2021 International Conference on Computational Science and Computational Intelligence (CSCI) | 978-1-6654-5841-2/21/$31.00 ©2021 IEEE | DOI: 10.1109/CSCI54926.2021.00115

Mining the Impact of Social Media on High-Frequency Financial data

Ray R. Hashemi
Department of Computer Science
Georgia Southern University
Savannah, GA, USA
rayhashemi@gmail.com

Omid M. Ardakani
Department of Economics
Georgia Southern University
Savannah, GA, USA
oardakani@georgiasouthern.edu

Jeffrey A. Young
Department of Computer Science
Clemson University
Clemson SC, USA
alanyoung10101@gmail.com

Chanchal Tamrakar
Department of Marketing
Georgia Southern University
Savannah, GA, USA
ctamrakar@georgiasouthern.edu

***Abstract*—**Establishing the relationship between stock price changes of a fortune 500 company and events (such as political, social, and/or business) is a multi-dimensional complex problem. However, such events change the social mood, which manifests itself in social media communications. Therefore, we collected time-series high frequency financial (HFF) data alongside corresponding time-series tweets about the same company for six months in 2019. Five months of data was used to (a) mine impactful tweets (nuggets) on minute-by-minute stock price changes, (b) discover and validate the nuggets profile, (c) predict future impactful tweets prior to their effects on the stock price using the HFF data and tweets for the sixth month as a test set, and (d) maintain an up-to-date nuggets profile. The results revealed successful detection of nuggets of tweets with a certainty factor close to 80%. Such prediction may greatly affect the decisions regarding market analytics.

Keywords—*Data Mining, Social Networking, High-Frequency Financial data, Impact of Social Networking on stock market, Social Mood Index*

I. INTRODUCTION

Market analytics are driven by the economic indices, news, and social events. However, most recently two new elements of *high-frequency financial (HFF) data* and s*ocial mood* received a great deal of attention in the market analytics domain [1], [2], [3]. The *HFF data* refers to time series intraday (minute-by-minute) stock prices that is a major indicator of stock movement. The significance of *HFF data* lies on the fact that it is the basis for frequency domain inferences including volatility analysis of the market, trading volume of financial assets, stock price movement, and tracing and enforcement of regulatory standards to name a few [4].

The s*ocial mood* refers to the "collective *mood* of individuals" as a primary causal variable in financial and social trends" [5], [6], [7]. Social mood influences activities such as the number of initial public offering shares of a private corporation, merger activities, financial risk tolerance, expansion of business, and use of borrowed money in investments [8]. Widespread use of social media, such as Twitter, is one of the most effective driving forces behind the social mood, and because of that, tweets about a company (negative or positive) may affect the stock value and trade volumes related to that company [9]. Mining of impactful tweets can serve as a basis for the prediction of stock values. However, the volume of both daily tweets and HFF data about an S&P 500 company is high and, therefore, mining is quite challenging. For example, for the Disney company, which is listed in both the S&P 500 and the Dow, there are over 1.3 million tweets and over 42,000 HFF trades for the 6 months of May through October 2019 [10].

The goal of this research effort is three-fold: (a) to mine those groups of tweets that have an impact on HFF data behavior, (b) to discover and validate the profiles of the impactful groups of tweets based on their content and the strength of their impact, and (c) to predict price changes using the profiles.

The rest of the paper is structured as follows. The Methodology is covered in Section Two. The Empirical results are the subject of Section Three. The conclusion and future research are discussed in Section Four.

II. METHODOLOGY

To meet the goal, the following algorithm encapsulates all the necessary steps. The details of the four steps of the algorithm along with the necessary methodology for delivering them are the subjects of the next four sub-sections.

```
Algorithm ONE
  Input: A time-series HFF data file for a fortune 500 company,
      a time-series Tweet file for the same company and for the
      same duration that HFF data are collected.
  Output: Burst profiles, Nugget's Profile, and prediction of
      the future bursts
  Method:
    1: Identification of price-jumps (bursts) and their profiles.
    2: Identification and Extraction of the impactful groups
       (nuggets) of the tweets, based on the burst profiles.
    3: Creation and validation of the profiles for the nuggets
       based on the topics of their tweets
    4: Prediction of future bursts based on the nuggets' profile
       and maintaining the updated profiles daily.
  End.
```

978-1-6654-5841-2/21/$31.00 ©2021 IEEE
DOI 10.1109/CSCI54926.2021.00115

A: Identification of price-jumps (bursts) and their profiles.

