Stock Price Prediction Leveraging Reddit: The Role of Trust Filter and Sliding Window

Dennis Huynh
Department of Electrical & Computer Engineering
Queen's University
Kingston, Canada
dennis.huynh@queensu.ca

Garrett Audet
Faculty of Commerce
Faculty of Arts & Sciences
Queen's University
Kingston, Canada
garrett.audet@queensu.ca

Nikolay Alabi
Faculty of Arts & Sciences
Queen's University
Kingston, Canada
nikolay.alabi@queensu.ca

Yuan Tian
School of Computing
Queen's University
Kingston, Canada
y.tian@queensu.ca

***Abstract*—The increasing attraction of high-risk investments for retail investors has fuelled speculation across social media platforms on stocks with a perceived likelihood for disproportionate returns. This paper investigates the capacity of one such forum's (Reddit's r/WallStreetBets) speculation and the prevailing literature on how value should be maximally extracted from a community's insights. With comparative research focused on aggregating the sentiment of a community – we took an individualized approach – and sought to understand the role of forum member posts in stock price predictions. Utilizing a sentiment sliding window and trust filter enabled a 40% reduction in errors compared to not applying these techniques with the financial and Reddit data.**

***Index Terms*—Stock price forecast, trustworthiness, sliding window, social media**

I. Introduction

The volatility of the GameStop share price throughout the 2021 calendar year, particularly in January, has attracted substantial attention to the role of social media in financial markets. Most of the investors connect the dramatic ups and downs in share prices to retail investors gathering and investing via a discussion hub, the "r/WallStreetBets" (WSB) forum [1], at a social media platform named Reddit, where retail investors share opinions on stocks. After recent large fluctuations in prices and user activity on Reddit, many regulators and investors wonder what effect Reddit has on the volatility of stock market prices.

Several studies have analyzed the relationship between social media platforms and their influence on stock market movements. However, there is limited literature exploring if the historical success of a social media user adds more credibility to the likelihood of their stock prediction being accurate and if identifying the optimal window length of social media data to consider for stock analysis provides any advantage for improving predictions. There is value associated with investigating these aspects for retail investors so they can quickly navigate, identify, and focus on emerging stock opportunities. In the case that predictions improve through the selection of highly credible users, then retail investors utilizing this system can reduce risk by capitalizing on successful advisors' collective understanding. Similarly, instead of having to test different number of days or weeks in order to make a decision; knowing the optimal window of historical social media information to consider can be beneficial to the retail investor.

The impact of sentiment within financial forums influencing value perceptions of stocks is highly catalogued across studies [1]. Our assumption of this phenomenon is that investors optimism of an expected stock's financial return impact purchase decisions [2]. With various outlets commonly used by investors to facilitate stock awareness, we assume the type of media coverage across medians impact investor optimism [3], which is consequentially reflected in stock pricing. Akin to the Efficient Market Hypothesis [4] – news across medians is thus reflected in the sentiment & discussion of forums – as users utilize this information to make decisions thus demonstrating the interconnection of forum sentiment with price predictions.

Continual Deep Learning (DL) research within Natural Language Processing (NLP) has fuelled increasing interest within the intersection of these two topics and a recognition of the validity of this approach. This paper's methodology continues this focus by utilizing historical sentiment and financial data to evaluate trend interconnection. Through using both data components in unison – sentiment trends are contextualized by financial performance – enabling more robust holistic trend analysis. The main contributions of this work are:

- We propose a trust score per record and per user for Reddit data and then utilize them as a trust filter to distinguish potential reliable stock-related discussions on Reddit from others.
- We analyze if aggregating social media (reddit) data using an incremental sliding window approach affects model performance.
- We evaluate which regression model performs the best in predicting both short-term and long-term adjusted closing stock prices, with our proposed Reddit and financial features. We conduct experiments on two years' WSB data and demonstrate the benefit of considering Reddit discussions, the trust filter, and the sliding window mechanism, together with financial features, in the prediction of stock prices.

