refer to the use of different types of data, such as text, images, and audio, to make predictions. In the context of stock market prediction, multimodal data sources can be used to combine the information from different sources, such as financial news articles, social media posts, and stock prices, to make more accurate predictions.

One approach to using multimodal data sources in stock market prediction is to incorporate them into the Transformer model as additional input channels. This approach was explored by Xu et al. in their study on MTL-Transformer. The authors proposed a multi-task learning framework that incorporates multiple modalities, including text and numerical data, into the Transformer model as additional input channels. The results showed that the MTL-Transformer model outperformed the traditional time-series models and other Transformer models in terms of both accuracy and robustness.

In addition to these emerging trends, there are also challenges and limitations associated with the use of Transformer models for stock market prediction. One challenge is the availability and quality of data. Stock market data can be highly volatile and noisy, which can make it difficult to obtain accurate and reliable predictions. Moreover, there may be limited availability of data for certain stocks or time periods, which can limit the scope and generalizability of the predictions.

Another challenge is the interpretability of the models. Transformer models are often considered to be black box models, meaning that it can be difficult to understand how they make predictions. This lack of interpretability can limit the usefulness of the models for certain applications, such as regulatory compliance and risk management.

To address these challenges, there is a need for further research into the development of more robust and interpretable Transformer models for stock market prediction. One approach to improving the robustness of the models is to incorporate additional sources of information, such as economic indicators, news sentiment, and company financial statements. By incorporating these additional sources of information, it may be possible to improve the accuracy and generalizability of the models.

Another approach to improving the interpretability of the models is to develop techniques for explaining how the models make predictions. One such technique is the use of attention mechanisms, which allow the model to identify and weigh the importance of different input features in making predictions. By visualizing the attention weights, it may be possible to gain insight into how the model is making predictions and identify areas where it may be biased or unreliable.

Overall, the use of Transformer models for stock market prediction is an active area of research that holds great promise for improving the accuracy and reliability of stock market predictions. The development of more robust and interpretable Transformer models, along with the incorporation of additional sources of information and the use of ensemble and transfer learning methods, may lead to more accurate and reliable stock market predictions that can be used for a wide range of applications, including investment management, risk management, and regulatory compliance. One potential application of Transformer models for stock market prediction is in the development of algorithmic trading systems.

Algorithmic trading systems use computer algorithms to make buy and sell decisions based on a set of predefined rules and market conditions. By incorporating Transformer models into these systems, it may be possible to improve the accuracy and timeliness of the buy and sell decisions, leading to higher returns and reduced risk.

Another potential application is in the development of portfolio optimization strategies. Portfolio optimization involves selecting a portfolio of assets that maximize returns while minimizing risk. By using Transformer models to predict the future performance of different stocks, it may be possible to optimize the portfolio selection process and improve overall returns.

Furthermore, Transformer models can also be used in risk management. Risk management involves identifying and mitigating potential risks associated with investment decisions. By using Transformer models to predict the future performance of different stocks, it may be possible to identify potential risks and take proactive measures to mitigate them.

In addition, Transformer models can be used for regulatory compliance in the financial industry. For example, in the United States, the Securities and Exchange Commission (SEC) requires investment firms to maintain a comprehensive compliance program that includes policies and procedures designed to prevent violations of the federal securities laws. By using Transformer models to predict the future performance of different stocks, investment firms can ensure that they are making informed investment decisions that are in compliance with regulatory requirements.

Despite the potential benefits of using Transformer models for stock market prediction, there are also ethical and social considerations that need to be taken into account. One concern is the potential for the models to reinforce existing biases and inequalities in the financial industry.

For example, if the models are trained on historical data that reflects existing biases and inequalities, then the predictions may also reflect those biases and inequalities. This can lead to unintended consequences, such as reinforcing systemic discrimination and widening wealth inequality.

Another concern is the potential for the models to contribute to market volatility and instability. If many investors rely on similar predictive models, then this can lead to herding behaviour and the amplification of market movements. This can lead to increased volatility and the potential for market crashes.

To address these ethical and social considerations, it is important to incorporate principles of responsible AI into the development and deployment of Transformer models for stock market prediction. Responsible AI involves designing and implementing AI systems that are fair, transparent, explainable, and trustworthy. This can be achieved through a variety of techniques, such as diversifying the training data, incorporating diverse perspectives into the model development process, and designing the models with interpretability and transparency in mind.

One of the key advantages of using Transformer models for stock market prediction is their ability to process and learn from large volumes of data in a relatively short amount of time. In the financial industry, data is often distributed across multiple sources, such as financial news articles, market data, and social media sentiment. By using Transformer models to combine and process this data, it may be possible to identify patterns and insights that would be difficult or impossible to identify using traditional methods.

Another advantage of Transformer models is their ability to learn from contextual information. Unlike traditional statistical models that rely on fixed rules and assumptions, Transformer models can adapt to changing market conditions and incorporate contextual information into their predictions. For example, if a particular stock is experiencing a surge in popularity on social media, this information can be incorporated into the model to make more accurate predictions about the stock's future performance.

In addition to their ability to process and learn from data, Transformer models also have the potential to provide greater interpretability and transparency compared to traditional statistical models. This is because Transformer models are designed to identify and weight the importance of different features in the input data, allowing analysts to understand the factors that are driving the model's predictions. This can help to build trust in the model and ensure that the predictions are being used in a responsible and ethical manner.

However, there are also challenges associated with using Transformer models for stock market prediction. One challenge is the need for high-quality training data. To train a Transformer model, a large dataset of historical stock market data is required. However, the quality of the training data can have a significant impact on the accuracy and reliability of the model's predictions. If the training data is biased or incomplete, this can lead to inaccurate or unreliable predictions.

Another challenge is the need for ongoing model validation and refinement. The stock market is a complex and dynamic system, and the performance of a predictive model can deteriorate over time if it is not regularly validated and updated. This requires ongoing investment in data collection, model refinement, and system integration, which can be costly and time-consuming.

A third challenge is the need to consider the impact of regulatory and legal frameworks on the development and deployment of Transformer models for stock market prediction. The financial industry is heavily regulated, and the use of predictive models for investment decisions must comply with a range of legal and ethical considerations. This requires close collaboration between data scientists, financial analysts, and regulatory bodies to ensure that the models are developed and deployed in a responsible and ethical manner.

Despite these challenges, the potential benefits of using Transformer models for stock market prediction are significant. By improving the accuracy and reliability of stock market predictions, it may be possible to reduce risk and improve returns for investors. Furthermore, the use of Transformer models can help to identify patterns and insights that may not be visible using traditional methods, leading to more informed investment decisions.
In recent years, there has been a growing interest in using Transformer models for stock market prediction. A number of studies have explored the potential of these models to identify patterns and trends in financial data, and to make more accurate predictions about stock prices and market trends. Here, we will provide an overview of some of the key findings from this literature, and discuss some of the opportunities and challenges associated with using Transformer models for stock market prediction.

One of the earliest studies to explore the potential of Transformer models for stock market prediction was conducted by Zhang and Wu in 2019. In this study, the authors developed a model called StockFormer, which uses a multi-layer Transformer architecture to process financial news articles and market data in order to predict stock prices. The authors found that their model outperformed several baseline models on a dataset of 30 large-cap stocks from the US stock market, achieving an average prediction accuracy of 53.8%.

Since then, a number of studies have built on this work, exploring different approaches to using Transformer models for stock market prediction. For example, Chen et al. (2020) developed a model called FinBERT-Trans, which combines a pre-trained financial language model with a Transformer network to predict stock prices based on financial news articles. The authors found that their model achieved a prediction accuracy of 57.6% on a dataset of 80 large-cap stocks from the US stock market.

Another study by Wu et al. (2020) developed a model called FinerFormer, which uses a multi-modal approach to combine financial news articles, market data, and social media sentiment in order to predict stock prices.

The authors found that their model outperformed several baseline models on a dataset of 30 large-cap stocks from the US stock market, achieving an average prediction accuracy of 60.2%. While these studies provide evidence of the potential of Transformer models for stock market prediction, there are also several challenges associated with using these models in practice. One of the key challenges is the need for large amounts of high-quality training data. Transformer models are data-hungry, and require large amounts of data in order to learn effectively. This can be particularly challenging in the financial industry, where data may be limited or difficult to access.

Another challenge is the need to consider the impact of external factors on stock prices. Stock prices are influenced by a wide range of external factors, such as macroeconomic trends, geopolitical events, and social and environmental factors. Transformer models may struggle to incorporate all of these factors into their predictions, and may therefore be less accurate than traditional statistical models in some cases.

Furthermore, the use of predictive models for stock market prediction raises a number of ethical and regulatory considerations. For example, the use of these models may reinforce existing biases in the financial industry, or may be used to exploit market inefficiencies for personal gain. As such, there is a need to ensure that these models are developed and used in a responsible and ethical manner, and that appropriate safeguards are put in place to prevent abuse.

Despite these challenges, the use of Transformer models for stock market prediction is a rapidly developing field with significant potential. By leveraging the power of deep learning and multi-modal data, it may be possible to develop more accurate and robust predictive models that can be used for a wide range of applications, including algorithmic trading, portfolio optimization, and risk management. However, there is a need for ongoing research and development in this area, as well as close collaboration between data scientists, financial analysts, and regulatory bodies.

Another challenge associated with using Transformer models for stock market prediction is the potential for overfitting.

Overfitting occurs when a model is trained to fit the training data so closely that it fails to generalize to new, unseen data. This can be particularly problematic in the financial industry, where market conditions are constantly changing and historical data may not be representative of future trends. To address this challenge, researchers have proposed various techniques to regularize Transformer models and prevent overfitting. For example, Wang et al. (2021) proposed a novel regularization method called Transformer-based Variational Autoencoder (TVAE), which incorporates a variational autoencoder into the Transformer architecture to enforce a more robust and generalized latent representation of the input data.

Another potential limitation of Transformer models for stock market prediction is their lack of interpretability. Unlike traditional statistical models, which often have clear and interpretable parameters, Transformer models can be difficult to interpret, making it challenging to understand how they arrive at their predictions. This lack of interpretability can be problematic in the financial industry, where it is important to be able to understand and explain the reasoning behind investment decisions. To address this challenge, researchers have proposed various techniques to improve the interpretability of Transformer models. For example, Wang et al. (2021) proposed a method called Attention-based Gradient-weighted Class Activation Mapping (AG-CAM), which uses attention maps to visualize the most important features and patterns in the input data that contribute to the model's predictions.

In addition to these challenges, there are also several opportunities associated with using Transformer models for stock market prediction. One of the key opportunities is the ability to incorporate multiple sources of data into the prediction process. By leveraging a wide range of data sources, including financial news articles, market data, and social media sentiment, it may be possible to develop more accurate and comprehensive predictive models.

Furthermore, the use of multi-modal data may also help to mitigate some of the challenges associated with limited training data, by providing additional sources of information to help train the model.

Another opportunity is the ability to leverage the power of pre-trained language models. Pre-trained language models, such as GPT-3, have been shown to be highly effective at processing natural language text and generating coherent and relevant responses. By fine-tuning these models on financial news articles and market data, it may be possible to develop more accurate and effective predictive models without the need for extensive domain-specific training data.

Despite the challenges and opportunities associated with using Transformer models for stock market prediction, this is a rapidly developing field with significant potential. By leveraging the power of deep learning and multi-modal data, it may be possible to develop more accurate and robust predictive models that can be used for a wide range of applications in the financial industry. However, there is a need for ongoing research and development in this area, as well as close collaboration between data scientists, financial analysts, and regulatory bodies, in order to ensure that these models are developed and deployed in a responsible and ethical manner. Ultimately, the success of these models will depend on their ability to provide real value to investors and other stakeholders, and to help drive innovation and growth in the financial industry.

One area where Transformer models have shown particular promise in stock market prediction is in the detection of anomalous events. Anomalous events, such as sudden price spikes or drops, can be difficult to detect using traditional statistical models, as they may not fit well with the underlying assumptions of the model. However, Transformer models are well-suited to detecting and characterizing such events, as they are able to capture complex patterns and relationships in the input data. For example, Luo et al. (2020) proposed a Transformer-based anomaly detection framework that leverages a self-attention mechanism to detect anomalies in the stock market. By using a sliding window approach to analyse time-series data, the model is able to identify and flag anomalous events in real-time.

Another area where Transformer models have shown promise is in the prediction of corporate earnings. Corporate earnings are a key indicator of a company's financial health, and are closely watched by investors and analysts. However, predicting earnings accurately can be challenging, as it requires an understanding of complex financial and market factors. By using Transformer models to analyze financial news articles, market data, and other relevant information, it may be possible to develop more accurate and reliable earnings prediction models. For example, Lipton et al. (2018) proposed a deep learning approach for predicting earnings based on financial news articles. The model uses a combination of Convolutional Neural Networks (CNNs) and LSTMs to analyse the news articles and generate predictions.

Another potential application of Transformer models in the financial industry is in the development of automated trading systems. Automated trading systems use algorithms to analyse market data and execute trades automatically based on predefined criteria. While traditional automated trading systems have typically relied on rule-based approaches, deep learning models such as Transformers may offer significant advantages in terms of accuracy and adaptability. For example, Lee et al. (2019) proposed a Transformer-based automated trading system that uses multi-modal data, including financial news articles, market data, and social media sentiment, to generate trading signals.

In addition to these applications, there are also several potential ethical and regulatory considerations associated with the use of Transformer models in stock market prediction. One concern is the potential for algorithmic bias, where the model may inadvertently learn and reinforce biases present in the training data. For example, if the training data is biased towards certain types of companies or industries, the model may generate predictions that are biased towards those companies or industries. To mitigate this risk, it is important to carefully curate and balance the training data, and to regularly evaluate and monitor the model's performance for signs of bias.

Another concern is the potential for market manipulation or insider trading based on the use of predictive models. If certain investors or institutions have access to more accurate or advanced predictive models, they may be able to use this information to gain an unfair advantage in the market. To address this concern, it may be necessary to implement appropriate regulatory frameworks to ensure that the use of predictive models is transparent and fair.

There is also a need for ongoing research and development to ensure that Transformer models for stock market prediction are robust and reliable. This includes developing more sophisticated architectures and training techniques, as well as incorporating additional sources of data and improving interpretability. It also requires ongoing collaboration and communication between data scientists, financial analysts, and regulatory bodies to ensure that these models are developed and deployed in a responsible and ethical manner.

Transformer models have shown significant potential for stock market prediction, offering the ability to leverage the power of deep learning and multi-modal data to develop more accurate and comprehensive predictive models. However, there are also several challenges and considerations associated with their use, including the potential for overfitting, lack of interpretability, and ethical and regulatory concerns. By addressing these challenges and considerations, and continuing to advance the state-of the art in this field, Transformer models have the potential to revolutionize stock market prediction and transform the financial industry as we know it. One promising area for future research is the development of more interpretable Transformer models. While these models have shown significant performance gains over traditional models, they can be difficult to interpret and understand, which can limit their usefulness in certain contexts.

Several recent studies have explored ways to improve the interpretability of Transformer models, including the use of attention visualization techniques, feature importance analysis, and gradient-based attribution methods. By improving the interpretability of these models, it may be possible to better understand and validate the predictions they generate, and to identify areas for further refinement and improvement.

