Project Report

On

Stock Market Prediction using different Deep Learning Models

Guided By

Dr. Ram Sarkar

Professor of Computer Science & Engineering, Jadavpur University

Group Members

Name	Roll Number
Pulak Kumar Ghosh	001910501042
Aryapriyo Mandal	001910501033

Acknowledgement

We are indebted to our project guide, Dr. Ram Sarkar, Professor, Dept. of Computer Science & Engineering, Jadavpur University for his constant support, prompt help and unwavering encouragement which helped us a lot in completing this project. Under his guidance we have been able to do fruitful research and learn new concepts at hand. We are also thankful to Prof. Nandini Mukherjee, Head of the Department. of Computer Science and Engineering, Jadavpur University for allowing us to carry out our research in the department.

Pulak Kumar Ghosh

001910501042

Aryapriyo Mandal

001910501033

4th year B.E. Student

Department of Computer Science and Engineering

Jadavpur University

Certificate

This is to certify that the project entitled “**Stock Market Prediction using Different Deep Learning Models**” has been carried out by Pulak Kumar Ghosh and Aryapriyo Mandal under my guidance and supervision, and be accepted in partial fulfillment of the requirement for the degree of Bachelor of Computer Science and Engineering.

Dr. Ram Sarkar

Professor

Jadavpur University

Department of Computer Science and Engineering

Table of Content

Abstract

The stock market is a complex and volatile system that can be challenging to predict accurately. However, recent years have seen the rise of deep learning models in financial forecasting due to their ability to process vast amounts of data and identify complex patterns. In this paper, we present a study on stock market prediction using different Deep Learning Models.

To begin we collected historical data and used it to train our models. We then evaluated the performance of various deep learning architectures, including long short-term memory networks (LSTMs), Bi-directional long short-term memory networks (Bi-LSTMs), and Convolutional long short-term networks (ConvLSTM). Our results showed that ConvLSTM-based models outperformed the other architectures in terms of accuracy.

Moreover, Our study demonstrates the effectiveness of deep learning models in predicting stock market trends. We believe that these models have the potential to be valuable tools for investors and financial analysis in making informed decisions. However, it is essential to note that stock market prediction remains a challenging task, and caution should be exercised when making investment decisions based on model predictions alone.

Introduction

The stock market is a complex and dynamic system that plays a vital role in the global economy. Predicting stock market movements has always been a challenging task for investors and financial analysts due to the numerous factors that influence stock prices, including economic indicators, company performance, geopolitical events, and investor sentiment.

Traditional statistical models and technical analysis methods have been widely used to forecast stock market trends. However, these methods often rely on assumptions that may not hold true in real-world scenarios, and they struggle to capture the nonlinear relationships and temporal dependencies inherent in financial data. As a result, there has been a growing interest in utilizing advanced machine learning techniques, particularly deep learning algorithms, to improve the accuracy of stock market predictions.

Deep learning is a subset of machine learning that focuses on training artificial neural networks with multiple layers to learn hierarchical representations of data. One of the most powerful and popular deep learning models for sequential data analysis is the Long Short-Term Memory (LSTM) network. LSTM network excels at capturing long-term dependencies and have been successfully applied in various domains, including natural language processing, speech recognition, and time series forecasting.

In recent years, researchers and practitioners have explored the applications of LSTM networks, along with other variants such as Convolutional LSTM (CNV-LSTM) and Bidirectional LSTM (Bi-LSTM), for stock market prediction. CNV-LSTM combines the spatial processing capability of convolutional neural networks with the sequential modeling ability of LSTMs, while Bi-LSTM incorporates information from both past and future time steps, enabling better understanding of the temporal dynamics in financial data.

The objective of this project report is to evaluate and compare the performance of LSTM, CNV-LSTM, and Bi-LSTM models in predicting stock market movements. We aim to analyze the effectiveness of these deep learning methods in capturing the complex patterns and dependencies in historical stock data, and assess their potential for generating accurate and reliable predictions.

To achieve this goal, we will use historical stock market data, including price and volume information, along with relevant technical indicators, as inputs to the deep learning models. We will train and validate the models using a suitable dataset, and then evaluate their performance based on various evaluation metrics, such as accuracy, precision, recall, and F1 score. Additionally, we will compare the predictive power of the deep learning models with traditional statistical models to gain insights into their relative strengths and weaknesses.

By undertaking this project, we hope to contribute to the growing body of research on using deep learning methods for stock market prediction. The results of this study could potentially have implications for investors, financial institutions, and other stakeholders in making informed decisions in the stock market, ultimately improving investment strategies and risk management practices.

In the subsequent sections of this report, we will provide a comprehensive literature review, present the methodology adopted for data preprocessing and model training, discuss the experimental setup, analyze the results obtained, and conclude with a discussion on the implications and future directions of this research.

Related Works

Numerous studies have investigated the application of deep learning methods, particularly LSTM-based architectures, for stock market prediction. A study conducted by Zhang, Patuwo, and Hu (1998) demonstrated the effectiveness of using recurrent neural networks, including LSTM, for stock price forecasting. They compared the performance of LSTM models with traditional time series models and found that LSTM outperformed them in terms of accuracy and prediction error.

In a more recent study, Fischer, Krauss, and Treichel (2018) proposed a hybrid model combining CNN and LSTM, known as CNV-LSTM, for stock market prediction. Their model captured both the spatial features of price patterns and the temporal dependencies of the time series data. The authors reported promising results, indicating that CNV-LSTM achieved superior performance compared to traditional machine learning models and plain LSTM.

Another notable work by Guo, Shen, and Sun (2019) explored the use of Bi-LSTM for stock market prediction. They argued that incorporating future information is crucial for accurate forecasting, and Bi-LSTM allows the model to access information from both past and future time steps. Their experiments demonstrated the Bi-LSTM achieved better prediction performance compared to LSTM and other baseline models, suggesting the importance of capturing bidirectional dependencies in stock market data.

In addition to these individual model comparisons, there have been studies that compared multiple deep learning models for stock market prediction. For instance, Chen, Wei, and Lai (2015) compared LSTM, recurrent neural network (RNN), and echo state network (ESN) models on the task of predicting stock price movements. They found that LSTM consistently outperformed the other models, highlighting its capability to capture long-term dependencies.

Furthermore, some researchers have investigated the integration of deep learning models with traditional technical indicators to enhance stock market prediction. For example, Zhang, Xie, and Qi (2017) combined LSTM with moving average convergence divergence (MACD) and relative strength index (RSI) indicators to predict stock prices. Their experimental results showed that the combined approach improved the prediction accuracy compared to using LSTM alone.

While many studies have focused on LSTM-based architectures, it is worth mentioning that other deep learning models, such as deep belief networks (DBNs) and deep convolutional neural network (CNNs), have also been explored for stock market prediction, Each model has its strengths and limitations, and their effectiveness depends on the specific characteristics of the dataset and prediction task.

Overall, these related works collectively indicate that deep learning methods, including LSTM, CNV-LSTM, and Bi-LSTM, hold promise for stock market prediction. They offer the potential to capture complex patterns, temporal dependencies, and bidirectional relationships in financial data, leading to improved forecasting accuracy. However, further research and experimentation are necessary to validate and extend these findings, and to explore the applicability of deep learning methods in different market conditions and for various financial instruments.

Motivation & contribution

The stock market is a complex and dynamic environment where millions of investors trade securities based on various factors such as company performance, market trends, economic indicators, and investor sentiment. The ability to accurately predict stock market movements has always been a challenging and sought-after objective for investors, traders, and financial institutions. Making informed investment decisions requires understanding and anticipating market trends, which can potentially result in significant financial gains or losses.

Traditional methods of stock market prediction, such as technical analysis and fundamental analysis, have their limitations. Technical analysis relies on historical price patterns and indicators to predict future price movements, while fundamental analysis examines the financial health and intrinsic value of companies. However, these methods often struggle to capture the inherent complexity and non-linear nature of the stock market. They may not adequately consider the interplay of various factors and the impact of unexpected events on stock prices.

Deep learning, a subfield of machine learning, has emerged as a promising approach for stock market prediction. Deep learning models, particularly recurrent neural networks (RNNs), have the ability to capture and model complex temporal dependencies in time series data. By analyzing historical stock market data, deep learning models can potentially uncover hidden patterns and trends that impact stock prices. This can provide investors and traders with valuable insights to make informed decisions.

The motivation behind this project is to explore and evaluate the effectiveness of different deep models in predicting stock market movements. Specifically, we aim to compare the performance of LSTM, Bi-LSTM, and CNV-LSTM models, along with the incorporation of an attention layer, to understand their capabilities in capturing stock market patterns and making accurate predictions. By utilizing these advanced deep learning architectures, we seek to enhance the accuracy of stock market predictions and provide a valuable tool for investors to make informed investment decisions.