[1] https://www.reddit.com/r/wallstreetbets/

II. Related Work

In recent decades, with the growth and expansion of the stock market, more studies have been conducted to investigate stock price forecasts. These studies aim to try to analyze and predict fluctuations and price changes of the stock market with the price of an individual stock being affected by a variety of factors, such as the global economy, politics, investors' behaviors, natural disasters, etc. Consequently, stock prices are highly dynamic, non-linear, and noisy, making them very difficult to predict.

Li, Bu & Wu [5] described past stock market studies based on Random Walk Theory and the Efficient Market Hypothesis. These dated studies believed that stock price fluctuations were random, so it cannot be predicted. According to the efficient market hypothesis, stock prices are driven by information, rather than only past prices. Since news is unpredictable - stock prices are highly volatile to predict - consequentially resulting in stocks following a random walk pattern. Thus, these studies proposed that it is difficult to forecast the stock price with an accuracy above 50%. Despite this, several studies in recent years have demonstrated that the price of the stock market is not random and could be forecast to an extent.

With the development of natural language processing and TSA, some recent studies have analyzed non-structure stock related information to improve forecast accuracy. This information includes financial news, or posts on social networking sites. Bustos & Pomares-Quimbaya [6] and Nelson, Pereira & de Oliveira [7] have demonstrated the performance of using SVM, Random Forests, Neural Networks, and other models on this type of information for stock market forecasts, achieving a Mean Squared Error (MSE) of 0.0000253%. Moreover, using this additional type of information offers different aspects that can be leveraged to improve predictions. Jui-Sheng Chou & Thi-Kha Nguyen [8] use sliding-window meta-heuristic optimization for the purpose of predicting the stock prices of Taiwan construction companies.

Liu et al. [9] used RNN to predict stock volatility, however RNN will have the problem of gradient disappearance and explosion with multiple recursions. An LSTM has been developed to strengthen the operation of the RNN in the artificial intelligence fields. Gao, Chai & Liu [10] collected the historical trading data of the Standard & Poor's 500 (S&P 500) from the stock market in the past 20 days as input variables, they were opening price, closing price, highest price, lowest price, adjusted price and transaction volume. They used an LSTM as the prediction model, and then evaluated the performance by Mean Absolute Error (MAE), Root Mean Square Error (RMSE), Mean Absolute Error Rate (MSE), and Mean Absolute Percentage Error (MAPE). Their method was better than other prediction ones. Khare et al. [11] also found that LSTM can successfully predict the ups and downs of stock prices. Li, Bu & Wu [5] used an LSTM model to predict the China Securities Index 300 (CSI 300) by inputting the opening price, closing price and trading volume of the past ten days, achieving an accuracy of 78.57%.

III. Methodology

A. Overview of the Approach

Figure 1 provides an overview of our proposed approach. We model the stock prediction as a traditional regression model. Given any submission/comment mentioning a stock, we aim to predict the adjusted closing prices from the next day to the 20th day after the date of Reddit post, and the minimum and maximum from 1 to 4 open market weeks.

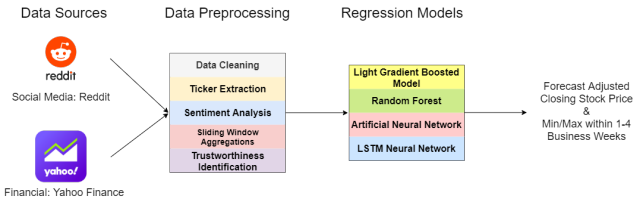

Fig. 1. The data flow of our proposed method to forecast stock price using historical stock transaction information and sentiments from Reddit