Let us introduce several definitions that are essential to our methodology.

Definition 1: Let V = {v_1, v_2, . . ., v_n} be a set of data points for a given stock price that are recorded at times T= {t_1, t_2, . . ., t_n}, respectively. And the difference between every two adjacent time t_{i-1} and t_i is one minute (HFF data). The difference between recorded values of every two adjacent minutes is calculated, $\delta_i = v_i - v_{i-1}$ (for i= 1 to n). We assume that there is a v_0 that is equal to v_1. The magnitude and sign of δ_i are referred to as the *price change, c_i,* and the *direction of the price change* (or simply *direction*, d_i), respectively. The direction could be positive (up), negative (down), or zero (no change). The time t_i is the starting time of the c_i.

A price change, c_i, has a *strength* defined by Formula 2 in which μ and σ are the mean and the standard deviation of all price changes.

$$\begin{array}{l} \text{If } [c_i \le (\mu - 3.09\sigma)]\ OR \\ \quad [c_i \ge (\mu + 3.09\sigma)] \\ \qquad \text{then } s_i = \text{High} \\ \text{If } [(\mu + 3.09\sigma) > c_i \ge (\mu + 2.57\sigma)] \text{ OR} \\ \quad [(\mu - 3.09\sigma) < c_i \le (\mu - 2.57\sigma)] \\ \qquad \text{then } s_i = \text{Medium} \\ \text{If } [(\mu + 2.57\sigma) > c_i \ge (\mu + 1.96\sigma)] \text{ OR} \\ \quad [(\mu - 2.57\sigma) < c_i \le (\mu - 1.96\sigma)] \\ \qquad \text{then } s_i = \text{Low} \\ \text{If } [(\mu + 1.96\sigma) > c_i \ge \mu] \text{ OR} \\ \quad [(\mu - 1.96\sigma) < c_i \le \mu] \\ \qquad \text{then } s_i = \text{Trivial} \end{array} \qquad (2)$$

To explain this formula further, the strength is decided based on the *confidence interval* computed as the sample average plus-minus its variation—the width of the confidence interval changes with the standard deviation [11]. The upper bound of the confidence interval is calculated as the sample mean μ plus a margin of error $\gamma\sigma$, where γ is the critical value found from the Student's t-distribution table. The confidence level can be written as CL=100(1-α)%, where α is the significant level. For significance levels of .002, .01, and .05, the confidence levels are 99.8, 99, 95 percent, and the critical values γ are 3.091, 2.576, and 1.960. Table 1 provides confidence levels with corresponding significance levels α and critical values γ.

Table1: Confidence levels

CL	α	γ
99.8%	0.002	3.091
99.0%	0.010	2.576
95.0%	0.050	1.960

The number of consecutive differences starting from δ_{i+1} for which, directions are the same as d_i and their strengths are the same as s_i, are considered the continuation of the burst b_i. Therefore, a burst starts at time t and may stay active for several minutes.

Definition 2: A price change that is not trivial is considered as a *price burst*, b_i. A burst (for short), therefore, has a price change, direction, starting time and a strength. We also assign a hidden initiation time (τ_i) to b_i that is equal to ($\tau_i = t_i - \lambda$), where λ is the natural time delay for a tweet's effect to be observed on the stock price changes (consumption time). The length of λ for different strength levels of b_i is given in Table 2, which is obtained by the trial and error and in consultation with two experts.

Definition 3: The profile of a burst b is a quintuplet of B = (c, d, t, s, τ), where:

c: is a price change,

d: is the direction of the price change,

t: is the starting time of the price change, and

τ is the hidden initiation time.

A *burst periphery, u,* (or simply *periphery*) starts from time τ and ends at time t, [τ, t), and it inherits the strength and direction of the burst b_i.

Table 2: Range of natural delay for different burst strengths

Strength of Burst	Range of λ in Minutes
High	3
Medium	5
Low	7

B: Identification and Extraction of the Nuggets of the Tweets, based on the burst profiles

Two adjacent peripheries of u_i and u_{i+1} may be situated in four possible ways of *separated* from each other, *touched* each other, *overlapped* each other*, or* one completely *masked* the other one. The intent is to separate the overlapped peripheries and separate the masked peripheries. This is done by the peripheries' resolution as follows

Peripheries' Resolution: Let the s_i and s_{i+1} be the strengths of two adjacent peripheries of u_i and u_{i+1}, respectively.

