Stock Market Prediction from Financial Indicators and Daily News Feed with Deep Learning

SAMRAT SENGUPTA

A thesis submitted in fulfillment of the
requirements for the award of the degree of
MASTER OF SCIENCE IN MACHINE LEARNING AND AI

LIVERPOOL JOHN MOORES UNIVERSITY (LJMU)

NOVEMBER 2020

TABLE OF CONTENTS

DEDICATION

The work aspires to contribute to the field of AI-based Stock market forecasting and is dedicated to my family, thesis guides, and the community of researchers in the field of AI and ML without whom, it would not have been materialized.

ACKNOWLEDGEMENTS

I would like to thank my supervisor, Ms. Harika V, for all the support, advice, and the time that she spent to help me construct this work.

I also want to express my gratitude to research advisor Dr. M Jayabalan for his constant mentoring by online sessions.

My sincere acknowledgment to the student mentor Ms. R Sonaminde for her excellent coordination and communication.

It would be a mistake not to thank all the professors and members of the department that helped me with their online teachings through my master's program here at LJMU.

I want to say a big "thank you" to my wife who allowed me to dedicate prolonged hours in the study without whom I wouldn’t be able to achieve the goal.

Finally, I want to thank my late parents without whom I would not have been the person that I am today. Samrat Sengupta, Bengaluru, November 2020

ABSTRACT

Forecasting stock prices has been a challenging problem, and it has attracted many researchers in the areas of the economic market, financial analysis, and computer science. The current study is associated with the use of deep learning models to predict the intraday stock price changes of the Dow Jones Industrial Average (DJIA) - a stock market index with the help of daily news feed from Reddit and stock technical indicators.

In recent years, Convolution neural network (CNN), Recurrent neural network (RNN), and their variants are applied successfully in this domain, the most prominent being a jointed Recurrent convolution neural network (RCNN) architecture. However, these architectures have certain limitations. To overcome these researchers introduced advanced models like Transformers and Capsule net. The premise of using these architectures is based on the fact that the transformer extracts the deep contextual features of the news headlines while the capsule network captures the structural relationship of the texts to derive meaningful features. Here, a jointed architecture of Capsule net and Transformers is studied for analyzing the impact of global events in form of news.

Another relatively new architecture Tabnet which uses sequential attention to decide feature importance is explored for processing stock technical indicators. Not only Tabnet captures sequential features like Long short-term memory (LSTM) but also uses attention mechanisms to deduce insights from structured tabular data.

Finally, an ensemble of Tabnet and Capsule Transformer variants is carried out to project the overall impact of the daily news feed as well as exchange-based Technical indicators on stock price change. The theme of our current study is the formulation that these neoteric architectures produce better performance than widely used existing models in the domain of stock market forecasting.

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

GRU	Gated Recurrent Unit
LSTM	Long Short-Term Memory
CNN	Convolutional neural network
MLP	Multi-Layer Perceptron
RNN	Recurrent Neural Network
TF-IDF	Term Frequency Inverse Document Frequency
DJIA	Dow Jones Industrial exchange
ANN	Artificial Neural Network
RCNN	Recurrent Convolutional neural network
CONVID	Convolutional 1 dimensional
STI	Stock Technical Indicator
CAP-TE	Capsule Transformer encoder
EMA	Exponential Moving Average
DNN	Deep Neural network
BERT	Bidirectional Encoder Representations from transformers
GLOVE	Global vectors for word representation
GLU	Gated Linear Units
ETS	Error Trend Seasonality
ADF	Augmented Dickey-Fuller
EDA	Exploratory Data Analysis
XLNET	Generalized Auto-Regressive model for NLU
GPT	Generative Pre-training Transformer
ROBERTA	Robustly-optimized BERT approach
ALBERT	A Lite BERT
SVM	Support Vector Machines
KNN	K-nearest neighbors
SARIMA	Seasonal Autoregressive Integrated Moving Average
GARCH	Generalized Autoregressive Conditional Heteroskedasticity

ARIMA	Autoregressive Integrated Moving Average
NSE	National Stock Exchange
NYSE	New York Stock exchange
RMSE	Root Mean square error
MAE	Mean absolute error

CHAPTER 1

INTRODUCTION

This chapter provides an introduction to the background of the thesis and the gaps seen in the earlier researches. Further sections highlight how this study will address the gaps through its aim, objectives, and research questions. This chapter also highlights the scope and limitations of this study.

1.1 Background of the Study

The ability to predict the movement of the stock market is considered an important ingredient in investing. The objective of any investor should be to forecast the market behavior with the closest approximation to make the best decision while buying or selling a stock that can return an optimal profit. Hence a reliable stock market prediction model becomes an important instrument in achieving financial goals for an investor.

The DJIA stock index is a price-weighted average of stock prices from 30 of the largest publicly-traded companies in the United States widely considered as the benchmark for stock market analysis including price movement and its change.

Traditionally, technical analysis has been used to make near-term forecasts on this index, based on the premise that future values of a financial time series are conditioned on its past values. The modeling techniques for stock market forecasting remained in the caveats of statistical methodologies, up until the arrival of digital computation. The era saw the rapid rise of machine learning algorithms in the domain with a focus on decision trees, regressions, and an earlier form of the neural network like ANN's. However, these techniques have their limitations while performing complex feature extractions and interpreting underlying relationships in the data.

With the advancement in computationally robust machines, researches started focusing on deep learning as a reliable alternative to the existing techniques to overcome the above-mentioned limitations. Deep learning operates on the principles of imitating the human brain in the processing of data and creating patterns to make decisions. The deep learning architectures like CNN, RNN, and their variants had been successfully applied by analysts and researchers around

the world with improved performance in the current domain, the most widely used among them being recurrent convolution network (RCNN). The RCNN networks have been successful to extract information from both news titles by converting them to event embeddings and technical indicators to make predictions that beat predictors that do not use news titles. Experiments on a jointed architecture of CNN LSTM demonstrated substantial improvement in accuracy while predicting stock price change from news articles alone by Wang[6] using pre-trained word embeddings.

More advanced architectures like Transformers with their variants are relatively rarely explored methodologies in the current area of study though these models have been successfully implemented in areas of Neural machine translation and sentiment analysis by Myagmar[10]. Transformers are included as a subject of our study here since they can capture a deep semantic relationship between the sentences. Another recently introduced variant of the deep learning model Capsule net also showed promises in text classification problems because they are trained to extract fine-grained structural and spatial information from complex text articles as shown by Zhao[12]. This architecture and variants are also explored here. Finally, a combination of both Transformer and capsule-Net is tried on pre-trained word embeddings to arrive at optimal performance for news headlines driven regression model.

The technical indicators for the stock price prediction model are based on statistical representation and the moving average of last n days exchange data. Apart from LSTM a recently introduced architecture namely Tab-net is tried to fit those indicators to produce optimum output. Tab-net chooses reasonable features at each decision step using sequential attention. Hence enables interpretability and better learning as the learning capacity is preserved for the most prominent features as been depicted by Arik[14].

Instead of following traditional approaches of feature concatenation from two different networks processing text and numerical data a unique ensemble technique using a linear model is introduced to calculate the final prediction.

1.2 Problem statement

It is a cumbersome task for financial analysts and traders who participate in stock market trading to appropriately predict the behavior of stock and arrive at a decision for buying/selling it. The volatility and extremely diverse factors affecting the stock market slide/surge makes it a formidably challenging task.

Hence it would be extremely beneficial for the investors if a reliable stock market forecasting model is made available that can predict the future change of stock price with optimum accuracy.

In the course of the current study, we would like to put forward a deep learning-based model that takes into account all the diversified factors like technical indicators derived from the stock and financial news captured in the form of global events that can predict intraday stock market price movement with considerable precision.

Hence the development of a robust AI-based stock market forecasting model to predict intraday stock price change would be an ideal problem statement.

1.3 Aims and Objectives

This work aims to provide a full-proof deep learning-based stock market forecasting model with technical indicators from the Dow Jones Industrial exchange and daily news headlines from Reddit news as input to predict the change of price in stock.

The objectives of the research are outlined as followed.

- In the context of the current study, we aim to derive stock-based technical indicators from DJIA as well as glove/Bert embedding based text features on Reddit news as input to the proposed deep learning model.

- We plan to enhance the dataset further by incorporating Reddit news articles and DJIA stock exchange from yahoo finance from date Aug-2016 to Jul 2020

- We scrutinize the baseline models like jointed ConvID-LSTM which is near the state of art in the current domain as per preceding researchers for analyzing the impact of textual information when taken as input, independently. Thereafter we focus to examine more advanced architectures like Transformer, Capsule-net, and their variants (Cap-Te) to analyze any improvement in prediction accuracy with their introduction.

- Similarly, we pursue to perform a comparative analysis of widely used LSTM based models with the more recently introduced tabnet on data from stock-based technical indicators.

- Our main objective is to put forward an ensemble prototype out of the best performing models one each from these two different input-based features. We would achieve that by taking the weighted average of test predictions as input to another linear model to arrive at the final prediction.

1.4 Scope and Significance of the Study

The study of advanced deep learning architectures for stock market prediction using both news headlines and technical indicators in this research can lead not only to the establishment of near state of art models for stock market forecasting but also open an entirely new outlook for using hybrid neural networks for further improvement by researchers using the current concepts as a baseline.

A reliable model that meticulously derives insights from global events in form of news and historical stock features and can perform stock forecasting with optimum accuracy is immensely helpful for financial investors in making decisive investments that can lead to considerable financial gains and minimize risking a considerable fortune. Therefore, it can be of utmost benefit to people related to investment banking and finance, shareholders and brokers, and stock market analysts.

The research though uses Dow-jones Industrial exchange with Reddit news headlines as a primary source that applies to any other similar stock exchange-based data like NSE, NYSE, etc. The deep learning architectures that are studied here include capsule net, transformers, and Tab-net which has been previously implemented successfully in the sentiment analysis and predictive analytics domain. The study is focused on the implementation, hybridization, and ensemble of these deep learning models to achieve an optimum outcome concerning the current domain of stock forecasting. It excludes detailed analysis of the previous state of art model (CNN-LSTM) in this domain as well as theories related to reinforcement learning which is based on more temporal data (hourly) and is also a current trend.

1.5 Structure of the Study

The study contains Literature review and Research methodology chapters mentioned in the table of contents.

The systematic literature review has been performed by referring to various conference papers, journals, articles, and books related to the datasets, methodologies, and evaluation metrics. It begins with a survey of influencing factors in the realms of Stock market forecasting namely Data mining of Stock indicators and Text mining of daily news articles. This leads to carrying out further reviews of the dataset to be used in the study, modeling techniques, and joint architectures to analyze the impact of both factors. After doing a brief thesis on the above we set to explore works based on advanced deep learning architectures to come out with a proposition for the current study in the stock market domain.

In the research methodology section data-preprocessing, visualization, word embeddings, and model architectures (considered as a baseline and proposed) mentioned in the literature review are explored in depth. We put forward a detailed analysis behind each of the theories formulated in the literature review and determine the methodologies of the work to be carried out in experiments. Finally, we derive a unique ensemble technique for our models to arrive at the result of our thesis.

CHAPTER 2

LITERATURE REVIEW

In the literature review section, we would be focusing on the overview of the existing and latest knowledge and studies related to the current domain. We would identify hypotheses and theories put forward in the relevant studies as well as gaps in them to lay a foundation of the background of our research including the novelty in our proposition.

2.1 Introduction

Although the Stock market prediction remains among one of the hottest topics of research in the financial domain, the volatile nature of stocks with diversified events and indicators affecting its price changes makes the task challenging. Typically, as in all other research areas, the methodologies for stock market prediction also walked the tide of continuous evolution.

The earliest forms of research in this domain included trend derivation and pattern recognition methods like exponential moving average (EMA), oscillators, support and resistance levels or momentum, and volume indicators as Candlestick patterns(1980's). These methods mainly considered stock exchange features to arrive at a prediction. With the increased availability of the internet, global events captured in the form of news headlines influencing slide and surge of stock prices were also taken into account.

With the arrival of digital computation, the stock market prediction has entered into the technological realm (1990's and 2000's). Artificial neural networks (ANNs) and Machine Learning Algorithms were among the most prominent techniques for some time.

However, with the advancement of computation powers emerged the era of deep learning and since then researches have been all been aimed at exploring different neural network-based architectures to solve the problem (2010's). The most successful and near state of the art having been variants of RNN/LSTM and CNN based hybrid architectures. In more recent times the Attention-based models like Transformers which capture deep semantic relations in text and Capsule net which captures structural information from has been applied successfully in fields

of neural machine translation and sentiment analysis. These models are studied concerning the current context over here. Another completely new form of architecture that also uses an attention mechanism for deriving the relationship between tabular predictor variables, namely Tabnet is also examined.

This study concentrates solely on deep learning-based methodologies for stock market prediction as emerging AI trends and techniques have proven to outperform ML-based or statistical models, now for a long time. The following sections provide analysis of input parameters, network architectures (both widely used and advanced), and recent trends in applications of stock market forecasting.

2.2 Factors affecting AI-based Stock Market forecasting

Contrary to the random walk theory, researchers suggest that the financial time series do not exhibit random behavior and that the stock price is predictable. Hence, several studies with different AI techniques were carried out over a while.

However, earlier researchers who used Fuzzy System Romahi[33], Genetic Algorithm Kim[32], Hidden Markov Model Hassan[34] or some hybrid combinations Leigh[35], Abraham[36] as well as statistical models, like moving average Hellstrom [37], had their shortcomings. Mainly, the role of information media (the news articles composed of both company-specific and relevant market sector details) which greatly affects the sale or purchase outlook of investors and plays a conclusive part in determining stock price surge/slides were not considered.

Mitchell[26] inspected the influence of collective data reported by Dow Jones and confirmed the existence of a direct relationship between released news articles and stock market activities. NofSinger [27] derived that in some cases buying of stocks and hence price surge is directly proportional to positive financial news and inversely proportional to the negative news resulting in price slide. Mittermayer [28] put forward a trading system that operates immediately after the publication of news articles through text mining and displayed positive outcomes. These researches significantly bore better results than those which ignored the impact of daily news.

However, relying only on investor behavior based on recently published news articles alone as a trading strategy is insufficient, as concluded by Brown [29]. The research conducted by Zhai [8], also proposes to combine the information from both the news release and technical indicators to enhance the predictability

of the daily stock price trends. Hence in this study, a combination of Stock Technical Indicators and news articles on daily basis is considered as input features.

The fundamental parameters of any stock exchange are the values of Open, Close, High, Low, Volume, Adj Close daily. However, these features though easily perceptible are too kaleidoscopic for deriving any substantial intuition. Hence different technical Indicators statistically derived from these variables (often over a certain period) are considered. They are represented as stock technical indicators or STI's.

STI's are independent of business fundamentals like revenue, profit/loss, or income. These are used by technical analysts and active traders to interpret long and short-term price movements with directions as well as entry and exit points to the stock as shown by Jabbarzadeh [39]. STI's also plays a significant role in the prediction of future price of the assets paving the way for easy integration into automated trading systems Chen[40], Oriani[41]. It has been experimentally proven from the above-mentioned studies that a combination of technical indicators particularly those that are represented by time lag functions can improve the accuracy of models remarkably compared to the fundamental indicators. The researchers conducted by Vargas[2],Zhai[8] uses seven sets of such technical Indicators, that achieved significantly improved predictive results when applied to deep learning models like ANN, LSTM, etc. These are mentioned in detail in Table1.

Feature	**Description**	**Formulae**
%K	Stochastic %K. It compares where a security's price closed relative to its price range over a given time.	C_t - L_{ln} / Hh_n - Ll_n
%D	Stochastic %D. Moving average of %K.	sum (i=0 to n) Kt_{-i} / n %
Momentum	It measures the amount that a security's price has changed over a given time	C_t - C_{t-n}

ROC	Price rate-of-change. It displays the difference between the current price and the price n days ago	$(C_t / C_{t-n)} * 100$
William's %R	Larry William's %R. It is a momentum indicator that measures overbought/ oversold levels	$(H_n - Ct/H_t - L_n) * 100$
A/D Oscillator	Accumulation/distribution oscillator. It is a momentum indicator that associates changes in the price	$(H_t - C_{t-1} / H_t - L_t)$
Disparity 5	5-day disparity. It means the distance of the current price and the moving average of 5 days.	$(C_t/MA_5) * 100$

Table 1. Technical Indicators ref Kim[38]

[where C_t is the closing price at day t, H_t is the highest price at day t, Lt is the lowest price at day t, MA_n is moving average of the past n days, and Hh_n and Ll_n are the highest high and the lowest low in the past n days, respectively].

These derived features were further normalized using z-score normalization The definitions in Table1 are inherited from the work of Kim[38]. The set of STI's mentioned above is used as inputs for the numerical feature-based deep learning model studied here.

Textual information collected in the form of daily news regarding global events plays a vital role in predicting the outcomes for stock market forecasting. The motivation for text mining approaches in stock market prediction has been derived from the work of Lee[1]. He demonstrated a significant improvement in the predictive power of a stock (relative to an index) based on textual data derived from daily news.

However, irrespective of how valuable might raw text data be it needs to be converted into a suitable format to be used as an input to different AI models. Hence different techniques of text mining adopted in related studies over the years become an interesting topic of discussion. Traditional approaches for depicting textual information used noun phrases, named entities, bags-of-words (Bow), and term frequency-inverse document frequency (TF-IDF). these approaches were not able to capture entity-relation information or the semantics of news headlines.

Deep Learning becomes effective once words are converted to high dimensional distributional vectors that capture syntactic, morphological, and semantic information. Word embeddings that convert word to vector form have achieved remarkable results when applied to neural networks. Hence a group of studies emerged to further the development of different word embeddings. Kim[3] used pre-trained word vectors for sentence-level classification. Sentence level vectors have also been constructed in sentiment analysis tasks as has been shown by Socher[15]. The paragraph-level encoding was put forward by Le[4]. However, these techniques had their shortcomings in either capturing of both semantics and positional occurrence of a word simultaneously or in sparsity resolution of sentence vectors, considering the entire corpus.

Pennington[23] combined two pioneering approaches in existing methodologies; global matrix factorization and local context window methods to formulate a new global log bilinear regression model or glove. This model efficiently leverages statistical information by training only on the nonzero elements in a word-based co-occurrence matrix to produce a vector space with meaningful substructure. Hence glove based word embedding for text mining is adopted in the current study.

2.3 Survey of DJIA and Reddit News as Data source

To analyze the directional as well as the differential movement of the stock market and the effect of global news on it, a reliable and widely accepted stock exchange as well as an authentic news content platform is needed to be examined in detail.

Hence in the current study Dow Jones Industrial Average (DJIA), which is the second oldest U.S. market index is chosen. DJIA created back in 1896 by Charles Dow Edward Jones, tracks 30 large, publicly-owned blue-chip companies trading on the New York stock exchange.

For studying the impact of global events in the form of daily news on the directional or differential movement of DJIA, Reddit news is chosen as a bulletin board system. Reddit.com founded in 2005 is an American social news aggregator and a web content rating website ranked as the 19th-most-visited in the U.S. and the world.

This work accredits the effort of Sun[58] for providing an accessible archived record for DJIA stock and Reddit news in the Kaggle website to be used in the present work.

Mitchell[60] studied the relation between the number of news announcements reported daily by Dow Jones & Company and aggregated measures of securities in market activity including trading volume and market returns. Their work revealed Dow Jones announcements and market activity are directly related and the results are robust enough to influence financial markets.

Bollen[59] was able to bring the mood of people posting on Twitter about the behavior of the stock market. They used the classified Twitter messages combined with the corresponding stock market values (Dow Jones Industrial Average) as a training set for a neural network. Their model achieved an accuracy of 86.7%. These researches proved the effectiveness of DJIA as a reliable data source for stock market predictive modeling when commensurate with news-based media that projects public sentiment or global events.

Ovadia[60] in his journal, did an introspective study of Reddit and concluded it to be a curated source of news and opinion mining. More so, as the quality of an article is determined by voting and subscriptions from the other users the social psychology of the public is also captured. Since these also reflect investor behavior it can be concluded that information mining from Reddit news can play a significant role in social media and financial analytics-based researches.

Onchareon[57] used News headlines from Reddit World News Channel to study the effect of financial events stock. The AI model performed moderately well while predicting the intraday direction movement of the DJIA.

The current study hence considers Dow Jones Industrial Exchange along with the Reddit news archive as the raw data source for Stock market forecasting models.

2.4 Deep Learning in Stock Market Forecasting

The high nonlinearity and randomness of Stock price data can set challenges on traditional time series methods like the ARIMA, SARIMA, and the GARCH.

These models are effective only when the series is stationary. This is again a restricting criterion that requires preprocessing of the series taking log returns or other such transforms to calculate predictions. In live trading systems as there is no guarantee of stationarity as new data is added, these techniques encounter a lot of issues.

This is counterbalanced by using Neural Networks which do not need stationarity in time series data. Furthermore, neural networks by nature are effective in finding the relationships between data and using it to predict (or classify) new data. Researches carried out by Choi[30], Quah[31] proved the efficacies of ANN; one of the rudimentary forms of deep learning architectures, in the current domain. This is not only true in the case of deriving outputs from time series-based stock Indicators but also in the instance of deriving semantically significant and structurally related information from news articles. Numerous significant researches were conducted related to widely used neural networks of current time like Multilevel perceptron, Convolution neural network (CNN), recurrent neural network (RNN), and their joint variants CNN-RNN, RCNN, etc. An analysis outlining the impacts of these group of architectures on the current study are detailed in the below sections.

The MLP's were the earliest among the variants of neural networks that worked on the principle of applying non-linear activation function between input and output layers. MLP's were extensively recommended in most earlier researches related to stock market predictions conducted by Wang[44], Guresen[45]. This is owing to their higher capabilities for function approximation enabling them to learn the underlying patterns from the data. Studies conducted by Shynkevich [42] also confirmed the improved results while using ANN's over traditional techniques like support vector machines (SVM) and K-nearest-neighbors (KNN).

However, ANN's designed as function approximators were not meant for memorizing sequential information over time. Recurrent neural networks (RNNs) introduced in the late '90s were suitable for processing sequential data such as sound, time-series data, or written natural language as shown by Lipton[46]. Some designs of RNNs were used to predict the stock market Rather[47] as well.

Long Short-Term Memory (LSTM) is one of the most powerful RNNs architectures. It introduces memory cells replacing traditional neurons of hidden

layers, which enables networks to effectively associate memories and input that are remote in time. Hence LSTM is suited to grasp the structure of data dynamically over time with high prediction capacity. Li[49] showed practical research value for applying LSTM in the field of stock market forecasting. The work of Zeng[48] confirmed the superiority of LSTM in capturing fluctuating stock values with respect to time. Therefore, LSTM based architectures are examined for STI based inputs and are considered baseline models in this work.