First, we collect all submissions and top-level comments (a top level is a comment that is a reply to a submission, rather than a reply to another comment) from the WSB forum that were posted during a specific time range. We then clean and preprocess the collected data to identify a set of submissions and comments that mention at least one stock in the content. We consider each mention of a particular stock in the submission/comment as a *record* (i.e., one data point in the regression model). Multidimensional information on the mentioned stock in each record, including the adjusted closing prices of the stock in the previous days and future days, statistics of the submission/comment containing the post, were extracted. Next, we apply sentiment analysis on the submission/comment associated with each record to get a sentiment score. We then create a set of Reddit features leveraging submissions/comments discussing the same stock in the past days and design a sliding window mechanism to determine how many days should be considered to represent the group sentiment on the mentioned stock for the price prediction task. Besides the sliding window, we also propose a trust filter to identify trustful discussions on stocks. Once collected all features, we apply state-of-the-art regression models and analyze the performance of various models in various setups to explore the role of the introduced trust filter and sliding window mechanism on the performance of regression models. We describe the details of the main steps in the following subsections.

B. Data Collection

The first data source is WSB, a public forum dedicated to the discussion of high-risk trading of equities (stocks). This subreddit proves user submissions and approval ratings in the form of likes and additional submission awards. Python Reddit API Wrapper (PRAW) and Pushshift API were used to collect information on all posts and first-level comments that were posted from January 1st, 2019 to December 31st, 2020. There are two types of posts on Reddit: submissions

and comments. The collected information for submissions includes: date of post, submission id, number of comments, score (number of likes), its text, title, upvote ratio (number of likes to sum of likes and dislikes), total awards, author comment karma (a metric defined and calculated by Reddit to reflect the contributions of each user), author id, and author name. Similarly, the collection information for comments are: date of post, comment id, number of replies, score, its text, total awards, author comment karma, author id, author name, and its submission id.

The second source is Yahoo Finance [2], a free public database which contains a stock's current and historical information. Given a specific stock (i.e., a stock mentioned in the collected Reddit data), all financial data, including its open, high, low, closing, adjusted closing prices, and volume were collected using Yahoo Finance API. The financial data ranges from 20 business days before the date of the Reddit post to 20 business days after the date of posting.

C. Preprocessing

We remove any rows that contained NA, removed, deleted, or banned in either the title or text columns. Text fields were then cleaned using the NLP spaCy package to remove emojis, links, and split sentences to tokens.

Next, we extract the company name or company ticker from the text in four steps. In the first step, we collect tickers and associated company names from NASDAQ. In step 2, we apply Named Entity Recognition using spaCy to identify candidate company names appearing in the collected posts and then search the tickers by name in the information collected in step 1. In step 3, we search for the appearance of tickers collected from NASDAQ in the text. In the final step, for comments that do not contain a ticker, we assign the ticker from the submission it belongs to. For instance, if a reply to a collected submission mentions a stock, the comment will also be associated with the identified stock in the submission. After the text of each post was cleaned using spaCy, the sentiment score of each post was obtained using VADER [3].

D. A Record Trust Filter

Online social media and forums like WSB are freely and publicly available, i.e., not all discussions on stock are trustworthy and the official websites do not employ any mechanism to identify trustworthy and credible content from others. In this work, we propose a novel trust filter to distinguish trustworthy and untrustworthy records. We assume that models trained from those trustworthy records will perform better than trained from all records.

We define two trust scores, one for each record and the other for each author of the identified records. Each trust score is a real number between 0 and 1 that measures the trustworthiness of a record and a user. Higher trust scores indicate higher levels of trustworthiness. The record trust score is either 0 or 1, which depends on the agreement of the sentiment expressed in the post/comment and the general trend (rising, no change, or declining) learned from the historical adjusted closing prices of the stock mentioned in the post/comment. Our hypothesis is that if the sentiment (positive, negative) expressed in the post/comment aligns with the trend predicted from adjusted closing prices, the post/comment may be credible as it follows basic finance patterns. Algorithm 1 shows how the trust score is calculated given a record.