Case 1: peripheries of u_i and u_{i+1} are *separated* or *touched*, then there is no need for further actions.

Case 2: Peripheries of u_i and u_{i+1} are *overlapped* (Figure 1.a) and

i) $s_i > s_{i+1}$, then $\tau_{i+1} = t_i$ (i.e., the periphery u_{i+1} is shortened, Figure 1.b)

ii) $s_i = s_{i+1}$, then $t_i = t_{i+1}$ (i.e., u_i is expanded and the two peripheries become one, Figure 1.c)

iii) $s_i < s_{i+1}$, then $t_i = \tau_{i+1}$(i.e., the periphery u_i is shortened, Figure 1.d).

Case 3: Periphery u_{i+1} masks the periphery u_i (Figure 2.a) and

i) $s_i > s_{i+1}$, then $\tau_{i+1} = t_i$ (i.e., the periphery u_{i+1} is shortened, Figure 2.b)

ii) $s_i \le s_{i+1}$, then dismiss the periphery u_i, Figure 2.c.

In Case 2, the assumption is that $\tau_i < \tau_{i+1}$. However, this is not always true. To explain it further, the λ value for a low burst is larger than the λ value for a higher burst, then there is a chance that $\tau_i > \tau_{i+1}$ when u_{i+1} belongs to a lower burst than u_i. By the same analogy, in Case 3, the assumption of $\tau_{i+1} < \tau_i$ is not always true. As a result, the cases of 2 and 3 grow into seven new valid cases.

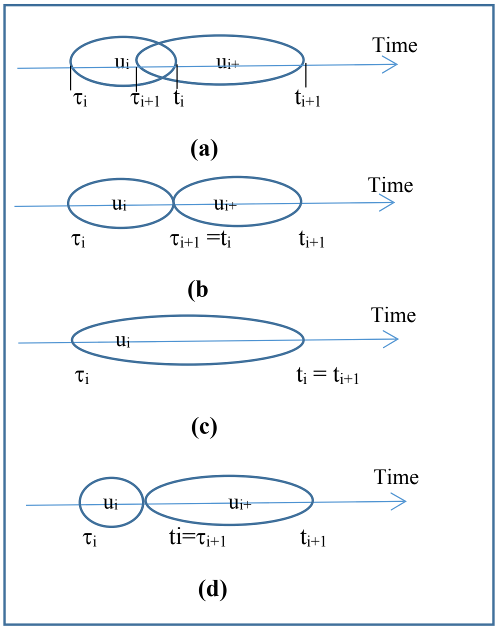

Figure 1: Overlapping: (a) Two overlaped peripheries, resolutions for: (b) $s_i > s_{i+1}$, (c) $s_i = s_{i+1}$, (d) $s_i < s_{i+1}$

C: Building the profiles of the nuggets based on the topics of their tweets and validating the profiles

All the social media messages that fall within the *resolved periphery* of burst b_i make the group of messages causing the burst b_i, and collectively referred to as the *nugget,* g_i, of the burst b_i. Therefore, a nugget for bi, includes all the messages published within the period of [τi, ti). The nugget g_i inherits the strength of the burst b_i, and it has the *nugget's length* of ABS(τi - ti), in minutes. Let us assume that these messages along with their corresponding HFF data are available for M months.

The nuggets responsible for the trivial, low, medium, and high bursts are identified using the bursts profile and extracted into four files of MonthTrivial, MonthLow, MonthMid and MonthHigh. Each file has a length in minutes which is the total nuggets length of in the file. The MonthTrivial length is even higher than the sum of the length of the other three files. As a result, any calculations and conclusions will end up in favor of the trivial bursts. To avoid such biased, nuggets in the MonthTrivial are sampled to bring its length down to, roughly, the average length of the other three files.

Sampling is done, first, by identifying the candidate nuggets. (A nugget with the length greater than μ minutes is a candidate nugget, where μ value is decided in a way that the number of candidate nuggets stay within the 8 to 10 nuggets.) Let us assume that the starting and ending times of a candidate nugget is Sc and Ec and its length is Lc. We change Sc and Ec to Sc' and Ec' such that Abs(Sc-Sc') = Abs(Ec-Eg). By doing so, we not only reduce the nugget's length, but we also stay far away from the adjacent nuggets with higher strengths housed in the other three files. The sampled file becomes the new MonthTrivial. We append MonthTrivial for all the months starting from month one and ending at month M-1 to make the file of All_Trivial. The last month is set aside for the testing of the entire system. By the same process we build the All_Low, All_Mid, and All_High files.

