Public opinion monitoring through collective semantic analysis of tweets

Dionysios Karamouzas[1†], Ioannis Mademlis[2*†] and Ioannis Pitas[2]

[1]Department of Electrical and Computer Engineering, Aristotle University of Thessaloniki, Thessaloniki, Greece.
[2]Department of Informatics, Aristotle University of Thessaloniki, Thessaloniki, Greece.

*Corresponding author(s). E-mail(s): imademlis@csd.auth.gr;
Contributing authors: dionkara@ece.auth.gr; pitas@csd.auth.gr;
†The first two authors contributed equally and are joint first authors.

Abstract

The high popularity of Twitter renders it an excellent tool for political research, while opinion mining through semantic analysis of individual tweets has proven valuable. However, exploiting relevant scientific advances for collective analysis of Twitter messages in order to quantify general public opinion has not been explored. This paper presents such a novel, automated public opinion monitoring mechanism, consisting of a semantic descriptor that relies on Natural Language Processing (NLP) algorithms. A four-dimensional descriptor is first extracted for each tweet independently, quantifying text polarity, offensiveness, bias and figurativeness. Subsequently, it is summarized across multiple tweets, according to a desired aggregation strategy and aggregation target. This can then be exploited in various ways, such as training machine learning models for forecasting day-by-day public opinion predictions. The proposed mechanism is applied to the 2016/2020 US Presidential Elections tweet datasets and the resulting succinct public opinion descriptions are explored as a case study.

Keywords: public opinion, Twitter analysis, social media analysis, sentiment analysis, opinion mining, Deep Neural Networks

1 Introduction

Social media have gradually risen to be central elements of modern life in the Western world. The easy access to on-line platforms and the benefits of constant social interaction keep the number of users steadily growing. They allow people to directly express their opinions and feelings using a variety of media, like text, images, videos, etc.

Consequently, these platforms can be used to monitor public opinion related to a subject of particular interest. Public opinion "represents the views, desires, and wants of the majority of a population concerning a certain issue, whether political, commercial, social, or other"[1]. Twitter seems to be the medium of choice for stating opinions regarding sociopolitical matters like COVID-19, elections, racism, etc. The massive number of people utilizing Twitter for staying up-to-date and expressing their views has provided politicians with the opportunity to put their message across quickly and cheaply, without going through the traditional media briefings and news conferences [2].

The potential of Twitter regarding political events was first highlighted during the US presidential elections of 2008, where Barack Obama used the platform efficiently for his campaign [3]. After that successful Twitter campaign, all major candidates and political parties quickly established a social media presence. Moreover, the popularity of Twitter provides a unique opportunity for e-government initiatives, especially with regard to simplified communication between government institutions and citizens [4]. This may allow for greater transparency and increased citizen confidence in local institutions.

Given this very high potential of Twitter, opinion mining of tweets can provide us with valuable information. Automated semantic text analysis tools, relying on modern Artificial Intelligence (AI)-based Natural Language Processing (NLP) algorithms, can identify the mood of a tweet (e.g., polarity/sentiment [5]) with remarkable precision. Deep Neural Networks (DNNs) have greatly advanced the relevant state-of-the art, especially in *sentiment analysis*: the task of assigning a class label to a corpus of written text, where each class expresses a possible sentiment of the author concerning the content of the text. Sentiment may simply be *polarity* (ranging from very negative to very positive attitude), or multi-dimensional (identifying the presence or absence of different emotions). Additional semantic text properties that are correlated with opinion, besides sentiment, can also be identified using almost identical algorithms (for instance, *bias* or *sarcasm*).

Thus, for instance, modern AI makes it possible to get a feeling about which candidate is more likely to win the next elections by analyzing the sentiment of related tweets [6]. Moreover, opinion mining can be used to determine the public's views regarding a crucial matter, e.g., a referendum [7], and help the government take the right decisions.

However, collective and multidimensional semantic analysis of tweets, based on state-of-the-art DNNs, in order to quantify and monitor general public opinion has not been significantly explored, despite the obvious potential. Most

related methods only extract limited amounts of semantic content (typically polarity), instead of multiple semantic dimensions, while the tweets are processed individually; the outcomes are simply summarized for manual human overview. Automated collective analysis of social media-extracted content from multiple semantic aspects, in order to identify tendencies in the overall public opinion as a whole, is rather scarce.