Algorithm 1 Calculating the trust score of a record

```
for each record do
    if (rd_sentScore ≥ 0.5 and rd_slope ≥ 0.5) or
    (rd_sentScore ≤ −0.5 and rd_slope ≤ −0.5) or
    ((−0.5 < rd_sentScore < 0.5) and (−0.5 < rd_slope <
    0.5)) then
        record_trust ← 1
    else
        record_trust ← 0
    end if
end for
```

In Algorithm 1, $rd\_sentScore$ refers to the sentiment score of a post/comment. The slope rd_slope is calculated by a linear regression of the previous 20 days historical adjusted closing prices of the mentioned stock, which represents the general trend of price movement. A trust score 1 will be assigned to a record if the sentiment score $rd\_sentScore$ and slope (rd_slope) satisfy the condition of being the same sign and pass the threshold condition of ± 0.5.

For an author to be considered trustworthy, we calculate an average trust score by taking the the sum of his/her comment trust scores divided by the number of posts made by the author. Note that we ignore any user who has posted only once because one record might randomly align/not align with financial trends.

Given the above two introduced trust scores, our trust filter is defined as: if the post is trustworthy (a post trustworthy score of 1), and the author trustworthy score is greater than or equal to 0.8, keep the record; otherwise, filter it out of the data set. The threshold is 0.8 as 80% is generally regarded as above average [12], though the threshold can be lowered or increased.

E. Sliding Window

Identifying the optimal window length of social media sentiment to consider for stock analysis provides an advantage for improving predictions. Using a sliding window, we can determine the effect each window size has on the prediction performance and identify the optimal sizes for short-term and long-term stock predictions. The sliding window is a technique that aggregates values over a continuous sequence. In this case, we apply the sliding window technique to aggregate the Reddit features, namely, number of comments, score, total awards, and sentiment score. We try varying window sizes from 1 to 20 days of the date of Reddit post and aggregate using the mean function. Mean is chosen as the aggregation function

[2]https://python-yahoofinance.readthedocs.io/en/latest/api.html

[3]https://github.com/cjhutto/vaderSentiment

as it normalizes the value. For instance, setting the sliding window parameter to five means that for each record, we select all records within the past 5 days that correspond to the same stock and aggregate these records to determine the final Reddit features for that record.

F. Regression Features

Table I shows the features we propose for the regression models. As described above, we consider three types of features, i.e., financial features (FIN1-FIN20), Reddit features at the single record level (RED1-4, SENT), and aggregated features from all discussions on the same stock (SW1-SW80). Note that not all aggregated Reddit features will be used in one model. For instance, when we set the value of the sliding window size to 1, only SW1-SW4 will be considered together with other features.

TABLE I
PROPOSED REGRESSION FEATURES

Feature ID	Feature Description
FIN1-FIN20	Adjusted closing prices of the considered stock for the day 1-20 previous with respect to the data of submission/comment posting.
RED1	Number of likes received by the submission/comment. (Score)
RED2	Number of replies to the submission/comment.
RED3	Number of awards received by the submission/comment.
RED4	Author comment karma (score given by Reddit for posting and commenting).
SENT	VADER compound sentiment score of the submission/comment's text.
SW1-SW80 (4*20)	Mean of each RED1, RED2, RED3, and SENT feature for window sizes from day of to 20 previous days with respect to the date of submission/comment posting

G. Regression Models

We use four prediction models: Random Forest (RF), Light Gradient Boosted Model (LGBM), Artificial Neural Network (NN), and Long Short-Term Memory Neural Network (LSTM). The RF was chosen as the baseline due to a recent study of models in predicting stock prices with social media, news, and financial data where it had the best performance (85% F1 Measure) out of 13 models [13]. In addition to the RF, GBM also performed well with the data (84% F1 Measure) [13]. In terms of which GBM to use, LightGBM was chosen due to computational speed and ability to handle large data sets better than XGBoost [14]. An ANN stood out in terms of deep learning models as ANNs are the preferred stock price prediction tool compared with other methods [15]. Another deep learning model chosen is the LSTM model, as the stock prices data is a time-series; LSTMs are widely used for NLP and time series forecasting [10]. Hence, LSTM models are prevalent in the domain of stock market analysis.

IV. EXPERIMENTS AND RESULTS

Our experiments are designed to answer the following questions:

- Which regression model performs the best?
- How can the Reddit features and financial features contribute to the performance of regression models?
- How would the sliding window design affect the performance of the regression models?
- How would the trust filter affect the performance of the regression models?