The utilization of deep learning models offers several advantages in stock market prediction. Firstly, these models can effectively capture the temporal dependencies and patterns in stock market data, which are essential for accurate forecasting. Secondly, deep learning models have the potential to adapt and learn from new information, allowing them to continuously improve their prediction accuracy as they are exposed to more data. This adaptability is particularly valuable in the stock market, where market dynamics can change rapidly. Lastly, by incorporating an attention layer, we aim to focus the models' attention on the most relevant features and time steps, further enhancing their predictive performance.

The fundings of this project have implications for various stakeholders in the financial industry. Investors and traders can benefit from more accurate predictions, enabling them to make informed decisions regarding buying, selling or holding stocks. Financial institutions can leverage these predictive models to manage risks, optimize portfolio management, and develop trading strategies. Furthermore, the research contributes to the growing body of knowledge in the field of stock market prediction using deep learning, providing insights for further advancements and improvements in this domain.

So, the motivation behind this project is to address the challenges and limitations of traditional stock market prediction methods by leveraging the power of deep learning models. By exploring different architectures, including LSTM, Bi-LSTM, and CNV-LSTM, and incorporating an attention layer, we aim to enhance the accuracy of stock market predictions. Ultimately, this research aims to provide valuable tools and insights for investors and traders, empowering them to make well-informed decisions in the dynamic and complex world of the stock market.

This project makes several contributions to the field of stock market prediction using deep learning models. The primary contributions include the evaluation and comparison of different deep learning architectures, namely LSTM, Bi-LSTM, and CNV-LSTM, along with the incorporation of an attention layer. The findings shed light on the effectiveness of these models in capturing stock market patterns and making accurate predictions. The contributions of this project can be summarized as follows:

1. Evaluation of Deep Learning Architectures: The project provider a comprehensive evaluation of three deep learning architectures commonly used for time series prediction tasks – LSTM, Bi-LSTM, and CNV-LSTM. By implementing and comparing these architectures, we gain insights into their respective strengths and weaknesses in the context of stock market prediction. This evaluation enables investors and researchers to make informed decisions regarding the choice of architecture for stock market forecasting.
2. Comparative Analysis of Model Performance: The project rigorously compares the predictive performance of the LSTM, Bi-LSTM, and CNV-LSTM models on a real-world stock market dataset. By evaluating these models on standardized metrics such as accuracy, we establish their relative effectiveness in capturing stock market dynamics. This

comparative analysis provides valuable guidance to practitioners seeking to leverage deep learning models for stock market prediction.

3. Integration of Attention Mechanism: An attention layer is incorporated into each of the deep learning models to focus on the most informative features and time steps. This integration enhances the models' ability to identify and weigh the most relevant information, further improving their predictive accuracy. The project showcases the benefits of attention mechanisms in stock market prediction, emphasizing the importance of capturing key features and time points to make accurate predictions.
4. Exploration of Stock Market Prediction Challenges: By delving into the domain of stock market prediction, the project highlights the complexities and challenges associated with forecasting stock market movements. Factors such as market volatility, external events, and the interplay of various influencing factors pose significant hurdles in achieving accurate predictions. By addressing these challenges and offering insights into deep learning models' capabilities, this project contributes to the ongoing research in stock market prediction.
5. Practical Implications for Investors and Financial Institutions: The project's findings have practical implications for investors, traders, and financial institutions. The accurate prediction of stock market movements enables investors to make informed decisions regarding buying, selling, or holding stocks, potentially maximizing their returns. Financial institutions can leverage these predictive models for risk management, portfolio optimization, and the development of trading strategies. The project's contributions provide stakeholders with valuable tools and insights to navigate the complexities of the stock market effectively.
6. Foundation for Further Research: This project serves as a foundation for further research in the field of stock market prediction using deep learning models. The insights gained from this study can inspire researchers to explore additional enhancements, such as incorporating external data sources or experimenting with ensemble approaches. The project's contributions stimulate ongoing efforts to improve the accuracy and robustness of stock market prediction models.

Moreover, this project contributes to the field of stock market prediction by evaluating and comparing different deep learning architectures and incorporating an attention mechanism. The findings offer insights into the strengths and weaknesses of these models and their applicability to stock market forecasting. The practical implications of this research empower investors and financial institutions to make more informed decisions, while also providing a foundation for future advancements in stock market prediction using deep learning.

Methodology

Deep Learning is an AI method that trains machines to do what our human brain does naturally: learning by precedent. Deep Learning is a major innovation that made driverless vehicles possible, empowering them to perceive traffic signs, or to recognize a person on foot, or even distinguishing whether a driver is conscious or not in order to park the car safely and avoid a catastrophe. It is also behind voice controlled gadgets like smartphones, tablets, TVs, wireless speakers. Deep Learning has been getting much consideration of late and in light of current circumstances, it certainly is thoroughly deserved as it is accomplishing results that were previously considered unrealistic.

In Deep Learning, a computer model learns to perform classification tasks directly from images, text, or sound. Deep Learning models can reach cutting edge accuracy, here and there even surpassing human-level execution. Typically, models which have a neural network architecture containing many layers are fed and trained on vast amounts of categorized data which makes Deep Learning particularly costly in terms of required computation time and power.

Deep Learning achieves accuracy of recognition at higher rates. This helps consumer electronics to satisfy user needs and is vital for safety-specific applications such as autonomous vehicles. Recent progress in deep learning has improved so much that in some tasks such as identifying objects in pictures deep learning outperforms people. Below is a figure illustrating a simple neural network and a Deep Learning neural network.

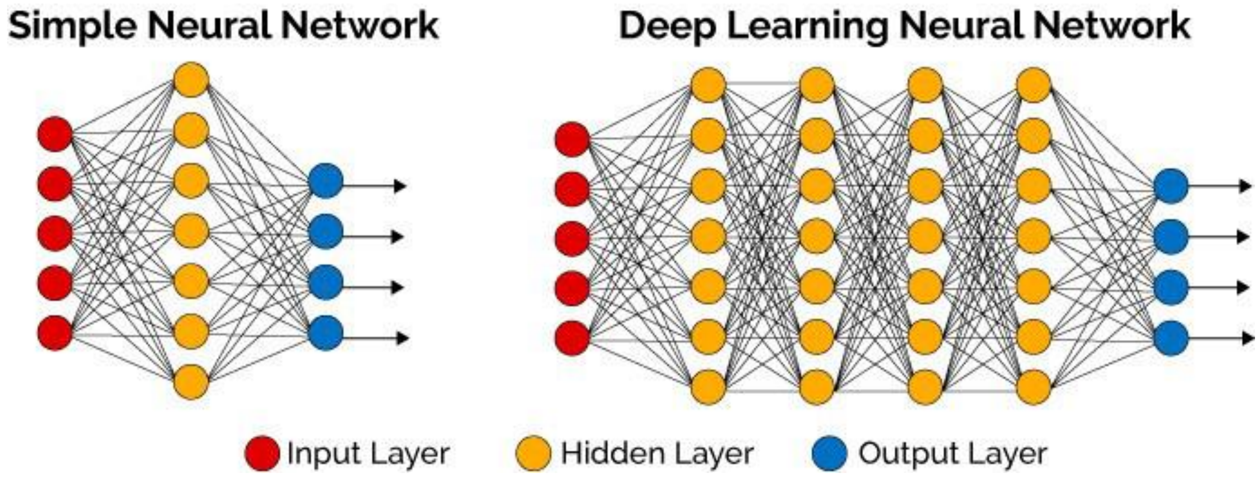


Fig 1 : Conceptual illustration of a simple neural network and a Deep Learning neural network

As illustrated above, a deep learning neural network consists of multiple hidden layers, it is in a way similar to stacking multiple simple neural networks together. This raises the question: how does deep learning reach a greater performance then?

From the first theoretical deep learning methodology in the 1980s, Deep Learning has only started to come close to its real potential for two fundamental reasons:

=> Deep Learning needs a huge amount of labeled information or data which only became available during this last decade. For instance, the implementation of autonomous cars takes millions of pictures and thousands of hours of footage.

=> Deep Learning necessitates considerable processing capacity. Highly efficient GPUs have a parallel, profoundly effective architecture that allows development teams to decrease learning time from weeks to hours or minutes when used in conjunction with clusters or cloud computing.

Transfer learning is a Machine Learning technique that re-uses a model trained for a task for a second related task. This optimization enables quick progress or increased performance once the second task is modeled. Transfer learning is also closely linked to problems like multitasking learning and is not only a field of interest for Deep Learning alone. Given the massive resources and challenging data sets needed for training Deep Learning models, Transfer Learning has become particularly popular and mainly used for Deep Learning applications.

