data set and an accuracy of 87.6% was obtained. They used worldwide tweets which matched their search without restricting it to the USA where the Dow Jones is.

A research conducted by [4] compared traditional blogs with microblogs to determine the predicative power on stock prices given the use of either data source. The research focused on sentiment analysis of blogs and micro-blogs and experiments concluded that micro-blogs consistently outperformed blogs in their predictive accuracy. The two data sources used were from Google Blogsearch11 and Twitter. The system used the stock name to filter and reduce the text after which sentiment analysis was performed using a lexicon of positive and negative terms. In the experiments, they predicted the actual stock price of the following day from the models of each data source. It was also found that the character limit of Twitter helped determine more concise sentiment results since one Twitter post usually relates to one topic.

In their paper, [5] described prediction of stocks using historical data together with twitter sentiment analysis. They used psychological states of twitter users using a lexicon based approach. The support vector machine and neural networks models were applied to over 755 million tweets. The dataset (tweets) were downloaded using twitter API for the period 13/02/2013 to 29/09/2013. The tweets were grouped into 8 states(categories) which are 'happy', 'loving', 'calm', 'energetic', 'fearful', 'angry', 'tired' and 'sad'.

A publication by [6] investigated the relationship between twitter feed content and stock market prices. They wanted to see how well sentiment information can predict future shifts in prices. Their model was successful in predicting stock movements. Alex Davies financial dictionary was used to calculate the sentiment of each text and they collaborated this by the stock prices .[7], found out a high negative correlation between words 'hope, 'fear' and 'worry' in tweets with the Dow Jones Average Index.

[8] introduced a novel approach for classification of twitter sentiment. The messages were classified into 2 classes, positive and negative. The research can be used for purchases of goods or by companies for the feedback of their products. Naïve Bayes and Maximum Entropy were used and they resulted in a very high accuracy in classification of twitter sentiment.

Rather than focusing on the content of news articles, [9] used news article headlines to classify stock movement in either up, down or steady directions. This was done by using different document and term weight techniques, including Term frequency-Inverse document frequency (Tf-Idf) and Term frequency-Category discrimination frequency (Tf-Cdf). Results showed better performance than random guessing.

3. Methodology

The framework used in this paper is divided into four major components: Data Pre-Processing, Feature Extraction and Selection, Prediction Model, and Evaluation as shown in Figure 1 below.

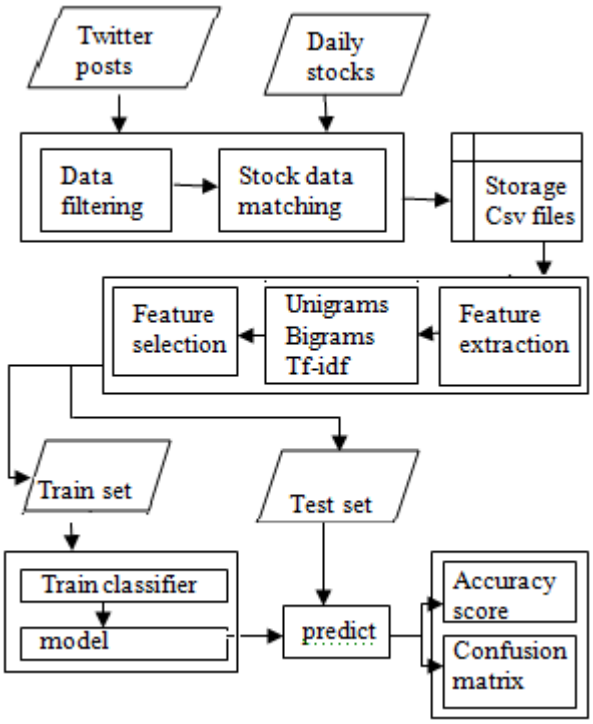

Figure 1: System Framework

3.1 Data gathering

The data set is a combination of tweets and stock quotes. The tweets collected were from January 5 2015 to June 30 2015 and over 200 000 tweets were collected during this time frame. Twitter posts or tweets are released within seconds from millions of users each day. Due to the huge number of tweets, more emphasis was put on problems concerning relevance and evaluation of information, including filtering out a lot of noise surrounding relevant information. The tweets were collected directly from twitter using Twitter API and filtered using keywords for example #airtel. The relevant stocks were downloaded directly from yahoo finance. One of the most important things which were kept track of was the time at which each tweet was created.