To answer the above questions, for each regression model, we explore the following cases for the x-inputs:

1) Financial attributes only (fin)
2) Reddit attributes only (red)
3) Reddit and financial attributes (redFin)
4) Reddit and financial attributes after the trust filter is applied (redFinT)
5) Reddit attributes aggregated for sliding window sizes of 1-20 with financial attributes (sw#)
6) Reddit attributes aggregated for sliding window sizes of 1-20 with financial attributes after the trust filter is applied (sw#T)

This totals to 44 (4+20+20) trials per model. Note that the red case did not apply to the LSTM model as LSTMs are typically used for sequential data and the Reddit-related attributes are not related in a sequential order.

To simplify the comparison of the results, only certain cases were chosen from each model. The common cases selected are fin, redFin, and redFinT when comparing adjusted closing stock prices; red is omitted as it consistently performed with the highest error (> 490). The common cases selected are fin, red, redFin, and redFinT when comparing the minimum and maximum within 1-4 open market weeks. The sw# and sw#T are selected based on the window size with the best performance per model per metric. Thus, sw# and sw#T are different window sizes as well as different per model.

A. Experiment Setup

The original WSB data set contained approximately 1 million posts and approximately 3.5 million comments. After data preprocessing (ref. Section III-C), we identified 2,371 unique stocks mentioned in the collected WSB posts/comments. Financial attributes defined in Table I are then collected for the identified stocks. Basic statistics of the final WSB data set are shown in Table II.

To evaluate the performance of the regression models, Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) are used. We further calculate the percent change of Reddit and financial data, application of trust filter, best sliding window size, and best sliding window size combined with a trust filter to just financial data per respective model.
Percent change is defined as the following:

$$Percent\,Change = \frac{(final\_value - initial\_value)}{initial\_value} \quad (1)$$

TABLE II
DATA SET STATISTICS

	All	With Trust Filter
# of Records	**940,785**	**36,578**
# of Unique Posts	484,171	15,577
# of Submissions	**40,882**	**819**
# of Unique Submissions	18,310	545
# of Comments	**899,903**	**35,759**
# of Unique Comments	465,861	15,032
# of Unique Authors	**67,092**	**8,930**
# of Unique Stocks	2,371	1,421

The models we will examine will be the RF and LSTM as RF has the best results for short-term movement and LSTM has the best results for long-term movement when considering RMSE. The model we will examine will be the RF as RF has the best results for short- and long-term movement when considering MAE. The model we will examine for predicting minimums and maximums is the NN as it has the best results.

B. Results and Analysis

To recognize the impacts of the sliding window technique, trust filters, and sliding window technique coupled with a trust filter, we compare the percent change in RMSE and MAE of redFin, redFinT, sw# and sw#T cases to fin from the best models per predicted future day.

As shown in Figure 2 and 3, the performance of models decreases as the models try to predict the adjusted closing price for days farther into the future. It is also apparent that the sliding window improves the performance for all considered models in comparison to financial only, especially when predicting later days with respect to RMSE. Similarly, when analyzing the MAE results, using the sliding window only slightly improves the performance for the RF in comparison to financial only. Much more noticeable is the increase in performance by using a trust filter for both RMSE and MAE.

Among all considered models, we find that RF and LSTM models consistently perform better than others. Specifically, the models best predict adjusted closing prices short-term (1-3 days ahead), where the RF dominates predicting up to and including the 9th day into the future, and then the best performance comes from the LSTM model onward when measuring by RMSE. With respect to MAE, the RF dominates predicting the adjusted closing price both short-term and long-term. We present the comparison among different settings for the best performing models in Table III-IV. It is apparent that each case improves the prediction from those without sliding window design and trust filter. The most notable results are when the trust filtered is applied to the data. When considering RMSE as the metric, the RF does not need a trust filter to be coupled with any sliding window aggregations, but for the LSTM, it does. However, when considering MAE as the metric, the best results are from the trust filter combined with a sliding window; albeit, the difference between using a sliding window aggregation and not is very minor.