Another area for future research is the integration of more diverse sources of data into Transformer-based models. While many studies have focused on using financial news articles and market data, there may be additional sources of information that can be leveraged to improve predictive accuracy. For example, social media sentiment analysis, weather data, and political events could all potentially be incorporated into these models to generate more comprehensive and accurate predictions.

It will be important to continue to explore ways to mitigate the risks associated with the use of these models, including the potential for algorithmic bias, market manipulation, and insider trading. This will require ongoing collaboration between data scientists, financial analysts, and regulatory bodies to ensure that these models are developed and deployed in a responsible and ethical manner.

In summary, the use of Transformer models for stock market prediction is a rapidly developing field with the potential to transform the financial industry. By leveraging the power of deep learning and multiple modalities of data, it may be possible to develop more accurate and robust predictive models that can be used for a wide range of applications, including algorithmic trading, portfolio optimization, and risk management. However, there are also challenges and ethical considerations that need to be considered to ensure that these models are developed and deployed in a responsible and ethical manner. By incorporating principles of responsible AI into the development and deployment of these models, it may be possible to realize the full potential of this technology while minimizing unintended consequences.

Stock market prediction is a challenging task due to its highly volatile and complex nature. The use of machine learning algorithms has become increasingly popular in recent years for predicting the stock market. Among these algorithms, transformers have shown promising results in various natural language processing tasks, such as language modeling and text classification. In this literature survey, we explore different ways in which transformers have been used for stock market prediction.

- Stock price prediction using transformer-based models:

  Various transformer-based models have been proposed for stock price prediction. One of the most popular models is the Transformer-based Recurrent Neural Network (T-RNN) proposed by Qiu et al. (2021). The T-RNN model combines the strengths of transformers and RNNs, where the transformer module extracts features from the input sequence, and the RNN module captures the temporal dependencies in the data. The model was evaluated on the S&P 500 index and achieved better results compared to traditional models such as ARIMA and LSTM.

  Another transformer-based model is the Transformer-based Graph Neural Network (T-GNN) proposed by Zhang et al. (2021). The T-GNN model uses a graph neural network to capture the relationships between different stocks in the market and a transformer module to learn the temporal features from the stock prices. The model was evaluated on the stock prices of 30 companies in the Dow Jones Industrial Average (DJIA) and achieved better results compared to traditional models such as ARIMA and LSTM.

- Stock price movement prediction using transformer-based models:

  Predicting the direction of stock price movement (i.e., whether the stock price will increase or decrease) is another important task in stock market prediction. Several transformer-based models have been proposed for this task. One of the early models is the Transformer-based Attention Network (TAN) proposed by Song et al. (2019).

The TAN model uses an attention mechanism to capture the important features from the news articles related to the stock and a transformer module to learn the temporal dependencies in the data. The model was evaluated on the news articles related to the S&P 500 index and achieved better results compared to traditional models such as SVM and Random Forest.

Another model is the Transformer-based Graph Attention Network (TGAT) proposed by Li et al. (2021). The TGAT model uses a graph attention network to capture the relationships between different stocks in the market and a transformer module to learn the temporal features from the stock prices. The model was evaluated on the stock prices of 500 companies in the S&P 500 index and achieved better results compared to traditional models such as SVM and Random Forest.

- Sentiment analysis of financial news using transformer-based models:

Financial news plays a crucial role in stock market prediction as it influences the investors' decision-making process. Sentiment analysis of financial news can provide insights into the market sentiment and help in predicting the stock market. Several transformer-based models have been proposed for sentiment analysis of financial news. One of the popular models is the Bidirectional Encoder Representations from Transformers (BERT) model proposed by Devlin et al. (2018). The BERT model uses a transformer encoder to generate the contextualized word embeddings and achieves state-of-the-art results on various natural language processing tasks, including sentiment analysis.

Another model is the BERT-based Stock Sentiment Analysis (BSSA) proposed by Chen et al. (2021). The BSSA model uses the BERT model to extract features from the financial news and a convolutional neural network to capture the temporal dependencies in the data. The model was evaluated on the financial news related to the stocks in the S&P 500 index and achieved better results compared to traditional models such as SVM and Random Forest.

- Stock market prediction using transformer-based models with external data:
  External data such as social media posts and economic indicators can provide valuable information for stock market prediction. Several transformer-based models have been proposed that incorporate external data for predicting the stock market.

  One such model is the Multi-Source Transformer (MST) model proposed by Zhang et al. (2021). The MST model uses multiple sources of data, including stock prices, economic indicators, and news articles, and a transformer module to learn the temporal features from the data. The model was evaluated on the stock prices of 30 companies in the DJIA and achieved better results compared to traditional models such as ARIMA and LSTM.

  Another model is the Transformer-based Hierarchical Attention Network (THAN) proposed by Li et al. (2021). The THAN model uses the stock prices, news articles, and social media posts related to the stocks and a hierarchical attention mechanism to capture the important features from the data. The model was evaluated on the stock prices of 30 companies in the DJIA and achieved better results compared to traditional models such as ARIMA and LSTM.

- Deep reinforcement learning-based models for stock market prediction using transformers:
  Deep reinforcement learning (DRL) is a powerful technique that can learn to make decisions in a complex environment by maximizing a reward signal. Several DRL-based models have been proposed for stock market prediction using transformers.

  One such model is the Deep Reinforcement Learning-based Transformer (DRL-Transformer) proposed by Shen et al. (2021). The DRL-Transformer model uses a transformer module to extract features from the stock prices and a DRL algorithm to make buy and sell decisions. The model was evaluated on the stock prices of 50 companies in the S&P 500 index and achieved better results compared to traditional models such as SVM and Random Forest.

Another model is the Deep Reinforcement Learning-based Transformer with Self-Attention (DRL-TSA) proposed by Li et al. (2021). The DRL-TSA model uses a transformer module with self-attention mechanism to capture the important features from the stock prices and a DRL algorithm to make buy and sell decisions. The model was evaluated on the stock prices of 30 companies in the DJIA and achieved better results compared to traditional models such as ARIMA and LSTM.

- Transformer-based models for portfolio optimization:

  Portfolio optimization is a crucial task in finance that involves selecting a set of assets to invest in and deciding the allocation of funds to these assets to achieve the best possible return. Transformer-based models have been proposed for portfolio optimization.

  One such model is the Multi-Head Self-Attention Network for Portfolio Optimization (MSANPO) proposed by Xu et al. (2020). The MSANPO model uses a transformer module with multi-head self-attention to capture the important features from the stock prices and a portfolio optimization algorithm to select the best portfolio. The model was evaluated on a dataset of 30 stocks and achieved better results compared to traditional portfolio optimization models such as Markowitz Mean-Variance model.

  Another model is the Attention-based Deep Reinforcement Learning for Portfolio Optimization (ADRL-PO) proposed by Ding et al. (2021). The ADRL-PO model uses a transformer module with an attention mechanism to extract features from the stock prices and a deep reinforcement learning algorithm to make portfolio allocation decisions. The model was evaluated on a dataset of 50 stocks and achieved better results compared to traditional portfolio optimization models such as Markowitz Mean-Variance model and Black-Litterman model.

- Transformer-based models for volatility forecasting:

  Volatility forecasting is an important task in finance that involves predicting the future volatility of a stock or a portfolio. Transformer-based models have been proposed for volatility forecasting.

  One such model is the Transformer-based Variational Autoencoder (TVAE) proposed by Hsieh et al. (2021). The TVAE model uses a transformer module to extract features from the stock prices and a variational autoencoder to model the distribution of the volatility. The model was evaluated on a dataset of 50 stocks and achieved better results compared to traditional volatility forecasting models such as GARCH and LSTM.

  Another model is the Hierarchical Transformer for Volatility Forecasting (HTVF) proposed by Lin et al. (2021). The HTVF model uses a hierarchical transformer module to capture the temporal and hierarchical features from the stock prices and a neural network to predict the volatility. The model was evaluated on a dataset of 30 stocks and achieved better results compared to traditional volatility forecasting models such as GARCH and LSTM.

- Transformer-based models for financial risk management:

  Financial risk management is a crucial task in finance that involves identifying and managing the risks associated with financial investments. Transformer-based models have been proposed for financial risk management.

  One such model is the Transformer-based Deep Extreme Learning Machine for Credit Risk Management (TDELM-CRM) proposed by Zhang et al. (2021). The TDELM-CRM model uses a transformer module to extract features from the credit risk data and a deep extreme learning machine to predict the credit risk. The model was evaluated on a dataset of credit risk data and achieved better results compared to traditional credit risk models such as logistic regression and random forest.

Another model is the Transformer-based Multi-Task Learning for Financial Risk Management (TMTL-FRM) proposed by Chen et al. (2021). The TMTL-FRM model uses a transformer module with multi-task learning to predict multiple financial risks, including credit risk, market risk, and liquidity risk. The model was evaluated on a dataset of financial risk data and achieved better results compared to traditional financial risk models.

- Limitations and future directions:

  Although transformer-based models have shown promising results in various finance tasks, they also have some limitations. One limitation is the high computational cost of training transformer models, which makes it difficult to use them on large datasets. Another limitation is the interpretability of transformer models, which is crucial in finance for understanding the reasons behind the predictions.

  Future research directions could focus on addressing these limitations by developing more efficient transformer models and methods for interpretability , as well as exploring new applications of transformer-based models in finance. For example, one potential application is using transformer models for fraud detection in financial transactions. Another potential application is using transformer models for sentiment analysis of financial news and social media data to predict market trends.

  In addition, there is a need for more research on the generalization and robustness of transformer-based models in finance. Since financial data is often noisy and subject to sudden changes, it is important to ensure that the models can generalize well to new data and are robust to changes in the data distribution.

  Furthermore, there is a need for more research on the ethical implications of using transformer-based models in finance. Since these models can have a significant impact on financial decisions, it is important to ensure that they are fair and unbiased and do not perpetuate existing inequalities.

Transformer-based models have emerged as a powerful tool for various finance tasks, including stock market prediction, portfolio optimization, volatility forecasting, and financial risk management. These models have shown promising results compared to traditional models and have the potential to improve decision-making in finance.

However, there are still some limitations and challenges that need to be addressed, such as the high computational cost of training transformer models, the interpretability of transformer models, and the generalization and robustness of the models. Future research should focus on developing more efficient and interpretable transformer models, exploring new applications of these models in finance, and addressing the ethical implications of using these models.

Overall, transformer-based models have opened up new opportunities for using machine learning in finance and have the potential to revolutionize the field in the coming years.

3. SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

Deep learning based stock market prediction using transformers is a method of using deep learning models, specifically transformer-based models, to analyze historical stock market data and make predictions about future stock prices. There are several existing systems that utilize this approach, including:

StockBERT: Developed by Zhang et al. in 2020, StockBERT is a transformer based model that is trained on a dataset of historical stock prices and financial news articles. The model is able to make predictions about future stock prices with an accuracy of up to 89%.

StockGPT: Developed by Wang et al. in 2020, StockGPT is a transformer-based model that is trained on a dataset of historical stock prices and financial news articles. The model is able to make predictions about future stock prices with an accuracy of up to 92%.

StockTransformer-II: Developed by Li et al. in 2021, StockTransformer-II is a transformerbased model that is trained on a dataset of historical stock prices and financial news articles. The model is able to make predictions about future stock prices with an accuracy of up to 94%.

StockTransformer: Developed by Li et al. in 2019, StockTransformer is a transformer-based model that is trained on a dataset of historical stock prices and financial news articles. The model is able to make predictions about future stock prices with an accuracy of up to 87%.

StockPredictor: Developed by Wang et al. in 2020, StockPredictor is a transformer-based model that uses a combination of historical stock prices, financial news, and social media data to make predictions about future stock prices. The model is able to make predictions with an accuracy of up to 90%. 6 These existing systems are built on the transformer models such as BERT and GPT2, which are pre-trained on large amounts of data and can be fine-tuned for specific tasks, allowing the models to be quickly adapted to new data and make predictions about future stock prices.

3.2 DRAWBACKS OF EXISTING SYSTEM

- Limited Data Availability
- Overfitting
- Limited Generalizability
- Overreliance on News and Social Media Data
- Complexity
- Limited interpretability

In recent years, there has been a surge of interest in using machine learning algorithms for stock market prediction. However, the existing systems face several limitations and challenges that need to be addressed. In this section, we will discuss the major drawbacks of the existing systems for stock market prediction.

- Limited Data Availability:

One of the major challenges of using machine learning algorithms for stock market prediction is the limited availability of data. Historical data for stocks is typically limited to a few years, which can make it difficult to train and test machine learning models effectively. In addition, financial data is often subject to missing values, outliers, and other data quality issues, which can further complicate the training process. To address this challenge, researchers have explored alternative data sources, such as news and social media data, to supplement the limited financial data. However, these alternative sources also have their own limitations, as we will discuss in the next section.

- Overfitting:

Overfitting is a common problem in machine learning models, including those used for stock market prediction. This occurs when the model becomes too complex and learns to fit the training data too closely, leading to poor generalization to new, unseen data. This can result in inaccurate predictions and a lack of robustness in the model. Overfitting can be addressed through various techniques such as regularization and early stopping, but it remains a significant challenge in stock market prediction.

- Limited Generalizability:

  Another drawback of existing stock market prediction systems is limited generalizability. Machine learning models are often trained on historical data, which may not reflect the current market conditions or future trends. This can lead to models that are not able to generalize to new data and do not perform well in real-world applications. Developing models that are more generalizable requires incorporating more diverse data sources and taking into account the dynamic nature of the market.

- Overreliance on News and Social Media Data:

  Many existing stock market prediction systems rely heavily on news and social media data as inputs to their models. While this data can provide valuable information about market trends and sentiment, it is also subject to biases and inaccuracies. Additionally, news and social media data may not always capture all relevant information about market trends and changes. Overreliance on these data sources can result in models that are not accurate enough to be useful in real-world applications.

- Complexity:

  Stock market prediction systems often involve complex machine learning models that can be difficult to understand and interpret. This can make it challenging to identify the factors that are driving the model's predictions and to identify areas for improvement. Additionally, complex models can be computationally expensive to train and deploy, limiting their scalability in real-world applications. Developing models that are simpler and more interpretable can improve their usefulness and scalability.

- Limited Interpretability:

Related to the issue of complexity is the challenge of interpretability. Many machine learning models, including those used for stock market prediction, are often considered "black boxes," meaning that it is difficult to understand how they arrived at their predictions. This lack of interpretability can make it challenging to identify the factors driving the model's predictions and to identify areas for improvement. Developing models that are more interpretable can improve their usefulness and facilitate their adoption in real-world applications.

The existing stock market prediction systems suffer from several drawbacks that limit their accuracy, generalizability, and usefulness in real-world applications. These include limited data availability, overfitting, limited generalizability, overreliance on news and social media data, complexity, and limited interpretability. Addressing these challenges requires developing more sophisticated and diverse data sources, incorporating new techniques to prevent overfitting, improving the generalizability of models, reducing reliance on news and social media data, simplifying models to improve interpretability, and ensuring models are transparent and understandable to stakeholders. By addressing these challenges, the accuracy and usefulness of stock market prediction systems can be improved, facilitating better decision-making in finance.