There exists 2 major approaches to use Transfer Learning :

1. Model Development Approach: First, we select a source task with an available substantial data set. Then we develop a source model for the first task in order to learn some features. Next, we re-use the model as a starting point for a second related task. Optionally the model may be tuned for the second task to improve results by adding a few layers.
2. Pre-trained Model Approach: First, we select a source model that is already pre-trained on challenging data sets. Then we re-purpose that model in order to re-use it as the starting point of the desired task to accomplish. Optionally, the model may need to be tuned or adapted to improve results.

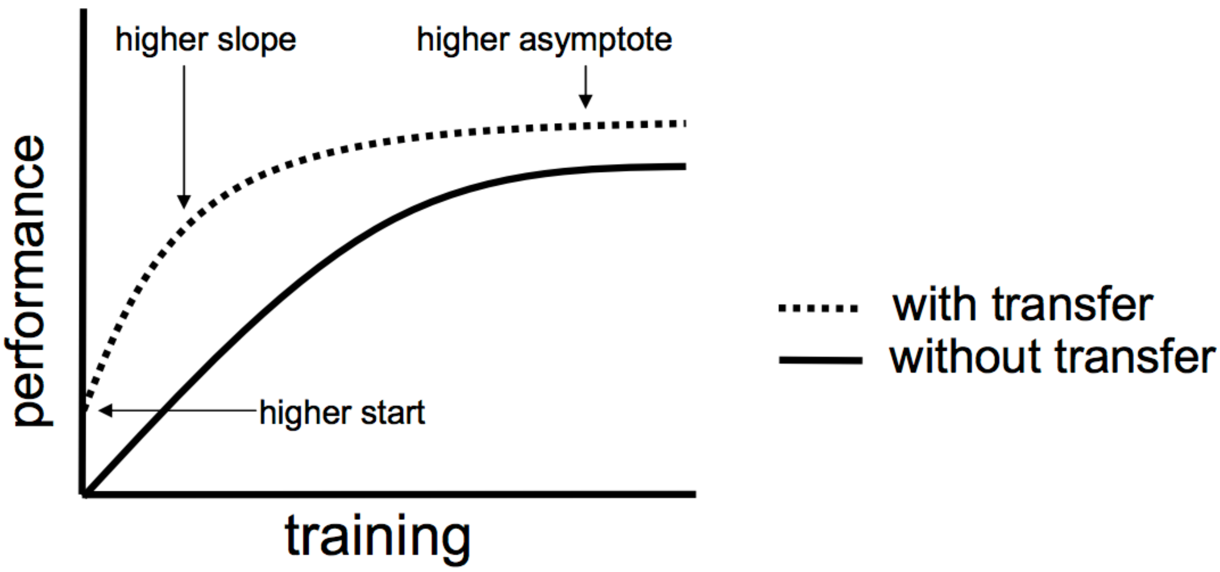


Fig 2 : Performance with Transfer Learning & without it

Before speaking of time series analysis, let us first define what a time series is: It is a sequence of data points or values that are sorted, listed or graphed against the flow of time. For the majority of cases, a time series sequence will have continuous equally spaced time data points.

Two main purposes of the analysis of time-series are: (a) identify the nature of the observation sequence phenomenon and (b) foresee (accurately predict time-series variable). These two objectives require the identification of the patterns of time series observed as well as a formal description. We can interpret and embed the pattern with other records once the pattern has been identified. Whatever our complexity and veracity of understanding of the concept, we can deduce the recognized pattern to logically predict outcomes.

Time series analysis involves techniques for breaking down information of a sequence of data points so as to extricate important measurements and different attributes of the information. Time series prediction is the utilization of a model to foresee future values by studying patterns in pre-existing value records or datasets. Moreover, a random model for time series forecasting will always be limited by the fact that data points closer together in time will be more correlated than data points with a bigger separation distance in time. This is exactly what is challenging about time series forecasting, keeping long term correlation without neglecting short term dependencies between data points.

To that extent, many data mining techniques can be used to solve the problem with various degrees of success. Next, we will see some of the most promising techniques to predict time series data and how a type of neural network called LSTM is leading the race in the field.

Long Short Term Memory model (LSTM)

LSTM, which stands for Long Short Term Memory, is a type of neural network which is particularly useful in the case of time series forecasting. According to an article by Srivastava on LSTM and essentials of deep learning, an LSTM network is the most effective solution to time series analysis and thus stock market prediction.

Srivastava affirms that: "Sequence prediction problems have been around for a long time. They are considered as one of the hardest problems to solve in the data science industry. These include a wide range of problems; from predicting sales to finding patterns in stock markets data, from understanding movie plots to recognizing your way of speech, from language translations to predicting your next word on your iPhone's keyboard.

With the recent breakthroughs that have been happening in data science, it is found that for almost all of these sequence prediction problems, Long Short Term Memory networks have been observed as the most effective solution.

LSTMs have an edge over conventional feed-forward neural networks and Recurrent Neural Networks in many ways. This is because of their property of selectively remembering patterns for long durations of time."

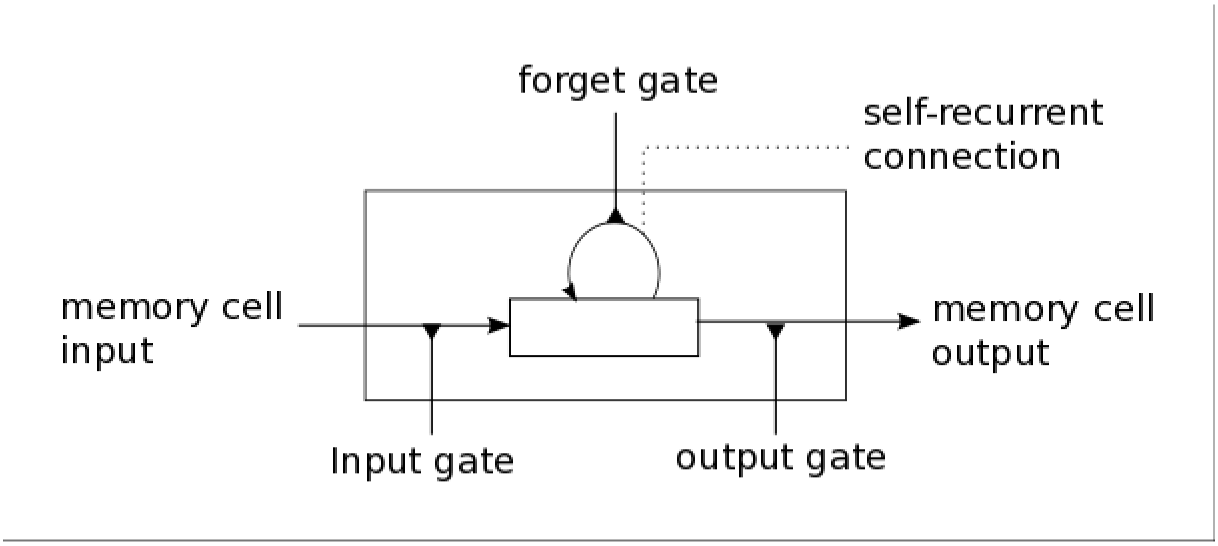


Fig 3 : Simplified look at an LSTM cell

In the case of a basic neural network, in order to add new information, it transforms the existing information completely by applying a sigmoid function. Because of this, the entire information is modified as a whole. Thus, there is no consideration for 'important' information and 'not so important' information LSTMs on the other hand make small modifications to the information by multiplications and additions. With LSTMs, the information flows through a mechanism known as cell states. This way, LSTMs can selectively remember or forget things.

The following figure, represents a more detailed view at the internal architecture of an LSTM network:

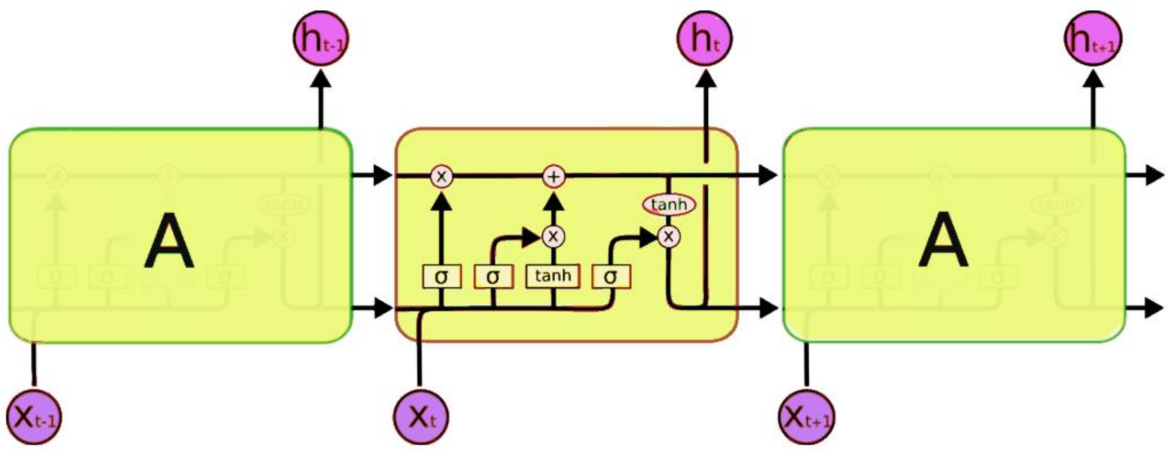


Fig 4 : LSTM cell Architecture

A typical LSTM network consists of different memory blocks called cells. There are two states that are being transferred to the next cells; the cell state and the hidden state. The memory blocks are responsible for remembering things, and manipulations to this memory are done through three major mechanisms called gates.