TABLE III
PERCENT CHANGE FOR RMSE VALUES FOR SELECT CASES COMPARED TO FINANCIAL DATA

Future Day	Model	redFin	redFinT	*sw#	**sw#T
1	RF	-9.07E-03	-34.49	-1.22	-33.67
2	RF	-4.3E-02	-37.28	-1.76	-36.78
3	RF	-1.1E-01	-38.65	-3.09	-38.94
4	RF	-9.3E-02	-37.70	-3.14	-38.04
5	RF	-8.2E-02	-37.44	-3.10	-37.93
6	RF	-8.0E-02	-36.90	-3.16	-37.10
7	RF	-7.4E-02	-40.77	-2.97	-40.66
8	RF	-6.8E-02	-40.36	-2.70	-40.40
9	RF	-6.5E-02	-41.22	-2.73	-40.99
10	LSTM	-11.20	-38.96	-6.75	-50.35
11	LSTM	-11.87	-38.84	-10.22	-50.48
12	LSTM	-10.98	-33.23	-13.27	-51.98
13	LSTM	-11.08	-31.33	-13.61	-52.39
14	LSTM	-12.81	-40.90	-15.62	-55.62
15	LSTM	-14.24	-39.90	-17.01	-56.29
16	LSTM	-13.01	-42.43	-14.96	-55.53
17	LSTM	-13.57	-44.56	-15.28	-58.49
18	LSTM	-12.86	-42.26	-19.73	-57.68
19	LSTM	-16.90	-42.15	-17.24	-54.31
20	LSTM	-11.65	-39.48	-18.34	-57.27

*Note: Best sw# for RF is 18 and for LSTM is 8
**Note: Best sw#T for RF is 9 and for LSTM is 19

TABLE IV
PERCENT CHANGE FOR MAE VALUES FOR SELECT CASES COMPARED TO FINANCIAL DATA FOR RF

Future Day	redFin	redFinT	sw18	sw9T
1	0.17	-39.30	1.11	-39.66
2	0.09	-40.12	0.53	-40.61
3	0.01	-40.33	-0.23	-41.23
4	0.02	-39.85	-0.39	-40.46
5	0.01	-39.80	-0.49	-40.45
6	0.02	-40.20	-0.72	-40.46
7	0.06	-42.48	-0.25	-42.65
8	0.06	-42.34	-0.27	-42.66
9	0.04	-43.44	2.55E-03	-43.26
10	0.02	-42.11	-0.62	-42.11
11	0.03	-40.42	-0.23	-40.46
12	0.03	-40.70	-0.61	-41.09
13	0.01	-41.00	-1.17	-41.67
14	0.06	-41.30	-0.34	-41.61
15	0.03	-43.60	-0.30	-43.78
16	0.02	-44.23	-0.39	-43.93
17	0.01	-45.75	-0.57	-45.51
18	0.01	-45.32	-0.89	-45.35
19	0.04	-44.23	-1.23	-44.65
20	0.02	-46.00	-1.47	-45.87

Experiments are also conducted to determine the optimal methodology for predicting minimum and maximum stock closing dates. Our main observations include:

- The best model for predicting minimum and maximum regardless of business week is the neural network (NN), results of which are shown in Table V.
- The most accurate predictions unilaterally with few exceptions were through the utilization of Reddit data decoupled from trust filters, any sliding window aggregation, or incorporation of historical stock data.