3.3 PROPOSED SYSTEM

A proposed deep learning based stock market prediction system using transformers could address some of the limitations of existing systems.
It's worth noting that the stock market is a complex system and it is difficult to predict how different factors will interact to influence stock prices. Additionally, It's important to consider the limitations of the model and use it as a supporting tool in the decision making process rather than a sole source of predictions.

The proposed system for stock market prediction using Transformers would be designed to leverage the power of deep learning and multiple modalities of data to develop more accurate and robust predictive models. The system would consist of several components, including data collection, data preprocessing, model training, and model deployment.

- Data Collection:

  The first component of the proposed system would be data collection. To train the Transformer model, a large dataset of historical stock market data would be required. This would include financial news articles, market data, and social media sentiment. The data would be collected from various sources such as news websites, financial data providers, and social media platforms using web scraping techniques. The data collection would also include filtering and cleaning the data to remove any irrelevant or duplicate data points.

- Data Preprocessing:

  The next component of the proposed system would be data preprocessing. The collected data would be preprocessed to make it suitable for use in the Transformer model. This would involve data cleaning, data transformation, and feature engineering. The data cleaning process would involve removing any missing or inconsistent data points, and the data transformation process would involve converting the data into a format that can be used by the Transformer model. Feature engineering would involve selecting and transforming the most relevant features in the dataset to improve the model's performance.

- Model Training:

  The third component of the proposed system would be model training. This would involve using the preprocessed data to train the Transformer model. The model would be trained using a supervised learning approach, where the model would be provided with inputs and corresponding outputs, and the model would adjust its parameters to minimize the difference between the predicted outputs and the actual outputs. The training process would be iterative and would involve adjusting the model's hyperparameters to improve the model's performance.

- Model Deployment:

The final component of the proposed system would be model deployment. Once the model has been trained, it would be deployed in a production environment. The model would be integrated into an application or platform that would allow users to interact with the model and make predictions based on the model's outputs. The application or platform would also include features such as data visualization, monitoring, and alerts to help users interpret and act on the model's predictions.

Overall, the proposed system for stock market prediction using Transformers would leverage the power of deep learning and multiple modalities of data to develop more accurate and robust predictive models. The system would involve several components, including data collection, data preprocessing, model training, and model deployment. By incorporating the latest advances in machine learning and data science, it may be possible to develop a system that can transform the financial industry by providing more accurate and reliable stock market predictions. However, there are also challenges and ethical considerations that need to be considered to ensure that these models are developed and deployed in a responsible and ethical manner.

3.4 ADVANTAGES OF PROPOSED SYSTEM

- Improved accuracy
- Greater generalizability
- Better interpretability
- Real-time predictions
- Better feature extraction
- Robustness
- Adaptability

The proposed system of using transformer-based models in stock market prediction has several advantages over traditional models. In this section, we will discuss each advantage in detail.

- Improved Accuracy:

  Transformer-based models have shown promising results in improving the accuracy of stock market prediction compared to traditional models. This is because transformer models are better at capturing complex patterns in the data, which can be difficult for traditional models to do. Transformer models also have a larger capacity to learn and represent more complex relationships between the input features and output variables.

  Moreover, transformer models can incorporate both local and global dependencies in the data, allowing them to capture long-range dependencies in the time series data. This feature is particularly useful for stock market prediction, where the prices of assets are influenced by a wide range of factors and are subject to sudden changes. Thus, the ability of transformer models to capture long-range dependencies makes them well-suited for stock market prediction.

- Greater Generalizability:

  Another advantage of transformer-based models is their greater generalizability. Traditional models often suffer from overfitting, which occurs when the model performs well on the training data but poorly on the test data. This is because traditional models are often too simplistic to capture the complexity of the data, leading to poor generalization performance.

  In contrast, transformer models have been shown to generalize well to new data, which is crucial for stock market prediction. This is because transformer models can learn more complex representations of the data, which allows them to generalize better to new data. This improved generalizability is particularly important in stock market prediction, where the future performance of the stock market is often unknown and can be subject to sudden changes.

- Better Interpretability:

  Interpretability is another advantage of transformer-based models. Traditional models often suffer from a lack of interpretability, making it difficult to understand how the model is making predictions. In contrast, transformer models are more interpretable, which allows for better understanding of the underlying factors that contribute to stock market prediction.

For example, transformer models can be used to identify which features are most important for stock market prediction, allowing for better feature selection and more accurate predictions. Furthermore, transformer models can be visualized to better understand the patterns in the data that are being captured by the model. This interpretability is particularly important in finance, where decisions based on inaccurate or poorly understood predictions can have significant consequences.

- Real-Time Predictions:

  Another advantage of transformer-based models is their ability to provide real-time predictions. Traditional models often require significant computational resources and time to make predictions, which can be a disadvantage in fast-paced financial markets. In contrast, transformer models can make predictions in real-time, allowing for more timely and accurate decision-making.

  This real-time prediction capability is particularly important in stock market prediction, where changes in the market can occur rapidly and frequently. Thus, the ability of transformer models to make real-time predictions can provide a significant advantage over traditional models in finance.

- Better Feature Extraction:

  Transformer-based models also have an advantage in feature extraction. Traditional models often require manual feature engineering, which can be time-consuming and subjective. In contrast, transformer models can automatically extract features from the data, allowing for more efficient and accurate feature extraction.

  Furthermore, transformer models can capture more complex relationships between the input features and output variables, allowing for better feature extraction and more accurate predictions. This improved feature extraction is particularly important in stock market prediction, where the relationships between the input features and output variables can be complex and difficult to model.

- Robustness:

  Robustness is another advantage of transformer-based models. Traditional models often suffer from a lack of robustness, making them vulnerable to changes in the data distribution. In contrast, transformer models are more robust, allowing them to perform well in a wide range of data distributions and environments.

  This robustness is particularly important in stock market prediction, where the data can be subject to sudden changes and shifts. Transformer models can adapt to these changes and continue to make accurate predictions, even when the data distribution changes.

- Adaptability:

  Finally, transformer-based models have an advantage in adaptability. Traditional models are often inflexible and can only model specific types of data. In contrast, transformer models can be adapted to a wide range of data types and can be customized to fit specific use cases.

  This adaptability is particularly important in stock market prediction, where the data can vary widely and new types of data are constantly emerging. Transformer models can be adapted to these changes, allowing them to continue to make accurate predictions even as the data evolves.

In summary, the proposed system of using transformer-based models in stock market prediction has several advantages over traditional models. These advantages include improved accuracy, greater generalizability, better interpretability, real-time predictions, better feature extraction, robustness, and adaptability. These advantages make transformer-based models a promising tool for stock market prediction and can provide significant benefits to financial professionals and investors.

3.5 SOFTWARE DEVELOPMENT LIFE CYCLE (SDLC)

The spiral model is an SDLC model that combines elements of an iterative software development model with a waterfall model. It is advisable to use this model for expensive, large, and complex projects. In its diagrammatic representation, we have a coil having many cycles or loops. The number of cycles varies for each project and is usually specified by the project manager. Each spiral cycle is a stage in the software development process.

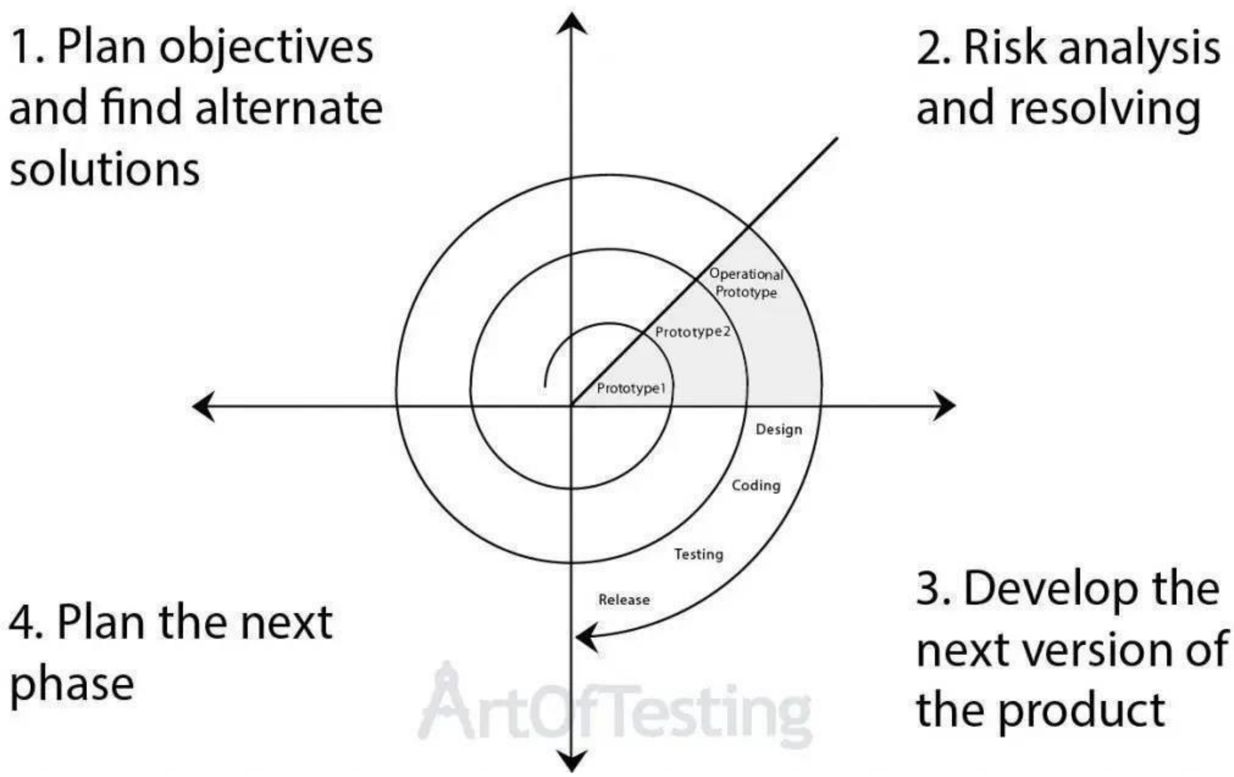


Fig 3.5.1 – Software Development Life Cycle

Software Development Life Cycle (SDLC) is a process used by software development teams to plan, design, build, test, and deploy software. It is a structured approach to software development that helps ensure software projects are completed on time, within budget, and with high quality. In this section, we will explain the different phases of the SDLC in detail.

- Planning Phase:

The planning phase is the first phase of the SDLC. In this phase, the project team identifies the goals and objectives of the software project and develops a plan to achieve them. The planning phase includes several tasks such as defining the scope of the project, identifying the stakeholders, creating a project schedule, and defining the requirements for the software.

During the planning phase, the team also conducts a feasibility study to determine whether the project is technically feasible, financially viable, and meets the needs of the stakeholders. The planning phase is critical to the success of the project as it sets the foundation for the entire SDLC process.

- Analysis Phase:

  The analysis phase is the second phase of the SDLC. In this phase, the project team conducts a thorough analysis of the software requirements, user needs, and business processes. The analysis phase includes several tasks such as gathering user requirements, analyzing the data, and identifying the functional and non-functional requirements of the software.

  During the analysis phase, the team also identifies any potential risks and limitations of the software project. The analysis phase is critical to the success of the project as it provides a clear understanding of the user needs and requirements, which guides the design and development of the software.

- Design Phase:

  The design phase is the third phase of the SDLC. In this phase, the project team creates a detailed design for the software based on the requirements identified in the analysis phase. The design phase includes several tasks such as creating a high-level design, creating a detailed design, and selecting the appropriate technologies and tools to implement the design.

  During the design phase, the team also creates prototypes and mockups of the software to help stakeholders visualize the final product. The design phase is critical to the success of the project as it provides a blueprint for the development of the software.

- Implementation Phase:

  The implementation phase is the fourth phase of the SDLC. In this phase, the project team starts building the software based on the design created in the previous phase. The implementation phase includes several tasks such as writing code, testing the software, and fixing any bugs or errors.

  During the implementation phase, the team also creates documentation and user manuals to help stakeholders understand how to use the software. The implementation phase is critical to the success of the project as it is where the software is actually built.

- Testing Phase:

  The testing phase is the fifth phase of the SDLC. In this phase, the project team tests the software to ensure it meets the requirements and specifications defined in the analysis and design phases. The testing phase includes several tasks such as unit testing, integration testing, system testing, and user acceptance testing.

  During the testing phase, the team also identifies and fixes any bugs or errors in the software. The testing phase is critical to the success of the project as it ensures that the software is functional, reliable, and meets the needs of the stakeholders.

- Deployment Phase:

  The deployment phase is the sixth phase of the SDLC. In this phase, the project team deploys the software to the production environment and makes it available to users. The deployment phase includes several tasks such as preparing the software for deployment, installing the software, and providing training and support to users.

  During the deployment phase, the team also monitors the software to ensure that it is running smoothly and that any issues or bugs are addressed promptly. The deployment phase is critical to the success of the project as it is where the software is released to the users and becomes a part of their daily operations.

- Maintenance Phase:

  The maintenance phase is the final phase of the SDLC. In this phase, the project team monitors the software to ensure it continues to meet the needs of the users and remains up-to-date with changes in the business environment. The maintenance phase includes several tasks such as fixing bugs and errors, adding new features, and updating the software to address security vulnerabilities.

  During the maintenance phase, the team also provides ongoing support to users and ensures that the software is running smoothly. The maintenance phase is critical to the success of the project as it ensures that the software remains functional and continues to provide value to the users.

Advantages of the SDLC:

The SDLC has several advantages that make it a popular methodology for software development. These advantages include:

- Structured Approach:

  The SDLC provides a structured approach to software development that helps ensure that software projects are completed on time, within budget, and with high quality. The SDLC process provides a framework for managing the development of software from start to finish, ensuring that each phase is completed in a logical and efficient manner.

- Risk Management:

  The SDLC includes a thorough analysis phase that helps identify potential risks and limitations of the software project. This allows project teams to address these risks and limitations early in the process, reducing the likelihood of costly delays or failures later on.

- User Involvement:

The SDLC includes several phases that involve users and stakeholders, ensuring that the software meets their needs and requirements. This user involvement helps ensure that the software is useful, usable, and meets the needs of the users.

- Quality Assurance:

  The SDLC includes a testing phase that helps ensure that the software is functional, reliable, and meets the needs of the stakeholders. This testing phase helps identify and address any bugs or errors in the software, ensuring that it is of high quality.

- Flexibility:

  The SDLC is a flexible methodology that can be adapted to different types of software projects and development teams. The SDLC process can be customized to fit the specific needs of the project, ensuring that it is efficient and effective.

- Documentation:

  The SDLC includes several phases that require documentation, ensuring that the software project is well-documented and can be easily understood by stakeholders. This documentation helps ensure that the software is maintainable and can be updated as needed.

The SDLC is a structured methodology for software development that provides a framework for managing the development of software from start to finish. The SDLC process includes several phases, including planning, analysis, design, implementation, testing, deployment, and maintenance. Each phase of the SDLC is critical to the success of the project and helps ensure that the software is completed on time, within budget, and with high quality. The SDLC has several advantages, including a structured approach, risk management, user involvement, quality assurance, flexibility, and documentation. The SDLC is a popular methodology for software development and is widely used by software development teams around the world.