3.2 Pre-Processing

The gathered data was processed and only relevant data was retained. Data processing involved a lot of stages which included removing noise and words which did not provide any useful information to the problem. The noise in this scenario was spam tweets and in some cases retweets. After removing junk tweets, stop words were also removed since they did not provide any additional information necessary for building the model in question. Url and digit removal are other pre-processing steps which were carried out on the data.

3.3 Causality test

A causality test is done to check if one time series has an effect on another time series. In this research, the granger causality test was used to check whether tweets sentiments

can forecast the stock movements on the stock exchange. It was found out that in some cases it was possible to forecast the movement using the tweets sentiment. The mood scores of the tweets were quantified using the Alex Davies list of words by taking the logarithmic probability of being happy or sad. For each tweet we implemented the traditional bag of words. The probabilities of being sad or happy were calculated as shown in equation 1.:

$$(\text{happy}, \text{sad}) = (p, 1-p) \quad (1)$$

Log odd score, S were calculated as follows:

$$S = log(\frac{p}{1-p}) \quad (2)$$

Thus a score greater than 1 is positive and negative otherwise.To calculate the granger causality test the following formula was used:

Let *y* and *x* be stationary timeseries. To test the null hypothesis that *x* does not granger cause *y*, it was necessary to first find the proper lagged values of *y* (stock movement) to include in a univariate auto regression of *y*:

$$y_t = a_0 + a_1 y_{t-1} + a_2 y_{t-2} + \cdots + a_m y_{t-m} + \text{residual}_t \quad (3)$$

Next the autoregression is augmented by including lagged values of *x*:

$$y_t = a_0 + a_1 y_{t-1} + a_2 y_{t-2} + \cdots + a_m y_{t-m} + b_1 x_{t-1} + \ldots + b_q x_{t-q} + \text{residual}_t \quad (4)$$

All lagged values of *x* that are individually significant are retained according to their t-statistic provided they add explanatory power to the regression. In the notation above *p* is the shortest and *q* is the longest lag length for which lagged value of *x* is significant. Lagged time of between 1 and 3 days depending on the type of data (company data) was found and used for the experiments.

3.4 Feature extraction and selection

After creating the bag of words, the features were created using unigrams and bigrams. Thus the bag of words was a collection of most frequent unigrams and bigrams. Term frequency–Inverse document frequency (Tf-Idf) was then applied for feature weighting. Tf–Idf is a numerical statistic that is intended to reflect how important a word is to a document in a collection or corpus. The occurrences of every query term appearing in each Twitter post as well as the total number of times each word appeared in the entire corpus was counted. With these counts the Tf-Idf algorithm was used to calculate the weight of each post. Thus:

$$tf_{ik} = f_{ik} \quad (5)$$

Where tf_{ik} is the term frequency weight of term *k* in a post and f_{ik} is the number of occurrences of the term *k* in the tweet.

The inverse document frequency was calculated as shown in the equation below:

$$idf_k = log\ (\frac{N}{n_k}) \quad (6)$$

Where idf_k is the inverse document frequency weight for term *k* and *N* is the number of tweets in the twitter dataset. N_k is the number of posts in which term *k* occurs. The two equations are then multiplied to get the important features.

The result was a very large feature vector and this had a potential of slowing down and overfitting when building the model. To reduce the features, Principal Component Analysis (PCA) was applied. PCA is a dimensionality reduction technique used to transform high dimensional datasets into a smaller dimensional subspace.