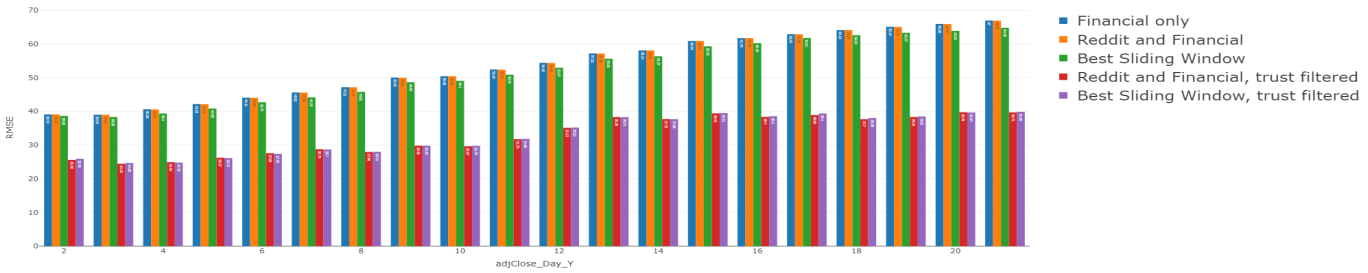

Fig. 2. RMSE of Adjusted Closing Prices Predictions, RF

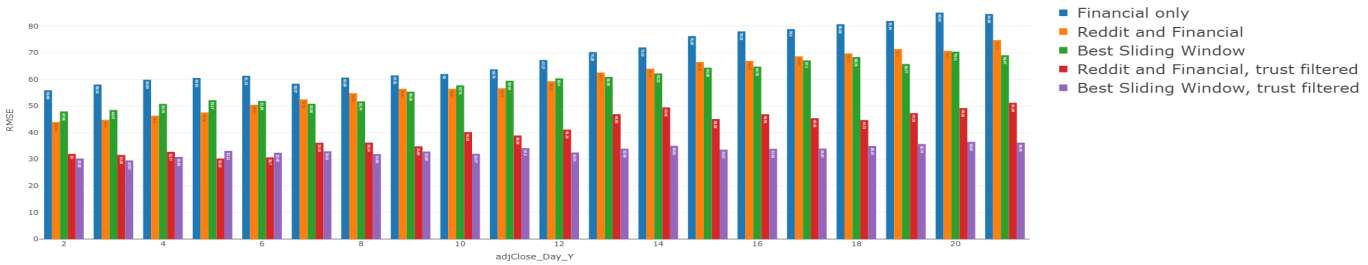

Fig. 3. RMSE of Adjusted Closing Prices Predictions,LSTM

TABLE V
RMSE VALUES FOR SELECT CASES FOR NN

Day	fin	red	redFin	redFinT	sw12	sw10T
5day_min	1.54	1.54	1.55	1.57	1.59	1.65
10day_min	3.28	3.27	3.28	3.36	3.26	3.29
15day_min	4.85	4.80	4.92	5.06	4.70	4.83
20day_min	6.47	6.33	6.53	6.72	6.82	6.81
5day_max	1.62	1.61	1.62	1.66	1.62	1.67
10day_max	3.34	3.26	3.31	3.38	3.27	3.30
15day_max	4.94	4.90	4.96	5.03	4.82	4.94
20day_max	6.59	6.46	6.56	6.36	7.34	7.23

- Predictions associated with stock purchasing decisions (minimums) were easier to predict than comparative selling decisions (maximums).
- Optimal non-filtered data generally required more information within the sliding window aggregation compared to filtered data prediction.

These insights align with the research conducted by Atkins, Niranjan, and Gerding who discerned that non-financial news better predicts stock market volatility than close price [16]. This paper predicts that this scenario can be explained as per follows:

- Social media is a distillation of retail investors interests and are primarily oriented towards purchase as opposed to sell behaviors.
- Retail investors primarily learn stocks from buying as opposed to selling and have more access to long-oriented knowledge.
- More social media data helps build consensus among community members in the absence of filtering whereas it serves as 'noise' when outdated data is considered by reputable users.