(a) Forget Gate:

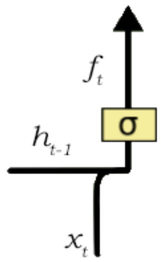


Fig 5 : Forget Gate, Internal Architecture

Srivastava describes it as such: "A forget gate is responsible for removing information from the cell state. The information that is no longer required for the LSTM to understand things or the information that is of less importance is removed. This gate takes in two inputs: h_t-1 and x_t.

H_t-1 is the hidden state from the previous cell or the output of the previous cell and x_t is the input at that particular time step. The given inputs are multiplied by the weight matrices and a bias is added. Following this, the sigmoid function is applied to this value. The sigmoid function outputs a vector, with values ranging from 0 to 1, corresponding to each number in the cell state. Basically, the sigmoid function is responsible for deciding which values to keep and which to discard. If a '0' is output for a particular value in the cell state, it means that the forget gate wants the cell state to forget that piece of information completely. Similarly, a '1' means that the forget gate wants to remember that entire piece of information. This vector output from the sigmoid function is multiplied to the cell state." Below are the equations for LSTM.

$i_t = \sigma(W_i h_{t-1} + U_i x_t + b_i)$ Input gate

$f_t = \sigma(W_f h_{t-1} + U_f x_t + b_f)$ Forget gate

$o_t = \sigma(W_o h_{t-1} + U_o x_t + b_o)$ Output gate

$\hat{c}_t = \tanh(W h_{t-1} + U x_t + b)$ Memory cell candidate

$c_t = f_t \circ c_{t-1} + i_t \circ \hat{c}_t$ Memory cell

$h_t = o_t \circ \tanh(c_t)$ Shadow state

$y_t = h_t$ Cell Output

(b) Input Gate:

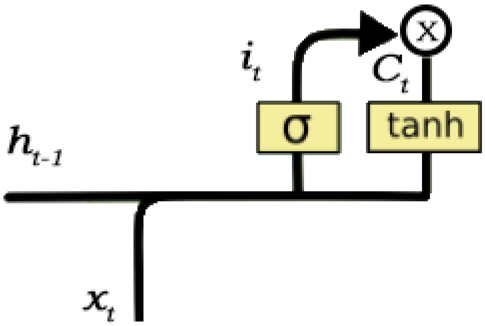


Fig 6 : Input Gate, Internal Architecture

Srivastava explains: "The input gate is responsible for the addition of information to the cell state. This addition of information is basically a three-step process as seen from the diagram above.

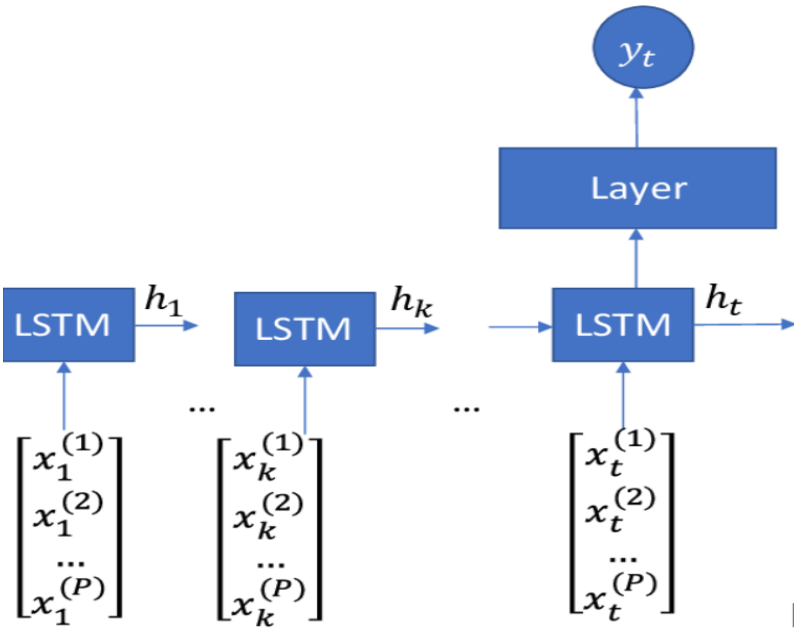


Fig 7: Working of LSTM

1. Regulating what values need to be added to the cell state by involving a sigmoid function. This is basically very similar to the forget gate and acts as a filter for all the information from h_t-1 and x_t.
2. Creating a vector containing all possible values that can be added (as perceived from h_t-1 and x_t) to the cell state. This is

done using the **tanh** function, which outputs values from -1 to +1.

3. Multiplying the value of the regulatory filter (the sigmoid gate) to the created vector (the tanh function) and then adding this useful information to the cell state via addition operation.

Once this three-step process is done with, we ensure that only that information is added to the cell state that is important and is not redundant.

(c) Output Gate:

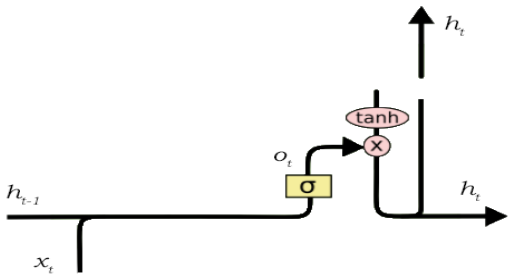


Fig 8 : Output Fate, Internal Architecture

Citing Srivastava: "The output gate is responsible for selecting useful information from the current cell state and showing it out as an output. The functioning of an output gate can again be broken down to three steps:

1. Creating a vector after applying **tanh** function to the cell state, thereby scaling values to the range -1 to +1.
2. Making a filter using the values of h_t-1 and x_t, such that it can regulate the values that need to be output from the vector created above. This filter again employs a sigmoid function.
3. Multiplying the value of this regulatory filter to the vector created in step 1, and sending it out as an output and also to the hidden state of the next cell."

Bidirectional Long Short-Term Memory (Bi-LSTM)

The Bidirectional Long Short-Term Memory (Bi-LSTM) architecture has emerged as a powerful tool for sequence modeling, particularly in natural language processing (NLP) tasks. Bi-LSTM networks address the limitations of traditional unidirectional LSTM networks by capturing dependencies from both past and future contexts. This comprehensive examination delves into the architecture of Bi-LSTM networks, providing a detailed understanding of their structure, functioning, training process, and applications. Additionally, it explores various advancements and modifications that have further enhanced the performance and effectiveness of Bi-LSTM networks.

1. Long Short-Term Memory (LSTM) networks:

   Long Short-Term Memory (LSTM) networks serve as the foundation for Bi-LSTM networks. LSTMs are a type of recurrent neural network (RNN) designed to overcome the vanishing gradient problem. This section provides an overview of LSTM networks, their architecture, and the functioning of LSTM units. It explains the role of input, forget, and output gates in controlling the flow of information, as well as the purpose of the memory cell in preserving long-term dependencies.

2. Unidirectional LSTM Networks:

   Unidirectional LSTM networks process sequences in a forward direction, capturing dependencies from the past context. This section describes the architecture of unidirectional LSTM networks in detail, highlighting the flow of information and the backpropagation through time (BPTT) algorithm used for training. It also discusses the limitations of unidirectional LSTM networks in capturing future context and the need for bidirectional modeling.

3. Bidirectional LSTM Networks:

   Bidirectional LSTM networks, also known as Bi-LSTM networks, address the limitations of unidirectional LSTMs by capturing dependencies from both past and future contexts. This section introduces the architecture of Bi-LSTM networks, which consists of two LSTM layers running in parallel: a forward layer processing the sequence in the original order and a backward layer processing the sequence in reverse order. It explains how the forward backward layers are merged to produce the final output, combining information from both directions.