This paper attempts to investigate the relevant unexplored possibilities of multidimensional collective tweet analysis through state-of-the-art DNNs. Thus, a novel, automated public opinion monitoring mechanism is proposed, consisting of a composite, quantitative, semantic descriptor that relies on DNN-enabled Natural Language Processing (NLP) algorithms. A four-dimensional vector, i.e., an instance of the proposed descriptor, is first extracted for each tweet independently, quantifying *text polarity, offensiveness, bias and figurativeness*. Subsequently, the computed descriptors are summarized across multiple tweets, according to a desired aggregation strategy (e.g., arithmetic mean) and aggregation target (e.g., a specific time period). This can be exploited in various ways; for example, aggregating the tweets of each days separately allows us to construct a multivariate timeseries which can be used to train a forecasting AI algorithm, for day-by-day public opinion predictions [8]. In order to evaluate the usefulness of the proposed mechanism, it has been applied to the large-scale 2016 US Presidential Elections tweet dataset. The resulting succinct public opinion descriptions are explored as a case study.

The remainder of this paper is organized in the following manner. Section 2 discusses related previous literature, focusing not on NLP or timeseries forecasting methods (which are exploited by us in a black-box manner) but on various existing mechanisms for extracting and monitoring public opinion by applying NLP and/or timeseries forecasting on social media posts. Section 3 presents the proposed semantic descriptor of public opinion. Section 4 discusses experimental evaluation on the 2016/2020 US Presidential Elections tweet datasets and the succinct public opinion descriptions that were derived using the proposed mechanism. Subsequently, Section 5 discusses key-findings on the employed datasets which were extracted using the proposed mechanism, as well as the latter's main novel contributions. Finally, Section 6 draws conclusions from the preceding presentation.

2 Related Work

This Section presents a brief overview of existing AI-based (NLP and/or DNN) approaches to semantic analysis of social media text for: a) quantitative description of public opinion, and b) timeseries forecasting.

2.1 Public opinion description

There is growing scientific interest on analyzing social media posts since the early 2010s, with Twitter dominating relevant research. For instance, [9] evaluated current public opinion regarding political advertising on Facebook, by

automatically extracting topics of discussion from relevant tweets published in October 2019. Manual inspection of these topics was conducted, leading to the conclusion that user perception of Facebook advertising is gradually decreasing, as a result of privacy concerns related to trust in the platform. Similarly, having the goal of analyzing tweets from the candidates' perspective, [10] and [11] processed the content of Donald Trump's and Hilary Clinton's tweets during the US 2016 presidential elections. The goal was to qualitatively identify which issue each candidate emphasized and what communication strategies they used.

However, this paper does not concern topic modeling and manual inspection of identified topics. Instead, it relates to methods that exploit semantic content attributes of tweets to quantify public opinion in an automatic manner. These methods can be broadly categorized into: a) non-semantic ones, which do not perform AI-based semantic analysis on tweets, and b) semantic ones, which typically perform a type of AI-enabled sentiment analysis/opinion mining on tweets.

2.1.1 Non-semantic methods

Non-semantic methods only consider keyword frequencies and/or tweet volume, resulting in rather inaccurate and/or purely qualitative insights. For instance, [12] performed a statistical analysis on tweets from the Spanish 2019 presidential campaign, which were selected based on keywords and quantified through their volume over time. The goal was to reveal political discourse of the parties engaged and highlight the main messages conveyed and their resulted impact in the share of candidates' voice. Machine learning classifiers were only used to detect spammers and the conclusions were purely qualitative. Similarly, [13] investigated social media activity during a political event, by analyzing the US 2016 GOP debate and observing the volumes of special keywords in Twitter posts.

Evidently, the mechanism proposed in this paper is entirely different in nature from these non-semantic approaches: it assigns each tweet a 4D semantic numerical vector acquired with DNN/NLP-based text classifiers, thus going a step further from conventional statistical approaches, and then aggregates these outputs for all tweets in order to construct an overall, quantitative public opinion descriptor.