While Technical Indicators indeed have a role to play in deciding outcomes of stock market forecasting a major effect is attributed to the investor behavior and global financial events captured in form of daily news, thereby revealing the vast scope of AI-driven techniques that uses textual information as input in the current domain.

Theories put forward by Kim[3] revealed excellent results in sentence classification tasks using a convolution neural network (CNN) on top of pre-trained word2vec encoding. Kalchbrenner [50] proposed a dynamic CNN based sentence modeling network that handles input sentences of varying lengths and induces a feature graph that is capable of explicitly capturing short and long-range relations. These researches established the effect of CNN's in capturing the semantic and structural relationship of text-driven inputs for language modeling tasks. Since textual information is sequential, the need for interpreting the underlying sequential relationship has arisen.

RNN/LSTM's are the bunch of deep learning architectures that exactly does the same. Studies conducted by Wang[53], Li[54] confirms the improved accuracy of Text classification models with LSTM when used with suitable embeddings. A more advanced version of LSTM, BiLSTM has been scrutinized with respect to sentiment analysis of comments by Xu[55]. BiLSTM being capable of propagating information both forward and backward in time captures better contextual information that resulted in improved accuracy in the mentioned research.

Taking into consideration the impact of capturing structural information from words/phrases by CNN's as well as interpreting sequential order/long term contextual dependencies by LSTM's a coupled architecture comprising both the variants were put forward by Wang[6]. This Conv1d-LSTM has been accepted as a baseline model for the current scope of analysis.

There is a substantial number of researches done over a period highlighting the superior performance of joint variants of CNN and RNN's.

Vargas[2] proposed Recurrent-Convolution neural network (RCNN) for predicting the intra-day directional movements of the Standard & Poor's 500 Index (S&P 500). His model uses word2vector based embeddings for textual input. He also used a set of seven technical indicators as proposed by Zhai[8] like [Stochastic %K, Stochastic %D, Momentum, Rate of Change, William's %R, A/D Oscillator, and Disparity 5] which are commonly used to describe assets, as another different set of input to an LSTM based network that predicts the same directional stock movement. Finally, he combined the prediction probabilities of the 2 networks with a SoftMax layer to arrive at the final output.

This theme of jointed architectures, taking into account both STI and text-based news has also been observed in the work of Oncharoen[57], who used a deep CNN with a fully connected dropout layer for text processing as well as LSTM with SoftMax for historical data processing and finally passed the concatenated output to another fully connected layer for achieving the result. Up until, very recently, a recurrent convolutional neural kernel (RCNK) model was proposed by Liu[56], which learned complementary features from different sources of data, namely, historical price data and text data in the message board, to predict the stock price movement. Fig 1 given below depicts the model proposed by Onchareon.

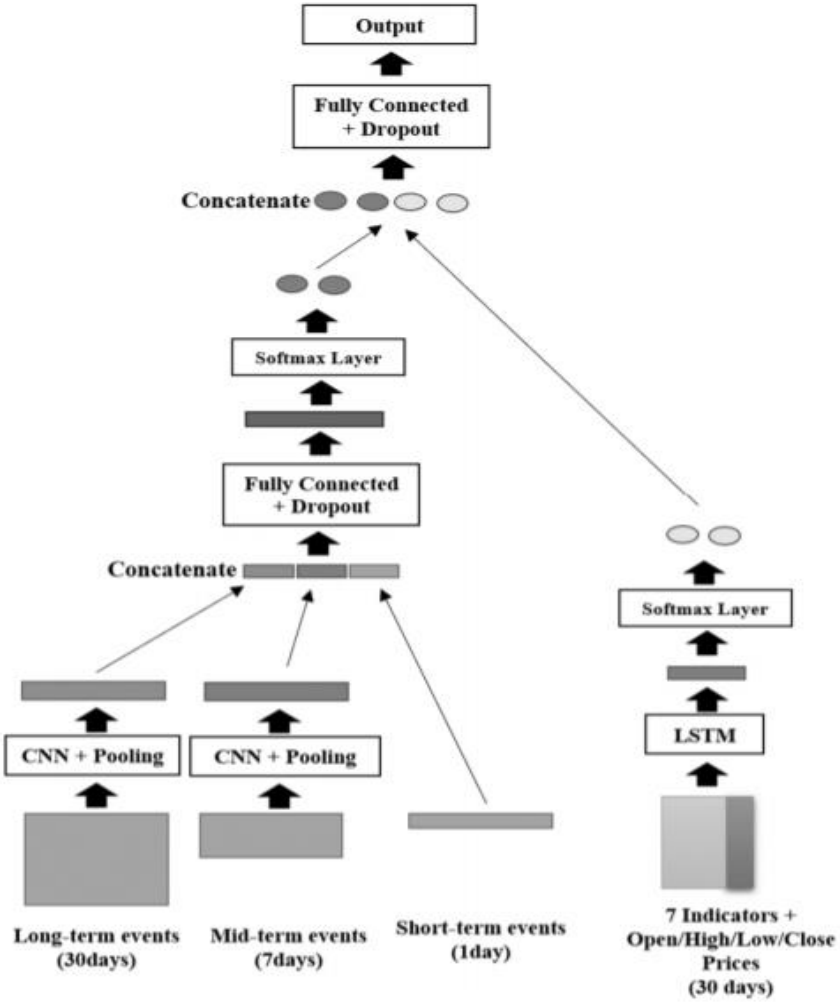


Fig 1. The model was proposed by onchareon[57].

[The idea of ensemble architecture put forward in this thesis is envisioned based on this jointed network.]

The above studies demonstrated the 360-degree impact of all the inputs considered and produced superior results as claimed in the papers. However, these were implemented for a binary classification task where the output is a set of probabilities. The output layers of both these networks do not consider the scope for predicting a cumulative continuous output. The removal of output layers and concatenating networks at the hidden layer level may lead to negate the impact of STI's due to the incompatibility of generated feature maps. The current work though motivated on studying the cumulative impact of both stocks based technical indicators and Text-based news articles for its outcome is concerned with predicting a continuous output that is the change of price. Hence, instead of concatenating outputs/features from different networks, a unique

ensemble technique that emanated from the above researches is used to predict a continuous output.

2.5 Advanced Deep Learning Architectures

The architectures of not so recent era like CNN/RNN have all has their limitations which gave rise to the advancement of deep learning techniques and the introduction of new architectures like capsule net, transformers, and the tabnet. CNN through its mechanism of kernel filtering and pooling had one serious backdrop. The filtering method failed to capture the directional or vector properties of each element like rotation, angle, etc. which are more directional.

On the other hand, RNNs become ineffective when the gap between the relevant information and the point where it is needed becomes very large while processing long texts. LSTM's seemed to solve that problem by introducing cell states to preserve the information passed by previous words to a large extent. However still when the documents are long enough the model often forgets the context of distant positions in the sequence. To overcome these limitations researchers recently introduced cutting edge models like Transformers/Capsule and the tabnet which are adopted in the current study and explored in detail.

A relatively new form of architecture namely capsule net proposed by Sabour[20] seemingly overcomes these problems faced by CNN's by replacing scalar outputs of the convolved feature maps with vector outputs which take into account more directional features. Though primarily designed for image analytics this has been successfully experimented on for text classification purposes and several studies have been conducted to prove its effectiveness on textual information. The current work takes motivation from earlier studies conducted by Zhao[12] and Rathnayaka[13]. Here variants of capsule net are applied to analyze the impact of news headlines on stock price change.

To solve the issues encountered by LSTM's, researchers created a technique for paying attention to specific words. Attention was presented by Bahdanau[19]which reads as a natural extension of their previous work on the Encoder-Decoder model. In short attention in deep learning can be broadly explained as a vector of significant weights. The prediction of an element, for example, a pixel of an image or a word in a sentence are estimated using an attention vector which calculates a degree of correlation or attention with respect to other elements and considers their weighted sum as the approximation for the target variable. With the advent of attention mechanisms, there rose variants of

architectures based on the theory like self-attention, hierarchical attention, and multiheaded attentions.

The focus of the current study is based on multi-head self-attention or Transformer architecture proposed by Vaswani[9]. The multi-head self-attention layer in the Transformer aligns words in a sequence with other words in the sequence, thereby calculating a representation of the sequence. It is more effective in representation and less computationally efficient than convolution and recurrent networks. The intuition behind transformers inspired several researchers, leading to the development of self-attention-based models such as Bidirectional Encoder Representations from Transformers (BERT) by Devlin et al. Along with Bert, there have several transformer-based language models that showed breakthrough results such as XLNet, Roberta, GPT-2, and ALBERT.

With sentiment and opinion expressions becoming widespread throughout social and e-commerce platforms, the applications of Transformer variants mentioned above slowly found their applications in this works as shown in studies done by Myagmar[10] focusing on Cross-domain sentiment classification applying transfer learning methods which is an inspiration to the current work.

The combination of capsule-based Transformer network which is the final approach in current work is still relatively unexplored in the domain of text-based analytics in Stock market forecasting.

The attention theory was not only restricted in Textual/Visual information but found its implementation in realms of tabular data as well. Arik[14]proposed a novel architecture with sequential attention modules for tabular learning. An attention module is trained to select some key elements from the (normalized) input feature, and a feature transformer takes the selected features for overall feature embedding. The philosophy behind this architecture is driven by the fact that Tab-Net is designed to learn a decision-treelike mapping to inherit the valuable benefits of tree-based methods (interpretability and sparse feature selection) while providing the key benefits of DNN-based methods (representation learning and end-to-end training).

2.6 Discussion

From the above section, we can conclude that optimum results in stock market forecasting can be obtained by considering the impact of both technical indicators as well as daily news articles. The above theory sets the main theme of current

work. The choice of the dataset (DJIA-Reddit), feature derivation, and embedding techniques, as well as study of baseline models like CNN-RNN and LSTM as well as advanced models like Transformer and Capsules, has all been hypothesized from the preceding researches that conform to the idea. However, the originality of the current work lies in the introduction of a hybrid network of Transformer and Capsule namely Cap-Te for prediction from news articles as well as the Tabnet for prediction from technical indicators with respect to the stock market domain. The results obtained from this study are expected to demonstrate superior prediction outcomes by Cap-Te and Tabnet when compared to baseline models. Hence this work can be considered as a trend-setter in the realms of AI-driven stock market forecasting.

2.7 Summary

The literature review starts with a summary of the inception, evolution, and applications of the stock market forecasting problem indicated in the introduction. Thereafter factors affecting stock market predictions are analyzed with references drawn to past studies and intuitions derived from them.

This study goes along with the proven concept that stock predictions give optimum results when the impact of both technical indicators (statistically derived stock features from base parameters like high, open, close) and global events captured in form of daily news is taken into consideration.

For modeling purposes, as per the subject of this study, we start with exploring works done on deep learning-based techniques keeping in mind it's superior performance over statistical and machine learning-based methodologies. We conduct an extensive study of variants of some of the widely used architectures like RNN and CNN like (Conv1d-LSTM) with respect to textual information-based modeling as well as LSTM in the case of Technical Indicator based modeling. We also observe from the studies of past researchers that a jointed architecture taking into consideration both stream of models yields better results. However, encouraged by the superior performance of recently introduced deep learning models like Transformers and Capsule nets in the fields of sentiment analysis, this work also implements the mentioned models in the domain of stock market forecasting based on news articles. This is done to examine the attention-based contextual interpretation capability of transformers and a more text-structure oriented semantic feature derivation capability of capsule nets.

Going by the same note, another recently introduced architecture, Tabnet is also investigated here. Tabnet has been claimed to combine the capabilities of a decision tree-based model with a DNN and have yielded high accuracies for predictions based on tabular data.

As the outcome of this research instead of concatenating a text-based and numerical data-based model, we derived an ensemble technique that combines the weighted predictions of both forms of input with the help of a linear model. The research methodology section below provides a meticulous analysis of the models studied here, their implementations, and ensemble techniques.

CHAPTER 3

RESEARCH METHODOLOGY

This section covers the methodology followed for data analysis, model building, model evaluation, and model ensemble based on the CRISP-DM framework. Here the entire outline of the thesis conducted has been analyzed, examining each of the steps taken and justifying the reason at arriving at them.

3.1 Introduction

A detailed flow-chart depicting End-to-end processes in research methodology is shown in Fig 2 below.

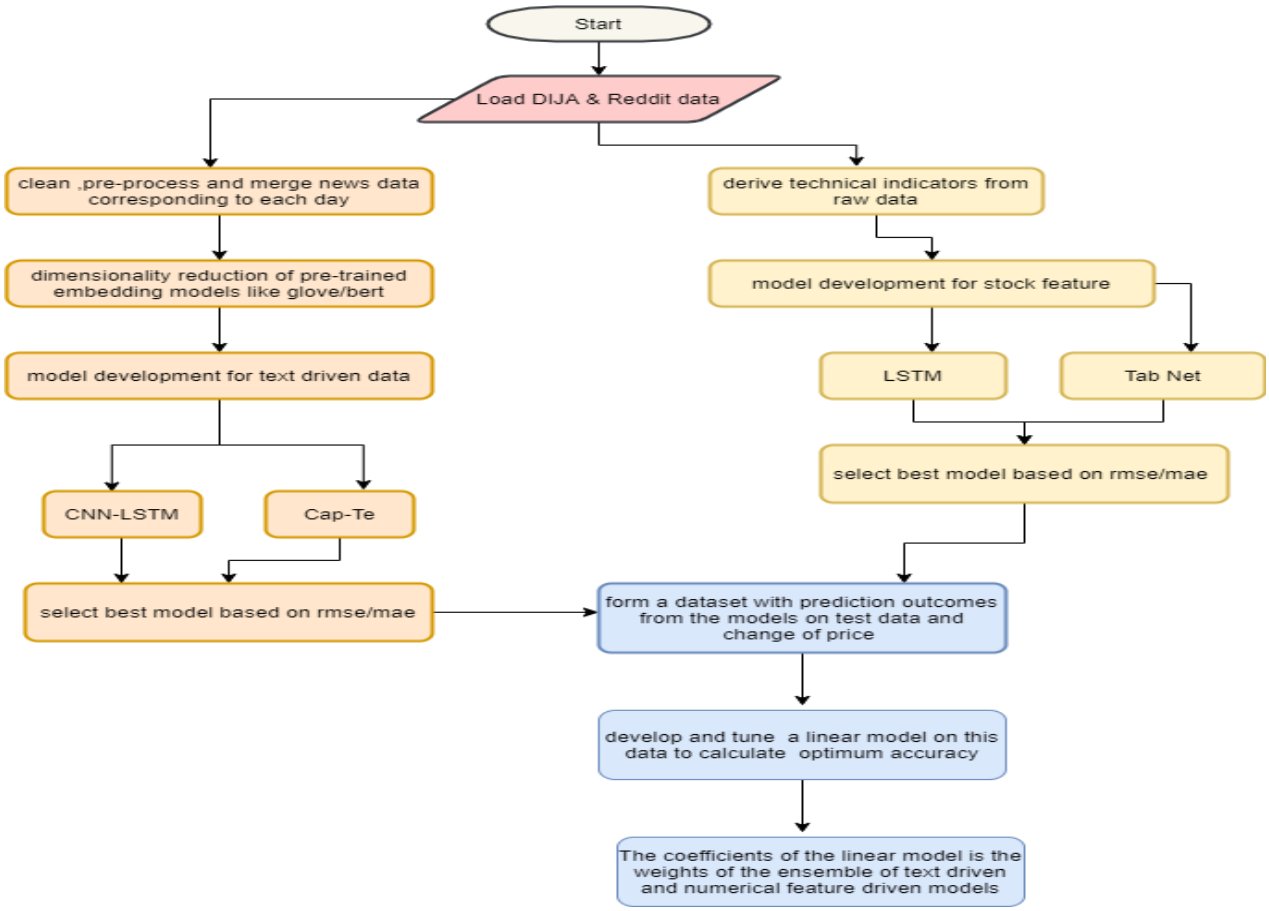


Fig 2. Research Methodology Flow-chart

3.2 Data Selection

The dataset contains approximately 74k Reddit news article titles with their corresponding dates, as well as data about the Dow-Jones Industrial average per day from 8-Aug-2008 to 1st-Jul-2016. The Source of data files is https://www.kaggle.com/aaron7sun/stocknews. Dataset is broadly divided into 2 files.

The dataset has been further enriched by adding data from 2nd Jul 2016 to 1st July 2020. For Reddit news, we used multithreaded Push Shift API enabled crawlers to crawl the top 25 daily news headlines between the afore-mentioned date.

The corresponding Dow Jones stock exchange data is manually downloaded from the Yahoo Finance site's archive. https://finance.yahoo.com/quote/%5EDJI/history/

These data files (both news and historical stock values) are merged with the main dataset programmatically to produce the final dataset keeping the format the same.

Hence the final size of the dataset is as follows:

➢ RedditNews.csv: 2 columns 109600 rows. The first column is the date, and the second column is the news headlines. All news is ranked from top to bottom based on user voting on hotness quotient Hence, there are 25 lines for each date

➢ DJIA_table.csv: 7 columns and 4384 rows. 7 columns and 1990 rows. The columns for DJIA data sources are date, open, high, low, close, volume, Adjacent close.

3.3 Data Pre-processing and Feature Derivation

The entire daily news articles from 7-Aug-2008 to 8-Jul-2020 are to be merged to form the master text corpus. Word cloud and word frequency histograms would be projected on this corpus for visualization of entire raw textual data. We would carry out data cleaning techniques like *filtering out non-ASCII chars*, *de-contractions* as well as *stopword removal* and do a further projection of word

cloud and frequency histogram. Topic modeling would also be carried out to interpret broader prospects of global news categories and their impacts on stock price changes.

The raw data from DJIA _table.csv is to be examined to observe for null rows. Since it's a daily stock data no outlier's detection is required (as sudden slides or surges in stock prices are to be taken into account to study the market behavior in detail).

While performing EDA on stock exchange data first we would propose the following techniques.

- Rolling means of Opening price with adjusted windows smoothens the noise over a single day and helps to capture the trend more efficiently than the original value over a day. we need to come to an optimum window value

- The Expanding function helps to understand whether the trend of the stock prices is increasing, decreasing, or stationary.

- ETS decomposition models to help us visualize time series data with respect to the components like Trend, Seasonality, Cyclical patterns, irregular patterns.

- Finally, the Augmented Dickey-Fuller (ADF) test needs to be conducted to reject or not reject the null hypothesis that the time series has a unit root, indicating whether it is stationary or non-stationary. We can take the differential intraday value of any Stock Open/Close parameters to revaluate p values obtained from ADF and carry on till we get a stationary time series.

The above methodologies would help us to deduce a differential/actual stock price that can be formulated as a response variable and set up a context for deriving STI's. [This is due to the fact that dynamic regression and hence the derivation of Technical Indicators for that purpose becomes applicable on a time series data that exhibits seasonality and trends. Also, the response variable which is the 1st order differential value of stock Opening price demonstrates stationarity implying that it can be used as a stable response variable for regression]

Technical Indicator derivation from Stock Exchange Parameters

We would formulate the technical indicators from DJIA as per the hypothesis adopted in the literature review and insights derived from the previous section. For stock price data in Dow Jones Index table which contains the date, open, high, low, close, volume, adjacent close as base columns, the following features are derived and normalized.

➢ $Stochastic\ \%K\ =\ Ct\ -\ Lln\ /\ Hhn\ -\ Lln$

➢ $Stochastic\ \%D\ =\ sum\ (i = 0\ to\ n)\ Kt - i\ /\ n\ \%$

➢ $Momentum\ =\ Ct\ -\ Ct - n$

➢ $Rate\ of\ Change\ =\ Ct\ /\ Ct - n\ *\ 100$

➢ $William's\ \%R\ =\ (Hn\ -\ Ct/Ht\ -\ Ln)\ *\ 100\ 12$

➢ $A/D\ Oscillator\ =\ (Ht\ -\ Ct - 1\ /\ Ht -\ Lt)$

➢ $Disparity\ 5\ =\ Ct/MA5\ *\ 100$

[where Ct is the closing price at day t, Ht is the highest price at day t, Lt is the lowest price at day t, MAn is moving average of the past n days, and Hhn and Lln are the highest high and the lowest low in the past n days, respectively].

The description of the parameters is already given in detail in Table 1.

These values are taken as predictors and intra-day stock price change as the response variable. While doing so the data from the last row in the dataset is ignored as we are considering the change of price with respect to the next day.

3.4 Predictive Modelling from DJIA Stock Exchange Data

This section deals with deep learning predictive models based on technical indicators that are derived from basic stock exchange parameters.

3.4.1 Baseline LSTM model

Among the most widely used architectures for stock market forecasting from original parameters or technical indicators of the stock exchange are the LSTM's

and their variants. In the current study, we examine the model put forward by Zeng[48] as our baseline.

Here, 2 layers of LSTM with dropouts are stacked up and finally connected with a dense layer having linear activations. This is connected to the output layer with a single node and linear activation to get the final output as shown in Fig 3. Adam Optimizer is chosen to oversee backward propagation

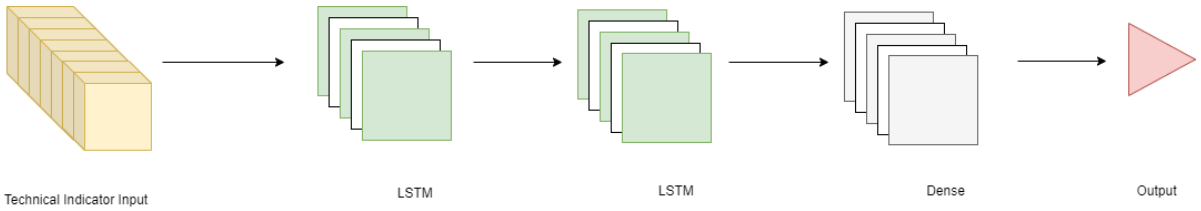


Fig 3. LSTM model proposed by Zeng[48]

The details of the layers are listed in Table 2.

Layers	**Output shape**
LSTM	None, 1, 5
LSTM	None, 5
Dense	None,1

Table 2. LSTM Model Layout

The reason behind LSTM's popularity in time series-based predictions is because it can memorize sequential inputs over a large number of steps. Since it is considered as a baseline, we have skipped a detailed description of it's working principles.

3.4.2 Proposed Tabnet Model

The Tab-Net introduced by Arik[14] is the Integration of DNNs into decision trees that can outperform the tree-based algorithms while reaping many of their

benefits by applying principles of transformer architecture while doing feature extraction.

The tabnet , like decision trees, follows a top-down approach of feature elimination and selection through a multi-step decision-making process. The building blocks of tabnet comprises of transformer blocks performing sequential attention while extracting features from a tabular data. This type of sequential attention is used for step by step elimination of redundant features using a sparse matrix-based function. The most significant property of the tabnet is that feature preprocessing is redundant here due to its built-in inter-operability. Hence, we need not consider applying iterative feature extraction techniques like RFE or Shap. An overview of the architecture is given in Fig 4.