3.5 Prediction

The prediction model was built using support vector machines (SVM). SVM is a supervised learning algorithm that analyzes data and recognizes patterns. SVM was used because it is the mostly widely used algorithm for text analysis. SVM models require tuning 2 parameters which influence the performance of the model. Gamma and C parameter were tuned and the model resulted in slight increase in performance. C is the parameter for the soft margin cost function, which controls the influence of each individual support vector; this process involves trading error penalty for stability. Gamma can be seen as the inverse of the radius of influence of samples selected by the model as support vectors. To get these 2 parameters cross validation was applied until optimum values were obtained. To train the learning algorithm, data from January 2015 to May 2015 was used. The selected features were a combination of unigrams and bigrams whilst the target value was the movement of stocks that is 1 for upward movement and -1 for downward movement. For testing, tweets for the month of June were used. The accuracy of the model varied with each company dataset as shown in section 4 below.

4. Experimental Evaluation and Analysis

Model performance was measured based on confusion matrix performance measures of sensitivity and specificity. The accuracy was calculated using the formula below:

$$Accuracy = \frac{TP+TN}{(TN+TP+FN+FP)} \quad (7)$$

where: *TP*-True Positives, *TN*- True Negatives, *FN* –False Negatives, *FP* –False Positives

Table 1 Accuracy on the data sets after applying PCA

Name	Tweets in train set	Tweets in test set	Trainset accuracy	Test set accuracy
Airtel	88815	27295	0.77	0.54
TCS	46050	10795	0.98	0.57
Infosys	30280	10300	0.96	0.59

The accuracy scores for the three datasets were in the fifties as shown above. The model's accuracy was able to beat the baseline accuracy of random walk through which is about 50% (random walk through states that stock prediction is by chance).

4.1 Confusion Matrix

A confusion matrix is a useful tool for analyzing how well a classifier can recognize tuples of different classes. It contains information about actual and predicted classifications done by a classification model [10],[11].