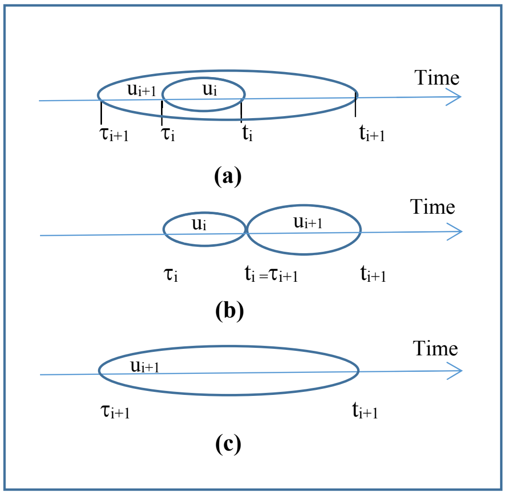

Figure 2: Masking: (a) Two masked peripheries. (b) and (c) Resolutions for (b) $s_i > s_{i+1}$, and (c) for $s_i \leq s_{i+1}$.

If we look at the All_Low, for example, there are a number of nuggets and each nugget has a starting minute (S) and an ending minute (E). This means that all the tweets (may be in tens, hundreds or more) between minutes [S-E) are the actual content of the nugget.

The tweets corresponding to the nuggets in All_Low are split into two files of All_Low_Pos and All_Low_Neg and they carry only nuggets with positive and negative directions in change of stock price, respectively. By the same process, we generate All_Mid_Pos, All_Mid_Neg, All_High_Pos, and All_High_Neg. The files of All_Trivial_Pos and All_Trivial_Neg cannot be created because the direction for the trivial bursts is zero).

For each file of *_Pos and *_Neg a pair of training and test sets are created such that the training set includes 90% of the tweets in the nugget and the test set includes the remaining 10% of the tweets. Also, a pair of training and test sets are generated from file All_Trivial.

Discovery of profiles for each one of the All_* files is a two steps process. In the first step the two algorithms of LED and PRUNE deliver a list of tokens for the file. The list goes through extra reduction to generate the profiles.

Before we introduce the algorithm LIST, some background about the *Latent Dirichlet Allocation (LDA)* [12] model is needed. This model accepts a body of documents and produces a number of possible topics based on the contents of the documents, describes each topic by a distribution of a set of tokens borrowed from the documents, and assigns a weight to each token. The number of topics and the number of tokens per topic are prescribed by the user. The outcome of this model is the three-dimensional array of: (number of topics) X (number of tokens) X 2.

The algorithm LIST uses the LDA model to generate the 3-D array for an input file of tweets, and then prune the unwanted tokens in the array. The unwanted tokens are numerated below:

i. All the emojis;
ii. All the tokens that are made up of one letter;
iii. All the common English tokens;
iv. All the tokens preceded by a hash tag;
v. Multiple copies of a token;
vi. The special tokens of {retweet, <. . .>, company name);

```
Algorithm LIST (F, N1, N2)
Input:  F is a body of documents (messages) in the training
        set of an All_*_*_Training. variables N1, N2 are the
        number of topics and the number of tokens per topic
        prescribed by the user .
Output: A 3-D array for file F.
Method:
  1. Apply the LDA model to deliver the 3-D array of Top(N1,
     N2, 2);
  2. Return pruned Top(N1, N2, 2) as a set of tokens;
End;
```

The Trivial profile is built by two following steps:
KP = LIST(All_Trivial_train);
Trivial_Profile = KP− (tokens with weight less than one);

Using the algorithm LIST, one can build the Positive and Negative profiles as follows.
KP=LIST(All_*_*_Train)
*_*_Profile = KP – (Trivial_Profile)

These profiles, however, go through one more reduction by removing all the tokens that they have in common with each other. The validity of each profile is also checked against its corresponding test set.

To validate the Low_POS_Profile, for example, the N1 topics with the distribution of N2 tokens per topic is generated for the test set of the All_Low_POS_Test, TW. Let Q be the number of common tokens between TW and Low_POS_Profile. The probability of the Low_POS_Profile being valid is calculated by formula 3.

$$P_{LowPOS}= Q/Card(\mathit{Low\_POS\_Profile}) \qquad (3)$$

If P_{LowPOS} is greater than a threshold, then the profile is valid with the *certainty factor* of P_{LowPOS}; Otherwise, profile is not valid, and it is dismissed.