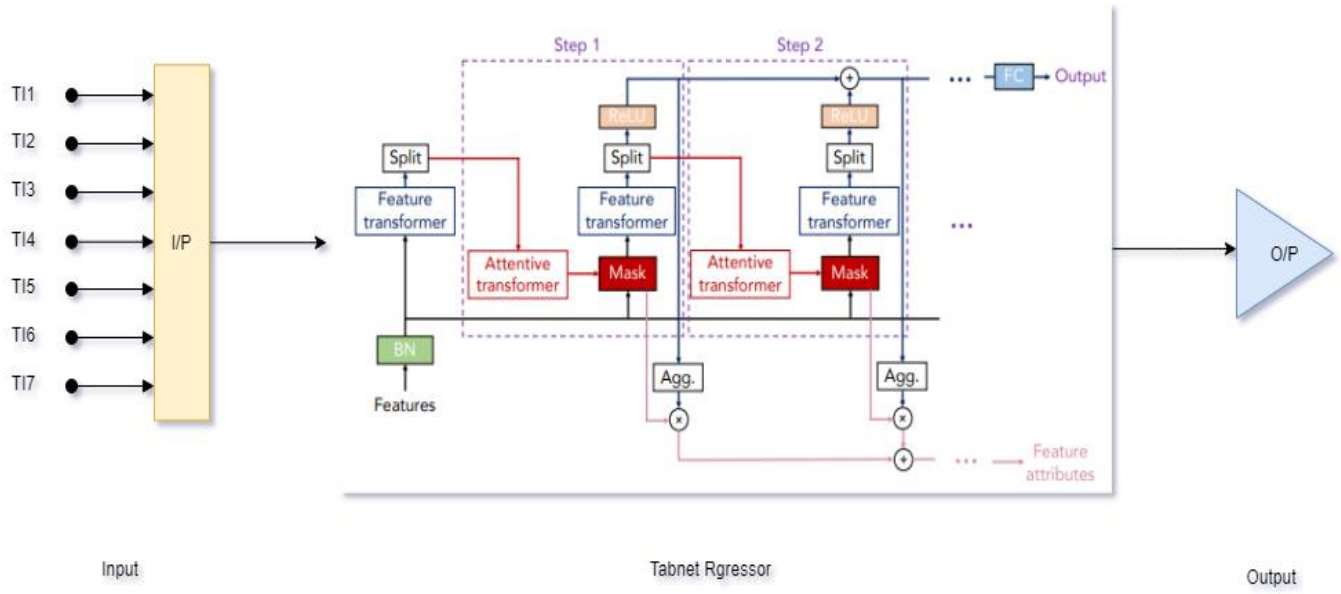


Fig 4. Tabnet Encoder architecture with Technical Indicators as input

[T1-7 are the 7 technical indicators in the discussion. The figure is referenced from Arik[14]]

The input features are passed through the Batch normalization layer into a feature transformer block which splits them further and routes them to transformer blocks. Each Transformer block consists of attention transformers that create masked features to be fed into another feature transformer (within the block) which splits the inputs into 2 features. One of them is passed to the activation function (ReLu) to produce a predictive output and the other is used as input to a similar block. The predictive outputs over the encoder blocks are summed up to provide the final output.

Given in Fig 5 is a view of the feature transformer unit of tabnet.

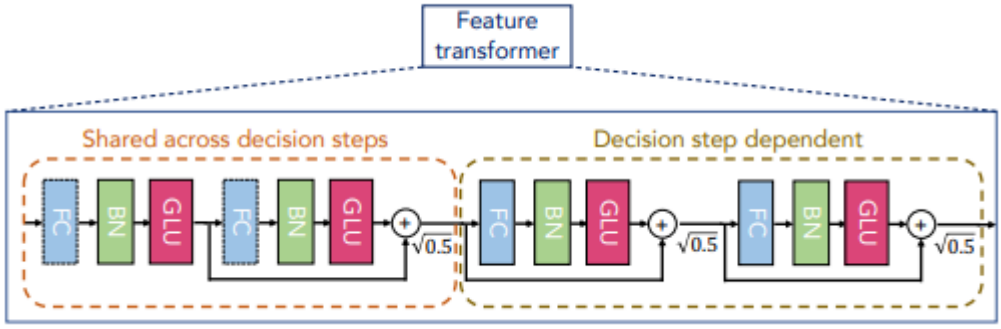


Fig 5. Feature Transformer unit in Tabnet ref Arik[14]

The feature Transformer block consists of a series of 4 consecutive GLU blocks which in turn can be broken down to units of Fully Connected - Batch Norm - GLU in that order.

The Glu function can be thought of as the product of the sigmoid output of a feature and the feature itself.

$$GLU(x) = \sigma(x) \cdot (x)$$

There are 2 shared and 2 independent GLU blocks with skip connections between 2 independent blocks as shown in the diagram. The dimension of the output from the block is the summation of the decision and attention features dimension.

Given Fig 6 is the pictorial depiction of the attention transformer unit of Tabnet.

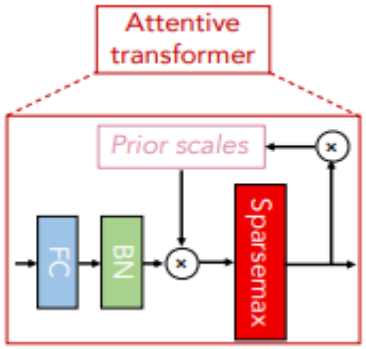


Fig 6. Attention Transformer unit in Tabnet ref Arik[14]

The attention unit consists of a fully connected layer, followed by a batch normalization layer. It is then passed to a prior function and then to a Sparse matrix function which generates feedback to the prior scales.

The prior function is going to tell about the intrinsic properties of the features and how many times they are used. In the beginning, the prior for all features are going to be 1. However, for the next iterations, prior function is going to be the multiplication of all previous steps which is equal to a relaxation parameter minus the previous mask. It can be represented by the below equation.

$$P0 = 1\ ,\ P[i] = i\prod j = 1(\gamma - M[j])$$

[P = prior function, γ = relaxation parameter usually >1, M = musk,

i, j = current and previous features]

A sparsemax function is a sparser version of softmax that calculates sparsity by mapping Euclidean projection into unidirectional probabilities.

$$M[i] = sparsemax(P[i-1] \cdot hi(a[i-1]))$$

Here h_i is a trainable function using fully connected layer followed by batch normalization.

We propose tabnet as one of our principle models for calculating stock price change from technical indicators and claim these to be a brand-new technique tried in domain of Stock market forecasting.

3.5 Predictive Modelling from Reddit News articles

In this section, the methodologies in the current study related to global events captured in form of news articles on stock market predictions have been analyzed meticulously.

3.5.1 Text Embedding

Text data cannot be fed directly to neural networks and needs some encoding into the numerical format. Word embeddings are a low dimensional version of sparsely populated co-occurrence matrices that represent documents in a corpus.

These mappings are done with respect to the vector space to preserve their distributional semantics. Google research first published the neural-based word embedding - word2vec model in 2013. After that, researchers developed several embeddings based on frequency-based or prediction-based approaches like Glove, Elmo, Bert. In the current study, we take into consideration Glove and Bert.

3.5.1.1 Glove Embedding

GloVe (Global Vectors for Words) is an unsupervised learning algorithm for obtaining vector representation for words, developed by a Stanford research group. The basic idea behind the GloVe is factorizing the co-occurrence probability matrix into two smaller matrices. Initially word pairs i,j are formed based on a context window from the raw corpus. The probability that the word 'j' is in the context of the word 'i' is calculated for all words in the corpus. All such occurrences are counted and normalized to get the final probability P(j|i). The dot product of vector notation of each word denoted by Wi and Wj w.r.t the corpus are then compared to the probability P(j|i) and optimized through a suitable loss function. The resultant vectors are considered as those that encapsulate similarities and differences among individual words.Penington[23]'s work shows Glove to outperform CBOW and skip-gram models on word similarity, named entity recognition, and word analogy-based tasks achieving state of art accuracy of 75%. Fig 7 demonstrates the comparsion.

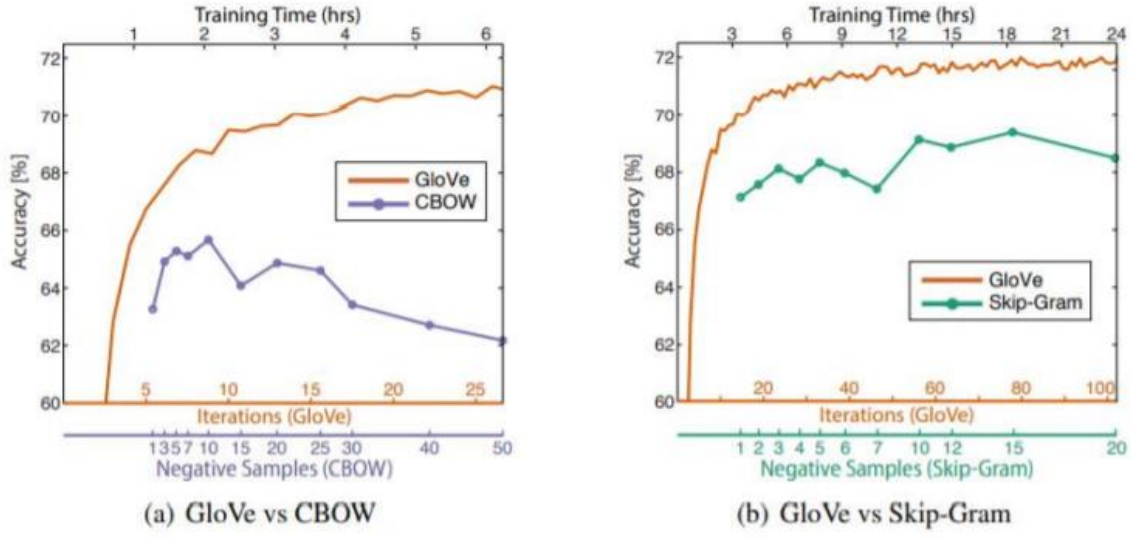


Fig 7: Accuracy vs Number of iterations plot on word analogy task as a function of training time ref Penington[23]

In the current study, we have used glove embedding for creating inputs to a CNN-LSTM model which is our baseline for stock market prediction from text-based data.

3.5.1.2 Bert Embedding

Bert stands for Bidirectional Encoder representations from transformers. It was published by researchers at Google AI language.

Bert is capable of extracting both semantic and contextual meaning from a sentence. It is based on transformer architecture which applies an attention mechanism to learn semantic and contextual meaning from text. The vanilla Transformer consists of 2 different techniques - the encoder which reads a text input and a decoder that calculates prediction from the text. Bert uses Masked LM (MLM) and Next sentence prediction (NSP) mechanisms to achieve it's training objectives which are based on the transformer encoder mechanism. The Bert mechanisms are explained in Fig 8.

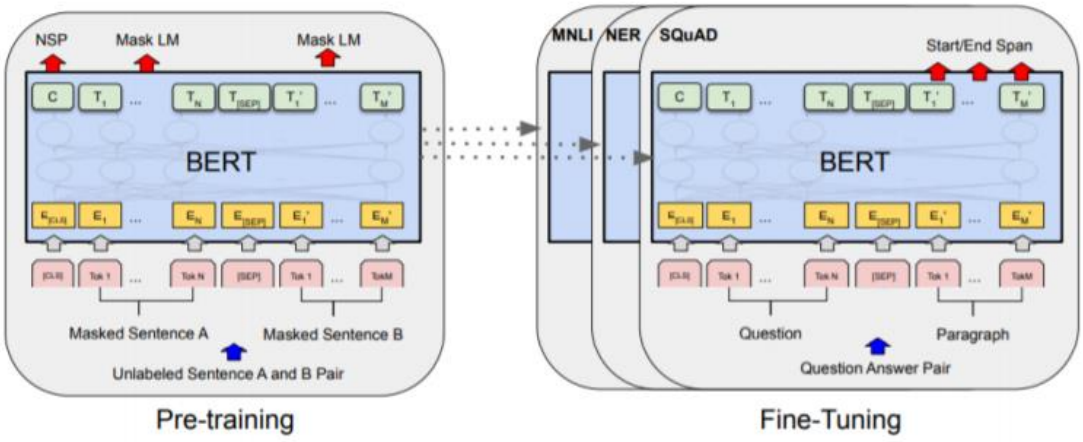


Fig 8. Bert pre-training and fine-tuning mechanisms Devlin[16]

The working principles of Bert are discussed in more detail in the modeling section where we have proposed Bert – capsule based model for stock market prediction from news articles.

The embeddings generated by the Bert encoder is fed to that model proposed in our study.

3.5.3 Baseline CNN-LSTM Model

The CNN-LSTM model chosen as baseline architecture is envisaged from the work of Wang[6]. This model designed for sentiment classification of texts performed extremely well on movie review and Stanford sentiment treebank datasets. The intuition behind using a jointed Convolution-LSTM architecture arises from the fact while the convolution layer is capable of deriving semantic relation between input sentences of a text the LSTM layer helps in interpreting sequential or contextual features. Hence a jointed architecture is proven to carry more impact in deriving complex features from text documents. The model architecture is depicted in Fig 9.

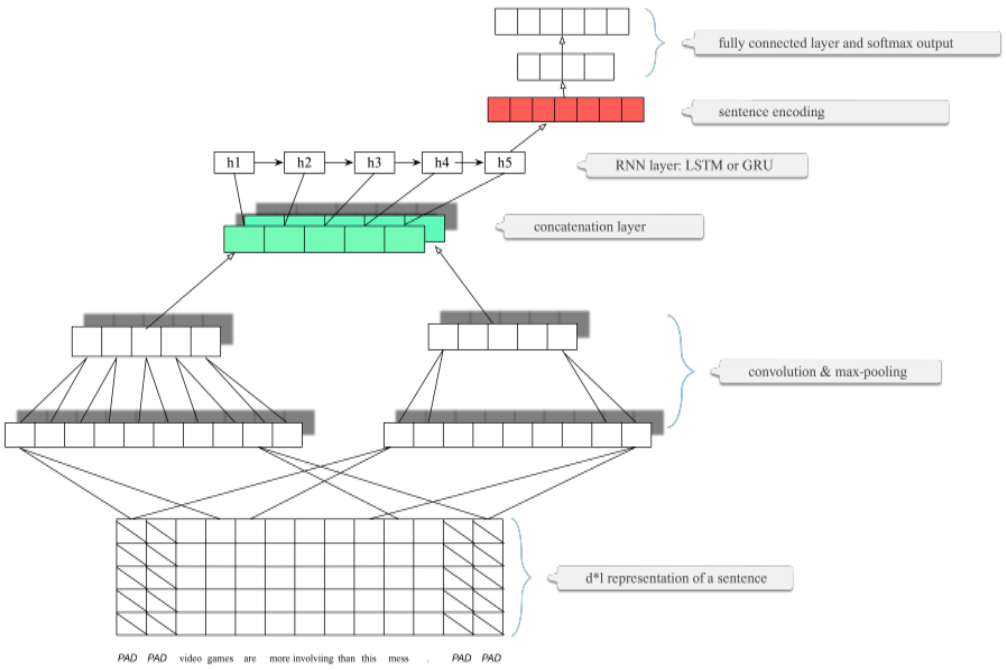


Fig 9. Conv-LSTM model ref Wang[6]

The original model in the architecture consists of the following components like word embeddings and sentence-level representation, convolutional and pooling layers, concatenation layer, RNN layer, fully connected layer with softmax output. In the current context of the study Dense layer is recommended to be of linear activations since the continuous output is processed, also dropouts can be added as per convenience instead of max pooling while hyperparameter tuning keeping the overall skeleton the same. An overview of possible layers with outputs is given below.

A layer-wise Model structure is given in Table 3.

Layers	**Output Shape**
Embedding	(None,200,300)
Dropout	(None,200,300)
Conv1d	(None,200,16)
Dropout	(None ,200,16)
Conv1D	(None,200,16)
Dropout	(None,200,16)
LSTM	(None,128)
Dense	(None,128)
Dropout	(None,128)
Output(Dense)	(None,1)

Table 3. CNN-LSTM model layout

3.5.4 Proposed Transformer-Capsule Model

The capsule Transformer architecture, proposed in the current study is a combination of a pre-trained Bert model with a capsule layer, conjoined together.

Before arriving at the jointed architecture, extensive studies have been conducted individually on both the Transformer (Bert base) as well as the Capsule net which

is comprehended in the following sections. Cap-Te architecture is also analyzed in detail in the last section.

Transformer (Bert)

The Transformer model with their variants like Bert/Xlnet/GPT-2 etc. rose to prominence after displaying the state of the art results in the domain of text classification and sentiment analysis. Here we aim to apply Bert architecture for determining outcomes of daily news-based Stock market forecasting.

The Bert is based on Transformer architecture uses an attention mechanism to solve the problem of remembering the long-range dependency of words in a document. These are discussed here briefly.

The transformer uses a positional encoder that creates positional embeddings along with word embeddings so that words will be closer to each other based on both their meaning and position in a sentence. The padded tokens are masked. From the masked input vector, 3 vectors namely query(Q), Key(K), and Value(V) are created. They are updated and trained during training. Next self-attention is calculated for every word in sequence. Self-attention function is represented as

$$Attention(x) = Attention\,(Q, K, V) = SoftMax\,(QKT\,/\,\sqrt{dk})\,V$$

where dk is the dimension of keys and query. The output representing the multiplication of the attention weights and the V (value) ensures that focused words are kept as-is and irrelevant words are flushed out. Q, K, and V are split into multiple heads to jointly attend to information at a different position based on representational spaces, hence the term multi-head attention. The output from multi-head attention is passed to a pointwise feed-forward network consisting of two fully connected layers with a ReLu activation in between. The Transformer encoders can be stacked up to complete a block. Multiple such blocks constitute a layer. The Transformer mechanism is inspired by the work of Vaswani[9].A high level overview is depicted in Fug 10.

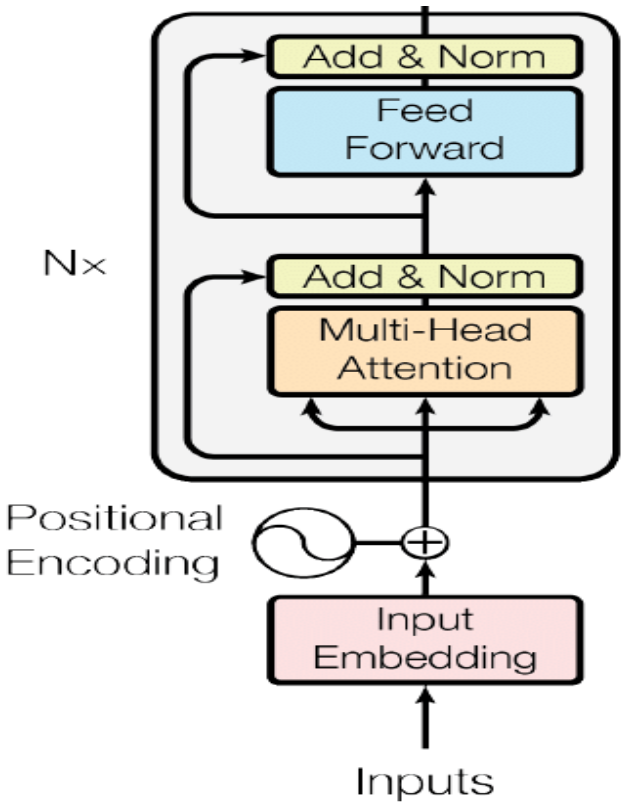


Fig 10. Transformer Encoder ref Vaswani[9]

Bert, therefore, uses Masked LM (MLM) and Next sentence prediction (NSP) mechanisms to achieve it's training objectives which are based on the above principle. A brief description is given below.

Masked LM - In the research carried out by Devlin[16] About 15% of the words in a sequence of text are replaced with Mask token before feeding to Bert. The model attempts to predict the value of masked words based on the non-masked word's context in the sequence. The output word prediction implements a classification layer on top of the encoder output. The output vectors are multiplied by the embedding matrix to transform them into the vocabulary dimension. Then softmax function is applied to calculate the probability of each word in the vocabulary.

The Bert loss function considers only the prediction of masked values, ignoring the non-masked ones resulting in slower convergence than directional models.

Fig 11 shows an ablation study to evaluate the results of various masking strategies with accuracy achieved in the y-axis and epochs (training steps)in the x-axis.

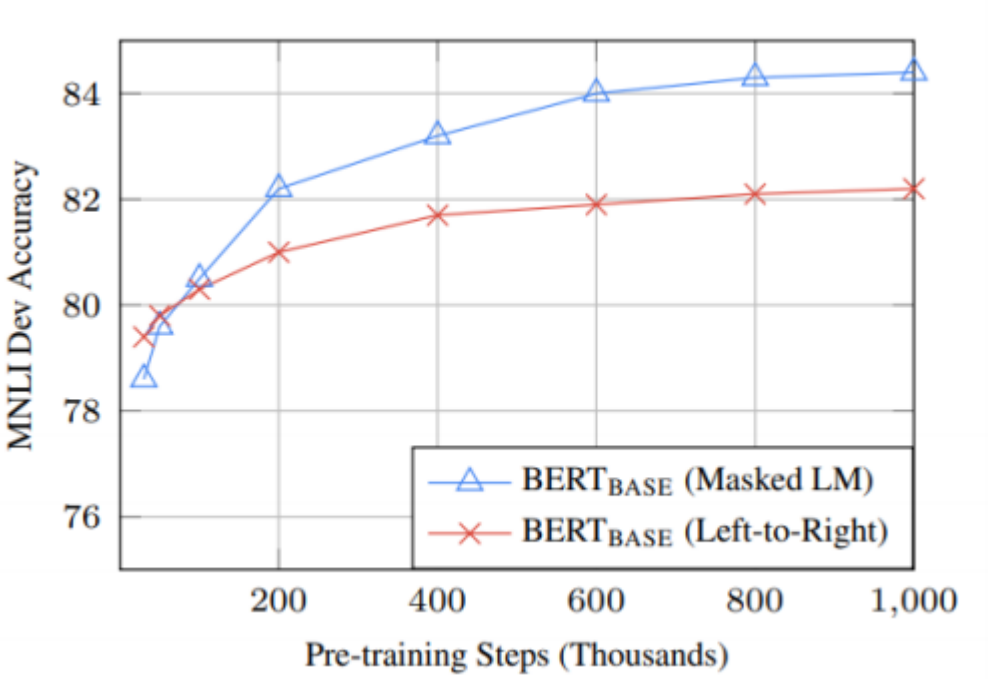


Fig 11. Ablation over a number of training steps Devlin[16]

NSP - BERT uses a special [SEP] token as a sentence delimiter. During training, the model uses two input sentences such that half the time the second sentence is preceded by the corresponding first sentence in the corpus, and half the time it is a random sentence. Bert is required to predict whether the next sentence is random or not, under the assumption that random sentences are disconnected from the first input. To do so, a complete input sequence is passed through the Transformer encoder, the output of the [CLS] token is transformed into a 2×1 shaped vector using a simple classification layer, and the Is Next-Label is assigned using softmax. Bert Layers are given in Fig 12

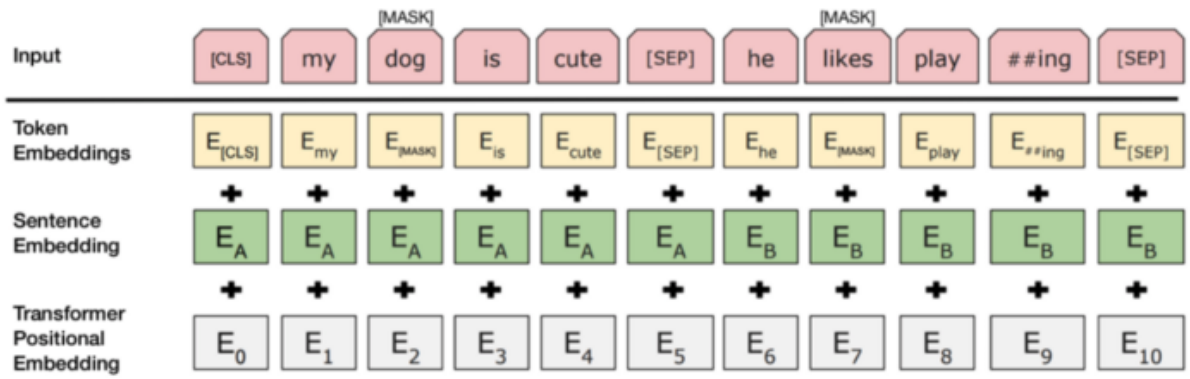


Fig 12. BERT with modifications, Devlin[16]

When training the BERT model, Masked LM and Next Sentence Prediction are trained together, with the goal of minimizing the combined loss function of the two strategies. We would use the pre-trained Bert model (published by google AI researchers) with its input and hidden layers as one of our main components in Cap-TE architecture.