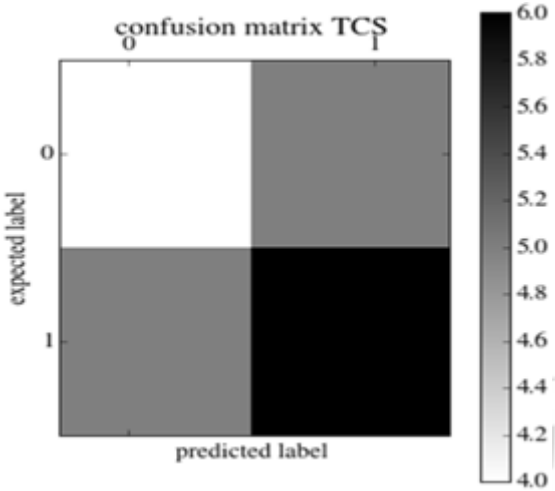

Figure 2: Confusion matrix for TCS

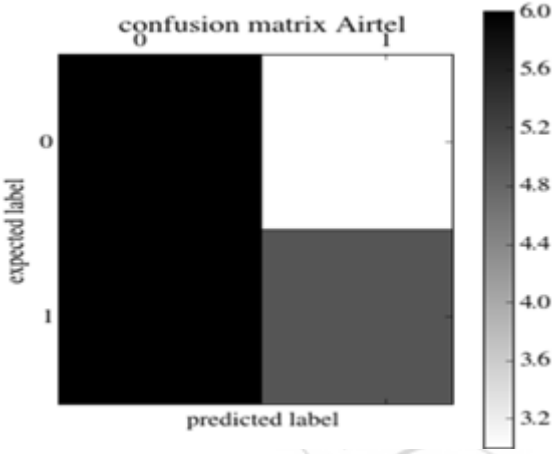

Figure 3: Confusion matrix for Airtel

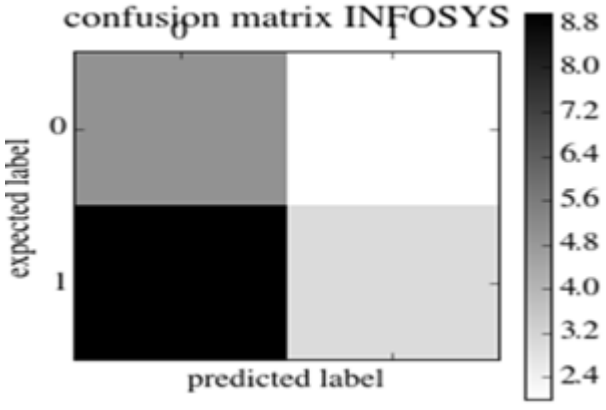

Figure 4: Confusion matrix for Infosys

Looking at the different confusion matrixes it can be noted that in some instances number of true positives and true negatives is way more than false positives thus although we are making wrong predictions in some instances, the rate of true predictions outweighs that of false predictions. Results for Infosys data showed more false negatives which is better than predicting false positives since less money is lost using our model. The difference in the accuracy of this model was due to the random variations across the different datasets. Due to these random variations, we got both a high sensitivity and specificity on some data sets while other data sets resulted in low values as shown by the different confusion matrixes. Sensitivity and Specificity were calculated as shown in Equations 7 and 8 respectively.

$$Sensitivity = \frac{TP}{TP+FN} \quad (7)$$

$$Specificity = \frac{TN}{TN+FP} \quad (8)$$

5. Challenges

Getting a large dataset was a problem because twitter has a limit on the number of tweets which an individual can get for free. At the time of writing this paper the number was 3500.

6. Conclusions and Future Work

In this paper a model to predict stock movement based on analysis of tweets for the Indian market was built. The results obtained were a slight improvement on the baseline model which is based on chance. Due to the random variations of the three datasets the accuracy for each dataset was also slightly different. The accuracy could have been low due to the small datasets used.

For future work, the accuracy can be improved by using methods that capture the context of the entire tweet as opposed to the bag of words used here. Also getting a bigger dataset will improve the model and combining stock data and twitter sentiment may also improve the prediction.

References

[1] J. Bollen, H. Mao, and X. Zeng, "Twitter mood predicts the stock market," *J. Comput. Sci.*, vol. 2, no. 1, pp. 1–8, 2011.
[2] "What is Twitter?," 2009. [Online]. Available: http://www.socialmediaoracle.com. [Accessed: 10-Aug-2015].
[3] B. G. Malkiel, "The efficiente market hypothesis and its critics," *J. Econ. Perspect.*, vol. 17, no. 1, pp. 59–82, 2003.
[4] D. Tayal and S. Komaragiri, "Comparative Analysis of the Impact of Blogging and Micro-blogging on Market Performance," *Int. J.*, vol. 1, no. 3, pp. 176–182, 2009.
[5] R. P. Schumaker and H. Chen, "Textual analysis of stock market prediction using breaking financial news," *ACM Trans. Inf. Syst.*, vol. 27, no. 2, pp. 1–19, 2009.
[6] R. Chen and M. Lazer, "Sentiment Analysis of Twitter Feeds for the Prediction of Stock Market Movement," *Cs 229*, pp. 1–5, 2011.
[7] L. Zhang, "Sentiment Analysis on Twitter with Stock Price and Significant Keyword Correlation," pp. 1–30, 2013.
[8] A. Go, R. Bhayani, and L. Huang, "Twitter Sentiment Classification using Distant Supervision," *Processing*, vol. 150, no. 12, pp. 1–6, 2009.

[9] D. Peramunetilleke and R. K. Wong, "Currency exchange rate forecasting from news headlines," *Aust. Comput. Sci. Commun.*, vol. 24, no. 2, pp. 131–139, 2002.
[10] K. Vijayarekha, "Classifier Performance," *SASTRA University*. [Online]. Available: http://nptel.ac.in. [Accessed: 10-Aug-2015].
[11] "Confusion Matrix," *University of Regina*, 2013. [Online]. Available: http://www2.cs.uregina.ca. [Accessed: 10-Aug-2015].

Author Profile

Phillip Tichaona Sumbureru received a BSC Honors in Computer Science from Midlands State University, Zimbabwe in 2011. He is currently pursuing an MTech in Information Technology at Jawaharlal Nehru Technological University, Hyderabad, India. His research areas include Big Data Analytics, Machine Learning and Natural Language Processing.