V. CONCLUSION

This paper has compared the impacts of Reddit, the sliding window technique, a credibility filter, and their combinations compared to the sole use of financial data on the regression performances of the Light Gradient Boosted Model, Random Forest, Artificial Neural Network, and Long Short-Term Memory Neural Network. We found that each case improved the RMSE of the models, where the greatest improvement to performance is with the sliding window technique and credibility filter. For the RF, when predicting 1-9 days into the future, the performance boost is 34-41%, and for the LSTM, when predicting 10-20 days into the future, the performance boost is 50-58%. Similarly, a sliding window size of 9 and credibility filter improves the MAE most notably, ranging from 39-46%, with the RF. When predicting minimums and maximums within 1-4 business weeks, we find that the NN does the best for all cases, and using only Reddit information consistently, though only a small difference, performs the best. Minimums are also easier to predict than maximums. Therefore, it can be inferred that predicting when to buy is easier than predicting when to sell.

ACKNOWLEDGMENT

We acknowledge the support of the Natural Sciences and Engineering Research Council of Canada (NSERC), [funding reference number RGPIN-2019-05071].

References

[1] F. Audrino, F. Sigrist, and D. Ballinari, "The impact of sentiment and attention measures on stock market volatility," Oct 2019.

[2] M. Buxbaum, W. Schultze, and S. Tiras, "Target price optimism, investor sentiment, and the informativeness of target prices," 05 2019.

[3] C.-H. Wu and C.-J. Lin, "The impact of media coverage on investor trading behavior and stock returns," *Pacific-Basin Finance Journal*, vol. 43, 04 2017.

[4] A. Degutis and L. Novickytė, "The efficient market hypothesis: A critical review of literature and methodology," vol. 93, 06 2014.

[5] J. Li, H. Bu, and J. Wu, "Sentiment-aware stock market prediction: A deep learning method," in *2017 International Conference on Service Systems and Service Management*, pp. 1–6, 2017.

[6] O. Bustos and A. Pomares-Quimbaya, "Stock market movement forecast: A systematic review," *Expert Systems with Applications*, vol. 156, p. 113464, 2020.

[7] D. M. Q. Nelson, A. C. M. Pereira, and R. A. de Oliveira, "Stock market's price movement prediction with lstm neural networks," in *2017 International Joint Conference on Neural Networks (IJCNN)*, pp. 1419–1426, 2017.

[8] J.-S. Chou and T.-K. Nguyen, "Forward forecast of stock price using sliding-window metaheuristic-optimized machine-learning regression," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 7, pp. 3132–3142, 2018.

[9] L. Y., Q. Z., L. P., and W. T, "Stock volatility prediction using recurrent neural networks with sentiment analysis," *Advances in Artificial Intelligence: From Theory to Practice*, vol. 10350, 2017.

[10] T. Gao, Y. Chai, and Y. Liu, "Applying long short term momory neural networks for predicting stock closing price," in *2017 8th IEEE International Conference on Software Engineering and Service Science (ICSESS)*, pp. 575–578, 2017.

[11] K. Khare, O. Darekar, P. Gupta, and V. Z. Attar, "Short term stock price prediction using deep learning," in *2017 2nd IEEE International Conference on Recent Trends in Electronics, Information Communication Technology (RTEICT)*, pp. 482–486, 2017.

[12] J. Charles E. Olson, "Is 80% accuracy good enough?," 11 2008.

[13] W. Khan, M. A. Ghazanfar, M. A. Azam, A. Karami, K. H. Alyoubi, and A. S. Alfakeeh, "Stock market prediction using machine learning classifiers and social media, news," *Journal of Ambient Intelligence and Humanized Computing*, 2020.

[14] H. H. Rebwar M. Nabi, Soran Ab. M. Saeed, "A novel approach for stock price prediction using gradient boosting machine with feature engineering (gbm-wfe)," *Kurdistan Journal of Applied Research*, vol. 5, pp. 28–48, 2020.

[15] Q. Mingyue, L. Cheng, and S. Yu, "Application of the artifical neural network in predicting the direction of stock market index," in *2016 10th International Conference on Complex, Intelligent, and Software Intensive Systems (CISIS)*, pp. 219–223, 2016.

[16] A. Atkins, M. Niranjan, and E. Gerding, "Financial news predicts stock market volatility better than close price," *The Journal of Finance and Data Science*, vol. 4, no. 2, pp. 120–137, 2018.