Capsule-net

The capsule networks were introduced 1st by Hinton[18] to overcome the shortcomings of the convolution neural network which encounters information loss regarding the structural relationship of data due to its pooling operations. The caps-net comprises of locally invariant units called capsules that try to solve the problem by learning the structural and directional properties of the visual entities and encoding them into vectors. The Capsule net has 2 levels of capsules, the primary or lower level, and the digits or higher-level capsules. It employs a dynamic routing mechanism Sara[20] between upper and lower level capsules to help them learn the features in data. The dynamic routing is a repetitive dispatching mechanism through a coupling coefficient utilized to measure the similarity of vector representations of entities in upper and lower capsules. The Capsule network with components is described below.

Capsules - Capsules are ***vectors*** (with both magnitude and direction) that represent the ***features*** of an object and it's a ***likelihood***. The features can be orientation, position, size, deformation, texture, etc. and any such structural component. The main difference between a neuron and a capsule is that while neural networks can learn to have one neuron to learn the feature of an object say "eye" and all of its variation, the capsule net deploys different capsules to learn about different eye variations like angle, size, etc. more specifically.

The generic architecture of the capsule net consists of 2 layers of capsules the primary which learns very basic features from an entity (shape & size of the eye, gaze direction) and the digit capsule which learns higher features (distinct features of the eye of human and other animals). Based on the type of architectures the number of capsules in the Primary and Digit layer can vary.

Iterative dynamic routing - the capsule network updated the weight of coupling coefficients through an iterative routing process and determined the degree to which lower capsules were directed to upper capsules. The coupling coefficient is determined by the degree of similarity between the standard-upper and prediction-upper capsules. The working principles of dynamic routing are explained in the paper of Sara[20] in detail. The algorithm is depicted in Fig 13.

Procedure 1 Routing algorithm.

procedure ROUTING($\hat{\mathbf{u}}_{j|i}, r, l$)
  for all capsule i in layer l and capsule j in layer $(l+1)$: $b_{ij} \leftarrow 0$.
  for r iterations **do**
    for all capsule i in layer l: $\mathbf{c}_i \leftarrow \texttt{softmax}(\mathbf{b}_i)$ ▷ `softmax` computes Eq. 3
    for all capsule j in layer $(l+1)$: $\mathbf{s}_j \leftarrow \sum_i c_{ij}\hat{\mathbf{u}}_{j|i}$
    for all capsule j in layer $(l+1)$: $\mathbf{v}_j \leftarrow \texttt{squash}(\mathbf{s}_j)$ ▷ `squash` computes Eq. 1
    for all capsule i in layer l and capsule j in layer $(l+1)$: $b_{ij} \leftarrow b_{ij} + \hat{\mathbf{u}}_{j|i}.\mathbf{v}_j$
  return $\mathbf{v}_j$

$$v_j = \frac{\|s_j\|^2}{1+\|s_j\|^2}\frac{s_j}{\|s_j\|}$$

Fig 13. Dynamic routing with squash function between capsules

The above diagram ref Sara[20] explains the working algorithm of iterative dynamic routing between capsules.[layer l+1,l are primary and digit capsule layers, b is biased and u is the weights of the input. The code within iteration is

the pass for dynamic routing and v is the squash function]The detailed working principles are explained in the paper of Sara[20]

In the current study, we aim to implement this variant of capsule net as a capsule layer to be one of the main components in our architecture.

Cap-Te model for Stock prediction from news articles

The model used in the current study is a conjoined implementation of the above 2 advanced deep learning architectures. The idea behind the formulation of Cap-Te network is that while Transformers are capable of interpreting the contextual and sequential relationship between text inputs, the capsule net can help to capture structural components of the corpus and together produce an impactful representation of the features for regression analysis of stock market prediction. A detailed working principle of the model is depicted below stepwise.

1. The text input (X_i)is tokenized using a pre-trained Bert tokenizer and word embeddings (Y_i)are produced. Let the tokenization process be considered as a function f_tokenize ()

$$Yi = f\_tokenize(Xi)$$

2.The word embeddings are fed to pre-trained Bert model with output hidden states and output attentions made trainable for transfer learning. The output from last hidden state of the model (H_i) is passed to a capsule layer. The Bert processing explained in previous sections is represented f_bert()

$$Hi = f\_bert(Yi)$$

4. The capsule layer is a variant of the Capsule network which consists of a set of capsules trainable to interpret high-level features. The capsules generate a vector each of which represents the probability of the existence of a feature. A prediction vector ($U_{j|i}$) is calculated by multiplying the output from the Bert layer with a weight matrix (W_{ij})

$$Uj|i = WijHi$$

5. The total input to the capsule layer (S_i) is taken to be a weighted sum of all the prediction vectors and the coupling coefficient. The coupling coefficient (C_{ij}) is generated through iterative dynamic routing between capsules.

$$Sj = \sum CijUj|i$$

5. Finally, A non – linear squash function is used to scale the vectors such that magnitude is mapped between 0 and 1. The squash function (V_j)is represented as below.

$$Pj = ||Sj||2 / (1 + ||Sj||2)$$

$$Qj = Sj / ||Sj||$$

$$Vj = Pj . Qj$$

6. The flattened output from the capsule layer (say S_f) is passed to the dense layer with linear activation (W_{dense}) to generate continuous output.

$$Z = Wdense(Sf)$$

3.6 Ensemble of Stock Exchange and Text-based Models

Since the historical data is on daily basis for the past 12 years, selected days would be chosen and kept aside as training, test, and ensemble-test data from the augmented dataset of both Reddit news and DJIA stock parameters. The data chosen are in chronological order as Training data being earliest and of largest volume (80%), the test and ensemble test data consists of 10% each of the dataset and are in chronological order as well. The Text-based models are trained on the Reddit train dataset and that based on STI are trained on the DJIA train dataset. Next, the proposed models out of these groups are chosen. Let's assume model M_t is a Reddit news trained model and model M_n is DJIA feature trained model. For a given date say M_t gives a prediction x_t and M_n gives another prediction say x_n . These predictions are registered on all test data points which also have labels against them let it be y. As a result of the above operation, we would generate a new dataset with x_n, x_t as predictor features, and y as the response. We would figure out a linear model (L_m) which gives minimal loss fit on the above test set. Let the coefficients of the model be a_1 and a_2 so the model will find an equation with

$$y = a1 * xn + a2 * xt .$$

Here we use a1 and a2 predicted from the above result aggregator model as our ensemble coefficient. In other words, our new model will calculate prediction P() on new datapoint as the final prediction

$$Y = P(Mn) * a1 + P(Mt) * a2 .$$

Finally, the ensemble outcome would be tested on ensemble-test data with trained model weights and the ensemble coefficients obtained from the linear model would be considered as the weighted average of each of the model's predictions.

3.7 Model Evaluation

For evaluating the models, we have selected Root Mean Square Error (RMSE) and (Mean absolute percentage error) MAPE as our principal model evaluation techniques. This is due to the fact that the outcome of our models is of a continuous value (stock price change over a day) and hence we can treat it as a regression problem. The losses are calculated, and the training process is halted when a minimum is reached (even if it means before the specified number of epochs executed). This is because we implemented a step learning rate scheduler with a dynamic learning rate that decays with epochs. These models are then validated on test data which were previously segregated, and metrics calculated are based on test data only.

We will choose the model and incorporate it into our ensemble architecture based on our observation of best values from both the criteria.

3.8 Tools Used

Tools and libraries used in the current work are mentioned in Table 4

System	Environment	Libraries
CPU: I7-9750H	Python 3.7	Pytorch 1.5
RAM: 16GB	Anaconda 4.8.2	NLTK 3.4.5
OS: Windows 10	CudaToolkit 10.1	Tensorflow 1.14
GPU : NVIDIA RTX 2060	CUDNN 7.6.4	Keras 2.1.2

Table 4 : Tools to be used for experiments

3.9 Summary

In this chapter, the methodologies for developing a robust AI-based stock market prediction model from Dow Jones Industrial exchange and Reddit news are discussed.

At first, we analyze techniques related to technical Indicators based on Stock prediction. We hint to carry out time series-based EDA on DJIA data and derive technical indicators based on the theories adapted in literature review and observations from EDA. Next, we briefly analyze LSTM as our baseline model and dive deep into tabnet architecture which is our proposed model for numerical analysis.

Then we would move on to examine the exploratory analysis for Reddit news-based articles and decides on the data cleaning operations to be performed. Post that we examined Bert and Glove based embedding for text data followed by an outline of CNN-LSTM architecture which is our baseline model and a more critical analysis of capsule Transformer based network which is our proposed model.

Finally, we proceed to describe the ensemble technique for our proposed models (text and numerical based) to arrive at the outcome of our study.

CHAPTER 4

ANALYSIS

This chapter deals with the implementation of the methodologies practiced in the research. It is divided into sections like data augmentation, data preparation and feature extraction, model analysis, and evaluation, and finally ensemble of STI-driven and news-driven models.

4.1 Data Augmentation

The main datasets are DowJones.csv and News.csv which are downloaded from Kaggle https://www.kaggle.com/aaron7sun/stocknews.The raw datasets are in the following format depicted in Table 5 and Table 6. The dates range from 8-Aug-2008 to 1st-Jul-2016.

DowJones.csv

Date	Open	High	Low	Close	Volume	Adj Close
01-07-2016	17924.2402	18002.3808	17916.9101	17949.369141	82160000	17949.369141

Table 5. Dow Jones dataset

[Date – date of the registered stock index, Open – the opening value of the stock for the day, High – The highest value of the stock for the day, Low - The lowest value of the stock for the day, Close – Closing value of the stock for the day, Volume – Number of shares traded for the day, Adj Close – stock value post dividend]

News.csv

Date	News
01-07-2016	A 117-year-old woman in Mexico City finally received her birth certificate, and died a few hours later. Trinidad Alvarez Lira had waited years for proof that she had been born in 1898.

Table 6. Reddit News dataset

[Date – date of the registered Reddit news articles, News – Reddit world news for that date]

After Collecting the original dataset from the Kaggle link, we download the DowJones data (in the same format) from yahoo finance manually for the date 2nd Jul 2016 to 1st Sep 2020.

For collecting Reddit News data we used the Push Shift API of Python to download data from the Reddit website. we applied Thread pool executor with 50 workers to download data for each year starting from 2016 (post 2nd July) to 2020 (till 1st September). We used the subreddit parameter as *'world news'* and filter as [*'score', 'title'*] with limits up to 35 news per day sorted by score. (The reason being we needed at least 400 words of news data per day for modeling excluding all non-Ascii character containing words and non-English words)

We merged DowJones and Reddit news data and sort it in date wise ascending order to for 2 new datasets dowjones_modified.csv and news_modified.csv.

4.2 Exploratory Data Analysis and Feature Derivation

For both DowJones and Reddit News, we needed to scrutinize the data separately to formulate parameters for Stock Technical Indicators as well as implement data cleaning techniques for preparing inputs to models. The implementation details are given in the following sections.

4.2.1 EDA and STI derivation from DowJones data

From the dowjones_modified.csv, we plot the opening price w.r.t Date as well as the rolling mean of the opening price with a window of 7 days to understand the trends captured. We can observe from the graph below that orange lines representing rolling windows are less noisy than the blue line meaning we can capture the trend more effectively using a rolling window. The smoothness increases with increasing rolling window length as noticed in Fig 14 below

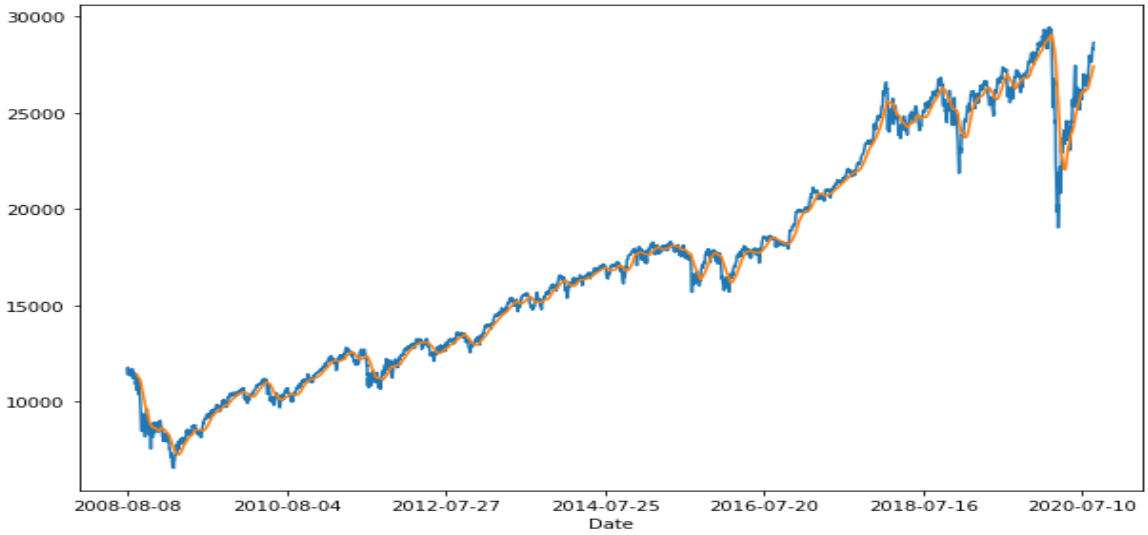


Fig 14 opening vs rolling mean plot [rolling window7]

The expanding function along with the aggregate function means works as a rolling function in that it computes the average mean of all the previous stock closing values at specific time steps. With the expanding function, we will be able to understand whether the trend of the stock prices is increasing, decreasing, or stationary as been manifested in Fig 15

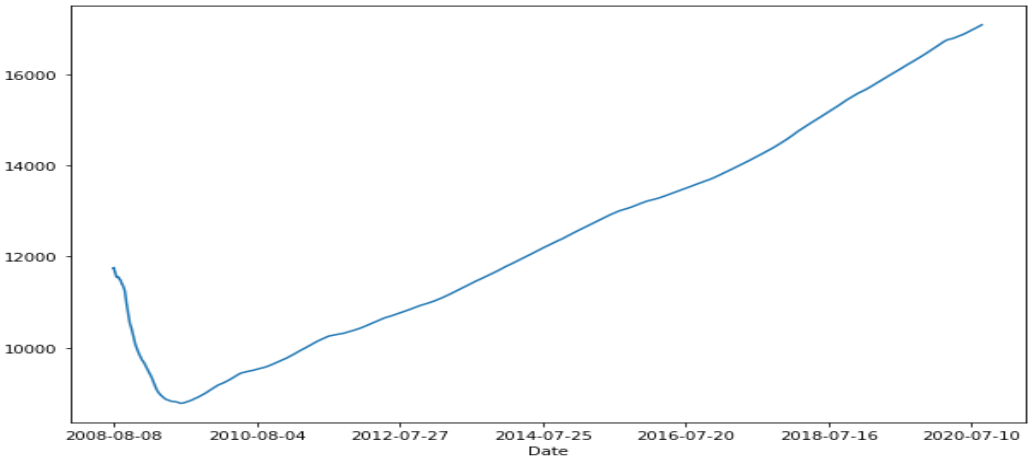


Fig 15 Expanding mean of the closing prices

From the above figure, we can notice that the stock trend went decreasing from Aug-2008 to reach a slide on Aug 2009 from whereupon it gradually follow an incremental trend till the end of 2020

ETS decomposition models help us to visualize time series data concerning the components like Trend, Seasonality, Cyclical patterns, irregular patterns. We

have applied the method here to study the above properties concerning the opening price of Stock. Fig 16 refers to the related graphs.

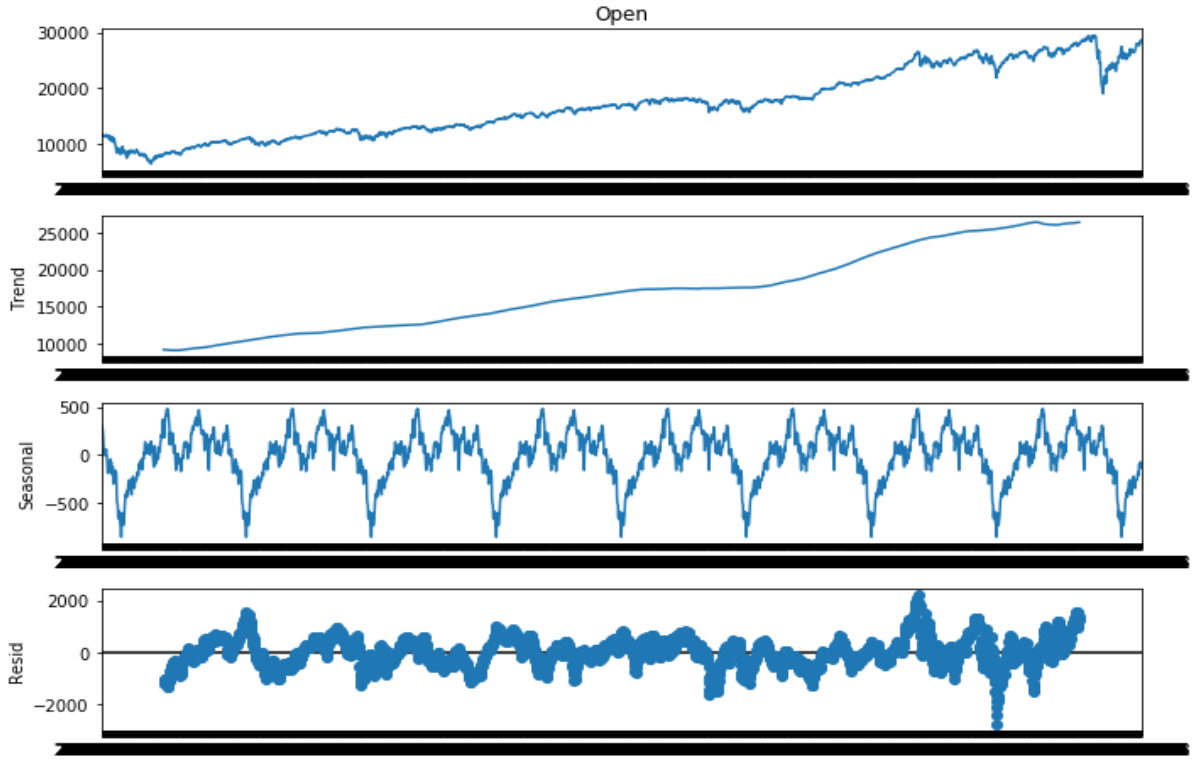


Fig 16 ETS decomposition of Stock Opening price

From the above figure, we can see the important components of time-series data such as trend, seasonality, residuals (error difference), and the observed data. ETS decomposition helps us to get more insights into stock patterns and trends. It also provides information on whether the data is seasonal or not. We infer that Dow jones industrial exchange follows a seasonal pattern and that there are both uptrends and downtrends in the pattern exhibited.

To find out whether the dataset is also stationary or not, we would conduct an Augmented Dickey-Fuller unit root test. In statistics and econometrics, an augmented Dickey-Fuller test (ADF) tests the null hypothesis that a unit root is present in a time series sample. The alternative hypothesis is different depending on which version of the test is used but is usually used to determine stationarity or trend-stationarity. We are trying to conclude whether to accept the Null Hypothesis H0 (that the time series has a unit root, indicating it is non-stationary) or reject H0 and go with the Alternative Hypothesis (that the time series has no unit root and is stationary). We end up deciding this based on the p-value return.

A small p-value (typically ≤ 0.05) indicates strong evidence against the null hypothesis, so you reject the null hypothesis. A large p-value (> 0.05) indicates weak evidence against the null hypothesis, so we fail to reject the null hypothesis. We observe that the p-value is more than 0.05 for ADF on opening price hence we fail to reject the null hypothesis. Thereafter we calculate the daily differential opening price with and conduct a second ADF test on that data. Here we see the p-value is reduced below 0.05 and hence we can safely reject the null hypothesis which states that the time series value has a unit root, indicating it is non-stationary.

We also plotted the differential Intra-day price to find out its distribution to be somewhat normal as shown in Fig 17

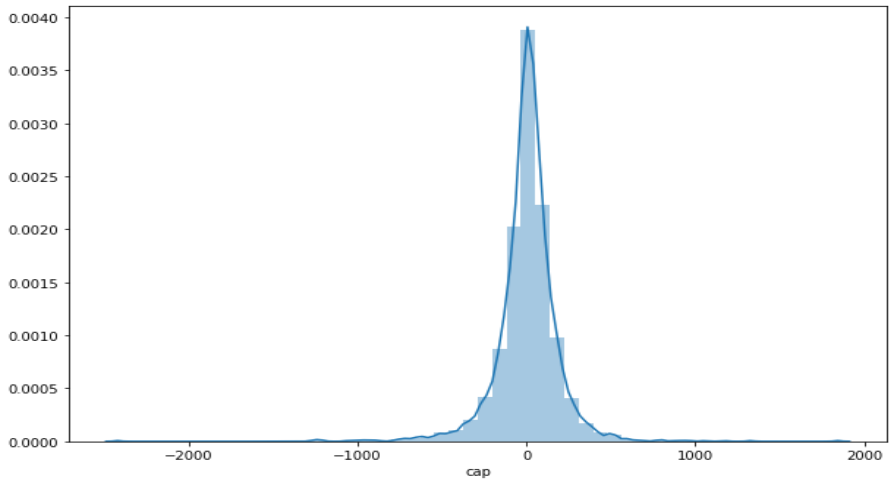


Fig 17 Distribution of differential opening price

From the EDA done above, we can conclude the following.