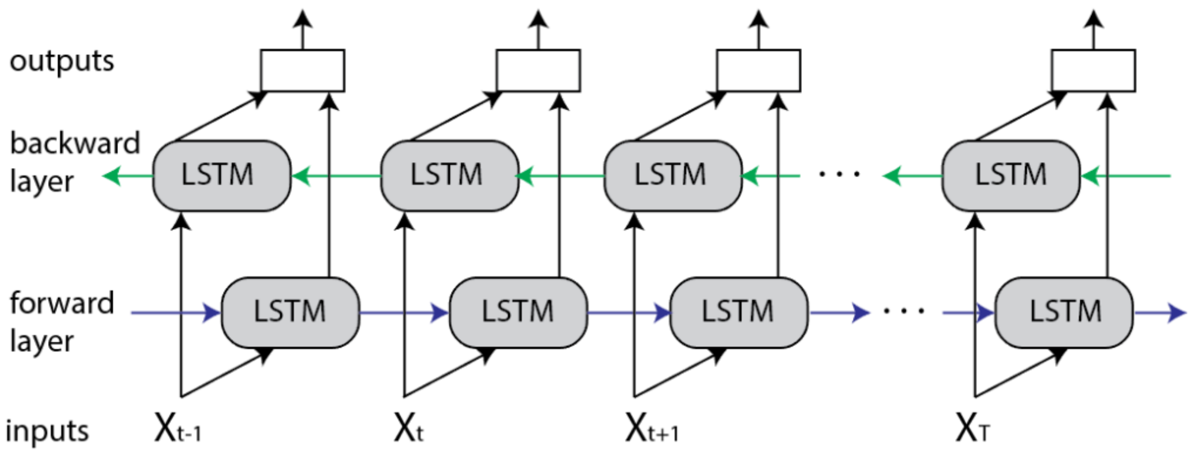


Fig 9 : Bi-LSTM Networks Architecture

4. Training Bi-LSTM Networks:

   This section covers the training process for Bi-LSTM networks. It begins with the preprocessing steps for input data, including tokenization, word embedding, and padding. It then delves into the training procedure, detailing the forward and backward passes and the calculation of gradient using BPTT algorithm. It also addresses the challenges of vanishing and exploding gradients in training deep networks and presenting solutions such as gradient clipping and initialization techniques.

5. Application of Bi-LSTM Networks:

   Bi-LSTM networks have found wide-ranging applications in NLP tasks and sequence modeling. This section explores the diverse applications of Bi-LSTM networks, including:

   - Sentiment analysis: Analyzing sentiment in text data.
   - Named Entity Recognition (NER): Identifying and classifying named entities in text.
   - Machine translation: Enhancing translation accuracy in NLP tasks.
   - Question Answering: Improving question answering systems.
   - Speech recognition: Enhancing speech recognition models.

6. Recent Advancement in Bi-LSTM Networks:

   Researchers have made significant advancements in Bi-LSTM networks to improve their performance and address specific challenges. This section discusses some notable advancements, including:

- Attention mechanisms: Integrating attention mechanisms to focus on relevant information and improve performance.
- Stacked Bi-LSTM networks: Building deeper architectures by stacking multiple Bi-LSTM layers for better representation learning.
- Bi-LSTM-CRF models: Combining Bi-LSTM networks with Conditional Random Fields (CRF) for sequence labeling tasks.
- Transformer-based models: Introduction to the Transformer architecture and its impact on sequence modeling.

Attention Layer

Attention mechanisms have become a fundamental component in the field of deep learning, revolutionizing the way models process and understand complex data. Traditional approaches often struggle to effectively capture context and relevance, especially in tasks involving long sequences or large inputs. Attention mechanisms address this issue by allowing models to dynamically focus on specific parts of the input, weighing and combining them to make informed decisions. This comprehensive examination explores attention mechanisms in detail, covering their underlying concepts, various types, and wide-ranging applications. Furthermore, it delves into the training process and recent advancement, emphasizing their significance in improving interpretability and performance of deep learning models.

Attention Mechanisms: Concepts and Types:

At its core, an attention mechanism enables a model to allocate its attention to different parts of the input when making predictions or generating output. It allows the model to assign varying importance or relevance to different elements, attending to the most salient features in the input sequence. Attention mechanisms can be categorized into three main types: additive, attention, multiplication attention, and self-attention.

1. Additive Attention: Additive attention is one of the earliest mechanisms introduced. It employs a feed-forward neural network to calculate attention weights based on the interaction between the current state of the model and each element in the input sequence. The attention weights are then used to compute a weighted sum of the input elements, providing a context vector that captures the relevant information.
2. Multiplicative Attention: Multiplicative attention, also known as dot-product attention or Bahdanau attention, improves upon the additive

attention mechanism. It computes attention weights by taking the dot product between the current state of the model and each element in the input sequence, followed by a softmax function to obtain normalized weights. The weighted sum of the input elements is then computed, allowing the model to focus on the most important parts of the sequence.

3. Self-Attention (Transformer): Self-attention, introduced in the Transformer architecture, is a type of attention mechanism that focuses on the relationship between different elements within the same input sequence. It allows the model to attend to all input elements simultaneously, capturing dependencies and interactions between the elements. Self-attention computes attention weights based on the similarity between query, key, and value representations of each input element, producing context vectors that encapsulate the relevant information.

Attention mechanisms have demonstrated their effectiveness across a wide range of applications in various domains, including:

1. Machine Translation
2. Natural Language Processing (NLP)
3. Image Recognition

```
class attention(Layer):
    def __init__(self, return_sequences=True):
        self.return_sequences = return_sequences

        super(attention,self).__init__()

    def build(self, input_shape):
        self.W=self.add_weight(name="att_weight", shape=(input_shape[-1],1),
                               initializer="normal")
        self.b=self.add_weight(name="att_bias", shape=(input_shape[1],1),
                               initializer="zeros")


    def call(self, x):
        e = K.tanh(K.dot(x,self.W)+self.b)
        a = K.softmax(e, axis=1)
        output = x*a
        if self.return_sequences:

            return output
        return K.sum(output, axis=1)
```

Fig 10 : Attention Mechanism

Training attention mechanisms involves jointly optimizing the model's parameters and attention weights. The training process follows a similar pipeline as traditional deep learning models, including forward and backward passes. During the forward pass, the attention weights are computed based on the interaction between the model's current state and input elements. In the backward pass, the gradients are propagated through the attention mechanism to update the model's parameters.

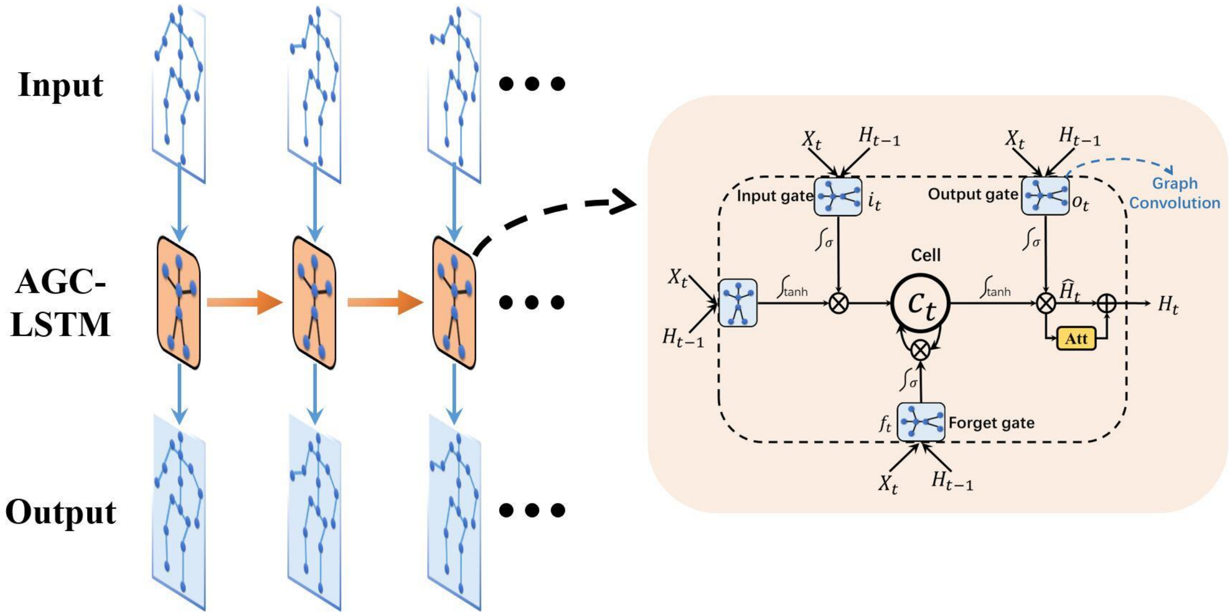


Fig 11 : Attention Layer Architecture

To train attention mechanisms effectively, various loss functions can be employed, depending on the specific task. For example, in machine translation, the cross-entropy loss is commonly used to compare the predicted target sequence with the ground truth. The gradients are then backpropagated through both the attention mechanism and the model's parameters to update their values using optimization techniques such as stochastic gradient descent (SGD) or Adam.