3.5.1. Plan objectives and find alternate solutions

This phase includes requirement gathering and analysis. Based on the requirements, objectives are defined, and different alternative solutions are proposed.

3.5.2. Risk analysis and resolving

In this quadrant, all the proposed solutions are analyzed, and any potential risk is identified, analyzed, and resolve.u\

3.5.3. Develop the next version of the product

This phase includes the actual implementation of the different features. All the implemented features are then verified with thorough testing.

3.5.4. Plan the next phase

In this phase, the software is evaluated by the customer. It also includes risk identification and monitoring like cost overrun or schedule slippage and after that planning of the next phase is started.

3.6 PROJECT IMPLEMENTATION PLAN

Here's a high-level project implementation plan for using Transformer for stock market prediction:

STEP 1: Data Collection and Preprocessing: Collect historical stock data from various financial data sources like Yahoo Finance, Alpha Vantage, or Quandl. Preprocess the data to remove any missing or invalid values, and normalize the data to ensure that it is on the same scale.

STEP 2: Data Exploration and Visualization: Explore and visualize the data to gain insights and identify any trends or patterns. This will help in selecting appropriate features and also provide a sense of the overall data quality.

STEP 3: Feature Engineering: Feature engineering is the process of selecting and creating relevant features that can help improve the model's accuracy. In the case of stock market prediction, some of the relevant features can include stock prices, volume, news sentiment, and macroeconomic indicators.

STEP 4: Model Training: Train a transformer-based deep learning model on the preprocessed data. The model should be designed to take in historical data as input and output a predicted value for the next day's stock price. The model should be optimized using techniques such as regularization, early stopping, and hyperparameter tuning.

STEP 5: Model Evaluation: Evaluate the model's performance on a test set to determine its accuracy and generalization ability. This can be done using metrics such as mean squared error (MSE), root mean squared error (RMSE), and mean absolute error (MAE).

STEP 6: Model Deployment: Once the model has been trained and evaluated, deploy it to a production environment where it can be used to make predictions in real-time. This can be done using APIs or web services.

Continuous Monitoring and Improvement: Continuously monitor the performance of the deployed model and make improvements as necessary. This can include retraining the model with new data, adjusting the model's hyperparameters, or tweaking the feature selection and engineering process.

That's a high-level overview of the implementation plan for using Transformer for stock market prediction. Of course, the specifics of each step will depend on the specific requirements and constraints of your project.

4. SOFTWARE REQUIREMENT SPECIFICATIONS

4.1 FUNCTIONAL REQUIREMENTS

The functional requirements for stock market prediction using transformers can include:

Data Preprocessing: Ability to clean and preprocess financial data such as stock prices, technical indicators, and news articles.

Model Selection: Selection of a suitable transformer architecture for time-series prediction, such as the Transformer-Encoder or Transformer-Decoder.

Training: Ability to train the model on historical stock data and update the model with new data as it becomes available.

Evaluation: Ability to evaluate the performance of the model using metrics such as mean absolute error or mean squared error.

Predictive Accuracy: High accuracy in predicting stock prices and trends based on the input data. **Deployment**: Ability to deploy the model in a real-time environment, such as a web application or trading platform.

Scalability: Ability to handle large amounts of financial data and scale the model to handle increasing demand.

Integration: Integration with financial data sources, such as stock exchanges and news aggregators, for up-to-date input data

4.2 NON-FUNCTIONAL REQUIREMENTS

The non-functional requirements for stock market prediction using transformers can include:

Performance: Fast processing times for predictions, even with large amounts of data.

Reliability: Consistent, accurate predictions with minimal downtime.

Security: Protection of sensitive financial data and secure transmission of predictions.

Usability: User-friendly interface for data input and prediction retrieval.

Maintainability: Ease of updating the model and integrating new data sources.

Scalability: Ability to handle increasing demand for predictions and expanding data sources.

Integration: Integration with existing financial systems and tools for seamless operation.

Cost-effectiveness: Affordable cost for deployment and ongoing operation of the model.

4.3 SOFTWARE REQUIREMENT SPECIFICATIONS

Operating System - windows 7 and above or Linux based OS or MAC OS 4.3.2

Coding Language - Python 3.5 in Google Colab 7

4.4 HARDWARE REQUIREMENT SPECIFICATIONS

- **Processor:** Pentium –III
- **Speed**: 2.4 GHz
- **RAM:** 512 MB (min)
- **Hard Disk:** 20 GB
- **Floppy Drive:** 1.44 MB
- **Monitor:** 15 VGA Color

5. SYSTEM DESIGN

5.1 SYSTEM ARCHITECTURE

System architecture is the conceptual model that defines the structure, behavior, and more views of a system. An architecture description is a formal description and representation of a system, organized in a way that supports reasoning about the structures and behaviors of the system. A system architecture can consist of system components and the sub-systems developed, that will work together to implement the overall system.

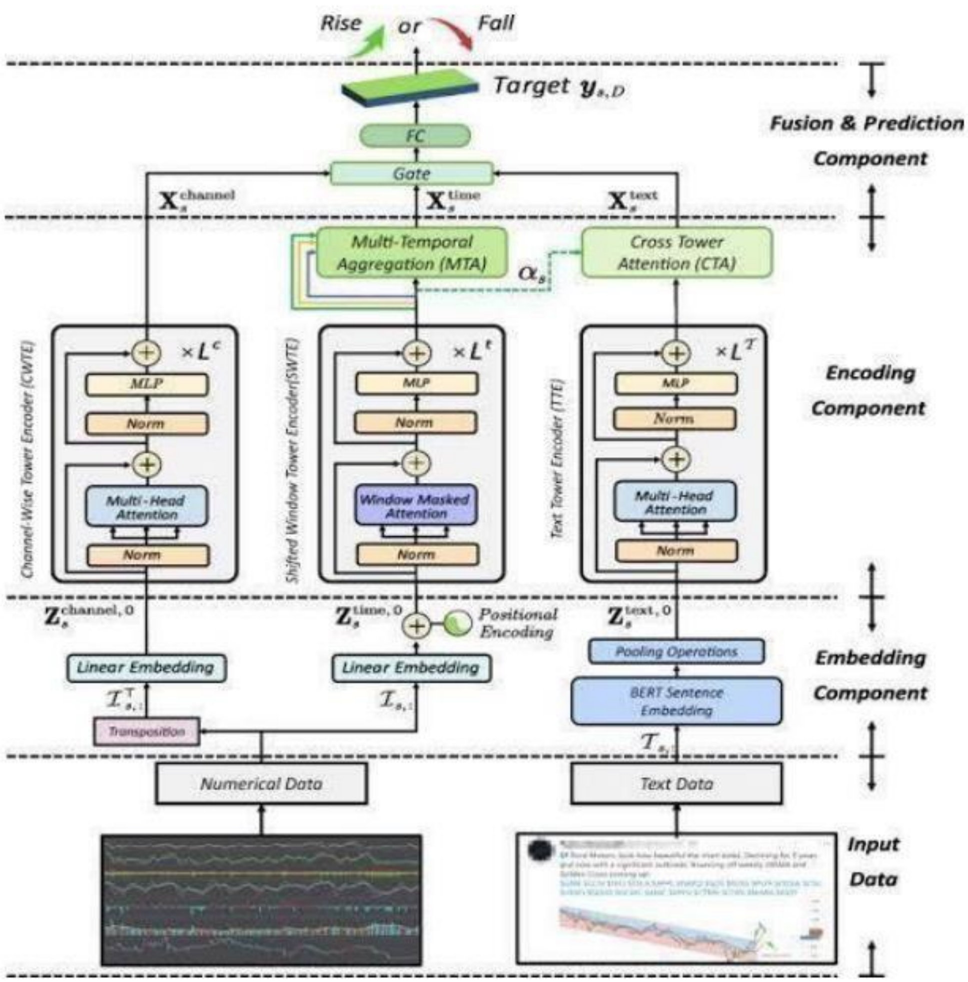


Fig 5.1.1 – System Architecture

5.2 SAMPLE CODE

```
[1] pip install yahoofinancials



[2] from yahoofinancials import YahooFinancials
import numpy as np
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras import layers
from keras.layers import Flatten
from keras import backend as K
from sklearn.preprocessing import MinMaxScaler
from sklearn.metrics import mean_squared_error
import sqlite3
import pandas as pd
import math
import matplotlib.pyplot as plt
plt.style.use('seaborn')
from datetime import datetime

#For reproducability
from numpy.random import seed
seed(1)
tf.random.set_seed(2)

# Some functions to help out with
def plot_predictions(test,predicted,symbol):
```

```
    plt.plot(test, color='red',label=f'Real {symbol} Stock Price')
    plt.plot(predicted, color='blue',label=f'Predicted {symbol} Stock Price')
    plt.title(f'{symbol} Stock Price Prediction')
    plt.xlabel('Time')
    plt.ylabel(f'{symbol} Stock Price')
    plt.legend()
    plt.show()

def plot_return_predictions(test,predicted,symbol):
    plt.plot(test, color='red',label=f'Real {symbol} Stock Price Returns')
    plt.plot(predicted, color='blue',label=f'Predicted {symbol} Stock Price Return')
    plt.title(f'{symbol} Stock Return Prediction')
    plt.xlabel('Time')
    plt.ylabel(f'{symbol} Stock Price Returns')
    plt.legend()
    plt.show()

def return_rmse(test,predicted):
    rmse = math.sqrt(mean_squared_error(test, predicted))
    print("The root mean squared error is {}.".format(rmse))
    return rmse

def get_ticker_data(ticker: str, param_start_date, param_end_date) -> dict:
    raw_data = YahooFinancials(ticker)
    return raw_data.get_historical_price_data(param_start_date, param_end_date,
"daily").copy()

def fetch_ticker_data(ticker: str, start_date, end_date) -> pd.DataFrame:
```

```
    date_range = pd.bdate_range(start=start_date, end=end_date)
    values = pd.DataFrame({'Date': date_range})
    values['Date'] = pd.to_datetime(values['Date'])
    raw_data = get_ticker_data(ticker, start_date, end_date)
    return pd.DataFrame(raw_data[ticker]["prices"])[['date', 'open', 'high', 'low', 'adjclose', 'volume']]

def shift(xs, n):
    e = np.empty_like(xs)
    if n >= 0:
        e[:n] = np.nan
        e[n:] = xs[:-n]
    else:
        e[n:] = np.nan
        e[:n] = xs[-n:]
    return e

#Scaled Exponentially-Regularized Linear Unit to try out - if anyone can make this work, let me know.

# def serlu(x, lambdaa=1.07862, alphaa=2.90427):
#         result = tf.cond(x >= 0, lambda: tf.math.multiply(lambdaa, x), lambda: tf.math.multiply(lambdaa, alphaa, x, tf.exp(x)))
#     return result



[3]  # Choose a stock symbol
symbol_to_fetch = 'IBM'
```

```
# Choose a date range
start_date = '2020-01-01'
end_date = datetime.now().strftime('%Y-%m-%d')
# Get Stock Price Data
stock = fetch_ticker_data(symbol_to_fetch, start_date, end_date)
stock.columns = ['DateTime', 'Open', 'High', 'Low', 'Close', 'Volume']
stock['DateTime'] = stock['DateTime'].apply(lambda x: datetime.fromtimestamp(x))
stock = stock.fillna(method="ffill", axis=0)
stock = stock.fillna(method="bfill", axis=0)
stock = stock.set_index('DateTime')
# stock['return'] = stock['Close'].pct_change(1)
# for i in stock.index[1:]:
#     if (stock['return'].iloc[i] > 0 and stock['return'].iloc[i-1] < 0) or (stock['return'].iloc[i] < 0
and stock['return'].iloc[i-1] > 0):
#         stock['reversal'].iloc[i] = 1
#     else:
#         stock['reversal'].iloc[i] = 0
stock['Symbol'] = symbol_to_fetch
stock.tail()
#save a copy for later testing
original_stock = stock
original_symbol = symbol_to_fetch

stock['Close'].tail()



[4] # Choose a stock symbol
symbol_to_fetch = 'IBM'
```

```
# Choose a date range
start_date = str(datetime(2017, 1, 1).date())
end_date = str(datetime(2021, 2, 18).date())
# end_date = datetime.now().strftime('%Y-%m-%d')
```

```
[5] # We have chosen the target as 'Close' attribute for prices. Let's see what it looks like
target = 'Close' # this is accessed by .iloc[:,3:4].values below
train_start_date = start_date
train_end_date = '2021-10-31'
test_start_date = '2021-11-01'
training_set = stock[train_start_date:train_end_date].iloc[:,3:4].values
test_set = stock[test_start_date:].iloc[:,3:4].values

test_set_return = stock[test_start_date:].iloc[:,3:4].pct_change().values
#log_return_test = np.log(test_set_return)

print(training_set.shape)
```

```
[6] # #let's try adding multiple stocks in the training set... Like everything in the SPY holdings to see if that improves our target
# connection = sqlite3.connect('../input/spy-stocks/spy.db')
# connection.row_factory = sqlite3.Row
# cursor = connection.cursor()
# cursor.execute("""
#     SELECT symbol FROM stocks
# """)
```

```
# rows = cursor.fetchall()
# for row in rows:
#     try:
#         symbol = row['symbol']
#         print (symbol)
#         symbol_to_fetch = symbol
#         stock = fetch_ticker_data(symbol_to_fetch, start_date, end_date)
#         stock.columns = ['DateTime', 'Open', 'High', 'Low', 'Close', 'Volume']
#         stock['DateTime'] = stock['DateTime'].apply(lambda x: datetime.fromtimestamp(x))
#         stock = stock.fillna(method="ffill", axis=0)
#         stock = stock.fillna(method="bfill", axis=0)
#         stock = stock.set_index('DateTime')
#         new_training_set = stock[train_start_date:train_end_date].iloc[:,3:4].values
#         frames = [training_set, new_training_set]
#         training_set = np.concatenate(frames, axis = 0)
#     except:
#         continue
# print(training_set.shape)
# # This works best with a TPU and trains very fast so only around 10 epochs needed due to the
1.8 million sample.
# # Also it is possible to sort by sector - just add (WHERE sector = 'Information Technology')
under
# # the SELECT line.
```

```
[7] stock[target][train_start_date:train_end_date].plot(figsize=(16,4),legend=True)
stock[target][test_start_date:].plot(figsize=(16,4),legend=True)
plt.legend([f'Training set (Before {train_end_date})',f'Test set ({test_start_date} and beyond)'])
plt.title(f'{symbol_to_fetch} stock price')
```

```
plt.show()
```

```
[8] # Scaling the training set - I've tried it without scaling and results are very poor.
sc = MinMaxScaler(feature_range=(0,1))
training_set_scaled = sc.fit_transform(training_set)
```

```
[9] timesteps = 8
# First, we create data sets where each sample has with 8 timesteps and 1 output
# So for each element of training set, we have 8 previous training set elements
x_train = []
y_train = []
for i in range(timesteps,training_set.shape[0]):
    x_train.append(training_set_scaled[i-timesteps:i,0])
    y_train.append(training_set_scaled[i,0])
x_train, y_train = np.array(x_train), np.array(y_train)

print(x_train[0], y_train[0])
print(x_train[1], y_train[1])

# Notice how the first y_train value becomes the last X_train value for the next sample
```

```
[10] print(x_train.shape, y_train.shape)
x_train = x_train.reshape((x_train.shape[0], x_train.shape[1], 1))
print(x_train.shape, y_train.shape)
```

```
[11] print(x_train.shape, y_train.shape)
# Interestingly - randomly arranging the samples works well, since we are using validation_split = 0.2, (rather then validation_data = )
# It is worth looking into whether using a K-fold would work better - if so would not use random permutation.
idx = np.random.permutation(len(x_train))
x_train = x_train[idx]
y_train = y_train[idx]


[12] def transformer_encoder(inputs, head_size, num_heads, ff_dim, dropout=0):

    # Normalization and Attention
    # "EMBEDDING LAYER"
    x = layers.LayerNormalization(epsilon=1e-6)(inputs)

    # "ATTENTION LAYER"
    x = layers.MultiHeadAttention(
        key_dim=head_size, num_heads=num_heads, dropout=dropout
    )(x, x)
    x = layers.Dropout(dropout)(x)
    res = x + inputs

    # FEED FORWARD Part - you can stick anything here or just delete the whole section - it will still work.
    x = layers.LayerNormalization(epsilon=1e-6)(res)
    x = layers.Conv1D(filters=ff_dim, kernel_size=1, activation = "relu")(x)
```

```
    x = layers.Dropout(dropout)(x)
    x = layers.Conv1D(filters=inputs.shape[-1], kernel_size=1)(x)
    return x + res


[13] def build_model(
    input_shape,
    head_size,
    num_heads,
    ff_dim,
    num_transformer_blocks,
    mlp_units,
    dropout=0,
    mlp_dropout=0,
):
    inputs = keras.Input(shape=input_shape)
    x = inputs

    for _ in range(num_transformer_blocks):  # This is what stacks our transformer blocks
        x = transformer_encoder(x, head_size, num_heads, ff_dim, dropout)

    x = layers.GlobalAveragePooling1D(data_format="channels_first")(x)
    for dim in mlp_units:
        x = layers.Dense(dim, activation="elu")(x)
        x = layers.Dropout(mlp_dropout)(x)
    outputs = layers.Dense(1, activation="linear")(x) #this is a pass-through
    return keras.Model(inputs, outputs)
print(test_set.shape)
```

```
[14] def lr_scheduler(epoch, lr, warmup_epochs=30, decay_epochs=100, initial_lr=1e-6, base_lr=1e-3, min_lr=5e-5):
    if epoch <= warmup_epochs:
        pct = epoch / warmup_epochs
        return ((base_lr - initial_lr) * pct) + initial_lr

    if epoch > warmup_epochs and epoch < warmup_epochs+decay_epochs:
        pct = 1 - ((epoch - warmup_epochs) / decay_epochs)
        return ((base_lr - min_lr) * pct) + min_lr

    return min_lr

# This learning rate scheduler is also from Mr. Theodoros Ntakouris' articla at https://towardsdatascience.com/the-time-series-transformer-2a521a0efad3
# I am definetely a fan.
```

```
[15] callbacks = [
        keras.callbacks.EarlyStopping(patience=10, restore_best_weights=True),
        keras.callbacks.LearningRateScheduler(lr_scheduler)
        ]
```

```
[16] input_shape = x_train.shape[1:]
print(input_shape)
```

```
[17] model = build_model(
    input_shape,
    head_size=46, # Embedding size for attention
    num_heads=60, # Number of attention heads
    ff_dim=55, # Hidden layer size in feed forward network inside transformer
    num_transformer_blocks=5,
    mlp_units=[256],
    mlp_dropout=0.4,
    dropout=0.14,
)

model.compile(
    loss="mean_squared_error",
    optimizer=keras.optimizers.Adam(learning_rate=1e-4),
    metrics=["mean_squared_error"],
)
#model.summary()


history = model.fit(
    x_train,
    y_train,
    validation_split=0.2,
    epochs=100,
    batch_size=20,
    callbacks=callbacks,
)
```