To calculate the value of Q, the semantic similarity score Wu-Palmer [13] is used. Semantic similarity is a concept whereby tokens are assigned a metric based on the likeness of their meaning/semantic content (e.g., punch and hit have a high similarity score). Because these are tweets written by people, it is highly probable that two or more tweets will convey similar ideas while using different language to express them. This technique increases the accuracy of the validation. At the end of this subsection, one profile for the burst type of Trivial and a pair of profiles (Positive and Negative) for the other three bursts' type (Low, Mid, and High) are generated using the topics of the corresponding tweets.

One may ask what should be done in the case that a profile becomes empty during the pruning process or dismissed during the validation process? The answer is that by having no profile it is impossible to predict the corresponding bursts by the content of tweets. In fact, in such a case, we conclude that the corresponding burst is possibly triggered by an external event that did not have a significant effect on the social mood.

D. Prediction of future bursts based on the nuggets' profile and maintaining daily updated profiles.

Let us assume that the stock market opens at the adjusted minute of X. Let us also assume that there are three different windows of WL, WM, and WH that can contain all the published tweets for 7, 5, and 3 minutes prior to X, in accordance with Table 1. If we consider the three windows as periphery for three bursts of 1, 2, and 3 and resolve them, then the three windows contain the tweets for the adjusted times of [X-7, X-5), [X-5, X-3), and [X-3, X), respectively. For example, if the opening of the stock market happens at the adjusted minute of 30 for the first day of the month, then the three windows of WL, WM, and WH contain tweets for adjusted minutes of [23-25), [25-27), [27-30), respectively. The windows can slide on the adjusted minutes one step at a time. Therefore, for the next adjusted minute the three windows contain all the published tweets for the adjusted minutes of [24-26), [26-28), and [28-31).

We treat the tweets in the window of WL, for example, as a training set and build its profile as follows. KP=LIST(WL) and WL_Profile= KP-(Trivial_Profile). By the same process we create WM_Profile and WH_Profile and then remove from these profiles those tokens that appear in more than one of them.

The WL_Profile is checked against Low_POS_Profile, and its certainty factor is calculated using Formula 4.

$$P_{LowPOS} = [Card(WL\_Profile \cap Low\_POS\_Profile)] / Card(\mathit{Low\_POS\_Profile}) \qquad (4)$$

The certainty factor for P_{LowNEG} is also calculated and if the Max(P_{LowPOS}, P_{LowNEG}) is greater than a threshold of interest (ε), then it is kept. There is a chance that both certainty factors are dismissed, or both are eligible to be kept. In the case that both survive, they represent a contradiction that is resolved by dismissing them both.

Using the same process, one can identify the kept certainty factors for WM and WH. The highest among the certainty factor kept for WL, WM, and WH is the winner, and its corresponding burst strength and direction are the predicted burst and direction for the coming minute X. In the case of a tie, we go with the higher strength. If there is no winner, then the Trivial by default is the winner and the predicted burst for X is Trivial and direction is zero. The winner Profile is appended to a designated storage space for future use.

The designated storage space needs further explanation. There are seven designated storage spaces of STR, SLP, SLN, SMP, SMN, SHP, SHN, and STR that are initially holding the Trivial_Profile, High_POS_Profile, High_NEG_Profile, Mid_POS_Profile, Mid_NEG_Profile, Low_POS_Profile, and Low_NEG_Profile, respectively. These seven storages are used to accommodate the winning profile among WL, WM, WH profiles. Each designated space is also assigned a counter that is initialized with zero and has the same name as the storage space followed by "_K". When a designated storage space is appended by the winner profile, the counter for the storage space is incremented by one. In fact, the counter holds the number of times that the storage space is updated.

One may ask what the profile is for the case that the winning is by default (i.e., Trivial is the winner). In this case, the profile is (WL_Profile $\cup$ WM_Profile $\cup$ WH_Profile) from which tokens with weight less than one are removed.

The critical issue that needs to be addressed here is that all the profiles created out of the M-1 month need to be updated because the content of the tweets (used as the basis for creation of these profiles) constantly changing.