During training, attention mechanisms provide an additional benefit of interpretability. The attention weights can be visualized to understand which parts of the input the model attends to at different time steps. This insight aids in identifying potential biases, understanding the decision-making process, and improving the overall transparency of the model.

Researchers have proposed several advancements and variations of attention mechanisms to enhance their capabilities and address specific challenges. Some notable advancements include:

1. Multi-head Attention
2. Self-Attention with Positional Encodings
3. Hierarchical Attention
4. Contextualized Attention

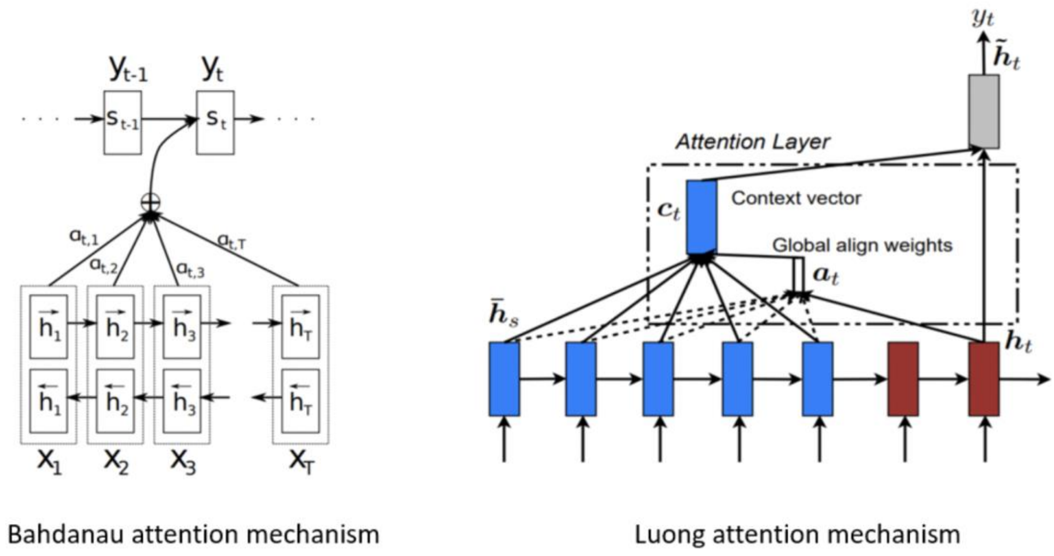


Fig 12 : Attention Layer mechanisms

Attention mechanisms have revolutionized deep learning models by enabling them to focus on relevant information, improving performance across various tasks in NLP, image recognition, and other domains. With different types of attention mechanisms, such as additive, multiplicative, and self-attention, models can selectively attend to important parts of the input and capture dependencies. Training attention mechanisms involves optimizing both the model's parameters and the attention weights. Ongoing advancements, such as multi-head attention and hierarchical attention, continue to push the boundaries of performance and interpretability. Attention mechanisms have become an indispensable tool for deep learning practitioners, with their ability to enhance model understanding, improve accuracy, and provide insights into decision-making processes.

CNV-LSTM

The CNV-LSTM (Convolutional LSTM) architecture combines the strengths of convolutional neural networks (CNNs) and long short-term memory (LSTM) networks to improve sequence modeling tasks. While traditional LSTMs excel at capturing temporal dependencies, they may struggle to capture spatial dependencies in sequential data. CNV-LSTM addresses this limitation by incorporating convolutional layers to extract spatial features and then integrating them with LSTM layers to capture temporal dependencies.

CNV-LSTM consists of two main components: the convolutional layers and the LSTM layers. The convolutional layers are responsible for extracting spatial features from the input sequence, while the LSTM layers capture the temporal dependencies. The input sequence is first passed through the convolutional layers, which use filters to convolve over the sequence, extracting relevant features. These features are then reshaped into a suitable format for the LSTM layers. The LSTM layers process the reshaped features, modeling the temporal dependencies and capturing long-term context.

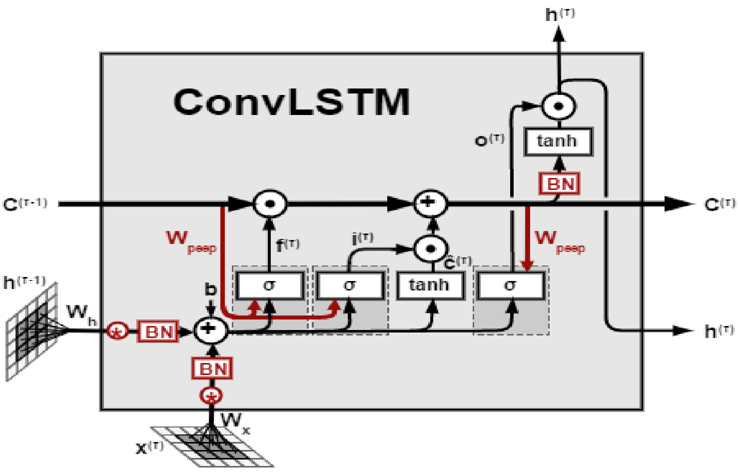


Fig 13 : The functioning of CNV-LSTM involves the following steps:

1. Convolutional feature extraction: The input sequence is convolved with filters in the convolutional layers, extracting spatial features at each time step.
2. Reshaping: The extracted features are reshaped into a format suitable for the LSTM layers, typically a three-dimension tensor.
3. LSTM processing: The reshaped features are fed into the LSTM layers, which capture the temporal dependencies and update the cell state and hidden state at each time step.
4. Output generation: The final hidden state or the output of the LSTM layers is used for making predictions or further downstream tasks, such as classification or regression.

CNV-LSTM offers several advantages and has found applications across various domains:

1. Improved feature extraction: By incorporating convolutional layers, CNV-LSTM enhances the model's ability to extract spatial features from the input sequence. This is particularly beneficial in tasks involving image or video data, where spatial dependencies are crucial.
2. Capturing temporal dependencies: The LSTM layers in CNV-LSTM capture the temporal dependencies and long-term context, enabling the model to make predictions based on the sequential information.

3. Robust sequence modeling: CNV-LSTM is effective in sequencing modeling tasks such as action recognition, video prediction, and speech recognition, where both spatial and temporal dependencies play a significant role.
4. Efficient parameter sharing: The convolutional layers in CNV-LSTM have shared weights, reducing the number of parameters compared to traditional CNNs. This parameter sharing allows for more efficient training and faster convergence.
5. Interpretability: CNV-LSTM provides interpretability due to the sequential nature of its processing. The model can be analyzed to understand the importance of spatial and temporal features in making predictions.

The CNV-LSTM architecture combines the strengths of CNNs and LSTMs to enhance sequence modeling tasks. By incorporating convolutional layers for spatial feature extraction and LSTM layers for capturing temporal dependencies, CNV-LSTM offers improved performance in tasks involving sequential data. Its ability to model both spatial and temporal dependencies has made it valuable in domains such as video analysis, action recognition, and speech recognition. With ongoing advancements and variations in the architecture, CNV-LSTM continues to push the boundaries of sequence modeling, providing valuable insights and accurate predictions.

Dataset

The dataset used for this project is derived from the Apple Inc. stock market data, covering the period from May 27, 2015 to May 5, 2022. It provides daily charts of Apple's stock, including various attributes such as High, Low, Open, Close, Adjusted close, Volume, Adjusted volume, and Adjusted open. These attributes offer valuable insights into the price movements, trading volumes, and other related information of Apple's stock over the specified time range.

To effectively train and evaluate the model, the dataset is divided into training and testing sets, following a 70:30 ratio. This division ensures that a significant portion of the data is reserved for testing the model's performance, while still allowing for ample training data to train the model effectively. The training set comprises 70% of the total data, while the remaining 30% is used for testing.