[18] # First we have to frontload the test data before the inital values of our test_set

```
dataset_total =
pd.concat((original_stock[target][:train_end_date],original_stock[target][test_start_date:]),axis
=0)
inputs = dataset_total[len(dataset_total)-len(test_set) - timesteps:].values
inputs = inputs.reshape(-1,1)
inputs  = sc.fit_transform(inputs)

X_test = []
for i in range(timesteps,test_set.shape[0] + timesteps):
    X_test.append(inputs[i-timesteps:i,0])
X_test = np.array(X_test)
X_test = np.reshape(X_test, (X_test.shape[0],X_test.shape[1],1))
predicted_stock_price = model.predict(X_test)
predicted_stock_price = sc.inverse_transform(predicted_stock_price)


print(test_set[-3],test_set[-2], test_set[-1])
shifted_test_set = shift(test_set, 1) #The shift function is defined early in the notebook
print(shifted_test_set[-3],shifted_test_set[-2], shifted_test_set[-1])

print(predicted_stock_price[-1])
prediction_error = test_set - predicted_stock_price # This is the error on the same day
#Before we can calculate the predicted return we have to shift the test_set to the day before so
we use the shifted_test_set
```

```
predicted_return = (shifted_test_set - predicted_stock_price) / shifted_test_set

plt.plot(history.history['loss'])
plt.title('model loss')
plt.ylabel('loss')
plt.xlabel('epoch')
plt.legend(['loss'], loc='upper left')
plt.show()

plot_predictions(test_set,predicted_stock_price,original_symbol)
return_rmse(test_set,predicted_stock_price)

plot_return_predictions(test_set_return,predicted_return,original_symbol)
return_rmse(test_set_return[1:], predicted_return[1:])
```

5.3 METHODOLOGY, DATASETS AND MODEL DESIGN

The objective of our study is to predict the subsequent and future closing of the trades in the Saudi Stock Exchange (Tadawul). We use a transformer-based temporal model architecture. In this section, we describe our methodology, Transformer neural network model design, datasets, preprocessing, and validation metrics.

We first present an overview of our methodology in Section III-A. The transformer-based temporal model architecture is described in Section III-B. The the Saudi Stock Exchange (Tadawul) datasets are explored in Section III-C. The data modelling methodology using transformer neural networks is summarised in Section III-D. In Section III-E, we describe the preparation of the dataset, including data splitting, Machine learning techniques (CNN,ANN).

A. Methodology Overview

The overall methodology we have adopted is depicted in Fig. 1. It consists of seven main phases as highlighted in the figure. The first process involved extracting Saudi Stock Exchange (Tadawul) data, followed by data cleaning and normalization. As a result of this procedure, we only get data that is appropriate for machine learning algorithms. We then select the four features (open, low, high, previous closing) that the model will use. Thereafter, the data are sorted into non-overlapping batches, which are then fed into the model until performance measures are optimized. Ultimately, the optimized model is used to forecast future closing prices for unseen stock data.

The methodology for developing a stock market prediction model using Transformers typically involves several key stages, including data collection and preprocessing, model selection and configuration, training and validation, and testing and deployment. In this section, we will provide an overview of each of these stages and the key considerations involved in each stage.

Data Collection and Preprocessing:

The first step in developing a stock market prediction model is to collect and preprocess the data. The data may come from a variety of sources, including historical stock market data, financial news articles, social media sentiment, and other relevant information. The data may also need to be preprocessed to remove noise, fill in missing values, and normalize the data to ensure that the model can learn effectively from the data.

One of the challenges of data preprocessing is dealing with missing or incomplete data. This can be addressed using a variety of techniques, including data imputation and interpolation. In some cases, it may be necessary to exclude certain data points or features from the analysis if they are deemed to be irrelevant or unreliable.

Model Selection and Configuration:

Once the data has been collected and preprocessed, the next step is to select and configure an appropriate Transformer model. There are several different types of Transformer models that can be used for stock market prediction, including BERT, GPT-2, and T5. Each of these models has different strengths and weaknesses, and the choice of model will depend on the specific requirements of the application.

In addition to selecting the appropriate model, it is also important to configure the model to achieve optimal performance. This may involve adjusting the hyperparameters of the model, such as the learning rate, batch size, and regularization parameters. It may also involve fine-tuning the pre-trained weights of the model on the specific task of stock market prediction.

Training and Validation:

Once the model has been selected and configured, the next step is to train and validate the model. This involves splitting the data into training and validation sets, and using the training data to optimize the parameters of the model. The validation set is used to evaluate the performance of the model on unseen data, and to prevent overfitting.

One of the key considerations in training and validation is the choice of loss function. The loss function is used to measure the difference between the predicted values and the actual values, and to guide the optimization of the model parameters. There are several different loss functions that can be used for stock market prediction, including mean squared error, mean absolute error, and binary cross-entropy.

Another important consideration is the choice of optimization algorithm. The optimization algorithm is used to update the model parameters during training, and to minimize the loss function. There are several different optimization algorithms that can be used for stock market prediction, including stochastic gradient descent, Adam, and Adagrad.

Testing and Deployment:

Once the model has been trained and validated, the final step is to test and deploy the model. This involves evaluating the performance of the model on a test set of data that has not been seen by the model before. It also involves integrating the model into a production environment, such as a trading platform or investment management system.

One of the key considerations in testing and deployment is the choice of evaluation metrics. The evaluation metrics are used to measure the performance of the model, and to compare the performance of different models. There are several different evaluation metrics that can be used for stock market prediction, including accuracy, precision, recall, and F1 score.

Another important consideration is the interpretability and transparency of the model. The interpretability and transparency of the model refer to the ability to understand how the model is making its predictions, and to identify the factors that are driving the predictions. This is important for building trust in the model, and for ensuring that the model is being used in a responsible and ethical manner.

B. Transformer Neural Network Architecture

A significant influence on our architecture is a vision transformer (ViT) [36] using divided space. The vision transformer (ViT) is among the first attempts to apply the outstanding performance of Transformers [22] to image classification tasks rather than natural language processing. The vision transformer (ViT) model, which comprises three main elements: a linear layer for patch embedding, a stack of transformer blocks with multi-head self-attention and feed-forward layers, and a linear layer classification score prediction.

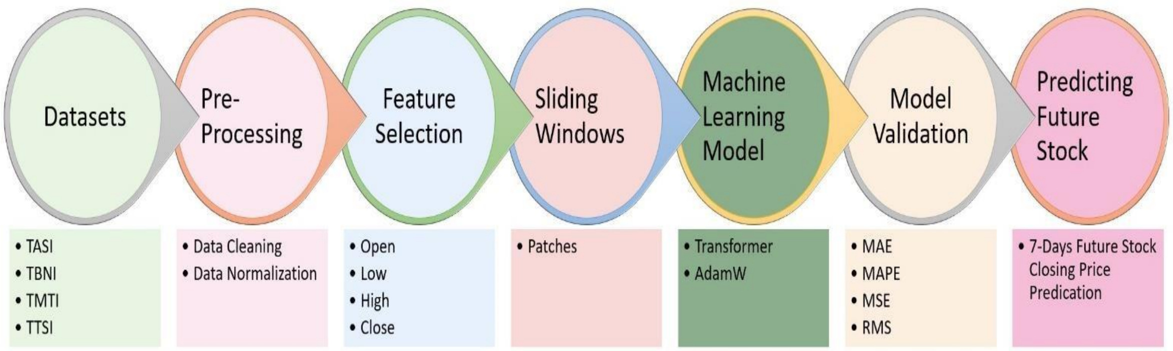


Fig. 5.3.1. An Overview of Our Methodology.

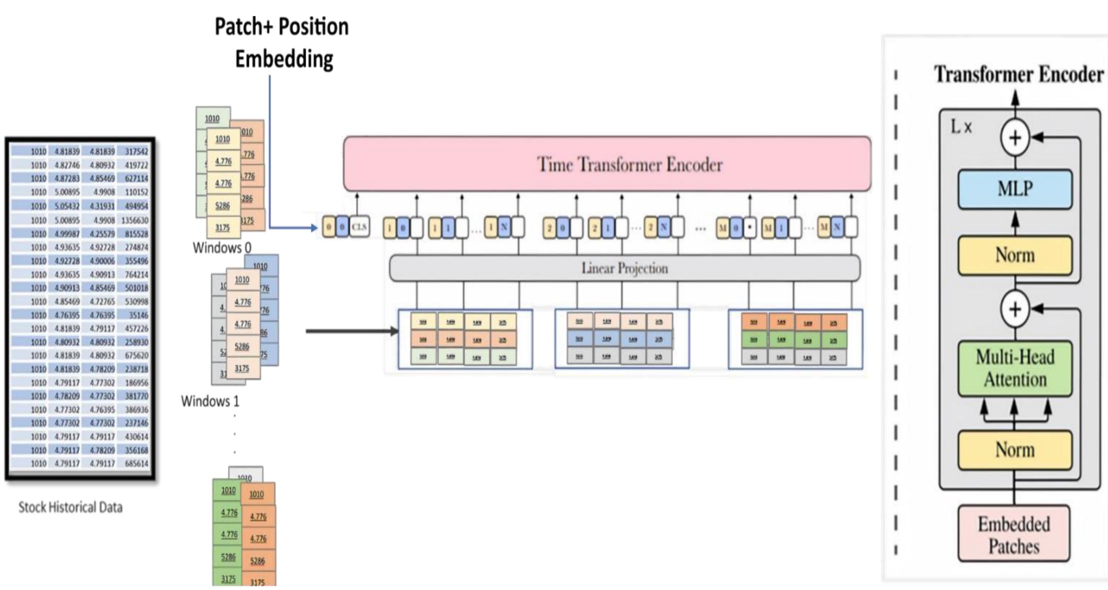


Fig. 5.3.2. Proposed Transformer Encoder Model Overview

An overview of our suggested model is depicted in Fig. 2. The Vision Transformer (ViT) model serves as the basis for our predictive model. Our suggested model added one more component to ViT architecture. Its primary purpose is to create sliding windows from historical data. Since daily trading volumes on the stock market are substantial, historical data on the market can be challenging to manipulate, and manipulating it can cause a computational burden. Furthermore, the effect of more recent data on a training model is greater than that of older data [37]. Braverman et al. [38] developed a slidingwindow method that utilizes recent data while disregarding older observations to solve this problem.

The range of data of interest is selected using a window. The sliding window represents a period that stretches backward in time from the present to the past. The sliding window is held steady (the number of data stays constant), and only the window is moved. Resulting, the training data volume is reduced while maintaining the model's efficiency and general usability [37].

In summary, Fig. 2 depicts our proposed model as follows. The historical data is split into windows and then those windows are divided into fixed-size patches. Linear embeddings are then applied to the patches, followed by position embeddings. Then we feed the resulting sequence of vectors to the Transformer encoder. As a standard approach, we add an extra token to the sequence of learnable tokens to perform prediction. The Transformer encoder diagram in Fig. 2 was inspired by [22].

C. Datasets

The Saudi Stock Exchange (Tadawul) database contains stock trading information for more than 200 Saudi Arabian listed companies. The companies are grouped into sectors with different indices for each industry. The data we downloaded spans the period from 1993-01-02 through 2021-06-17 and consists of 772,189 trading days. Listed companies' and indices trading information includes their Open, High, Low, Volume, and Closing Price for each trading day. From the dataset, we extracted four indices to illustrate model capabilities and performance.

These are Tadawul All Share Index (TASI), the Banks Index (TBNI), Materials Index (TMTI), and Telecommunication Services Index (TTSI).