1. The Time series-based stock data is seasonal
2. It has both uptrends and downtrends
3. The stock data w.r.t to daily opening value is non-stationery but w.r.t intra-day differential opening value it's stationary
4. The distribution of change in price is somewhat normal

From the above observation we can presume that the differential Intraday opening value can be formulated as the response variable. It makes sense to derive moving average/statistics based technical indicators from the above data due to properties 1,2 and 3.

Hence we derive our Stock Technical Indicators based on the hypothesis made in researches Kim[38],Vargas[2],Onchareon[57] after analyzing the assumptions

from a thoroughly conducted Exploratory Data Analysis on original Stock parameters of Dow Jones data. The details of these STI's are already mentioned in Table1 under Literature review section. We form the Stock_indicators.csv from Dow-Jones modified.csv as shown in Table 7.

Date	stochaistic_ k_percent	stochaistic_D	Moment um	Rate_of_ change	William_ R_percent	AD_oscillator	Disparity	cap
08-08-2008	93.106	13.3008	0	8649.194	0.0689	0.0689	100	297.5

Table 7. Stock Technical Indicators

4.2.2 EDA and Data cleaning for Reddit News

To understand the distribution of words and their characteristics, we accumulate the entire News data from news_modified.csv and plot word Cloud as well as frequency distribution shown in fig 18 and 19.

Fig 18 word cloud for raw Reddit news data.

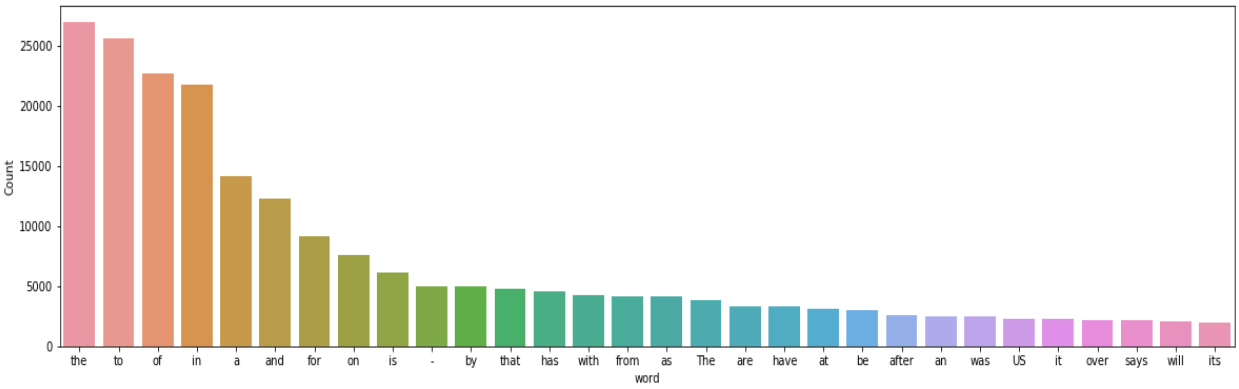


Fig 19. Word Frequency distribution for raw Reddit news data

From the graph and the word cloud, we could easily understand that we need to carry out extensive data cleaning operations to get meaningful representations. As the article ‘the’ appears to be the most frequent word in the frequency distribution graph and the word cloud contains lots of words that are either abbreviations or non-ascii chars or articles and prepositions.

Hence, we adopted the following methods for Data cleaning.

- We choose words that contain only ASCII characters to filter out non-English words and special characters.
- We removed punctuations from the sentences
- We did decontractions like converting phonetic short forms to full forms example: ‘ve’ to ‘have’
- We also did stopword removal according to the NLTK manual

After performing these operations, we again formed word cloud and frequency distribution graph and the results depicted in fig 20 and 21 shows better visibility

Fig 20 word cloud for cleaned Reddit news data.

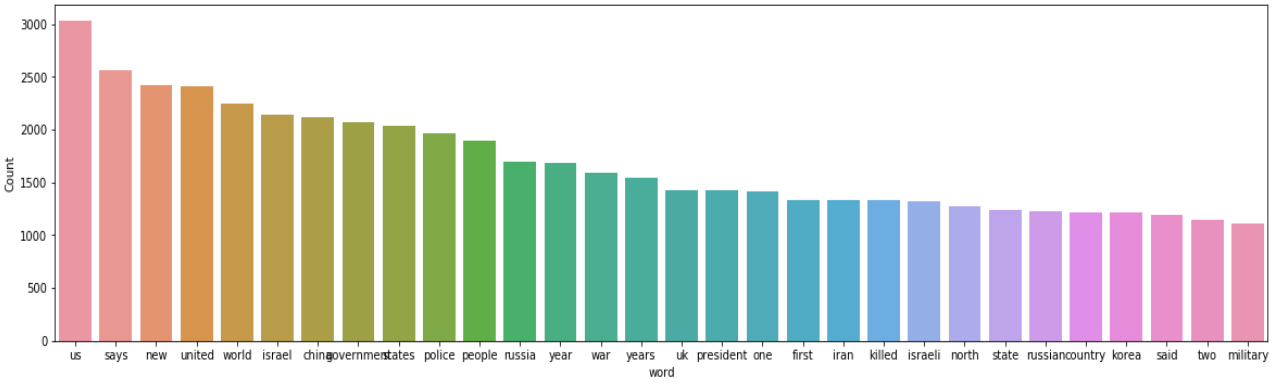


Fig 21 word Frequency distribution for cleaned Reddit news data

We also performed topic modeling using LDA to understand the overall distribution and categorization of news articles. However, the results obtained were merely exploratory and non-inferential as revealed in Fig 22

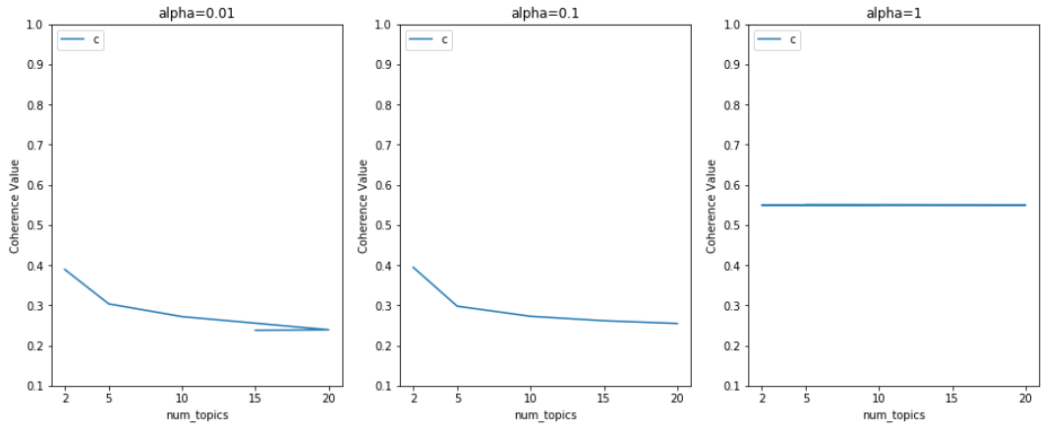


Fig 22 coherence plot for topic modeling

The number of optimum topics obtained as a result of the coherence plot was 5 with an alpha value of 0.01. Also, a distribution of most prominent words per topic is shown below in Fig 23

```
[(0,
  '0.010*"israel" + 0.008*"gaza" + 0.007*"israeli" + 0.006*"us" + '
  '0.004*"united" + 0.004*"nothe" + 0.004*"war" + 0.003*"one" + 0.003*"people" '
  '+ 0.003*"government"'),
 (1,
  '0.008*"" + 0.004*"israel" + 0.004*"us" + 0.004*"world" + 0.003*"year" + '
  '0.003*"people" + 0.003*"police" + 0.003*"new" + 0.003*"united" + '
  '0.003*"war"'),
 (2,
  '0.004*"says" + 0.004*"china" + 0.004*"us" + 0.004*"new" + 0.003*"people" + '
  '0.003*"government" + 0.003*"police" + 0.003*"world" + 0.003*"united" + '
  '0.003*"israel"'),
 (3,
  '0.005*"says" + 0.005*"us" + 0.005*"new" + 0.005*"united" + 0.004*"russia" + '
  '0.004*"states" + 0.004*"world" + 0.004*"china" + 0.003*"government" + '
  '0.003*"russian"'),
 (4,
  '0.007*"us" + 0.006*"wikileaks" + 0.004*"government" + 0.004*"police" + '
  '0.004*"assange" + 0.003*"war" + 0.003*"egypt" + 0.003*"united" + '
  '0.003*"states" + 0.003*"china"')]
```

Fig 23 Topic distribution

We can understand that most topics are based on international issues based on finance and politics. However, we do not conclude anything from topic modeling, it was done merely for exploratory purpose.

4.3 Dataset creation for Modeling and Ensemble

We merge the datasets stock_indicators.csv and news_modified.csv (cleaned) on Date and sort it in ascending order of Date. Post that we carefully divide the data such that the first 90% is chosen for Model development and the rest is chosen for the ensemble. We club the headlines for a day such that each headline does not cross the 20-word mark and the total length of all headlines collected for a single day is within 400 words. Thus the converging all the rows from news.csv into one single row containing cleaned headline corpus for the day.

So, the model.csv contains merged news and stock indicator data from 8-8-2008 to 17-06-2019. The columns are shown in Table 8

Columns	Description
Date	date of forecasting
News	Reddit news collected for the given date with 400 max length
cap	Intra-day change of price for DIJA
stochaistic_k_percent	Derived STI parameter
stochaistic_D	Derived STI parameter
Momentum	Derived STI parameter
rate_of_change	Derived STI parameter
William_R_percent	Derived STI parameter
AD_oscillator	Derived STI parameter
Disparity	Derived STI parameter

Table 8 columns for model.csv and ensemble.csv

The ensemble.csv contains merged news and stock indicator data from 8-8-2008 to 17-06-2019. The distribution is made in a chronological way as the model will be used to predict results on an ensemble set and the resultant prediction would be used to calculate ensemble coefficients for each of the news-based and STI based models. The missing data in news indicates news article score below a certain threshold and hence will be adjusted.

4.4 Model Analysis and Evaluation Techniques

We offer LSTM as a baseline model and Tabnet as a proposed model for STI based Stock market prediction. Similarly, CNN-LSTM a baseline, and Bert-Capsule as a proposed model for Reddit news-based Stock market prediction. We would now deep dive into each of the Model architecture and validation strategies.

4.4.1 Baseline Models

LSTM

For STI-based predictions we train an LSTM based model as our baseline. from model.csv we drop the *'news'* and *'Date'* column and use the STI as predictors and cap (change of price as response variable). we used a Standard scaler to normalize the change of price variable. We train split the data into 0.85 for training and 0.15 for testing. we used *'Adam'* optimizer with default learning rate accounting for '*mean_squared_error*' loss.

The model summary is given below in Fig 24

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm_1 (LSTM)                (None, 1, 100)            43600
_________________________________________________________________
dropout_1 (Dropout)          (None, 1, 100)            0
_________________________________________________________________
lstm_2 (LSTM)                (None, 100)               80400
_________________________________________________________________
dropout_2 (Dropout)          (None, 100)               0
_________________________________________________________________
dense_1 (Dense)              (None, 1)                 101
=================================================================
Total params: 124,101
Trainable params: 124,101
Non-trainable params: 0
```

Fig 24 LSTM Model summary

We tried varying LSTM hidden layers 1 and 2 sizes to calculate optimum RMSE and MAE keeping fixed the batch size as 32 and epochs as 25. Optimum hyperparameters are given below in table 9 and the test vs prediction results are depicted in Fig 25.

epochs	batch size	hidden layer 1 units	hidden layer 2 units	rmse	mae
25	64	25	25	0.623	0.278

Table 9 optimum hyperparameters for the LSTM model

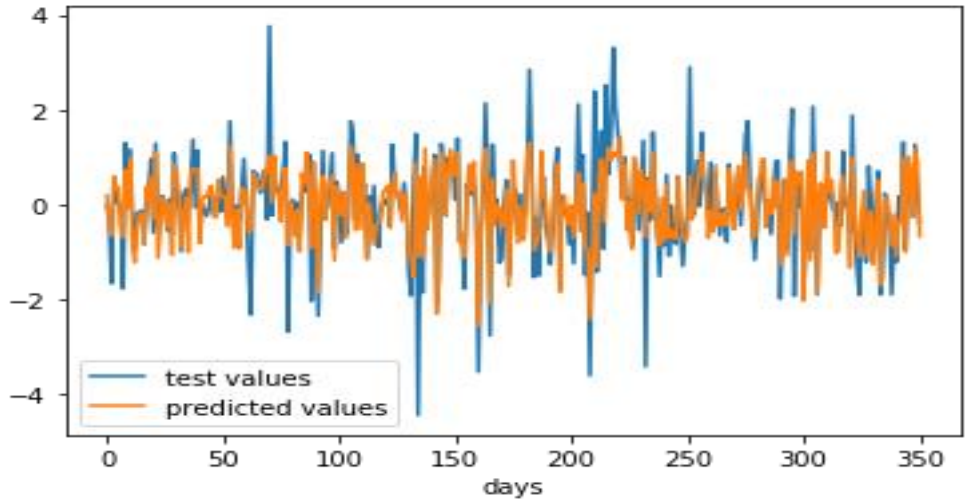


Fig 25 test vs Prediction results of LSTM model

The rmse and mae obtained were satisfactory but further rounds of testing did not see significant improvement. Hence, we registered the above parameters. The graph above indicates that the LSTM model can make predictions that are close to testing results except for a few spikes.

CNN-LSTM

For predictions from Reddit news, we put forward CNN-LSTM architecture as our baseline. We select only the '*News'* as a predictor and '*cap'* or change of price as response and drop all STI based predictors from model.csv. we used MinMax scaler to normalize the change of price variable. We apply *'<PAD>'* for the headlines that are under a 400-word mark and create embedding using the glove model. Next, we create input sequences from these embeddings and split train test data into a 0.85 to 0.15 ratio. The training set is further split to provide

0.15 fraction of data for validation. We train the model(shown in Fig 26) for 10 epochs with batch size 128. we used '*Adam'* optimizer starting with initial learning rate 0.001 following '*ReduceLROnPlateau'* strategy and early stopping for training and validation loss convergence accounting for '*mean_squared_error*' loss.

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_1 (Embedding)      (None, 400, 300)          10922100
_________________________________________________________________
dropout_1 (Dropout)          (None, 400, 300)          0
_________________________________________________________________
conv1d_1 (Conv1D)            (None, 400, 16)           24016
_________________________________________________________________
dropout_2 (Dropout)          (None, 400, 16)           0
_________________________________________________________________
conv1d_2 (Conv1D)            (None, 400, 16)           1296
_________________________________________________________________
dropout_3 (Dropout)          (None, 400, 16)           0
_________________________________________________________________
lstm_1 (LSTM)                (None, 64)                20736
_________________________________________________________________
dense_1 (Dense)              (None, 64)                4160
_________________________________________________________________
dropout_4 (Dropout)          (None, 64)                0
_________________________________________________________________
output (Dense)               (None, 1)                 65
=================================================================
Total params: 10,972,373
Trainable params: 10,972,373
Non-trainable params: 0
_________________________________________________________________
None
```

Fig 26 CNN-LSTM model summary

We tuned for CNN kernel size and LSTM hidden dimensions to arrive at the optimum model for RMSE and MAE. For the baseline CNN-LSTM, we tried adjusting hidden units of LSTM layers and CNN-Kernel size keeping fixed the batch and epoch. The optimum hyperparameters are mentioned in the below table 10. However further attempts to manipulate the parameters did not yield significantly different results from already conducted experiments. The test vs prediction graph is shown in Fig 27.

epochs	Batch size	CNN -Kernel size	RNN Hidden Dim	rmse	mae
10	128	16	64	0.45	0.448

Table 10 Optimum hyperparameters for CNN-LSTM

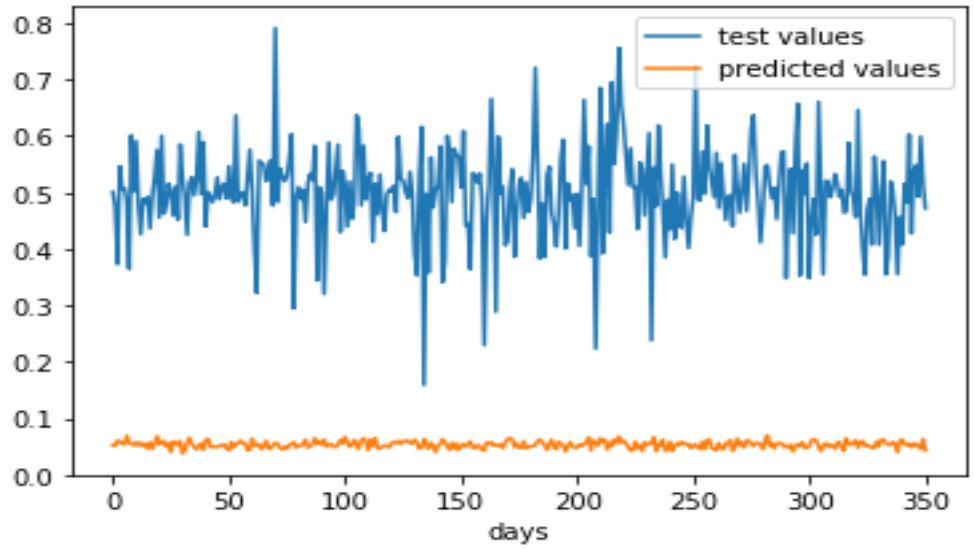


Fig 27 Test vs Prediction results of Conv-LSTM model

The model ignores most slides and surges, though it predicts somewhat close price changes per day in case of most of the day's data, but not by a very satisfactory margin as has been demonstrated in the graph.

4.4.2 Proposed Models

Tabnet

For STI based predictions we train the Tabnet as our proposed model. From model.csv we drop the *'news'* and *'Date'* column and use the STI as predictors and cap (change of price as response variable). we used a Standard scaler to normalize the change of price variable. We divide the data into train, valid, and test in the ratio 0.8:0.1:0.1. We choose the Tabnet Regressor model and train it varying the parameter grid variables given below in Table 11.

Parameters	**Definition**	**Range**
n_d	Width of the decision prediction layer	[24, 32]
n_a	Width of the attention embedding for each mask	[24]

n_steps	Number of steps in the architecture	[3, 4, 5]
gamma	This is the coefficient for feature reusage in the masks	[1,1.5,2]
lambda_sparse	This is the extra sparsity loss coefficient	[1e-2, 1e-3, 1e-4]
momentum	Momentum for batch normalization	[0.3, 0.4, 0.5]
n_shared	Number of shared Gated Linear Units at each step	[2]
n_independent	Number of independent Gated Linear Units layers at each step	[2]
clip_value	Float value for clipping gradient	[2]

Table 11 Tabnet Grid parameters

We keep the parameters like num_epochs as 25, patience as 7, batch_size as 128, and virtual_batch_size as 32 fixed with respective values. We calculate RMSE and MAE values for each of the combinations of parameter grid and finally choose the optimum one. For Tabnet regressor model from PyTorch which exhibits sklearn properties, we ran training cycles with combinations of parameter grid keeping epoch, batch size, patience, and virtual batch size as stationery parameters. While conducting iterations for parameter search, we fixed values for gamma and clip each with defined ranges (for each iteration) and lay down observations varying other parameters. Hyperparameters for best results are given below in table 12 and the test vs prediction plot is given in Fig 28.

lambda_sparse	momentum	n_a	n_d	n_independent	n_shared	n_steps	Clip-value	gamma
0.0001	0.3	24	24	2	2	4	2	1

Table 12 Optimum Tabnet hyper parameters

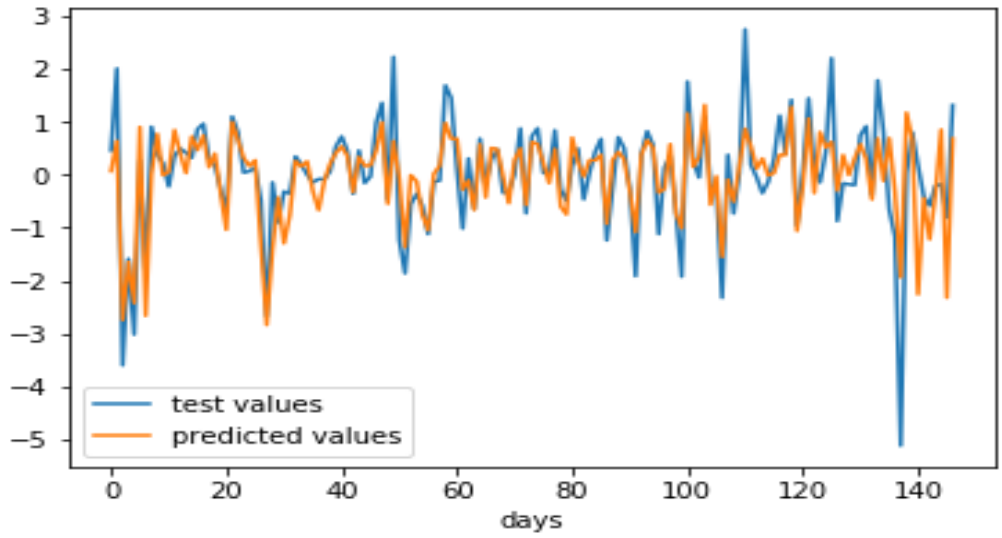


Fig 28 Test vs Prediction results of Tabnet model

The Tabnet gives slightly better results than LSTM as it is able to capture finer spikes . Hence we choose it for ensemble. The results obtained from the LSTM and Tabnet are very close if optimum values for RMSE and MAE are considered.