2.1.2 Semantic methods without aggregation

Semantic methods are more advanced and provide more accurate results. The majority of such methods operate on Twitter, but do not treat the relevant tweets collectively and mostly just consider the volume of tweets per sentiment class. For instance, [1] presented a framework for monitoring the evolution (16 months) of public opinion related to the topic of climate change. The proposed mechanism is able to on-the-fly identify and monitor sentiment in a desired set of individual tweets, possibly as they are being published, but only rudimentary analysis is performed to these outcomes as a collection. Thus, public

opinion as a whole is scarcely considered. Having a different goal in mind, [14] conducted sentiment analysis of tweets for predicting election outcome. A simple CNN model was employed for that task, while the total volume of tweets (non-semantic attribute) was compared against sentiment polarity (volume of positive tweets) to find out which is a better predictor of election results. The semantic descriptor was found to be more accurate. Advancing on this line of research, [15] aimed to predict not only the winner but also the voting share of each candidate in the 2019 Spanish Presidential elections, by considering the volume of positive tweets per candidate. Still, the semantic attributes themselves were not aggregated over the set of all tweets.

In a similar manner, [16] performed opinion mining on Italian tweets about vaccination for a 12-month period and simply counted the number of tweets per polarity class ("against", "in favor" or "neutral") for each month. It was found that vaccine-related events influenced the distribution of polarity classes, but no aggregation of the semantic content was performed. Under an almost identical general idea, [17] aimed to determine the critical time window of public opinion concerning an event, by applying multi-emotional sentiment classification to microblog posts in Sina Weibo (published within a short time period of approximately 10 days after certain events). The volume of tweets per class (among the employed 7 emotion classes) was simply examined to find out that monitoring the negative emotions trend is crucial for predicting the influence of events.

2.1.3 Semantic methods with aggregation

In contrast to these approaches, a set of more advanced semantic methods do perform aggregation of the semantic content and thus treat the tweets collectively, by constructing a semantic public opinion descriptor in the form of a low-dimensional timeseries. This is exactly the method family to which the mechanism proposed in this paper belongs to. For instance, [18] explored the change of public sentiment in China after "Wenzhou Train Collision", by performing sentiment analysis on posts from the Sina Weibo microblogging platform, by aggregating eight identified emotions per tweet over time in order to produce an 8D daily vector, being monitored as 8 separate timeseries for a 10 day interval. However, sentiment analysis accuracy at the time was not high and the results were not particularly useful. Similarly, in [19], tweets about the 2017 Anambra State gubernatorial election in Nigeria are semantically analyzed and the outcomes are aggregated for every two-hour interval posts. The produced time-series cover an 18-hour time-frame on the election day.

Operating also in this direction, [20] employed tweet semantic analysis and aggregation to construct an average daily sentiment timeseries for each party, covering a 21-day pre-election period during the US presidential elections. Finally, in [21], similar ideas were applied to the prediction of cryptocurrency price returns through collective semantic analysis of tweets. Relevant timeseries were constructed through day-by-day aggregation of individual tweet semantic outcomes augmented with financial data, covering a period of 2 months, and a

learning model was trained for timeseries forecasting. Twitter-derived public sentiment was found to indeed have predictive power, but it was not enough on its own for accurate forecasting. Overall, methods of this type are most similar to ours, but the scale of experimental evaluation (e.g., temporal duration of constructed timeseries) is significantly limited compared to this paper. Other important differences are discussed in the following Subsection.

2.1.4 Semantic analysis dimensions and algorithms

The vast majority of the semantic methods presented above only utilize tweet sentiment, thus they can be considered as exploiting one-dimensional text semantics. This is most obvious in cases where the semantic analysis outputs a polarity (e.g., binary or ternary classification into positive/negative tweets, or into positive/neutral/negative ones). This limited approach is the most dominant one [14] [15] [20] [16] [21]. However, one-dimensional sentiment analysis can be considered to be the case even when multi-emotional classifiers are being employed instead of simple polarity. E.g., in [18] (expect, joy, love, surprise, anxiety, sorrow, angry and hate), [17] (happiness, like, sadness, disgust, astonishment, anger, and fear) and [1] (joy, inspiration, anger, discrimination, support). Although it is a more nuanced approach, these emotions still fall under the general umbrella of sentiment, thus these methods keep ignoring other semantic text attributes. The only case where multidimensional semantics are considered is [19], where 2 different opinion dimensions (polarity and bias) are both taken into account. In contrast, this paper proposes *a 4-dimensional mechanism jointly considering polarity, bias, figurativeness and offensiveness, which are all different text attributes, and experimentally verifies their usefulness.*