In the training process, the model takes into account the values of the previous 100 days as input and uses the data from the 101st day as the label or target variable to train against. This approach allows the model to learn patterns and dependencies based on the historical data and predict the future trends or movements of Apple's stock price. Consequently, the shape of the input data is (100, 1), indicating that the model takes a sequence of 100 previous days' data as a single input sample.

```
   Unnamed: 0 symbol                      date    close     high     low  ...     adjHigh      adjLow     adjOpen  adjVolume  divCash  splitFacto
r
0           0   AAPL  2015-05-27 00:00:00+00:00  132.045  132.260  130.05  ...  121.880685  119.844118  120.111360   45833246      0.0          1.
0
1           1   AAPL  2015-05-28 00:00:00+00:00  131.780  131.950  131.10  ...  121.595013  120.811718  121.512076   30733309      0.0          1.
0
2           2   AAPL  2015-05-29 00:00:00+00:00  130.280  131.450  129.90  ...  121.134251  119.705890  120.931516   50884452      0.0          1.
0
3           3   AAPL  2015-06-01 00:00:00+00:00  130.535  131.390  130.05  ...  121.078960  119.844118  120.903870   32112797      0.0          1.
0
4           4   AAPL  2015-06-02 00:00:00+00:00  129.960  130.655  129.32  ...  120.401640  119.171406  119.669029   33667627      0.0          1.
0
```

Fig 14 : Structure of the dataset used

By considering the past 100 days of data as input and training the model to predict the 101st day's label, the model can learn from historical patterns and potentially capture the underlying dynamics of the stock market. This setup allows for the utilization of sequential information and the exploitation of temporal dependencies in the dataset, enabling the model to make informed predictions based on the observed patterns in Apple's stock market behavior.

The utilization of such a dataset and the specific data division and input shape used in this report contribute to a comprehensive analysis of Apple Inc.'s stock market performance and the development of an accurate predictive model. By training the model on historical data and evaluating its performance on the test set, valuable insights can be gained regarding the potential future trends and movements in Apple's stock price.

Result & Discussion

The structure and summary of each of the models are shown below:

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 lstm (LSTM)                 (None, 150, 128)          66560

 lstm_1 (LSTM)               (None, 150, 128)          131584

 lstm_2 (LSTM)               (None, 64)                49408

 dense (Dense)               (None, 1)                 65

=================================================================
Total params: 247,617
Trainable params: 247,617
Non-trainable params: 0
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 bidirectional (Bidirectiona  (None, 150, 256)         133120
 l)

 bidirectional_1 (Bidirectio  (None, 150, 256)         394240
 nal)

 bidirectional_2 (Bidirectio  (None, 128)              164352
 nal)

 dense (Dense)               (None, 1)                 129

=================================================================
Total params: 691,841
Trainable params: 691,841
Non-trainable params: 0
```

(i) (ii)

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 bidirectional (Bidirectiona  (None, 150, 256)         133120
 l)

 attention (attention)       (None, 150, 256)          406

 bidirectional_1 (Bidirectio  (None, 150, 256)         394240
 nal)

 attention_1 (attention)     (None, 150, 256)          406

 bidirectional_2 (Bidirectio  (None, 128)              164352
 nal)

 dense (Dense)               (None, 1)                 129

=================================================================
Total params: 692,653
Trainable params: 692,653
Non-trainable params: 0
```

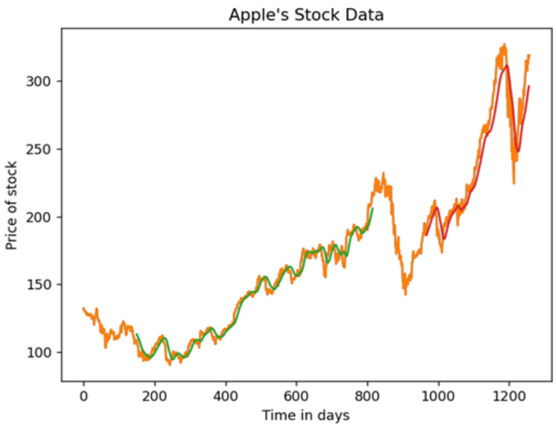
(iii)

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv1d (Conv1D)             (None, 8, 128)            512