Bert-Capsule

We selected Bert-Capsule as the proposed model for Reddit News based stock market predictions. Hence, we select only the '*News'* and '*cap'* as predictor and response variables and drop the remaining columns from model.csv. We used MinMax Scaler for the normalization of the *'cap'* variable. We prefix each headline with '*[CLS]'* and suffix with '*[SEP]'* and used '*bert-base-uncased'* tokenizer to create input sequences and attention masks. We divide the input sequences in the ratio of 0.9 and 0.1 for train and validation. We used pre-trained *'bert-base-uncased'* for the transformer layer and added a capsule layer to the last hidden states of the transformer. The model architecture is given below in Table 13

Component	Layer	Architecture	Units
Bert	BertEmbeddings	(word_embeddings): Embedding(30522, 768, padding_idx = 0)	1

		(position_embeddings): Embedding(512, 768) (token_type_embeddings): Embedding(2, 768) (LayerNorm): LayerNorm((768,), eps = 1e-12, elementwise_affine = True) (dropout): Dropout (p = 0.1, inplace = False)	
	Bert Encoder	BertLayer((attention): BertAttention((self): BertSelfAttention((query)Linear(in_features=768, out_features=768, bias=True) (key):Linear(in_features=768, out_features=768, bias=True) (value):Linear(in_features=768, out_features=768, bias=True) (dropout): Dropout(p=0.1, inplace=False)) (output): BertSelfOutput((dense): Linear(in_features=768, out_features=768, bias=True) (LayerNorm): LayerNorm((768,), eps=1e-12, elementwise_affine=True) (dropout): Dropout(p=0.1, inplace=False))) (intermediate): BertIntermediate((dense): Linear(in_features=768, out_features=3072, bias=True)) (output): BertOutput((dense): Linear(in_features=3072, out_features=768, bias=True)	12

		(LayerNorm): LayerNorm((768,), eps=1e-12, elementwise_affine=True) (dropout): Dropout(p=0.1, inplace=False)))	
	BertPooler	((dense):Linear(in_features=768, out_features=768, bias=True) (activation): Tanh())	1
Capsule	Caps_Layer	()	1
Dense	Dense_Layer	Sequential((0): Dropout(p = 0.25, inplace = True) (1): Linear(in_features = 640, out_features = 1, bias = True))	1

Table 13 Bert Capsule Model architecture

[The capsule Layer above only does dynamic routing operation; no modules are associated with it.]

We use Adam with weight Decay ('*AdamW*') optimizer with an initial learning rate of 0.001 for '*RMSE*' loss. We keep the batch size as 4 and epochs as 5 and do hyper-parameter on Capsule layer parameters like routing (iterations for routing), T_epsilon (routing activation parameter), num_capsule (primary capsule), dim_capsule (digit capsule) to get optimized RMSE and MAE.

For the Bert-Capsule model, we fixed the batch size and epochs and vary the parameters like Routing, T_epsilon, Num_capsule, Dim_capsule. The best set of hyperparameters obtained from the experiment are demonstrated in the below table 14 and test vs prediction results are shown in Fig 29

epoch	Batch size	Routing	T_epsilon	Num_capsule	Dim_capsule
5	4	3	1.00E-07	20	32

Table 14 Bert-Capsule optimum hyper-parameters

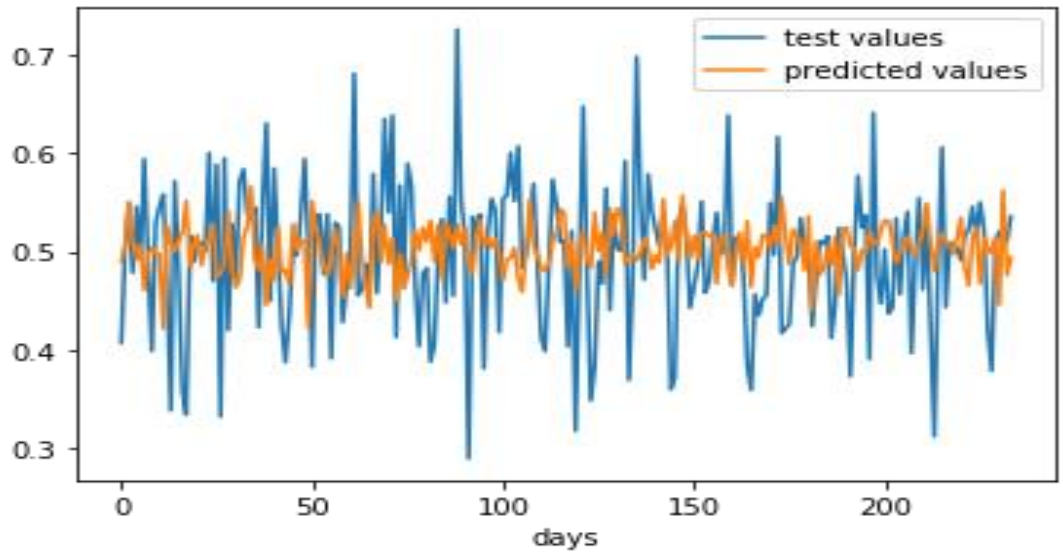


Fig 29 Test vs Prediction results of Bert-Capsule model

The Bert-Capsule model performs significantly better than the baseline in predicting the change of prices over a day. Further fine-tuning can lead to capturing the spikes even better.

4.5 Model Ensemble

The Bert-Capsule model and the Tabnet model are used to do predictions on ensemble linear features. The prediction results are somewhat closer to the actual change of price, though it fails to capture exact maxima's and minima for certain dates. The predictions are registered as Linear ensemble features and a linear model is built on top of the predicted values as predictors and change of price as the response.

For example, Let the coefficients of the model be a1 and a2 and predictions be xn (for bert -capsule) and xt(for tabnet). So the model will find an equation with $y = a1 * xn + a2 * xt$ + b. Here we use a1 and a2 are our ensemble coefficient and b is the intercept. In other words, our new model will calculate prediction P () on new datapoint as the final prediction $Y = P(Mn) * a1 + P(Mt) * a2$ + b. where Mn and Mt are bert-capsule and tabnet models respectively.

4.6 Summary

In this chapter, we discussed in detail about Data Augmentation for DowJones and Reddit news-based data files and Exploratory data analysis for both Reddit News and Stock Exchange parameters that lead us to feature derivation and data cleaning. Post that we put forward model architectures with implementation details for both baseline and proposed models like LSTM, CNN-LSTM, Tabnet, and Bert-Capsule vividly with model evaluation techniques. Finally, we detailed our ensemble process-based which introduces a linear combination of predictions from news-based and STI based models to formulate the final prediction.

All the information related to the practical experiments can be found in the following repository .

CHAPTER 5

RESULTS AND DISCUSSIONS

This chapter of the thesis aims to summarize and evaluate the results of all experiments conducted to build a deep Learning based stock market Prediction Model. Here the evaluation metrices of individual models as well as ensemble are discussed in details.

5.1 Model Evaluation and Results

We have chosen RMSE and MAE as the standard metrics for performance since the output is continuous. The following Table 15 shows the comparative analysis of the baseline and proposed models that yielded optimum values of rmse and mae with an experimentally chosen set of hyperparameters.

Model	RMSE	MAE
LSTM (BaseLine)	0.623	0.278
TABNET (Proposed)	0.564	0.290
CNN-LSTM (Baseline)	0.450	0.448
BERT-CAPSULE(Proposed)	0.079	0.037

Table 15 evaluation for baseline and proposed models

We observe that in the case of STI-based models there is not much difference in terms of performance between Tabnet and LSTM and the former only marginally outdoes the latter. However, since the Tabnet uses an attention-based mechanism for feature extraction if the columns or features are more it is safe to go with the Tabnet than LSTM which interprets sequential features.

Nevertheless, if the performance of the Conv-LSTM and Bert-Capsule is compared we see significant improvement for the proposed model. This can be attributed purely to the superiority of transformer-based Bert in capturing contextual information over LSTM as well as the ability of Capsule networks to infer deeper structural relation of textual information over CNN.

5.2 Model Ensemble Results

We noted down the best set of hyperparameters for proposed models like tabnet and bert-capsule and retrain them with the given set (model.csv) to come up with final models.

Next, we let the bert-capsule model figure out predictions for '*cap'* values in ensemble.csv for news-based data and Tabnet for STI's. We name these prediction columns as ebc_pred and tb_pred. We formulate *ensemble-linear-features.csv* (303 rows) with these data. The ensemble.csv is briefed in Fig 30

cap	ebc_pred	tb_pred
-189.779297	-905.1267126	-2203.041504
66.86914	-879.8002892	-1978.967407
-326.599609	-870.8874004	-1774.637573
180.86914	-827.1895955	-913.5997314
-773.929687	-847.5264043	-2861.465332
-77.029297	-886.8795034	-5383.361816
-205.25	-856.434631	167.5131378
27.730468	-862.6967348	-3888.755127
192.888672	-859.7879884	1296.44397
-36.509766	-907.9985075	-268.096344
472.429688	-862.9309795	173.9978638
-27.009766	-873.8451687	1636.493652

Fig 30 : Ensemble Linear Features

Post this data collection we developed a simple linear model using *ebc_pred* and *tb_pred* as predictors and *cap* as a response variable. We calculated the coefficients of this linear model to formulate our outcome.

After carrying out the exercises we obtain the value of a1 and a2 as -0.0669 and 0.0698 respectively with b as 28.5680

5.3 RealTime Prediction on Future Data

We downloaded Dowjones and Reddit news data from 2nd Sep2020 to 20th Sep 2020. We applied the chosen models with ensemble coefficients and predicted the change of price for 10 days (taking the intersection of Reddit and Dowjones available dates). On that final prediction, our ensembled models returned MAE of 361.27 and RMSE of 501.40

5.4 Summary

From the comparative analysis of baseline and proposed models, we choose Tabnet and Bert-capsule models for ensemble with the set of parameters that yielded optimum value for RMSE and MAE. The proposed models with optimum configurations perform better (although slightly) than the baselines. We register the best performing models and uses them for the ensemble. A linear combination of predictions of these models is finally presented as the outcome for Stock market prediction. This is again validated for real-time scenarios.

CHAPTER 6

CONCLUSIONS AND RECOMMENDATIONS

In this chapter, we discuss the proposed Stock Market Prediction methodology by comparing with existing approaches, justification of evaluated metrics for the STI based and News based proposed models and their ensemble, and future recommended work in the current domain.

6.1 Discussion and Conclusion

- In the current study, we use an augmented dataset for Dowjones and Reddit news to implement models each from News articles and Stock Technical Indicators. The original dataset is downloaded from Kaggle and augmented further by crawling data from the Reddit website and downloading from Yahoo Finance.

- We perform extensive Exploratory data analysis, mostly involving time series analysis and derived STI's from exchange data. At the same time, we implemented data cleaning operations like removal of stopwords, non-ASCII characters, and decontractions for news articles. With the processed and modified dataset available we merge them date-wise and divide them chronologically into model building and ensemble sets in the ration 9:1.

- We proceeded to build baseline models of LSTM and CNN-LSTM on STI based predictors and News articles with the change of price as a response variable. In the same way, we implemented our proposed models like Tabnet and Bert-Capsule on the STI and News-based predictors respectively keeping intra-day price change as a response. We used the glove model and Bert tokenizer for converting text to sequence inputs for CNN-LSTM and Bert-Capsule respectively.

- We measure each of the model performances based on RMSE and MAE since it's a regression problem and did hyper-parameter tuning of all the models accordingly for optimum values of evaluation metrics.

- The proposed models slightly outperformed baseline models in terms of evaluation metrics and hence we selected optimum hyper-parameters of proposed models (tabnet and Bert-Capsule) for the ensemble.

- Next, we used our final models trained from the modeling dataset to predict outcomes on the ensemble dataset. For a given period the tabnet model trained on STI was used to predict the change of price on the ensemble dataset and the Bert-Capsule model trained on news articles was used to predict the same.

- Hence, we formed a new dataset with 3 columns: actual price predicted price from tabnet, and predicted price from Bert-capsule. We build a linear model that used actual price as response and the remaining 2 fields as a predictor. The coefficients of the linear model are our ensemble coefficients.

- Our outcome is determined as a weighted sum of predictions from the models with respective ensemble coefficients.

6.2 Contribution to the knowledge

Stock market forecasting has been an ever-evolving domain although numerous researches have been conducted over a significant period contributing to knowledge growth both vertically and horizontally. The current study however takes deep learning-based approaches as a benchmark and tries to contribute to the knowledge enhancement by introducing a jointed transformer-capsule network for News articles-based prediction, scrutinizing the performance of the brand-new Tabnet model for STI driven predictions, and finally ensemble the two models by combining their predictions outputs linearly instead of putting up a jointed architecture. The idea of the transformer-capsule network transgressed from the jointed CNN-LSTM architecture relying on the superior performance of capsule-net over CNN and transformers over LSTM. The Tabnet which uses an attention-based mechanism to interpret tabular data is also proven to deliver better evaluation performance than predecessors like LSTM and time-series-based

techniques like ARIMA. Finally, the ensemble methodology of combining the outcome of both text-driven and numerical feature-based data is expected to outperform the jointed or concatenated deep learning architectures as it eliminates the scope of information loss (occurs during feature concatenation) and instead focusses on weighted individual contributions for predictions.

6.3 Future Recommendations

The current implementation can be considered as a pioneering effort in the realm of deep learning-based Stock Market Forecasting. However, treading on the path shown here further studies can be taken up on Long-transformer and XLnet based jointed architectures that perform well on the even longer document than considered here (max size 400 words). Likewise, several polynomial regression techniques can be studied to improve the ensemble process followed here with the help of a linear model. The most significant recommendation would be however to apply reinforcement Learning methodologies based on DL techniques applied here. To implement that we would need a more kaleidoscopic data like news and STI's collected over a minute per day rather than intra-day.

References

[1] Lee, H., Surdeanu, M., MacCartney, B. and Jurafsky, D., 2014, May. On the Importance of Text Analysis for Stock Price Prediction. In *LREC* (Vol. 2014, pp. 1170-1175).

[2] Vargas, M.R., dos Anjos, C.E., Bichara, G.L. and Evsukoff, A.G., 2018, July. Deep leaming for stock market prediction using technical indicators and financial news articles. In *2018 International Joint Conference on Neural Networks (IJCNN)* (pp. 1-8). IEEE.

[3] Kim, Y., 2014. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*.

[4] Le, Quoc, and Tomas Mikolov. "Distributed Representations of Sentences and Documents." *31st International Conference on Machine Learning, ICML 2014*, 2014.

[5] Tai, K.S., Socher, R. and Manning, C.D., 2015. Improved semantic representations from tree-structured long short-term memory networks. *arXiv preprint arXiv:1503.00075*.

[6] Wang, X., Jiang, W. and Luo, Z., 2016, December. Combination of convolutional and recurrent neural network for sentiment analysis of short texts. In *Proceedings of COLING 2016, the 26th international conference on computational linguistics: Technical papers* (pp. 2428-2437).

[7] Shankar, P., Arora, M.A., Kaushal, R. and Singh, I., 2019, September. Analyzing Varied Approaches for Forecast of Stock Prices by Combining News Mining and Time Series Analysis. In *2019 International Conference on Computing, Power and Communication Technologies (GUCON)* (pp. 434-441). IEEE.

[8] Zhai, Y., Hsu, A. and Halgamuge, S.K., 2007, June. Combining news and technical indicators in daily stock price trends prediction. In *International symposium on neural networks* (pp. 1087-1096). Springer, Berlin, Heidelberg.

[9] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł. and Polosukhin, I., 2017. Attention is all you need. In *Advances in neural information processing systems* (pp. 5998-6008).

[10] Myagmar, B., Li, J. and Kimura, S., 2019. Cross-domain sentiment classification with bidirectional contextualized transformer language models. *IEEE Access*, *7*, pp.163219-163230.

[11] Pappagari, R., Zelasko, P., Villalba, J., Carmiel, Y. and Dehak, N., 2019, December. Hierarchical Transformers for Long Document Classification. In *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)* (pp. 838-844). IEEE.

[12] Zhao, W., Ye, J., Yang, M., Lei, Z., Zhang, S. and Zhao, Z., 2018. Investigating capsule networks with dynamic routing for text classification. *arXiv preprint arXiv:1804.00538*.

[13] Rathnayaka, P., Abeysinghe, S., Samarajeewa, C., Manchanayake, I. and Walpola, M., 2018. Sentylic at IEST 2018: Gated recurrent neural network and capsule network based approach for implicit emotion detection. *arXiv preprint arXiv:1809.01452*.

[14]Arik, S.O. and Pfister, T., 2019. Tabnet: Attentive interpretable tabular learning. *arXiv preprint arXiv:1908.07442*.

[15] Socher, R., Pennington, J., Huang, E.H., Ng, A.Y. and Manning, C.D., 2011, July. Semi-supervised recursive autoencoders for predicting sentiment distributions. In *Proceedings of the 2011 conference on empirical methods in natural language processing* (pp. 151-161).

[16] Devlin, J., Chang, M.W., Lee, K. and Toutanova, K., 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

[17] Liu, J., Lin, H., Liu, X., Xu, B., Ren, Y., Diao, Y. and Yang, L., 2019. Transformer-based capsule network for stock movement prediction. In *Proceedings of the First Workshop on Financial Technology and Natural Language Processing* (pp. 66-73).

[18] Hinton, G.E., Sabour, S. and Frosst, N., 2018, February. Matrix capsules with EM routing. In *International conference on learning representations*.

[19] Bahdanau, D., Cho, K. and Bengio, Y., 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

[20] Sabour, S., Frosst, N. and Hinton, G.E., 2017. Dynamic routing between capsules. In *Advances in neural information processing systems* (pp. 3856-3866).

[21] Kim, Y., Denton, C., Hoang, L. and Rush, A.M., 2017. Structured attention networks. *arXiv preprint arXiv:1702.00887*.

[22] Dai, Z., Yang, Z., Yang, Y., Carbonell, J., Le, Q.V. and Salakhutdinov, R., 2019. Transformer-xl: Attentive language models beyond a fixed-length context. *arXiv preprint arXiv:1901.02860*.

[23] J. Pennington, R. Socher, and C. D. Manning, "GloVe: Global vectors for word representation," in EMNLP 2014 - 2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference, 2014, pp. 1532-1543.

[24] Agrawal, M., Khan, A.U. and Shukla, P.K., 2019. Stock price prediction using technical indicators: A predictive model using optimal deep learning. *Learning*, *6*, p.7.

[25] Abdullah, M.H.L.b, Ganapathy, V.: Neural Network Ensemble for Financial Trend Prediction. Proc. TENCON 3 (2000) 157-161.

[26] Mitchell, M.L., Mulherin, J.H.: The Impact of Public Information on the Stock Market. Journal of Finance 49 (3) (1994) 923–950.

[27 Nofsinger, J.R.: The Impact of Public Information on Investors. Journal of Banking & Finance 25 (2001)1339-1366.

[28]. Mittermayer, M.: Forecasting Intraday Stock Price Trends with Text Mining techniques. Proc. of the 37th Hawaii International Conference on System Sciences, Hawaii (2004).

[29]. Brown, G.W., Cliff, M.T.: Investor Sentiment and the Near-Term Stock Market. Journal of Empirical Finance 11 (2004) 1-27.

[30] Choi, J.H., Lee, M.K., Rhee, M.W.: Trading S& P 500 Stock Index Futures Using a Neural Network. Proc. of the Annual Int. Conference on Artificial Intelligence Applications on Wall Street, New York (1995) 63–72.

[31] Quah, T.S., Srinivasan, B.: Improving Returns on Stock Investment through Neural Network Selection. Expert Syst. Appl. 17 (1999) 295–301.

[32] Kim, K., Han, I.: Genetic Algorithms Approach to Feature Discretization in Artificial Neural Networks for the Prediction of Stock Price Index. Expert Syst. Appl. 19 (2) (2000) 125–132.

[33] Romahi, Y., Shen, Q.: Dynamic Financial Forecasting with Automatically Induced Fuzzy Associations. Proc. of the 9th Inter. Conf. on Fuzzy systems (2000) 493-498.

[34] Hassan, M.R., Nath, B.: Stock Market Forecasting Using Hidden Markov Model: a new approach. Proc. of 5th Int. Conf.on intelligent systems design and applications (2005).

[35] Leigh, W., Purvis, R., Ragusa, J.M.: Forecasting the NYSE Composite Index with Technical Analysis, Pattern Recognizer, Neural Network, and Genetic Algorithm: a Case Studying Decision Support. Decision Support Systems 32 (2002) 361–377.

[36] Abraham, A., Nath, B., Mahanti, P.K.: Hybrid Intelligent Systems for Stock Market Analysis. Proc. of the Inter. Conf. on Computational Science (2001) 337-345.

[37] Hellstrom, T., Holmstrom, K.: Predicting the Stock Market. Technical Report Series IMATOM-1997-07 (1998).

[38] Kim, K.J., 2003. Financial time series forecasting using support vector machines. *Neurocomputing*, *55*(1-2), pp.307-319.

[39]Jabbarzadeh, Armin, et al. "A Multiple-Criteria Approach for Forecasting Stock Price Direction: Nonlinear Probability Models with Application in S&P 500 Index." International Journal of Applied Engineering Research 11.6 (2016): 3870- 3878.

[40] Chen, K., Zhou, Y. and Dai, F., 2015, October. A LSTM-based method for stock returns prediction: A case study of China stock market. In *2015 IEEE international conference on big data (big data)* (pp. 2823-2824). IEEE.

[41] F. B. Oriani and G. P. Coelho, "Evaluating the impact of technical indicators on stock forecasting," 2016 IEEE Symposium Series on.

[42] Shynkevich, Y., McGinnity, T.M., Coleman, S.A., Belatreche, A. and Li, Y., 2017. Forecasting price movements using technical indicators: Investigating the impact of varying input window length. *Neurocomputing*, *264*, pp.71-88.

[43] Agrawal, M., Khan, A.U. and Shukla, P.K., 2019. Stock price prediction using technical indicators: A predictive model using optimal deep learning. *Learning*, *6*, p.7.

[44] Wang J. Z., Wang J. J., Zhang Z. G. and Guo S. P. (2011). ”Forecasting stock indices with back propagation neural network.” Expert Systems with Applications 38 (11): 14346-14355.

[45] Guresen, E., Kayakutlu, G. and Daim, T.U., 2011. Using artificial neural network models in stock market index prediction. *Expert Systems with Applications*, *38*(8), pp.10389-10397.

[46] Lipton, Z.C., Berkowitz, J. and Elkan, C., 2015. A critical review of recurrent neural networks for sequence learning. *arXiv preprint arXiv:1506.00019*.

[47] Rather, A.M., Agarwal, A. and Sastry, V.N., 2015. Recurrent neural network and a hybrid model for prediction of stock returns. *Expert Systems with Applications*, *42*(6), pp.3234-3241.

[48] Zeng, Y. and Liu, X., 2018, September. A-Stock Price Fluctuation Forecast Model Based on LSTM. In *2018 14th International Conference on Semantics, Knowledge and Grids (SKG)* (pp. 261-264). IEEE.