Another relevant aspect is how semantic tweet analysis is performed in such published methods. The vast majority among them employ outdated algorithms for text description, relying on lexicons or older representations. E.g., [18] uses the HowNet lexicon [22], [17] uses an emotional dictionary [23] and a negative word dictionary, [20] uses SentiStrength [24], [16] uses a Bag-of-Words approach, etc. Only a few rely on modern DNN-based word representation schemes, such as [17] and [1] which exploit Word2Vec [25]. The situation is even more dire when examining the type of learning models employed for actual semantic analysis. Almost all of the presented methods utilize outdated approaches, such as [17], [16] or [19], which exploit a simple K-Nearest Neighbours (KNN) classifier, a Support Vector Machine (SVM) or Textblob's Naive Bayes Classifier[1], respectively. [15] evaluates a variety of traditional (non-neural) machine learning algorithms, while [21] exploits a sentiment analysis rule set [26]. Recent DNN-based learning models were only exploited in [1] (Bidirectional Long Short-Term Memory network) and [14] (Convolutional Neural Network).

[1] https://textblob.readthedocs.io/en/dev/

Contrary to all of the above approaches, the mechanism proposed in this paper relies end-to-end on *state-of-the-art DNN solutions, both for word representation and for semantic analysis.*

2.2 Timeseries forecasting

Forecasting of timeseries derived by sentiment analysis of tweets has mainly been previously employed for predicting future financial indices. Thus, [27] explored the effect of different major events occurring during 2012–2016 on stock markets. A similar approach was followed in [28]. [29] examined the use of polarity values, extracted from tweets about the United States foreign policy and oil companies, in order to forecast the direction of weekly WTI crude oil prices. Finally, [30] investigated whether a public polarity indicator extracted from daily tweets on stock market or movie box office can indeed improve the forecasting of a related timeseries.

In all of these cases, the only exploited semantics dimension was polarity. It was shown that forecasting accuracy improves by using polarity, but public opinion extracted through tweets was only employed as an auxiliary information source for a financial-domain task, complementing a main source (e.g., stock exchange data in [27]). Moreover, outdated word representation and opinion mining algorithms were employed in all of these papers: [27] utilized SentiWordNet[2] lexicons, [29] exploited SentiStrength and the Stanford NLP Sentiment Analyzer[3], while [30] used a Naive Bayes classifier.

Similarly, older algorithms were mainly used for the timeseries forecasting task itself. [27] exploited linear regression and Support Vector Regression (SVR), [29] utilized SVM, Naïve Bayes and Multi-Layer Perceptron (MLP) learning models, while [30] relied on linear regression, MLP and SVMs to predict the target timeseries' immediate trajectory, considering the polarity or volume of tweets as input features.

In contrast to the above methods for financial forecasting exploiting tweet-derived polarity estimations, *this paper focuses on the proposed semantic public opinion descriptor itself and its potential uses for political analysis (including forecasting of public opinion).* This descriptor compactly captures *multiple opinion/semantic dimensions, instead of simply polarity.* Timeseries forecasting is utilized as an example application, among others, and *state-of-the-art DNN models are exploited during all stages, in contrast to previous methods in the literature.*

3 Proposed Mechanism

The proposed novel automated public opinion monitoring mechanism consists of a composite, quantitative, semantic descriptor that relies on NLP/DNN-based classifiers. By utilizing them, a four-dimensional vector, i.e., an instance of the proposed descriptor, is first extracted for each tweet independently, thus

[2] https://github.com/aesuli/SentiWordNet
[3] https://nlp.stanford.edu/sentiment/

Fig. 1 Quantitative public opinion forecasting using the proposed mechanism/semantic descriptor.

quantifying *text polarity, offensiveness, bias and figurativeness*. Subsequently, the computed descriptors are summarized across multiple tweets, according to a desired aggregation strategy (e.g., arithmetic mean) and aggregation target (e.g., a specific time period). The summarized descriptors can be exploited in various ways: for instance, by aggregating them on a day-by-day basis allows us to construct a multivariate timeseries which can be used to train a forecasting DNN for predicting future summarized descriptors. As an example, the pipeline of such a public opinion forecasting application/case study employing our proposed mechanism is depicted in Figure 1.