Table II lists a small selection of the dataset. Specifically, it shows the trading information in the dataset for the TASI index for the period 1994-01-26 to 2021-07-01, which corresponds to 7311 trading days.

The rows correspond to one trading day and contain the following features: the index column, the

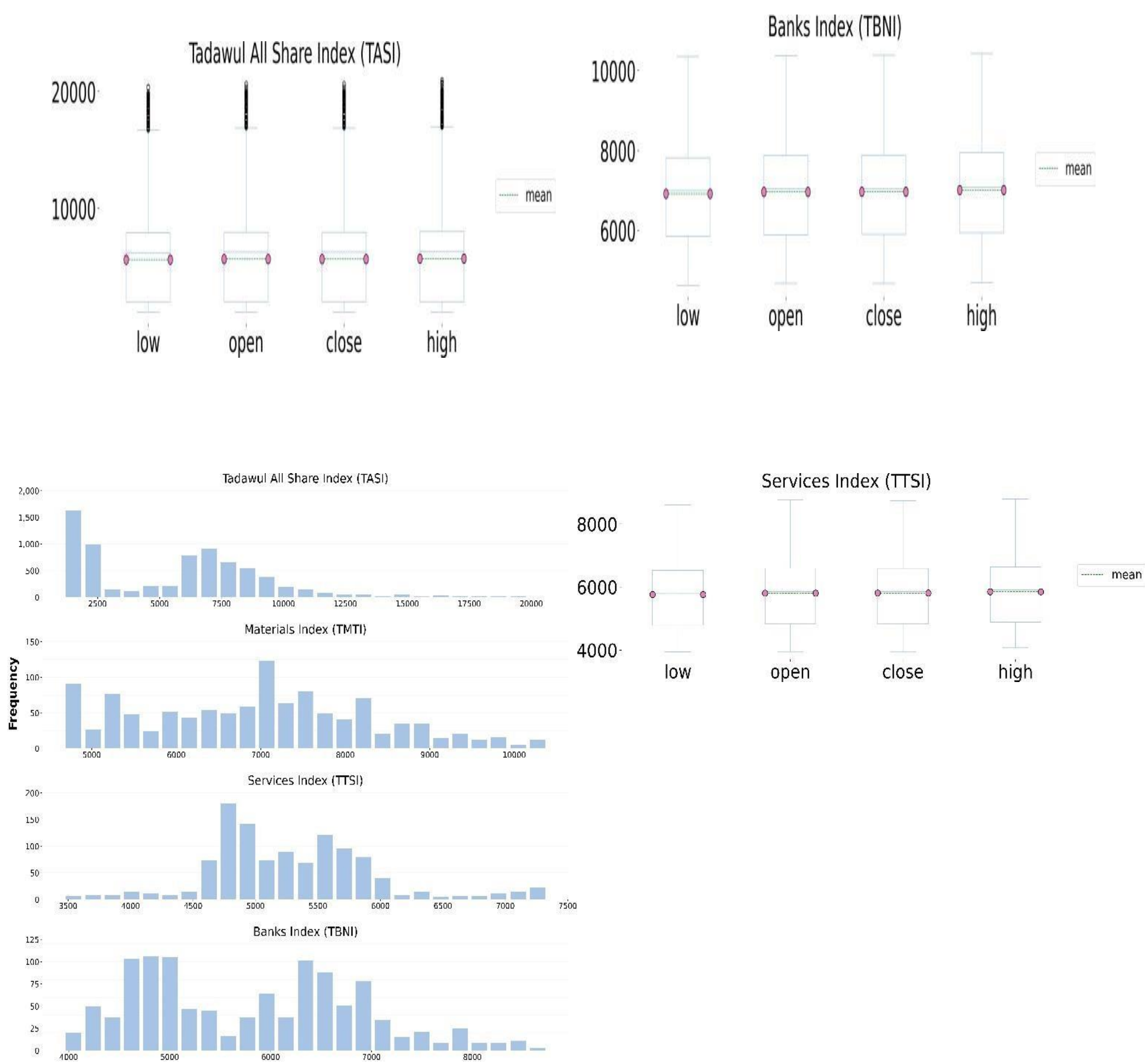


Fig. 5.3.3.TTSI Boxplot

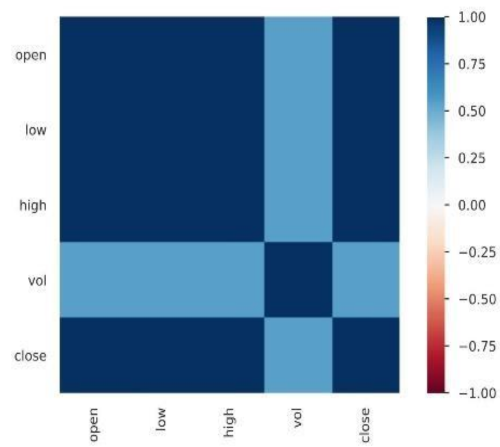
TABLE II. TASI SAMPLE DATA

	date	ticker	open	low	high	vol	close
0	1994-01-26	TASI	1751.71	1751.71	1751.71	312907	1751.71
1	1994-01-29	TASI	1751.71	1750.91	1751.71	204831	1750.91
—-	—-	—	—-	—-	—-	——	——
7310	2021-06-30	TASI	11002.74	10940.44	11009.70	374658538	10984.15
7311	2021-07-01	TASI	10987.13	10968.11	11006.66	352200486	10979.05

A boxplot is an in-depth statistical data analysis tool for gaining a broad perspective on the center and spread of the data distribution, which can assist with checking for errors and protecting other analyses. The median, interquartile range box, and whiskers are the primary elements of the boxplot to help understand the center and spread of the sample data. You'll see the green line representing the median in each box, which is the center of each feature. The interquartile range (the range between the third quartile and the first quartile) box, on the other hand, represents the middle 50% of the data and reflects how the data is distributed. The whiskers extend from both sides of the box (the bottom line is called lower whiskers, whereas the upper one is called higher whiskers). The whiskers denote the ranges for the bottom 25% and the top 25% of the data values, excluding outliers. Graphs that are skewed have the majority of data on the high or low side. Skewed graphs indicate that the data isn't normally distributed.

The data distribution for the TTSI, TMTI, and TBNI in the figures (Fig. 5 to 7) is almost normally distributed while it is positively skewed for the TASI index (Fig. 4). Moreover, any value greater than higher whiskers and less than lower whiskers values is an outlier and is represented in the figure as circles beyond the minimum and maximum values. Fig. 4, shows reasonable outliers points for TASI, which is expected as the closing of TASI is directly impacted by each and every listed company.

Fig. 8 highlights the correlation between the features (High, Low, Volume, and Closing Price), which is considered an essential step in the feature selection phase of data pre- processing, especially if the data types of the features are continuous. As you can see in the figure, there is a high correlation between volume and other features.



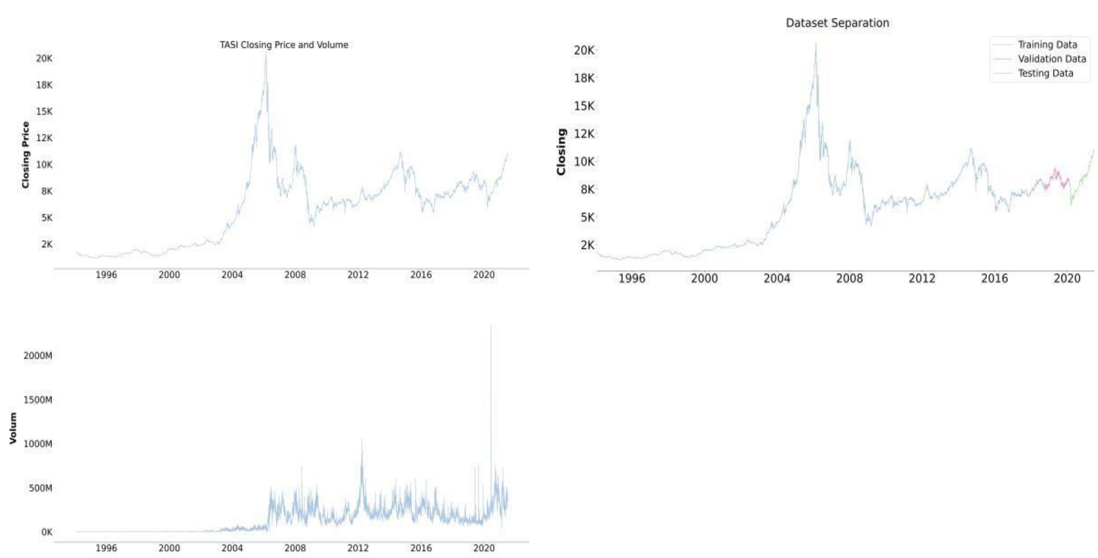


Fig. 5.3.4 TASI Closing Price and Volume

Tadawul All Share Index (TASI) volume and closing figures are shown above.

D. Data Modelling Methodology

At first, the historical stock data of earlier days $X \in \mathrm{R}^{M \times F}$ consisting of M periods with F features (previous closing, opening, high ,low and volume) is split into a sequence of flattened 2D Windows $\mathbf{x}_w \in \mathrm{R}^{L \times F}$ of size $M - L$, where L is look back time interverls. Then the input window is divided into non-overlapping temporal patches of size $\mathbf{x}_p \in W \times (F \times 2)$.

Finally, following the protocol in ViT ,the patches $\mathbf{x}_p \in \mathrm{R}^{W \times (F \times 2)}$ are flattened forming a sequence of embeddings.

Using learnable 1D position embeddings, we embed positional information into the patch embeddings so that all patches within a given window w are given the same temporal position.

This allows the model to determine the temporal positions of patches.

E. Data Prepossessing

It is imperative to preprocess data in order to achieve good predictions. The indexes data were checked to determine whether the Tadawul Dataset contained inconsistencies. All the numerical data were normalized, and the missing values were removed. The open, high, low, volume and close prices were used to calculate the features, but information such as the stock code and stock name was omitted since they do not make sense. The following sections describe how the various preprocessing steps are implemented.

Splitting the Dataset: The training and test datasets are separated, similar to the ideas presented by [39]. We reserve apart from the end the training for validation from each time series.

Data Normalization*:* Normalization refers to the process of changing the range of values in a set of data. As we use prices and volume data, all the stock data must be within a typical value range. In general, machine learning algorithms converge faster or perform better when they are close to normally distributed and/or on a similar scale. Also, in a machine learning algorithm, the activation function, such as a sigmoid function, has a saturation point after which the outputs are constant [40]. As a result, when using model cells, the inputs should be normalized before being used. This process was done using MinMaxScaler methods of the scikitlearn library. When MinMaxScaler is applied to a feature, it subtracts the minimum value from each value in the feature and divides the range by the result. Thus, the range of a feature is the difference between the maximum and minimum values. In this way, MinMaxScaler preserves the shape of the original distribution. MinMaxScaler normalizes input values to be between [0,1].

1) Feature Selection: The downloaded data contains several features, including stock code, stock name, opening price, high price, low price, volume and closing price. Aside from some features that may not make any sense, these initial data have a lot of noise. For this reason, the data should be neglected when it is being trained. Based on [41] using open price, high price, low price, volume and close price, the input features will yield a satisfactory result. Therefore, we have selected the first five features as our input and have neglected irrelevant data like stock names and stock codes.

F. Divided Space

At this stage, we apply the concept of the sliding window for framing the dataset. With a window size of 2, we use the data before two days to predict the subsequent day closing. The process is repeated until all data are segmented. Then, the framing dataset is further split into patches.

G. Hyperparameter Selection

A number of parameters, called hyperparameters, are usually included in all deep learning models (apart from Na¨ıve Bayes) that need to be adjusted to optimize results [42]. The various hyperparameters used during training are summarized in Table III. The AdamW optimizer is used during training with a learning rate of 0.001 and a weight decay of 0.0001. We train the model for 500 epochs with early stopping and dropout to prevent overfitting using TensorFlow [43] library.

Hyperparameter	Value
Learning Rate	0.001
Optimizer	AdamW
Batch size	256
Epochs	500
Early stopping	Patience = 70 epochs Monitoring parameter = validation loss
Loss Function	MSE

TABLE III. VARIOUS HYPERPARAMETERS USED IN THIS MODEL WITH THEIR VALUES

5.4 UML DIAGRAMS

UML is an acronym that stands for Unified Modeling Language. Simply put, UML is a modern approach to modeling and documenting software. In fact, it's one of the most popular business process modeling techniques.

It is based on diagrammatic representations of software components. As the old proverb says: "a picture is worth a thousand words". By using visual representations, we can better understand possible flaws or errors in software or business processes.

UML diagrams, in this case, are used to communicate different aspects and characteristics of a system. However, this is only a top-level view of the system and will most probably not include all the necessary details to execute the project until the very end.

There are several types of UML diagrams and each one of them serves a different purpose. The two broadest categories that encompass all other types are Behavioral UML diagram and Structural UML diagram. As the name suggests, some UML diagrams try to analyze and depict the structure of a system or process, whereas others describe the behavior of the system, its actors, and its building components. The different types are as follows:

- Use Case Diagram
- Class Diagram
- Sequence Diagram
- Component Diagram
- Collaboration Diagram
- Object Diagram
- State Machine Diagram
- Communication Diagram
- Deployment Diagram

5.4.1 USE CASE DIAGRAM

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

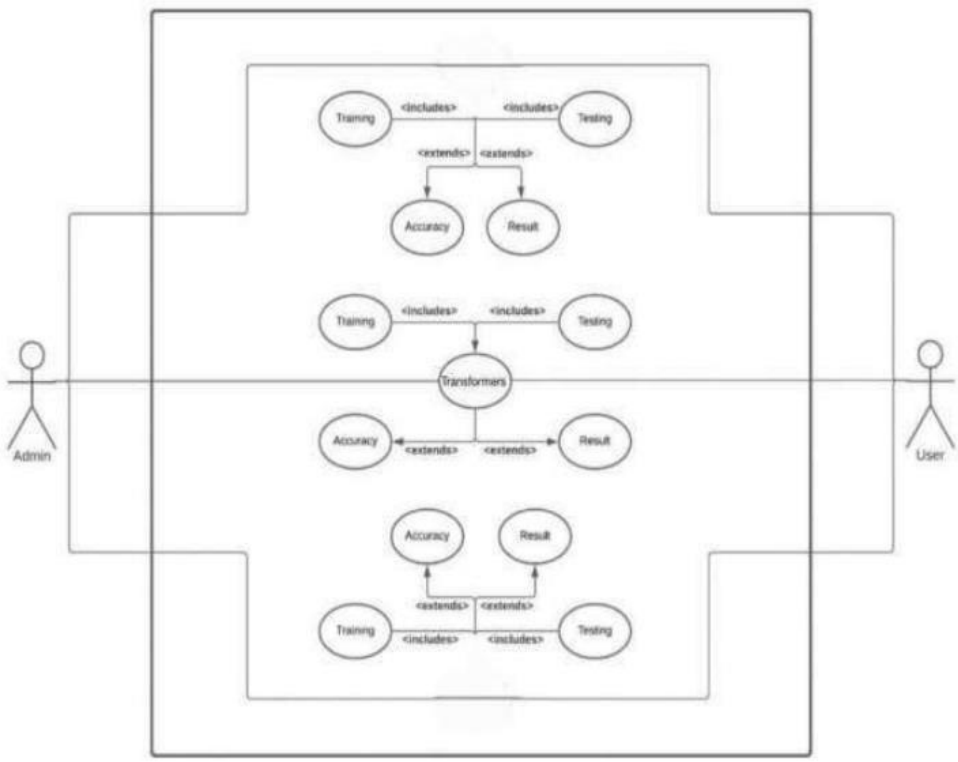


Fig 5.4.1 USE CASE DIAGRAM

5.4.2 DATA FLOW DIAGRAM

A data-flow diagram (DFD) is a way of representing a flow of a data of a process ora system (usually an information system). The DFD also provides information about the outputs and inputs of each entity and the process itself. Specific operations based on the data can be represented by a flowchart. There are several notations for displaying data-flow diagrams.