[49] Li, G., Xiao, M. and Guo, Y., 2019, October. Application of Deep Learning in Stock Market Valuation Index Forecasting. In *2019 IEEE 10th International Conference on Software Engineering and Service Science (ICSESS)* (pp. 551-554). IEEE.

[50] Kalchbrenner, N., Grefenstette, E. and Blunsom, P., 2014. A convolutional neural network for modelling sentences. *arXiv preprint arXiv:1404.2188*.

[51] Liu, S., Zhang, X., Wang, Y. and Feng, G., 2020. Recurrent convolutional neural kernel model for stock price movement prediction. *Plos one*, *15*(6), p.e0234206.

[52] Vargas, M.R., De Lima, B.S. and Evsukoff, A.G., 2017, June. Deep learning for stock market prediction from financial news articles. In *2017 IEEE International Conference on Computational Intelligence and Virtual Environments for Measurement Systems and Applications (CIVEMSA)* (pp. 60-65). IEEE.

[53] Wang, J.H., Liu, T.W., Luo, X. and Wang, L., 2018, October. An LSTM approach to short text sentiment classification with word embeddings. In *Proceedings of the 30th conference on computational linguistics and speech processing (ROCLING 2018)* (pp. 214-223).

[54] Li, D. and Qian, J., 2016, October. Text sentiment analysis based on long short-term memory. In *2016 First IEEE International Conference on Computer Communication and the Internet (ICCCI)* (pp. 471-475). IEEE.

[55] Xu, G., Meng, Y., Qiu, X., Yu, Z. and Wu, X., 2019. Sentiment analysis of comment texts based on BiLSTM. *Ieee Access*, *7*, pp.51522-51532.

[56] Liu, S., Zhang, X., Wang, Y. and Feng, G., 2020. Recurrent convolutional neural kernel model for stock price movement prediction. *Plos one*, *15*(6), p.e0234206.

[57] Oncharoen, P. and Vateekul, P., 2018, August. Deep learning for stock market prediction using event embedding and technical indicators. In *2018 5th International Conference on Advanced Informatics: Concept Theory and Applications (ICAICTA)* (pp. 19-24). IEEE.

[58] Sun, J. (2016, August). Daily News for Stock Market Prediction, Version 1. Retrieved [30.06.2020] from https://www.kaggle.com/aaron7sun/stocknews.

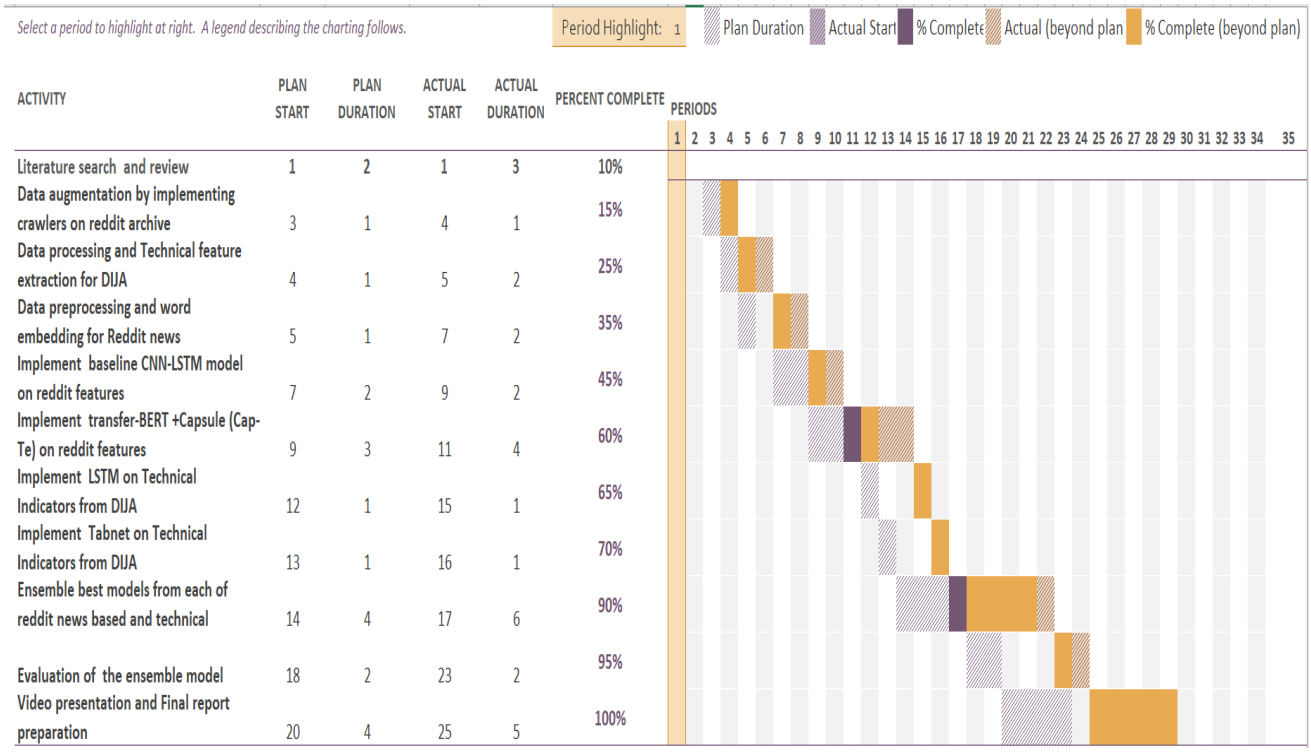
[59] J. Bollen and H. Mao, “Twitter mood as a stock market predictor”, Computer, vol. 44, no. 10, pp. 91–94, 2011.

[60] Ovadia, S., 2015. More than just cat pictures: Reddit as a curated news source. *Behavioral & Social Sciences Librarian*, *34*(1), pp.37-40.

[61] Mu, J., Bhat, S. and Viswanath, P., 2017. All-but-the-top: Simple and effective postprocessing for word representations. *arXiv preprint arXiv:1702.01417*.

[62] Kim, J., Jang, S., Park, E. and Choi, S., 2020. Text classification using capsules. *Neurocomputing*, *376*, pp.214-221.

APPENDIX A: RESEARCH PLAN



ACTIVITY	PLAN START	PLAN DURATION	ACTUAL START	ACTUAL DURATION	PERCENT COMPLETE
Literature search and review	1	2	1	3	10%
Data augmentation by implementing crawlers on reddit archive	3	1	4	1	15%
Data processing and Technical feature extraction for DIJA	4	1	5	2	25%
Data preprocessing and word embedding for Reddit news	5	1	7	2	35%
Implement baseline CNN-LSTM model on reddit features	7	2	9	2	45%
Implement transfer-BERT +Capsule (Cap-Te) on reddit features	9	3	11	4	60%
Implement LSTM on Technical Indicators from DIJA	12	1	15	1	65%
Implement Tabnet on Technical Indicators from DIJA	13	1	16	1	70%
Ensemble best models from each of reddit news based and technical	14	4	17	6	90%
Evaluation of the ensemble model	18	2	23	2	95%
Video presentation and Final report preparation	20	4	25	5	100%

APPENDIX B: RESEARCH PROPOSAL

Stock Market Prediction from financial indicators and daily news feed using deep learning

Samrat Sengupta, Liverpool John Moore's university U.K [2020]

Abstract

Forecasting stock prices has been a challenging problem, and it has attracted many researchers in the areas of the economic market, financial analysis, and computer science. The current study is associated with the use of deep learning models to predict the intraday stock price changes of the Dow Jones Industrial Average (DJIA) stock market index with the help of daily news feed from Reddit and stock technical indicators. In recent years, Convolution neural network (CNN), Recurrent neural network (RNN), and their variants are applied successfully in this domain, the most prominent being a jointed Recurrent convolution neural network (RCNN) architecture. Here, more advanced models like Capsule net and Transformers (which have been proven very successful in sentiment analysis) are studied for analyzing the impact of global events in form of news. The premise of using these architectures is based on the fact that the transformer Encoder extracts the deep semantic features of the news headlines while capsule network captures the structural relationship of the texts to derive meaningful features. Another relatively new architecture Tab-net which uses sequential attention to decide feature importance is explored for processing stock technical indicators. Not only Tab-net captures sequential features like Long short-term memory (LSTM) but also uses attention mechanisms to deduce insights from structured tabular data. Finally, an ensemble of Tab-net and Capsule Transformer variants is carried out to project the overall impact of the daily news feed as well as exchange Technical indicators on stock price change.

Table of Contents

1.Introduction

The ability to predict the movement of the stock market is considered an important ingredient in investing. The objective of any investor should be to forecast the market behavior with the closest approximation to make the best decision while buying or selling a stock that can return an optimal profit. Hence a reliable stock market prediction model becomes an important instrument in achieving financial goals for an investor.

The DJIA stock index is a price-weighted average of stock prices from 30 of the largest publicly-traded companies in the United States widely considered as the benchmark for stock market analysis including price movement and its change. Traditionally, technical analysis has been used to make near-term forecasts on this index, based on the premise that future values of a financial time series are conditioned on its past values. However recent work by Lee, Heeyoung, et al has

the shown impact of global events captured in form of daily news titles on stock price movements. The RCNN networks have been successful to extract information from both news titles by converting them to event embeddings and technical indicators to make predictions that beat predictors that do not use news titles. Experiments on a jointed architecture of CNN LSTM demonstrated substantial improvement in accuracy while predicting stock price change from news articles alone by Wang and Xingyou et al using pre-trained word embeddings. More advanced architectures like Transformers with their variants are relatively rarely explored methodologies in the current area of study though these models have been successfully implemented in areas of Neural machine translation and sentiment analysis by Myagmar, Batsergelen et al . Transformers are included as a subject of our study here since they can capture a deep semantic relationship between the sentences. Another recently introduced variant of deep learning model Capsule net also showed promises in text classification problems because they are trained to extract fine-grained structural and spatial information from complex text articles as shown by Zhao, Wei, et al . This architecture and variants are also explored here. Finally, a combination of both Transformer and capsule-Net is tried on pre-trained word embeddings to arrive at optimal performance for news headlines driven regression model. The technical indicators for the stock price prediction model are based on statistical representation and the moving average of last n days exchange data. Apart from LSTM a recently introduced architecture namely Tab-net is tried to fit those indicators to produce optimum output. Tab-net chooses reasonable features at each decision step using sequential attention. Hence enables interpretability and better learning as the learning capacity is preserved for the most prominent features as been depicted by Arik, S.O. and Pfister. Instead of following traditional approaches of feature concatenation from two different networks processing text and numerical data a unique ensemble technique is comprising the text-based and technical indicator-based models is followed.

2. Background and related research

Although the Stock market prediction remains among one of the hottest topics of research in the financial domain, the volatile nature of stocks with diversified events and indicators affecting its price changes makes the task challenging. Typically, as in all other research areas, the methodologies for stock market prediction also walked the tide of continuous evolution. The earliest forms of research in this domain included trend derivation and pattern recognition methods

like exponential moving average (EMA), oscillators, support and resistance levels or momentum, and volume indicators as Candlestick patterns(1980's). These methods mainly considered stock exchange features to arrive at a prediction. With the increased availability of internet, global events captured in form of news headlines influencing slide and surge of stock prices were also taken into account. With the arrival of digital computation, the stock market prediction has entered into the technological realm (1990's and 2000's). Artificial neural networks (ANNs) and Machine Learning Algorithms were among the most prominent techniques for some time. However, with the advancement of computation powers emerged the era of deep learning and since then researches have been all been aimed at exploring different neural network-based architectures to solve the problem (2010's). The most successful and near state of the art having been variants of RNN/LSTM and CNN based hybrid architectures. In more recent times the Attention-based models like Transformers which capture deep semantic relations in text and Capsule net which captures structural information from has been applied successfully in fields of neural machine translation and sentiment analysis. Most related previous studies to this work can be classified into 4 groups.

2.1 Deep Learning with Textual Information

Traditional approaches for depicting textual information used noun phrases, named entities, bags-of-words (Bow), and term frequency-inverse document frequency (TF-IDF). However, these approaches were not able to capture entity-relation information or the semantics of news headlines. The motivation for text mining approaches in stock market prediction has been derived from the work of Lee et al. This paper was one of the first to show a significant improvement in the predictive power of a stock (relative to an index) based on textual information. Even more important is that they show the variation in the predictive power of text over time from a relevant event. However, Lee et al. did not implement any deep learning methods. Deep Learning becomes effective once words are converted to high dimensional distributional vectors that capture syntactic, morphological, and semantic information. Word embeddings that convert word to vector form have achieved remarkable results when applied to CNN/RNN. Kim et al used pre-trained word vectors with CNN for sentence-level classification. Sentence level vectors have also been constructed from RNN in sentiment analysis tasks as has been shown by Socher et al. The paragraph-level encoding was put forward by Le and Mikolov et al . Tai et al proposed the tree-

structured LSTM networks to improve the semantic representations. The performance improvement owing to word embeddings has been attributed to the fact that Word embeddings can solve the semantic sparsity of short texts in a better way compared with the one-hot representation. While CNN takes into account local features from words or phrases in different places of texts RNN considers learns sequential order and long-term dependencies of texts. These approaches have been abridged in the work proposed by Wang, Xingyou et al where they came up with a joint CNN RNN model for sentiment analysis of the short text. This has been taken as a baseline in our study while processing Reddit news headlines.

2.2 Deep Learning with Textual Information and Technical Indicators

Vargas, Manuel R., et al proposed RCNN for predicting the intra-day directional movements of the Standard & Poor's 500 Index (S&P 500). While CNN has a superior ability to extract semantic information from texts, RNN is better at capturing the sequential and contextual information. RCNN utilizes both these properties to its advantage. A two-step process is applied to represent each news in the data set: first, a word2vec model is used to generate a word embedding; second, an averaging of all these word vectors of the same sentence is performed to address sparsity in word-based inputs. This RCNN model uses only news from the day before the forecasting and gives a better result than a set of models that uses news from the past day, week, and month. This reinforces Lee's idea that the impact of news on stock diminishes with time. This model also uses an input of seven technical indicators extracted from the stock exchange. The formulation of these technical indicators was originally proposed by Zhai et al . They include the Stochastic %K, Stochastic %D, Momentum, Rate of Change, William's %R, A/D Oscillator, and Disparity 5, which are commonly used to describe assets. Moreover, the papers confirms the positive outcome of the use of a hybrid input (news and technical indicators), leading to the conclusion that both sources of information are important. Here Vargas' model for processing technical Indicators is taken as our baseline while analyzing DJIA exchange indicators. Also, his work paved the way for using a combination of textual Information and technical indicators in the current context.

2.3 Advanced Deep Learning architectures for Textual information.

The architectures of not so recent era like CNN/RNN have all has their limitations which gave rise to the advancement of deep learning techniques and the introduction of new architectures like capsule net and transformers. These are discussed below in context to the current problem. CNN through its mechanism of kernel filtering and pooling had one serious backdrop. The filtering method failed to capture the directional or vector properties of each element like rotation, angle, etc which are more directional. A relatively new form of architecture namely capsule net proposed by Geoffrey Hinton et al seemingly overcomes these problems by replacing scalar outputs of the convolved feature maps with vector outputs which take into account more directional features. Though primarily designed for image analytics this has been successfully experimented on for text classification purposes and several studies have been conducted to prove its effectiveness on textual information. This work takes motivation from earlier studies conducted by Zhao, Wei, et al , and Rathnayaka, Prabod, et al . Here variants of capsule net are applied to analyze the impact of news headlines on stock price change. On the other hand, RNNs become ineffective when the gap between the relevant information and the point where it is needed becomes very large while processing long texts. LSTM's seemed to solve that problem by introducing cell states to preserve the information passed by previous words to a large extent. However still when the documents are long enough the model often forgets the content of distant positions in the sequence. To solve some of these problems, researchers created a technique for paying attention to specific words. Attention was presented by Dzmitry Bahdanau, et al, which reads as a natural extension of their previous work on the Encoder-Decoder model. In a nutshell, attention in deep learning can be broadly interpreted as a vector of importance weights: to predict or infer one element, such as a pixel in an image or a word in a sentence, we estimate using the attention vector how strongly it is correlated with or "attends to" other elements and take the sum of their values weighted by the attention vector as the approximation of the target. With the advent of attention mechanisms, there rose variants of architectures based on the theory like self-attention, hierarchical attention, and multiheaded attentions. The focus of the current study is based on multi-head self-attention or Transformer architecture proposed by Vaswani et al. The multi-head self-attention layer in the Transformer aligns words in a sequence with other words in the sequence, thereby calculating a representation of the sequence. It is more effective in representation and less computationally efficient than convolution and recurrent networks. The intuition behind the Transformer inspired several researchers, leading to the development

of self-attention-based models such as Bidirectional Encoder Representations from Transformers (BERT) by Devlin et al. Along with Bert, there have several transformer-based language models that showed breakthrough results such as XLNet, Roberta, GPT-2, and ALBERT. With sentiment and opinion expressions becoming widespread throughout social and e-commerce platforms, the applications of Transformer variants mentioned above slowly found their applications in this works as shown in studies done by Myagmar, Batsergelen et al focusing on Cross-domain sentiment classification applying transfer learning methods which is an inspiration to the current work. The combination of capsule-based Transformer network which is the final approach in current work is still relatively new and has been under experimentations in workshops, one such reference is given here Liu, Jintao, et al.

2.4 Deep Learning for Tabular data

The attention theory was not only restricted in Textual/Visual information but found its implementation in realms of tabular data as well. In this paper S.O. and Pfister, T, the authors propose a novel architecture with sequential attention modules for tabular learning. An attention module is trained to select some elements from the (normalized) input feature, and a feature transformer takes the selected features for overall feature embedding. The philosophy behind this architecture is driven by the fact that Tab-Net is designed to learn a decision-tree-like mapping to inherit the valuable benefits of tree-based methods (interpretability and sparse feature selection) while providing the key benefits of DNN-based methods (representation learning and end-to-end training). For Technical Indicator based stock prediction this approach is adopted in the current study.

3.Research Question

To study the performances of Transformer & Capsule networks based on textual information as well Tabnet based on Technical indicators for predicting intraday stock price change of Dow Jones Industrial Average and explore the outputs of an ensembled model for daily prediction from above approaches.

4. Aim and Objectives

This work aims to improve the earlier solutions to this problem by exploring deep learning techniques based on i) historical news headlines and ii) past technical indicators of the exchange to predict the movement of a particular stock over the coming period (daily).

The objectives of the research are outlined as followed.

- Derive word embedding based text features based on Reddit news headlines and moving average/stochastic descent-based technical indicators based on stock features.

- Investigate the state of art CNN-LSTM architecture in determining stock prediction on above text-based features

- Study the performance of state of art deep learning models like Transformer and it's variants like Bert/XLnet on news feed data for the current problem.

- Study the performance of the Capsule network on text-based features of the mentioned dataset.

- Study the performance of the model created by hybridizing variants of Transformers with Capsule

- Study the performance of Tab-net architecture for prediction of the stock price from technical indicators

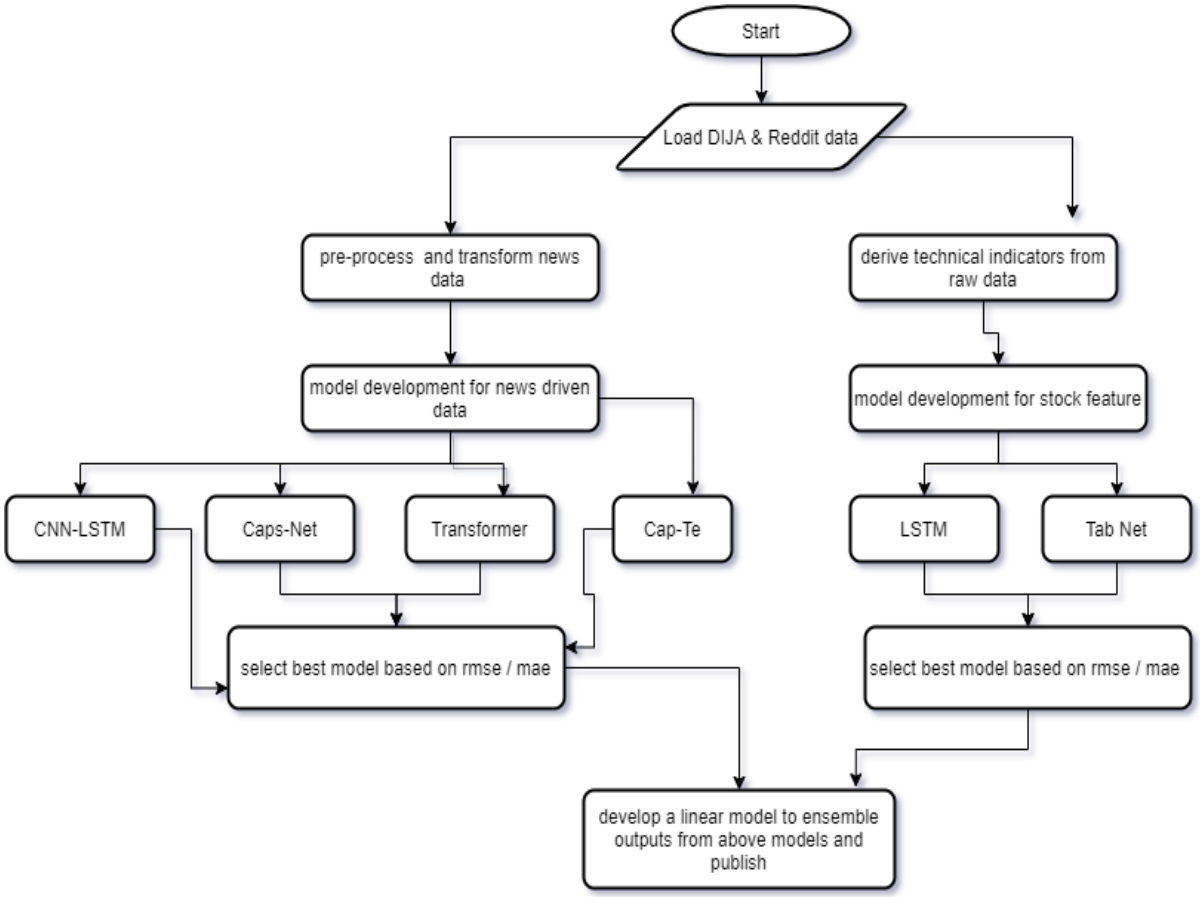
- Produce a hybrid ensemble model from both text-based news headlines and numerical technical indicators combining above and evaluate its performances to predict stock movement for a day.