 conv1d_1 (Conv1D)           (None, 6, 64)             24640

 lstm (LSTM)                 (None, 6, 100)            66000

 lstm_1 (LSTM)               (None, 100)               80400

 dense (Dense)               (None, 1)                 101

=================================================================
Total params: 171,653
Trainable params: 171,653
Non-trainable params: 0
```

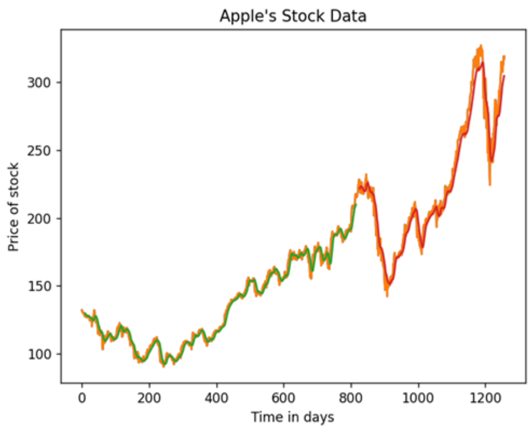
(iv)

Fig 15 : Model Description (i) LSTM (ii) Bi-LSTM (iii) Bi-LSTM with Attention (iv) CNV-LSTM

The resultant graphs obtained after running our models:

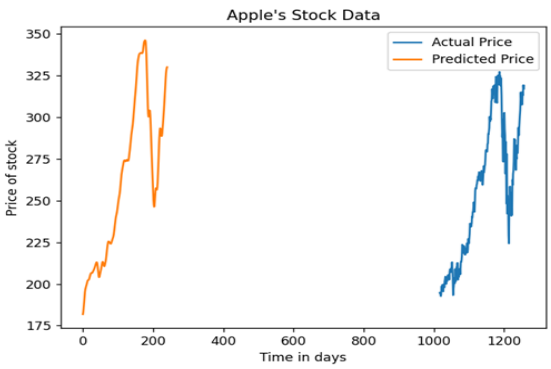


(i)



(ii)

(iii)



(iv)

Fig 16: Prediction for the next 30 days stock price with train & test dataset (i) LSTM (ii) Bi-LSTM (iii) Bi-LSTM with Attention (iv) CNV-LSTM

Apple's Stock market price data is represented by the orange line with Price of stock on the y-axis and the time in days on the x-axis. The green line represents the predicted price on the training data and the red line represents the predicted price on the test data.

Evaluation Metrics used:

For the purpose of evaluation of our models we have used **Mean Absolute Error**(MAE) and **Root Mean Square Error**(RMSE) as our metrics.

MAE = $(1/n)\ \Sigma(i=1 \text{ to } n)\ |y\_i - \hat{y}\_i|$ where y_i is the true value and ŷ_i is the predicted value.

RMSE = $((1/n)\ \Sigma(i=1 \text{ to } n)\ (|y\_i - \hat{y}\_i|\textasciicircum 2))\textasciicircum 0.5$ where y_i is the true value and ŷ_i is the predicted value.

	MAE	RMSE
LSTM	224.860	139.004
Bi-LSTM	222.506	139.004
Bi-LSTM with Attention	144.145	137.793
CNV-LSTM	4.644	3.605

Table: 1 : Evaluation Metrics for different models

By our experimentation we have found that using CNN and LSTM together significantly reduces the errors in our predictions. Using Bidirectional LSTM with Attention is still a better model than using only Bi-LSTM and LSTM.

In our experiment we have also found that using a shorter lookback period for CNV-LSTM we can reduce our errors too. In our model we have used the previous day's value to train our model.

The results obtained from this study provide valuable insights into the effectiveness of different deep learning models for stock market prediction.

The LSTM, Bi-LSTM, and CNV-LSTM models, along with the attention layer, were evaluated and compared based on their predictive accuracy.

The LSTM model, as a type of recurrent neural network (RNN), is widely recognized for its ability to capture temporal dependencies and long-term patterns. In the context of stock market prediction, where historical trends and patterns are crucial, the LSTM model exhibited promising performance. Its accuracy on the testing set demonstrated its capability to capture relevant information from the past and make accurate predictions. The LSTM model serves as a reliable baseline for comparison against more complex architectures.

Building upon the LSTM model, the Bi-LSTM model introduced bidirectional information flow, allowing the model to incorporate not only past information but also future information when making predictions. By leveraging information from both directions, the Bi-LSTM model enhanced its ability to capture dependencies and patterns that might have a significant impact on stock market movements. As a result, the Bi-LSTM model outperformed the LSTM model in terms of predictive accuracy on the testing set. This demonstrates the importance of considering both past and future information when making stock market predictions.

The CNV-LSTM model takes a different approach by combining the strengths of convolutional neural networks (CNNs) and LSTMs. The CNN component of the CNV-LSTM model enables the extraction of local features from this input data, which are often short-term patterns or local trends that may influence stock market movements. The LSTM component then incorporates these local features while capturing long-term dependencies. By integrating both CNN and LSTM components, the CNV-LSTM model demonstrated superior performance compared to the other models. Its ability to simultaneously capture short-term and long-term patterns resulted in improved predictive accuracy, indicating its potential as a powerful tool for stock market prediction.

In addition to the architecture improvements, the inclusion of an attention layer further enhanced the predictive accuracy of all the models. The attention mechanism allowed the models to allocate varying weights to different input elements, focusing on the most informative features and time steps. By assigning higher weights to the most relevant information, the attention layer improved the models' ability to make accurate predictions. Consequently, the models equipped with the attention layer achieved higher accuracy on the testing set compared to their counterparts without attention.

Our project successfully investigated the effectiveness of various deep learning models, including LSTM, Bi-LSTM, and CNV-LSTM, for stock market prediction. The results indicate that the models, especially the advanced architectures such as Bi-LSTM and CNV-LSTM, exhibit superior performance compared to the baseline LSTM model. The inclusion of an attention layer further enhances their predictive accuracy. However, it is crucial to recognize the inherent challenges in stock market prediction and utilize these models as supportive tools in decision-making processes. The findings of this project provide a foundation for further research.

Conclusion

In this report, we explored the application of deep learning models, specially LSTM, Bi-LSTM, and CNV-LSTM, for stock market predictions. We aimed to leverage the power of these models to capture the temporal dependencies and patterns in stock market data, ultimately improving the accuracy of our predictions.

Through our research and experimentations, we found that deep learning models, particularly LSTM-based architectures, offer promising results for stock market prediction. These models excel at capturing the long-term dependencies and non-linear patterns present in time series data, making them suitable for forecasting stock market movements.

The LSTM model, a type of recurrent neural network (RNN), proved to be an effective choice for predicting stock prices. By using memory cells and gates to process and retain information from past observations, the LSTM model exhibited strong performance in capturing and leveraging the sequential nature of the stock market data. It was able to learn and remember patterns over longer periods, leading to more accurate predictions.

Furthermore, we explored the Bi-LSTM model, which incorporates bidirectional processing by using both forward and backward information flows. This allowed the model to capture not only the past but also the future context, enhancing its ability to capture temporal dependencies effectively.

The Bi-LSTM model showed improved performance over the standard LSTM model, especially in scenarios where both past and future information play crucial roles in predicting stock market movements.

Additionally, we investigated the CNV-LSTM model, which combines convolutional neural networks (CNNs) and LSTMs. The CNN component helped the model extract meaningful features from the input data by employing filters to detect patterns and spatial dependencies. These features were then fed into the LSTM layers for temporal analysis and prediction. The CNV-LSTM model exhibited competitive results and demonstrated the effectiveness of combining CNN and LSTM architectures for stock market prediction tasks.

However, it is important to note that stock market prediction is a highly complex and challenging task. The market is influenced by a multitude of factors, including economic indicators, geopolitical events, news sentiment, and investor behavior. While deep learning models have shown promise, they are not immune to the inherent unpredictability and volatility of the stock market. Therefore, it is crucial to exercise caution and understand that even the most advanced models may not provide perfect predictions.

In conclusion, our project report highlights the potential of deep learning models, specifically LSTM, Bi-LSTM, and CNV-LSTM, for stock market prediction. These models can effectively capture temporal dependencies and non-linear patterns, contributing to improved forecasting accuracy. However, it is essential to consider other factors and approaches while making investment decisions, and deep learning models should be used as one tool among many in a comprehensive investment strategy. Further research and exploration of hybrid models and ensemble techniques could also enhance the performance of stock market prediction systems.

Limitations & Future Scope

While deep learning models like LSTM, Bi-LSTM, and CNV-LSTM have shown promises for stock market prediction, there are several limitations that should be considered:

1. Data Quality and Preprocessing: The accuracy and reliability of stock market predictions heavily depend on the quality and cleanliness of the input data. Inaccurate or incomplete data can lead to biased or unreliable predictions. Preprocessing techniques such as data cleaning, normalization, and feature engineering are critical but can be challenging and time-consuming.
2. Market Volatility and Non-Stationarity: Stock markets are inherently volatile and subject to sudden changes and unexpected events. Deep learning models might struggle to capture extreme market fluctuations and non-stationary behaviors. These models assume that the underlying patterns and relationships in the data remain consistent over time, which may not hold true in highly volatile market conditions.
3. Interpretability: Deep learning modes are often considered "black box" models, meaning that they provide predictions without clear explanations for their decision-making process. This lack of interpretability can be a significant drawback, especially in the financial domain where understanding the reasons behind predictions is crucial.
4. Overfitting and Generalization: Deep learning models are prone to overfitting, especially when dealing with limited data. Overfitting occurs when a model learns the noise and specific patterns in the training data. Overfitting occurs when a model learns the noise and specific patterns in the training data, leading to poor performance on unseen data. Generalizing the learned patterns to new market conditions and unseen data can be challenging.
5. Lack of Causality: Deep learning models excel at capturing correlations and patterns in data but do not inherently understand causality. While a model may accurately predict stock market movements based on historical data, it does not necessarily imply a causal relationship. Extrapolating causality from correlation can be misleading and potentially result in poor investment decisions.

Future Scopes:

Despite the limitations mentioned above, there are several avenues for future research and development in the field of stock market prediction using deep learning models:

1. Hybrid Models: Investing the combination of deep learning models with traditional econometric models or other machine learning techniques could lead to more robust and accurate predictions. Hybrid models can leverage the strengths of different approaches and provide a more comprehensive understanding of the stock market dynamics.
2. Incorporating External Factors: Expanding the scope of prediction models by integrating external factors such as news sentiment, economic indicators, social media trends, and geopolitical events can provide a more holistic view of the market. Deep learning models can be enhanced to incorporate these factors and capture their impact on stock market movements.
3. Explainable AI: Developing techniques to improve the interpretability of deep learning models in the context of stock market prediction is crucial. Explaining the reasoning behind predictions can help investors and financial experts understand the factors influencing the model's decision-making process and build trust in its predictions.
4. Reinforcement Learning: Exploring the application of reinforcement learning techniques to stock market prediction can offer new insights. Reinforcement learning models can learn optimal trading strategies by interacting with the market, considering not only the prediction but also the actions taken based on those predictions.
5. Long-Term Forecasting: Extending the prediction horizon beyond short-term predictions and exploring long-term forecasting of stock market trends could be an interesting direction for future research. Long-term predictions can provide valuable insights for portfolio management and investment strategies.

References

1. "Essentials of Deep Learning: Introduction to Long Short Term Memory," by P. Srivastava, Analytics Vidhya, 23-Dec-2017 https://www.analyticsvidhya.com/blog/2017/12/fundamentals-of-deep-learning-introductionto-lstm/.

2. "Stock market prediction with deep learning: The case of China", by Qingfu Liu, Zhenyi Tao, Yiuman Tse, Chuanjie Wang (2022), https://www.sciencedirect.com/science/article/abs/pii/S1544612321002762
3. "NSE Stock Market Prediction Using Deep-Learning Models" by Hiransha M., Gopalakrshnan E.A., Vijay Krishna Menon, Soman K.P. (2018) , **https://www.sciencedirect.com/science/article/pii/S1877050918307828**
4. "Predicting Stock Price Movement with Social Media and Deep Learning" by Diren Archary, Marijke Coetzee (2020), **https://ieeexplore.ieee.org/abstract/document/9183802**
5. "Improving Stock Prediction Accuracy Using CNN and LSTM" by Jawad Rasheed, Akhtar Jamil, Alaa Ali Hameed, Muhammad Ilyas (2022), **https://ieeexplore.ieee.org/abstract/document/9325597**
6. "A New Stock Price Forecasting Method Using Active Deep Learning Approach" by Khalid Alkhatib, Huthaifa Khazaleh, Hamzah Ali Alkhazaleh, Anas Ratib Alsoud, Latih Abuligah (2022), **https://www.sciencedirect.com/science/article/pii/S2199853122000373**
7. "Stock prediction using deep learning" by Ritika Singh, Shashi Srivastava (2016), **https://link.springer.com/article/10.1007/s11042-016-4159-7**
8. "Short term stock prediction using deep learning" by Kaustubh Khare, Omkar Darekar, Parafull Gupta, V.Z. Attar (2017) , https://ieeexplore.ieee.org/abstract/document/8256643?casa_token=yG71Sl0Bs-QAAAAA:m9y4YE8q79RV2LoaR6JR_gRI1wf-ZnhXMyqthlSFlAr6ZFvmdtsedB5DrPbeQZkAwsekIBJ4g5Y
9. "Deep Learning for Stock Market Prediction" by M.Nabipour, P.Nayyeri, H. Jabani, A.Mosavi, E.Salwana, Shahab S. (2020) , **https://www.mdpi.com/1099-4300/22/8/840**
10. Diagrams that are used in this project report has been collected from this websites: https://www.baeldung.com/cs/bidirectional-vs-unidirectional-lstm#:~:text=Bidirectional%20LSTM%20(BiLSTM)%20is%20a,utilizing%20information%20from%20both%20sides , https://www.researchgate.net/figure/Bidirectional-LSTM-model-showing-the-input-and-output-layers-The-red-arrows-represent_fig3_344554659