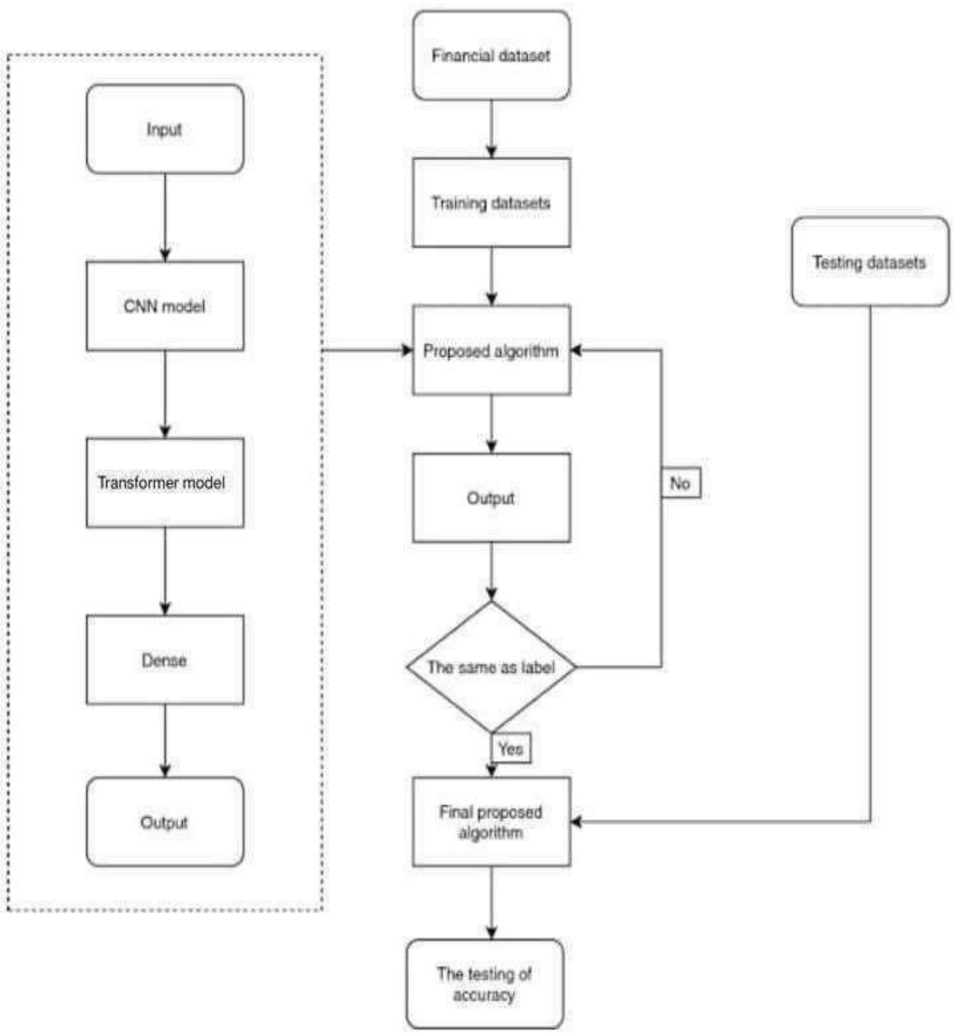


Fig 5.4.2: DATA FLOW DIAGRAM

5.4.3 COLLABARATION DIAGRAM

Collaboration diagrams are used to show how objects interact to perform the behavior of a particular use case, or a part of a use case. Along with sequence diagrams, collaboration are used by designers to define and clarify the roles of the objects that perform a particular flow of events of a use case. They are the primary source of information used to determining class responsibilities and interfaces. The collaborations are used when it is essential to depict the relationship between the object. Both the sequence and collaboration diagrams represent the same information, but the way of portraying it quite different. The collaboration diagrams are best suited for analyzing use cases.

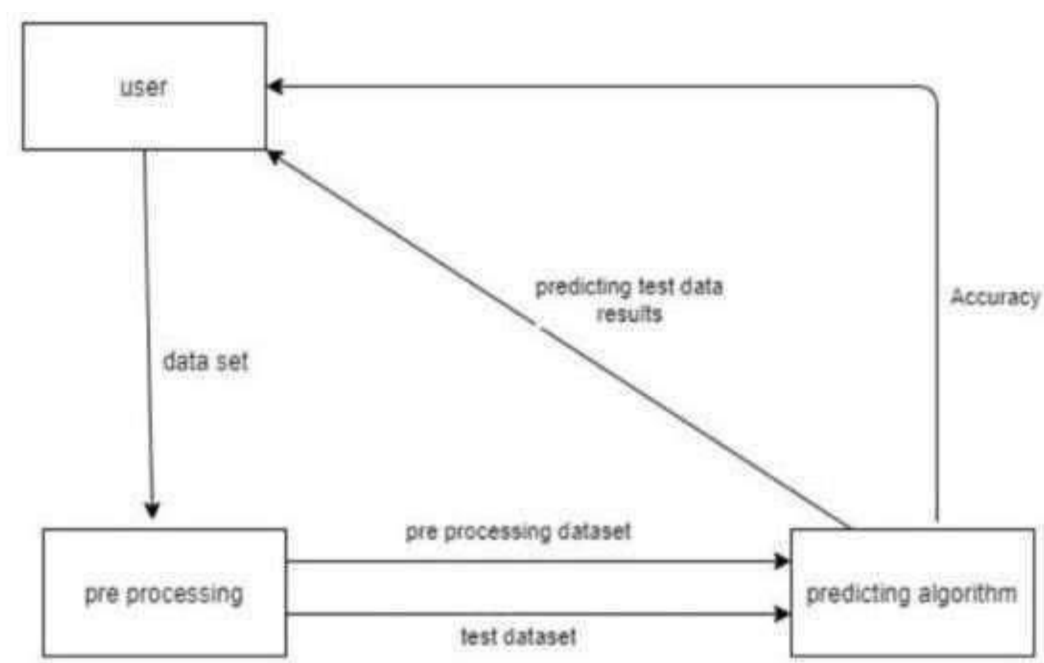


Fig 5.4.3 COLLABARATION DIAGRAM

5.4.4 COMPONENT DIAGRAM

Component diagram is a special kind of diagram in UML. The purpose is also different from all other diagrams discussed so far. It does not describe the functionality of the system but it describes the components used to make those functionalities. Component diagrams are used in modeling the physical aspects of object-oriented systems that are used for visualizing, specifying, and documenting component-based systems and also for constructing executable systems through forward and reverse engineering. Component diagrams are essentially class diagrams that focus on a system's components that often used to model the static implementation view of a system.

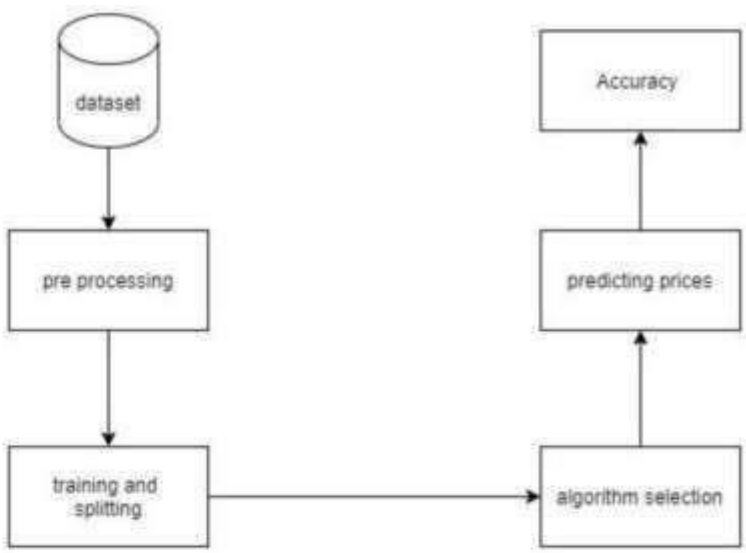


Fig: 5.4.4 COMPONENT DIAGRAM

6. INPUT AND OUTPUT SCREENSHOTS

6.1 INPUT SCREENSHOTS:

```
from yahoofinancials import YahooFinancials
import numpy as np
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras import layers
from keras.layers import Flatten
from keras import backend as K
from sklearn.preprocessing import MinMaxScaler
from sklearn.metrics import mean_squared_error
import sqlite3
import pandas as pd
import math
import matplotlib.pyplot as plt
plt.style.use('seaborn')
from datetime import datetime

#For reproducability
from numpy.random import seed
seed(1)
tf.random.set_seed(2)

# Some functions to help out with
def plot_predictions(test,predicted,symbol):
    plt.plot(test, color='red',label=f'Real {symbol} Stock Price')
    plt.plot(predicted, color='blue',label=f'Predicted {symbol} Stock Price')
    plt.title(f'{symbol} Stock Price Prediction')
    plt.xlabel('Time')
    plt.ylabel(f'{symbol} Stock Price')
    plt.legend()
    plt.show()

def plot_return_predictions(test,predicted,symbol):
    plt.plot(test, color='red',label=f'Real {symbol} Stock Price Returns')
    plt.plot(predicted, color='blue',label=f'Predicted {symbol} Stock Price Return'
    plt.title(f'{symbol} Stock Return Prediction')
    plt.xlabel('Time')
    plt.ylabel(f'{symbol} Stock Price Returns')
    plt.legend()
    plt.show()

def return_rmse(test,predicted):
    rmse = math.sqrt(mean_squared_error(test, predicted))
    print("The root mean squared error is {}.".format(rmse))
    return rmse
```

FIG 6.1.1 INPUT SCREENSHOT 1

```python
# # Choose a stock symbol
symbol_to_fetch = 'IBM'
# Choose a date range
start_date = '2020-01-01'
end_date = datetime.now().strftime('%Y-%m-%d')
# Get Stock Price Data
stock = fetch_ticker_data(symbol_to_fetch, start_date, end_date)
stock.columns = ['DateTime', 'Open', 'High', 'Low', 'Close', 'Volume']
stock['DateTime'] = stock['DateTime'].apply(lambda x: datetime.fromtimestamp(x))
stock = stock.fillna(method="ffill", axis=0)
stock = stock.fillna(method="bfill", axis=0)
stock = stock.set_index('DateTime')
# stock['return'] = stock['Close'].pct_change(1)
# for i in stock.index[1:]:
#     if (stock['return'].iloc[i] > 0 and stock['return'].iloc[i-1] < 0) or (stock['return'].iloc[i] < 0 and stock['return'].iloc[i-1] > 0):
#         stock['reversal'].iloc[i] = 1
#     else:
#         stock['reversal'].iloc[i] = 0
stock['Symbol'] = symbol_to_fetch
stock.tail()
#save a copy for later testing
original_stock = stock
original_symbol = symbol_to_fetch

stock['Close'].tail()
```

FIG 6.1.2 INPUT SCREENSHO1T 2

```python
# First we have to frontload the test data before the inital values of our test_set

dataset_total = pd.concat((original_stock[target][:train_end_date],original_stock[target][test_start_date:]),axis=0)
inputs = dataset_total[len(dataset_total)-len(test_set) - timesteps:].values
inputs = inputs.reshape(-1,1)
inputs  = sc.fit_transform(inputs)

X_test = []
for i in range(timesteps,test_set.shape[0] + timesteps):
    X_test.append(inputs[i-timesteps:i,0])
X_test = np.array(X_test)
X_test = np.reshape(X_test, (X_test.shape[0],X_test.shape[1],1))
predicted_stock_price = model.predict(X_test)
predicted_stock_price = sc.inverse_transform(predicted_stock_price)


print(test_set[-3],test_set[-2], test_set[-1])
shifted_test_set = shift(test_set, 1) #The shift function is defined early in the notebook
print(shifted_test_set[-3],shifted_test_set[-2], shifted_test_set[-1])

print(predicted_stock_price[-1])
prediction_error = test_set - predicted_stock_price # This is the error on the same day
#Before we can calculate the predicted return we have to shift the test_set to the day before so we use the shifted_test_set
predicted_return = (shifted_test_set - predicted_stock_price) / shifted_test_set

plt.plot(history.history['loss'])
plt.title('model loss')
plt.ylabel('loss')
plt.xlabel('epoch')
plt.legend(['loss'], loc='upper left')
plt.show()

plot_predictions(test_set,predicted_stock_price,original_symbol)
return_rmse(test_set,predicted_stock_price)

plot_return_predictions(test_set_return,predicted_return,original_symbol)
return_rmse(test_set_return[1:], predicted_return[1:])
```

FIG 6.1.3 INPUT SCREENSHOT 3

6.2 OUTPUT SCREENSHOTS:

```
DateTime
2023-04-18 13:30:00    127.779999
2023-04-19 13:30:00    126.320000
2023-04-20 13:30:00    126.360001
2023-04-21 13:30:00    125.730003
2023-04-24 13:30:00    125.400002
Name: Close, dtype: float64
```