5. Significance and Scope of the study.

The study of advanced deep learning architectures for stock market prediction using both news headlines and technical indicators in this research can lead to the not only establishment of near state of art models for stock market forecasting but also open an entirely new outlook for using hybrid neural networks for further improvement by researchers using the concepts as a baseline. A reliable model that meticulously derives insights from global events in form of news and historical stock features and can perform stock forecasting with optimum accuracy is immensely helpful for financial investors in making decisive investments that can lead to considerable financial gains and minimize risking a considerable fortune. Therefore, it can be of utmost benefit to people related to investment banking and finance, shareholders and brokers, and stock market analysts. The research though uses Dow-jones Industrial exchange with Reddit news headlines as a primary source that applies to any other similar stock exchange-based data like NSE, NYSE, etc. The deep learning architectures that are studied here include capsule net, transformers, and Tab-net which has been previously implemented successfully in sentiment analysis and predictive analytics domain. The study is focused on the implementation, hybridization, and ensemble of these deep learning models to achieve an optimum outcome concerning the current domain of stock forecasting. It excludes detailed analysis of the previous state of art model (CNN-LSTM) in this domain as well as theories related to reinforcement learning which is based on more temporal data (hourly) and is also a current trend.

6. Research Methodology.

An outline for the research methodology process flow is given below with steps mentioned in detail.



6.1 Dataset Description

The existing dataset contains approximately 74,000 Reddit news article titles with their corresponding dates, as well as data about the Dow-Jones Industrial average per day from 8-Aug-2008 to 1st-Jul-2016. The Source of data files is https://www.kaggle.com/aaron7sun/stocknews. The dataset has been further enriched by adding data from 2nd Jul 2016 to 29th Jun 2020 with Push Shift API

enabled crawlers for Reddit news. For Dow Jones, additional data have been directly downloaded from Yahoo Finance. Dataset is broadly divided into 2 files.

- RedditNews.csv: 2 columns 3360 rows

  The first column is the date, and the second column is the news headlines. All news is ranked from top to bottom based on user voting on hotness quotient Hence, there are 25 lines for each date.
- DJIA_table.csv: 7 columns and 3360 rows.

  The columns for DJIA data sources are date, open, high, low, close, volume, Adjacent close.

6.2 Preprocessing & Feature Extraction

For newsfeed driven modeling the Reddit news articles are clubbed together date wise in a data frame and a response column indicating intraday stock price change is calculated for 3360 dates. The news feed data is then cleaned by removing punctuations and stop words by python's nltk lib .from 25 headlines each news article is cropped based on word limit with padding and merged sequentially with all the headlines to create text body of particular word length to avoid excessively long training time and balance the number of headlines used and the number of words from each headline., post that each text body is converted to a sequence of integer tokens based on vocabulary and embedding matrix is created with pre-trained glove model. Let's name this Reddit-transformed.csv. For stock price data in Dow Jones Index table which contains the date, open, high, low, close, volume, adjacent close as base columns, the following features are derived and normalized.

- Stochastic %K = C_t - L_{ln} / Hh_n - Ll_n
- Stochastic %D = sum (i=0 to n) Kt_{-i} / n %
- Momentum = C_t - C_{t-n}
- Rate of Change = C_t / C_{t-n} * 100
- William's %R = (H_n - Ct/H_t - L_n) * 100
- A/D Oscillator = (H_t - C_{t-1} / H_t - L_t)
- Disparity 5 = C_t/MA_5 * 100

[where C_t is the closing price at day t , H_t is the highest price at day t, L_t is the lowest price at day t, MA_n is moving average of the past n days, and Hh_n and Ll_n are the highest high and the lowest low in the past n days, respectively].These values are taken as predictors and intra-day stock price change as the response variable.

6.3 Modeling

For prediction based on news feed data baseline model of CNN-LSTM is studied followed by variants of Capsule net and Transformer architectures (Transformer encoder/XLnet/Bert). Finally, a transformer encoded Capsule net is designed and its performance is analyzed. On a similar notation, the Tab-net model is also observed for stock market technical indicators. Cap-Te and Tab-net are finally ensembled for daily prediction.

6.3.1 Models for news Headlines based prediction

Apart from CNN-LSTM based baseline model, variants of Capsule net and transformers are studied. Also, a hybridization of Capsule and Transformer namely Cap-Te is be considered.

Baseline models

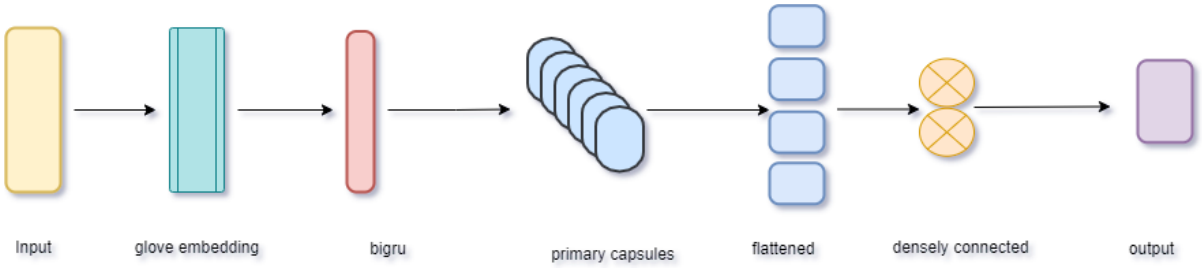
CNN – LSTM is studied as the baseline model for our work Wang, Xingyou, et al. The model uses glove embedding as it's weight in the embedding layer followed by conv1D layers (each with dropouts). The next single LSTM layer with dropouts is implemented. Finally, the output is connected to a dense layer of linear activations followed by the output layer with a single node (linear activation). This model gave an accuracy score of 0.82 in their experiments.

Proposed models

i) Bigru-Capsule

300-dimensional pre-trained glove embeddings are used as a word embedding matrix followed by a bidirectional GRU. The output from forward and backward GRU's are concatenated and passed to the Capsule Network. Each of the capsules in a capsule net corresponds to high-level structural features in the text. The output of each capsule is a vector whose magnitude represents the probability of corresponding feature existence. The prediction vector is calculated by multiplying the GRU layer's output with a weight matrix. The input to the capsule is a weighted sum of all these prediction vectors multiplied by coupling

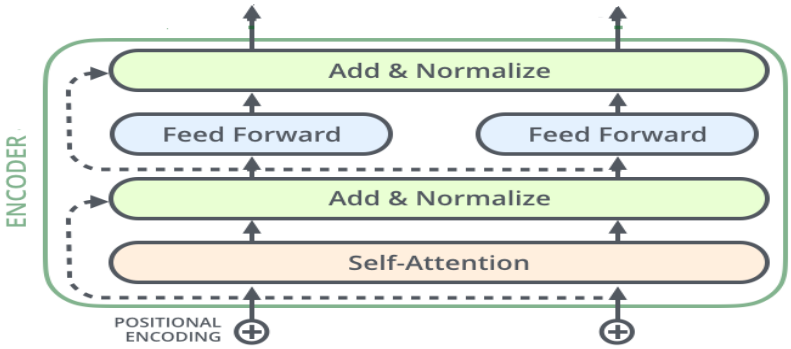
coefficient generated from dynamic routing between the capsules as mentioned by Sabour, Sara, et al. A non-linear Squash function is used to scale the vectors such that the magnitude is mapped to a value between 0 and 1. The flattened output from the capsule layer is fed to dense layers that have linear activations with a single output node for calculating continuous output.



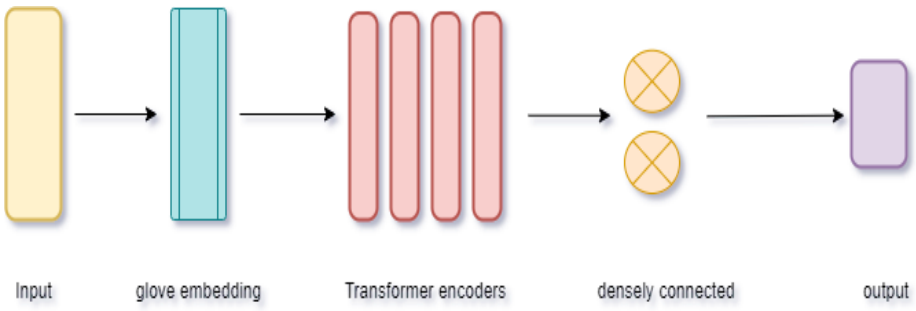
ii) Transformers and its variances

The Transformer architecture uses an attention mechanism to solve the problem of remembering the long-range dependency of words in a document. These are discussed here briefly; a positional encoder creates positional embeddings along with word embeddings in so that words will be closer to each other based on both their meaning and position in a sentence. The padded tokens are masked. From the masked input vector, 3 vectors namely query(Q), Key(K), and Value(V) are created. They are updated and trained during training. Next self-attention is calculated for every word in sequence. Self-attention function is represented as Attention(x) =Attention (Q, K, V) =SoftMax (QKT / $\sqrt{d_k}$) V where d_k is the dimension of keys and query. The output representing the multiplication of the attention weights and the V (value) ensures that focussed words are kept as-is and irrelevant words are flushed out. Q, K, and V are split into multiple heads to jointly attend to information at a different position based on representational

spaces, hence the term multi-head attention. The output from multi-head attention is passed to a pointwise feed-forward network consisting of two fully-connected layers with a Relu activation in between. The Transformer encoders can be stacked up to complete a block. Multiple such blocks constitute a layer. The Transformer mechanism is inspired by the work of Vaswani, Ashish, et al.



In the current model Transformer encoder layer is used to propagate output to densely connected layer with linear activations for generating continuous output.

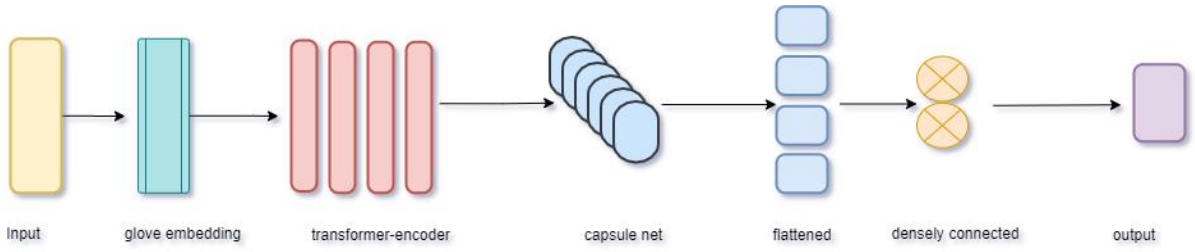


Additionally, the transformer variants like BERT and XLnet (all pre-trained models) are also experimented with by replacing the transformer encoder layer. The fine-tuned representation-based BERT achieved state-of-art performance on a Large suite of sentence-level and token-level tasks. However, it still had limitations on input length which was overcome with the introduction of XLnet

proposed by Dai, Zihang, et al , which is a modification on the original Transformer.

iii) Cap-Te networks

Finally, an architecture comprising of Transformer Encoder (pre-trained like XLnet/Bert) is coupled with the Capsule network.is proposed. For learning more valuable information from multiple news headlines, the transformer is used to encode the texts and then get the encoded representation as to the input of the capsule network. By capsule network, the relationship between different news appeared on the same trading day that belonging to one stock is captured. For each news feed, the pre-trained word embeddings (glove) to project each word token onto the model-dimensional space as the input of the Transformer encoder is utilized. The output from the Transformer and Capsule is then passed to a dense layer with linear activations for generating final continuous output. The ideation for the above model is adopted from Liu, Jintao, et al.



6.3.2 Models for Technical Indicator based prediction

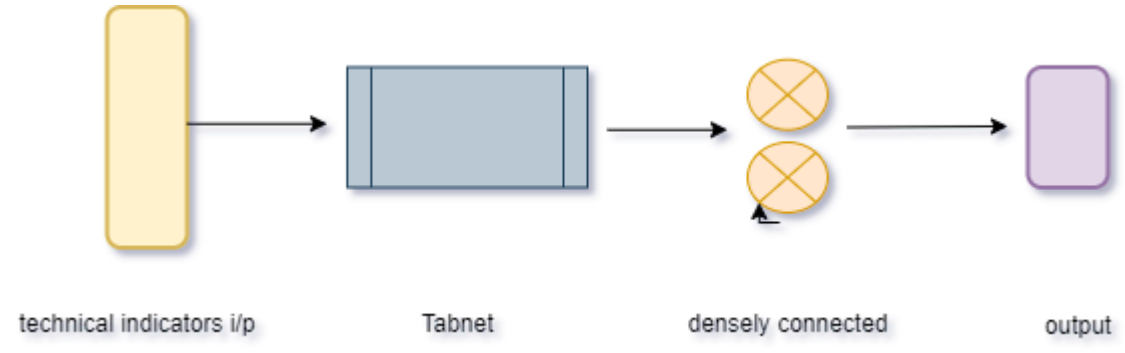
Baseline Models

The input layer is built to accept seven technical indicators as been mentioned in 6.2. The Next layers of LSTM with dropouts are stacked up and finally connected with a dense layer having linear activations. This is connected to the output layer with a single node and linear activation to get the final output.

Proposed Models

i)Tab-net

Tab-Net is the Integration of DNNs into decision trees that can outperform the tree-based algorithms while reaping many of their benefits. Typically, it consists of a feature transformer coupled with an attentive transformer with feature masking at each decision point. The processed representation is divided into two by a split block to be used by an attentive transformer in the next step as well for output construction. The feature selection mask provides interpretable information about the model's functionality at each step. The masks can be aggregated to obtain global feature important attributes. Here the Technical indicators in tabular form are passed directly to Tab-net encoder and then output is fed to densely connected layer with linear activation.



.

It's working principles are cited in detail over here Arik, S.O. and Pfister, T.

6.4 Model Evaluation

For both text-based and numerical feature-based models chosen loss functions are parameters Root mean squared error (RMSE) due to the continuous nature of the output. The losses are calculated and the training process is halted when a minimum is reached (even if it means before the specified number of epochs executed). These models are then validated on test data which was previously segregated. The metric for accuracy chosen is the R2 score due to the continuous nature of the output. For both text-based and numerical feature-based model, the ones which gave the best R2 score on test data are chosen for the ensemble.

6.5 Ensemble of Text-based and Numerical feature-based Models

Since the historical data is on daily basis for past 12 years, selected days can be chosen and kept aside as training, test and ensemble-test data from both modified dataset mentioned in section 6.2 [reddit-transformed and DJIA-transformed]The models from 6.3.1 are trained on Reddit-transformed train dataset and that from 6.3.2 are trained on DJIA-transformed train dataset. Next, the best performing model out these groups are chosen. Let's assume model $\mathbf{M_t}$ is a Reddit news trained model and model $\mathbf{M_n}$ is DJIA feature trained model. For a given date say M_t gives a prediction $\mathbf{x_t}$ and M_n gives another prediction say $\mathbf{x_n}$. These predictions are registered on all test data points which also have labels against them let it be **y**. As a result of the above operation, we can generate a new dataset with x_n, x_t as predictor features, and y as the response. We can figure out a linear model ($\mathbf{L_m}$)which gives minimal loss fit on the above test set. Let the coefficients of the model be a_1 and a_2 so the model will find an equation with $\mathbf{y = a_1 * x_n + a_2 * x_t}$ **.** Here we use a1 and a2 predicted from the above result aggregator model as our ensemble coefficient. In other words, our new model will calculate prediction on new datapoint as final prediction Y = (prediction of M_n) * a_1 + (prediction of M_t) * a_2 . The ensembled outcome is tested on ensemble-test data with trained model

weights and ensemble coefficients obtained from the linear model and the final result is noted.

7. Expected Outcomes

An ensemble model each from the technical indicator and textual information both based on advanced deep learning architectures are supposed to beat the previous state of art model in this domain. Not only it will predict stock price changes

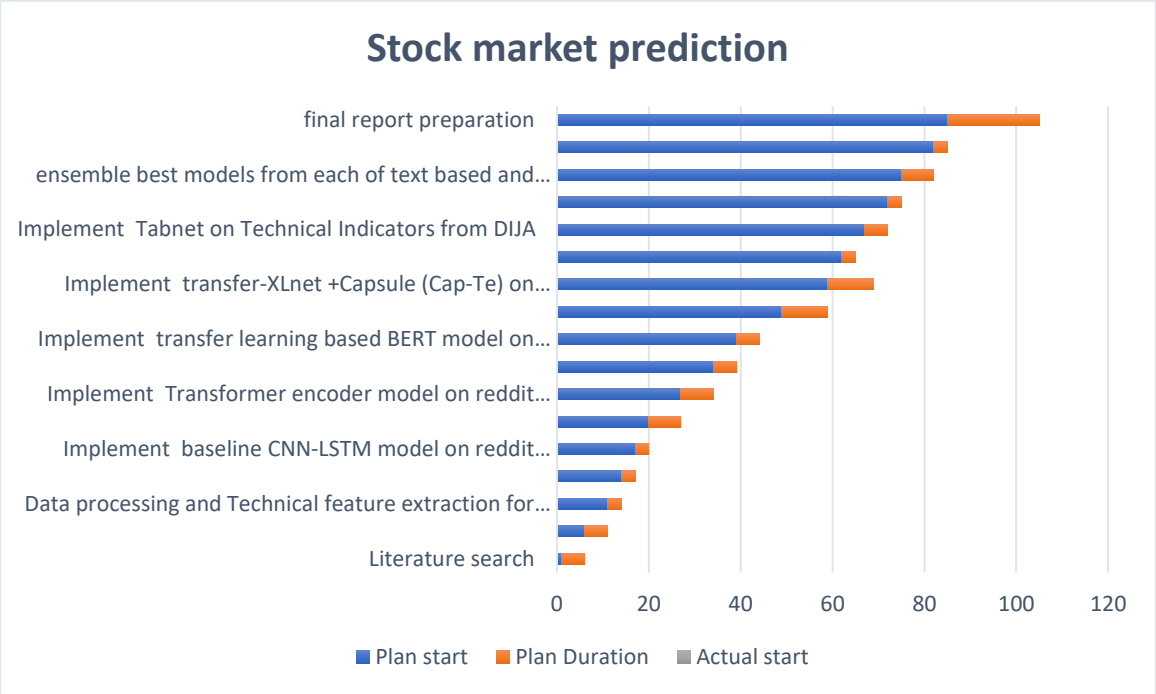
with optimum accuracy and serve the interest of stock buyers and sellers but also it can open a new set of research-based on variants of the CAP-TE network studied here. The CAP-Te model is supposed to perform better than either of Transformer or Capsule net taken separately as well as a jointed CNN-LSTM model. On the other hand, Tab-net is also expected to beat the regular LSTM model while predicting stock price based on Technical Indicators. Finally, an ensemble of the two instead of the concatenated network can set the trend in ensemble models with similar outcomes from different features rather than concatenating network processed outputs.

8. Requirements / resources

For carrying out the experiments following system configuration and libraries are used.

System	Environment	Libraries
CPU: I7-9750H	Python 3.7	Pytorch 1.5
RAM: 16GB	Anaconda 4.8.2	NLTK 3.4.5
OS: Windows 10	CudaToolkit 10.1	Tensorflow 1.14
GPU : NVIDIA RTX 2060	CUDNN 7.6.4	Keras 2.1.2

9. Research Plan



References

[1] Lee, Heeyoung, et al. "On the Importance of Text Analysis for Stock Price Prediction." *Proceedings of the 9th International Conference on Language Resources and Evaluation, LREC 2014*, 2014.

[2] Vargas, Manuel R., et al. "Deep Leaming for Stock Market Prediction Using Technical Indicators and Financial News Articles." *Proceedings of the International Joint Conference on Neural Networks*, 2018, doi:10.1109/IJCNN.2018.8489208.

[3] Kim, et al. "Convolutional Neural Networks for Sentence Classification." *EMNLP 2014 - 2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, 2014, doi:10.3115/v1/d14-1181.

[4] Le, Quoc, and Tomas Mikolov. "Distributed Representations of Sentences and Documents." *31st International Conference on Machine Learning, ICML 2014*, 2014.

[5]Tai, Kai Sheng, et al. “Improved Semantic Representations from Tree-Structured Long Short-Term Memory Networks.” *ACL-IJCNLP 2015 - 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, Proceedings of the Conference*, 2015.

[6] Wang, Xingyou, et al. “Combination of Convolutional and Recurrent Neural Network for Sentiment Analysis of Short Texts.” *COLING 2016 - 26th International Conference on Computational Linguistics, Proceedings of COLING 2016: Technical Papers*, 2016.

[7] Shankar, Pranav, et al. “Analyzing Varied Approaches for Forecast of Stock Prices by Combining News Mining and Time Series Analysis.” *2019 International Conference on Computing, Power and Communication Technologies, GUCON 2019*, 2019.

[8] Zhai, Yuzheng, et al. “Combining News and Technical Indicators in Daily Stock Price Trends Prediction.” *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2007.

[9]Vaswani, Ashish, et al. “Attention Is All You Need.” *Advances in Neural Information Processing Systems*, 2017.

[10] Myagmar, Batsergelen, et al. “Cross-Domain Sentiment Classification with Bidirectional Contextualized Transformer Language Models.” *IEEE Access*, 2019, doi:10.1109/ACCESS.2019.2952360.

[11] Pappagari, Raghavendra, et al. “Hierarchical Transformers for Long Document Classification.” *2019 IEEE Automatic Speech Recognition and Understanding Workshop, ASRU 2019 - Proceedings*, 2019, doi:10.1109/ASRU46091.2019.9003958.

[12] Zhao, Wei, et al. “Investigating Capsule Networks with Dynamic Routing for Text Classification.” *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, EMNLP 2018*, 2020, doi:10.18653/v1/d18-1350.

[13]Rathnayaka, Prabod, et al. *Sentylic at IEST 2018: Gated Recurrent Neural Network and Capsule Network Based Approach for Implicit Emotion Detection*. 2019, doi:10.18653/v1/w18-6237.

[14]Arik, S.O. and Pfister, T., 2019. Tabnet: Attentive interpretable tabular learning. *arXiv preprint arXiv:1908.07442*.

[15]Socher, Richard, et al. “Semi-Supervised Recursive Autoencoders for Predicting Sentiment Distributions.” *EMNLP 2011 - Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, 2011.

[16]Devlin, Jacob, et al. “BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding.” *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, 2019.

[17]Liu, Jintao, et al. “Transformer-Based Capsule Network For Stock Movements Prediction.” *First Workshop on Financial Technology and Natural Language Processing*, 2019.

[18] Hinton, Geoffrey, et al. “Matrix Capsules with EM Routing.” *6th International Conference on Learning Representations, ICLR 2018 - Conference Track Proceedings*, 2018.

[19] Bahdanau, Dzmitry, et al. “Neural Machine Translation by Jointly Learning to Align and Translate.” *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, 2015.

[20] Sabour, Sara, et al. “Dynamic Routing between Capsules.” *Advances in Neural Information Processing Systems*, 2017.

[21] Kim, Yoon, et al. “Structured Attention Networks.” *5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings*, 2019.

[22] Dai, Zihang, et al. “Transformer-XL: Attentive Language Models beyond a Fixed-Length Context.” *ACL 2019 - 57th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference*, 2020, doi:10.18653/v1/p19-1285.