Each designated storage space is cleaned at the end of each Stock market working day to update the profiles. Basically, in each designated storage space, we identify and remove the tokens that the ratio of their frequency of appearance in the space in respect to the number of times that the space is used is less than or equal to a threshold of interest. We also remove the tokens that are in more than one of the designated storage spaces. The tokens in each designated area, now, serve as an updated version of the corresponding profile. The algorithm Update encapsulates the daily updating process of the profiles.

```
Algorithm Update
Input:   Designated storage spaces S* and their counters.
Output:  Updated profiles
Method:
 1. At the end of each Stock market working day do:
      Repeat for every storage (S*)
      | Repeat for each token ($tok_i$)
      |   | F= frequency of $tok_i$ in S*;
      |   | If $F/S^*\_K^* < 0.65$, Then Remove $tok_i$ from S*;
      | End;
      End;
 2. Remove from S* all tokens of ($S^* \cap$ STR);
    Remove from S* all tokens of ($St^* \cap$ with other storage spaces);
 3. Each profile is replaced by the content of its corresponding
    designated storage as its updated version;
End;
```

As an example, let the designated storage for High_POS profile initially includes the following tokens SHP={tok1, tok2, tok3, tok4} that is the same as the High Profile. The SHP at the end of the day contains { tok1, tok2, tok3, tok4, tok1, tok2, tok5, tok6, tok1, tok2, tok5, tok1, tok2, tok4, tok6, tok1, tok5, tok6, tok1, tok5, tok6} and its counter = 6. The ratio of frequency to the counter for tok1 tok2, tok3, tok4, tok5, and tok6, are 1, 0.67, 0.17, 0.34, 0.67, 0.67, respectively. At the end of step1 of the Algorithm Update, SHP contains {tok1, tok2, tok5, tok6}. Let us assume that no more tokens are removed from SHP by steps 2 and 3. The new High_POS profile is {tok1, tok2, tok5, tok6}.

III. EMPIRICAL RESULTS

The HFF data (minute-by-minute stock price) for Disney corporation (a Fortune 500 company) are collected for the six months of May, June, July, August, September, and October of 2019 in six separate files that are named after their corresponding months. Each record in these files include the minute itself and the stock price for the minute (total of 42,035 records in all 6 files) . The minutes are repeated from 0 to 59 without recording the hours and the days for the entire month. The HFF data are recorded from 9:30 a.m. to 4:00 p.m. (6 hours and 30 minutes) on working days of the stock market. Of course, the first minute of every day starts with minute number 31. Some of the very early minutes are missing for some of the months. For example, the first minute for August is 38.

We have also collected minute-by-minute published tweets about the same company for the same six months, separately (the total of 1,321,557 tweets). For each month, tweets are collected for every day and on a 24-hour cycle. For some of the days, in a given month, tweets are missing.

Six processes (P1, P2, P3, P4, P5, and P6) are applied to each HFF file. The explanation of each process is as follows.

Process P1 (Minutes adjustment). To make sure that minute m_i for hour h_j is not confused with the same minute for hour h_k, the process P1 adjusts the minutes such that the minutes become compounding. For example, adjusted minute values for the month of May starts from minute 31 and ends with minute 9261. Process P1 with some modification is also applied on tweet data to adjust their minutes. By now the recorded time in each file of published tweets and its corresponding HFF data have the same meaning.

Process P2 (Analysis of the HFF data). For every file of the HFF data, stock bursts, their profiles, and their peripheries are identified, and the process of peripheries' resolution is applied. The outcome for each HFF data file is a new file that carries the resolved peripheries, peripheries' strength, direction, and the starting and ending points in minutes for each resolved periphery.

Process P3 (Nugget extraction). Resolved peripheries for a given month (produced in process P2) are used to extract the corresponding nuggets from the tweet file of the same month. Upon the completion of the extraction of nuggets, the process P3 generates the four files of MonthTrivial, MonthLow, MonthMedium, and MonthHigh, for every month as described in the methodology. The tweets of the MonthTrivial are sampled (as explained in the methodology.) The MonthTrivial for the first 5 months are appended to make the file of All_Trivial. The process P3 also generates the All_Low, All_Mid, and All_High files.

Process P4 (Training and test Sets). The tweets in All_Low, for example, are split into two files of All_Low_POS and All_Low_NEG and they carry only tweets for low bursts with positive and negative directions, respectively. Process P4 also generates All_Mid_POS, All_Mid_NEG, All_High_POS, All_High_NEG, The Total of 6 files generated. Every one of the 6 files and the file of All_Trivial is split into a pair of training set (90% of the tweets) and test set (10% of the tweets) and the total of 14 pairs are generated.