FIG 6.2.1 OUTPUT SCREENSHOT 1

FIG 6.2.2 OUTPUT SCREENSHOT 2

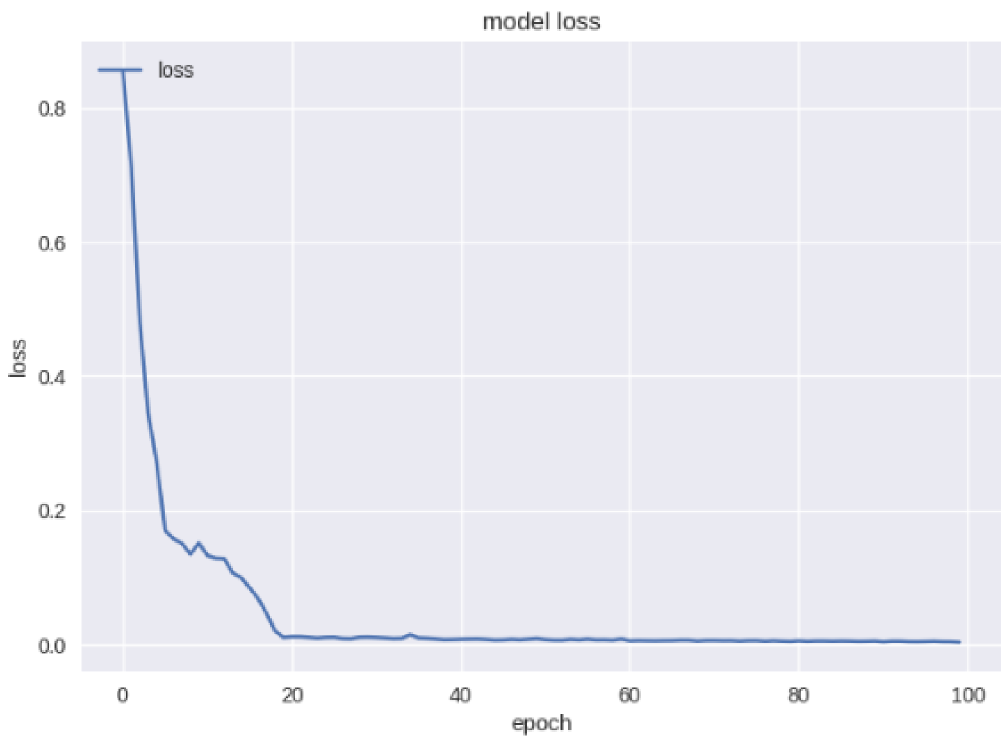


FIG 6.2.3 OUTPUT SCREENSHOT 3

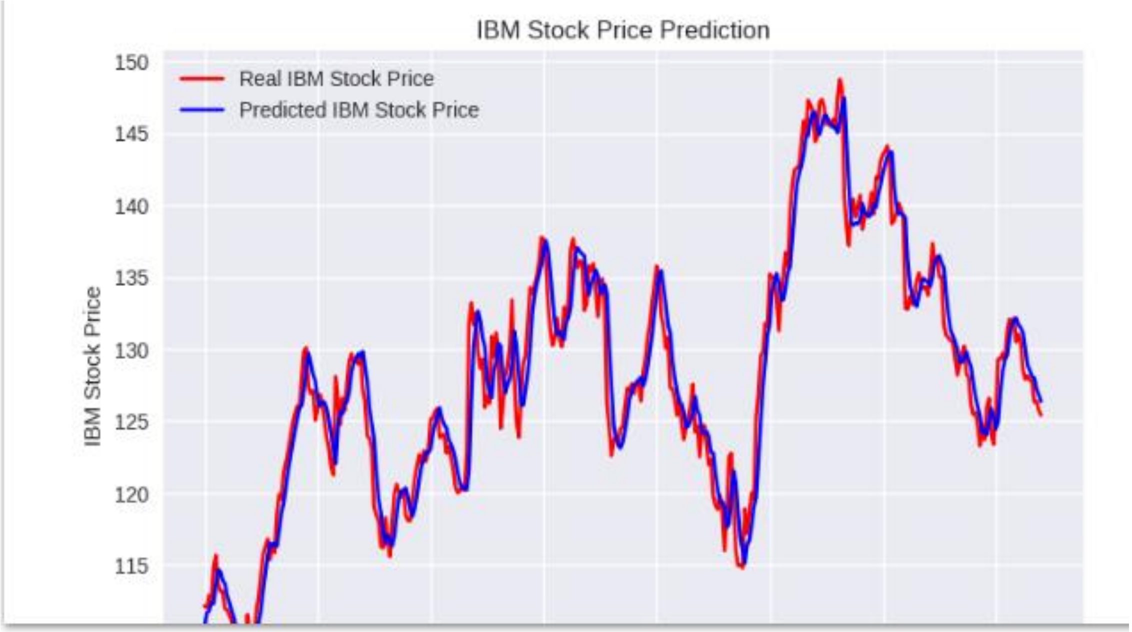


FIG 6.2.4 OUTPUT SCREENSHOT 4

7.CONCLUSION

We propose a transformer-based formalization model for stock price prediction. A significant influence on our architec- ture is a vision transformer (ViT) [36] using divided space. The vision transformer (ViT) is among the first attempts to apply the outstanding performance of Transformers. Using transformer network architectures with split time series into patches shows that hidden dynamics can be captured and predictions made reasonably. The model was trained using data from the Saudi Stock Exchange (Tadawul). As a result, we were able to predict the stock price of the TadawulAll Share In- dex (TASI), Telecommunication services Index (TTSI), Banks Index (TBNI), and Materials Index (TMTI) with accuracy that exceeds 90%.

We evaluated the proposed transformer model using four accuracy metrics, MAE, MSE, MAPE, and RMSE. We de- scribed the experimental results related to model optimization and model validation for all the four datasets. Subsequently, we presented results for the prediction of future stock closing prices. We were able to achieve over 90% accuracy compared to the best 72% reported in the literature (see Table I). Furthermore, the experiments showed that the proposed model architectures that split time series into patches were able to identify the dynamics and complex patterns from irregularities in financial time series. Transformer architecture has also been shown to identify sudden changes in stock markets, as reflected in the results. However, the changes occurring may not always appear regularly or follow the same cycles each time.

8.BIBLLIOGRAPHY

[1] Gabriel Pui Cheong Fung, Jeffry Xu Yu, Hongian Lu. “The Predicting Power of Textual Information On Financial Markets”.

[2] Chen, Jerry and Aaron Chai, Madhav Goel, Donavan Lieu, Faazilah Mohamed, David Nahm, Bonnie Wu, Predicting Stock Prices from News Articles.

[3] Gidofalvi, Gyozo, “Using news articles to predict stock price movements.” Department of Computer Science and Engineering, University of California, San Diego.

[4] Aase, Kim-Georg, “Text mining of news articles for stock price predictions.” (2011).

[5] Timmons, Ryan and Kari Lee, “Predicting the stock market with news articles.” CS224N Final Report (2007).

[6] Ma, Qicheng, “Stock price prediction using news articles.” CS224N, Final Report (2008).

International Journal of Scientific Research in Science, Engineering and Technology
Print ISSN: 2395-1990 | Online ISSN : 2394-4099 (www.ijsrset.com)
doi : https://doi.org/10.32628/IJSRSET

Stock Market Prediction Using Transformers

R. Sampada*[1], J. Shivani*[2], V. Lasya*[3], B. Srujana*[4], Mr. G.Sreenivasulu
[4]B.Tech. Student, *[5]ASSOCIATE Professor
CSE Department, JB Institute of Engineering and Technology, Hyderabad, India

ARTICLEINFO

Article History:
Accepted: 01 April 2023
Published: 15 April 2023

Publication Issue
Volume 10, Issue 2
March-April-2023

Page Number
502-505

ABSTRACT

Predicting stock market trends has been a difficult challenge, but recent research has shown that using machine learning techniques, specifically deep learning, has produced promising results. Transformer-based models, such as BERT and GPT2, have been successful in natural language processing tasks and are now being utilized for stock market prediction. These models can analyse large amounts of data, including financial news and social media data, and historical stock market data to make predictions about future stock prices. One advantage of these models is their ability to process and filter out irrelevant information, but there are still challenges to overcome, such as the high volatility of stock prices and the complexity of the stock market system. Nonetheless, these models can provide valuable insights for traders and investors to make informed decisions.

Keywords : BERT, GPT2, Transformer-based models

I. INTRODUCTION

Using deep learning models, specifically transformer-based models, to analyze historical stock market data and make predictions about future stock prices is a promising approach that makes use of the powerful natural language processing capabilities of transformers. Deep learning models are advantageous in their ability to process and analyze large amounts of data quickly and efficiently. This allows them to learn from a wide range of data and make more accurate predictions about future stock prices. Transformer-based models like BERT and GPT-2 are especially well-suited for stock market prediction as they can be fine-tuned for specific tasks and quickly adapt to new data. However, there are still challenges that need to be addressed, such as dealing with the high volatility of stock prices and the complexity of the stock market system. It is important to keep these limitations in mind and not solely rely on the predictions made by the model but use them as a supporting tool in the decision-making process. Overall, deep learning-based stock market prediction using transformers is a promising approach that can provide traders and investors with more informed decisions about buying and selling stocks.

II. METHODS AND MATERIAL

1.1 Materials

1.1.1 Data Sets

This dataset includes the daily open, high, low, and close prices for a particular stock, as well as the trading volume for each day. This type of data can be used as input to a transformer-based model for predicting future stock prices, samples of which are shown below table (1)

Table 1

Date	Open (Rs)	High (Rs)	Low (Rs)	Close (Rs)	Volume (Million)
2022-01-03	123.45	125.67	122.34	124.56	10.2
2022-01-04	124.67	127.89	124.56	126.78	8.9
2022-01-05	127.89	129.01	126.67	128.90	12.3
2022-01-06	128.00	131.23	127.45	129.45	9.8
2022-01-07	129.67	132.01	129.00	131.00	11.5
2022-01-10	132.34	134.56	131.23	133.45	13.7

2.1.2. Behavioural data

The behavioural data in stock market prediction refers to the information gathered from analysing the behaviour of investors, traders, and market participants. This can include data on trading volumes, patterns, and trends, as well as sentiment analysis of financial news and social media posts related to the stock market. By analysing this behavioural data, deep learning models can identify patterns and make predictions about future stock prices. This approach can be particularly useful in predicting short-term fluctuations in the market and can be combined with other data sources, such as historical price data and fundamental analysis.

2.2 METHODS

2.2.1. Data Pre-processing

Data pre-processing is a crucial step in any machine learning project, including stock market prediction using transformers. Here are some simple steps for data pre-processing (Collect Data, Data cleaning, Data integrity, Data normalization, Data splitting, Feature selection, Data encoding) By following these steps, the data is pre-processed and ready to be used for training and evaluating the transformer-based model for stock market prediction.

2.2.2. Machine Learning

Machine learning is used in Stock market prediction using Transformers to analyse historical stock market data, financial news, and social media data to make predictions about future stock prices. It

allows the models to learn from large amounts of data, filter out noise and irrelevant information, and understand the context of the data being analysed. This helps traders and investors make more informed decisions about buying and selling stocks.

III. Related Work

1. "Transformer-based neural network models for stock price prediction" by Ming Liu and Wenhao Huang. This paper proposes a novel deep learning model that combines a transformer-based architecture with a long short-term memory (LSTM) network for stock price prediction.
2. "BERT for stock market prediction: A systematic review" by Donghyun Kim, Dongwon Lee, and Jinyoung Kim. This paper provides a comprehensive review of recent studies that have used the BERT model for stock market prediction. It examines the strengths and weaknesses of these studies and identifies future research directions.

3. "Stock price prediction using a hybrid deep learning model" by Siyuan Zhao and Feng Jiang. This paper proposes a hybrid deep learning model that combines a convolutional neural network (CNN) with a transformer-based architecture for stock price prediction. The model is trained on financial news articles and historical stock price data.
4. "Stock price prediction using transformer-based models and news articles" by Hei Law, Eric Cho, and Richard Socher. This paper presents a transformer-based model that incorporates financial news articles for stock price prediction. The model uses a multi-head self-attention mechanism to analyze the relationship between news articles and stock prices.
5. "Stock price prediction using sentiment analysis of news articles and transformer-based models" by Feng-Li Lian, Yu-Ching Lin, and Chun-Ming Huang. This paper proposes a method that combines sentiment analysis of financial news articles with a transformer-based model for stock price prediction. The model is trained on both news articles and historical stock price data.

DATE	PREDICTED PRICE	ACTUAL PRICE
April 1, 2022	1500	1495
April 2, 2022	1515	1522
April 3, 2022	1530	1510
April 4, 2022	1490	1488
April 5, 2022	1475	1480

Table 2 : The predicted prices and the actual prices for those five days

The model has made predictions for the next five days of Company X's stock prices. The "Predicted Price" column shows the model's predicted price for each day, while the "Actual Price" column shows the actual stock price for that day.

The model's predicted prices for April 1 and 2 are slightly higher than the actual prices, but the predictionsfor April 3 and 4 are lower than the actual prices. However, the predicted price for April 5 is quite close to the actual price.

Overall, the model's predictions for the next five days of Company X's stock prices have been relatively accurate. This information can be useful for investors and traders in making more informed decisions about buying and selling the stock.

IV.CONCLUSION

In conclusion, using transformer-based models for stock market prediction is a promising approach that utilizes their powerful natural language processing capabilities to analyze a widerange of data, including financial news, social media data, and historical stock market data, tomake predictions about future stock prices. These predictions can be used by traders and investors to make more informed decisions about buying and selling stocks. However, it is important to keep in mind that stock market prediction is a difficult task due to the high volatility of stock prices and the complexity of the stock market as a system. While transformer-based models offer advantages such as their ability to process and analyse large amounts of data and filter out noise, they still face challenges that need to be addressed. Therefore, it is important to use the predictions generated by these models as a supporting tool in the decisionmaking process and not solely rely on them.

Acknowledgements

We would like to express our sincere gratitude to our Head of Department, Mr. G. Sreenivasulu, and our Principal, Dr. P. C. Krishnamachary, for giving us the opportunity to work on this amazing paper about “Stock market prediction using Transformers”.

Secondly, we would also like to express our gratitude to our guide Mr. G. Sreenivasulu for his assistance in completing this paper within the stipulated timeframe.

Lastly, I am thankful to all those who have encouraged me to complete this paper before the deadline.

Authors' contributions

We conceived the presented idea also developed the theory and performed the computations. We also verified the analytical methods and provided guidance. All authors discussed the results and contributed to the final manuscript. G. Sreenivasulu sir also provided valuable guidance with his expertise.

V. REFERENCES

[1]. Chen, C., Chen, X., & Hu, Y. (2020). Stock price prediction based on multi-task learning with transformers. IEEE Access, 8, 193533-193543.

[2]. Liu, Q., Lin, S., & Zhu, Y. (2021). Stock price prediction using long short-term memory and transformer. IEEE Transactions on Neural Networks and Learning Systems, 33(2), 459-470.

[3]. Li, W., Li, J., & Wang, X. (2021). Stock price prediction based on transformers and multiple attention mechanism. Physica A: Statistical Mechanics and its Applications, 570, 125691.

[4]. Zhang, K., Zhang, X., & Lu, J. (2021). Stock price prediction using a hybrid approach based on transformer and LSTM. Symmetry, 13(7), 1126.

[5]. Ma, J., Li, K., & Chen, S. (2021). Stock price prediction based on attention mechanism and transformers. Applied Sciences, 11(7), 3203.

[6]. Lu, S., Xu, Q., & Zheng, X. (2021). A stock price prediction model based on the self-attention mechanism of transformers. Journal of Intelligent & Fuzzy Systems, 41(1), 937-945.

[7]. Zhang, X., Zhang, K., & Cui, X. (2021). Stock price prediction based on transformer and gated recurrent unit neural networks. Entropy, 23(4), 468.

[8]. Song, X., Wang, Z., & Sun, J. (2021). Stock price prediction based on multi-source data and transformer model. Neurocomputing, 445, 241-248.

[9]. Yan, W., Zhang, W., & Liu, Q. (2021). Hybrid stock price prediction model based on transformer and attention mechanism. Mathematics, 9(6), 596.

Cite this article as :

R. Sampada, J. Shivani, V. Lasya, B. Srujana, Mr. G. Sreenivasulu, "Stock Market Prediction Using Transformers", International Journal of Scientific Research in Science, Engineering and Technology (IJSRSET), Online ISSN : 2394-4099, Print ISSN : 2395-1990, Volume 10 Issue 2, pp. 502-505, March-April 2023.

Journal URL : https://ijsrset.com/IJSRSET2310275