Process P5 (Generating and validating the nuggets' profile). The profile of each training set is built and validated against its corresponding test set.

Process P6 (Building windows, predicting bursts, and updating profiles). For each adjusted minute, in HFF file of the month of October (the sixth month) the windows of WL, WM, and WH are built out of the tweet file of October for that adjusted minute. The prediction of the burst for the adjusted minute is done using the seven profiles, and the winner is the one with the maximum certainty factor. In the case of a tie, the case is considered *unpredictable*. During the prediction process, some housekeeping is also done to perform the updating of the profiles, at the end of the day, in accordance with the algorithm Update. The results of predictions are shown in Table 3. The accuracy of prediction is 80%.

IV. CONCLUSION AND FUTURE RESEARCH

Although we had over 1.3 million tweets, but we have forced to use only 114,235 of them and for the following reasons: (i) sampling of tweets for Trivial bursts was necessary, and (ii) using only those tweets that were located by peripheries of the price bursts were needed.

The worst prediction belongs to Mid_NEG (23%) and Mid_POS (61%). These predictions suggest that the number of tweets with topics unique to Mid_* are not too many. The results also revealed that the social-mood may be used as a measurable index for market analytics-driven decisions. In addition, the proposed methodology is powerful enough to (a) support a platform for mining the impact of social media in other domains such as politics, sports, arts, etc., and (b) provide a framework for identifying mal-information and disinformation that are relentlessly released in cyberspace.

As future research, the investigation of the social media influence on local election is under consideration. The goal is to identify the relationship between social media messages (fake and real) and the outcome of a specific local elections.

Table 3: Predicting bursts for the month of October using the published tweets

	High-NEG	High-POS	Low-NEG	Low-POS	Mid-NEG	Mid-POS	Trivial	Sum
High-NEG	736	53	82	48	21	43	45	1029
High-POS	39	1227	82	37	24	54	92	1555
Low-NEG	67	35	1815	72	15	43	90	2137
Low-POS	54	49	108	1248	29	57	120	1665
Mid-NEG	49	38	91	86	141	68	150	623
Mid-POS	29	64	80	43	23	527	97	863
Trivial	53	56	75	79	25	72	4173	4533
Sum	1027	1522	2334	1613	278	864	4767	12405

REFERENCES

[1] Baker, M., Wurgler, J. (2007). Investor sentiment in the stock market. Journal of Economic Perspective, 21, 129–151.

[2] Olson, Kenneth R. (2006). "A Literature Review of Social Mood." Journal of Behavioral Finance, 7, 193–203.

[3] Edmans, A., Diego G., Oyvind N. (2007). Sports Sentiment and Stock Returns, The Journal of Finance, 62, 1967–98.

[4] Aït-Sahalia, Y., Jacod, J. (2014). High-Frequency Financial Econometrics, Princeton: Princeton University Press.

[5] Hirshleifer, D., Shumway, T. (2003). Good day sunshine: Stock returns and the weather. The Journal of Finance, 58, 1009–1032.

[6] Bollen, J., Mao, H., Zeng, X. (2011). Twitter mood predicts the stock market. Journal of Computational Science, 2(1), 1–8.

[7] Pak, A., Paroubek, P. (2010). Twitter as a corpus for sentiment analysis and opinion mining (pp. 1–7). Proceedings of the Seventh International Conference on Language Resources and Evaluation.

[8] Sanjiv R. Das, Mike Y. Chen. (2007). Yahoo! for Amazon: Sentiment Extraction from Small Talk on the Web. Management Science 53(9):1375-1388.

[9] Schumaker, R. P., Chen, H. (2009). Textual analysis of stock market prediction using breaking financial news: The AZFinText system, ACM Transactions on Information Systems, 27, 1–19.

[10 Kearney, (2019). rtweet: Collecting and analyzing Twitter data. Journal of Open Source Software, 4(42), 1829.

[11] Robinson, G. K. (1975). Some Counterexamples to the Theory of Confidence Intervals. Biometrika, 62(1), 155–161.

[12] Andrew Y. Ng, Jordan M. I. (2003). Latent Dirichlet Allocation. Journal of Machine Learning Research, 3, 993-1022.

[13] Wu, Z., Palmer, M. (1994). Verb semantics and lexical selection. arXiv preprint cmp-lg/9406033