Below, the steps of computing the proposed public opinion descriptor are analyzed in detailed, along with the algorithmic machinery for implementing the process.

3.1 Step 1: Selecting the desired pool of tweets

The Twitter API allows easy and automated extraction of tweets based on manually set criteria about their topic and date. For instance, the presence of specific keywords and/or hashtags, the tweet timestamp, the fact of having been published within a desired range of dates, etc. Monitoring public opinion concerning an issue (e.g., attitude towards the incumbent party during an extended pre-election period) evidently requires smart adjustment of these criteria, so that an actually relevant corpus of user messages can be obtained. However, in general, this is extremely straightforward and simple, therefore we will not elaborate further.

3.2 Step 2: Individual descriptor extraction per tweet

The second step of the proposed mechanism is to semantically describe each tweet from the selected message pool as a 4-dimensional (4D) real-valued opinion vector. This description vector is separately extracted for each Twitter message. A set of 4 pretrained DNN models are employed to this end. Based on the state-of-the-art in NLP, two different neural architectures were employed. The hybrid CNN-LSTM from [31] was separately trained three time ex nihilo, using three different public annotated datasets for recognizing *offensiveness*, *bias* and *figurativeness* (sarcasm, irony and/or metaphor) in tweets. Additionally, a state-of-the-art pretrained, publicly available neural model[4] was employed for *polarity* recognition. In all four cases, the desired

[4] https://github.com/DheerajKumar97/US-2020-Election-Campaign-Youtube-Comments-Sentiment-Analysis

task was posed as binary text classification, with corresponding tweet labels (offensive/non-offensive, biased/non-biased, figurative/literal, negative/positive). Importantly, forcing classification to be as uncomplicated as possible (discriminating between two classes is typically easier than discriminating between multiple classes) renders the employed DNNs more robust, accurate and dependable for the actual problem tackled by the proposed mechanism, i.e., public opinion monitoring. All trained classifiers output a real value within the range $[0, 1]$ for each test tweet, with a value of $0/1$ implying perfect and uncontested assignment of one of the two opposite labels (e.g., $0/1$ means definitely and fully negative/positive sentiment, respectively, in the case of polarity).

Practical details about training the four DNNs that form the backbone of the proposed mechanism follow below.

3.2.1 Training Datasets

SemEval-2019 Task 6 sub-task A (S19-T6) [32]: This dataset contains 14,100 tweets annotated as offensive/non-offensive and was used for training the offensiveness recognition DNN.

Political Social Media Posts (PSMP) from Kaggle[5] : This dataset contains 5,000 messages from Twitter and Facebook annotated as neutral/partisan and was used to create a bias recognition DNN. The presence of Facebook messages in the dataset did not pose a problem, as they also lie in the general category of short opinionated texts, similarly to tweets.

Tweets with Sarcasm and Irony (TSI) [33]: This dataset contains approximately 76,000 tweets annotated as ironic/sarcastic/figurative/literal. In the context of this paper, the first three classes were grouped in a single class called "figurative", so that a binary figurativeness recognition DNN could be trained.

YouTube Comments (YTC): Moving on to the pretrained polarity recognition DNN, it was originally trained on a dataset with 12,559 YouTube comments. The comments were scrapped from 8 different YouTube videos related to the 2020 US presidential elections. Annotation was performed automatically via TextBlob[6] and so a positive/negative label was assigned to each comment.

3.2.2 Text Preprocessing

Identical preprocessing was applied to the three training datasets, i.e., stopwords, hashtags, mentions and URLs were removed. These entities either provide us no semantic information, or only encode information about the discussed topic. However, the proposed mechanism assumes that the topic has been manually selected by the user (in Step 1); therefore, the presented automated individual tweet descriptor relying on the four pretrained DNNs

[5] https://www.kaggle.com/crowdflower/political-social-media-posts
[6] https://textblob.readthedocs.io/en/dev/

only captures complementary semantic information, such as polarity, bias, etc. Additionally, lemmatization was applied to avoid having multiple words with identical meaning. Finally, all words were converted into lower-case. These are typical text preprocessing options in NLP.

3.2.3 Neural Models

The DNN architecture that was separately trained for offensiveness, bias and figurativeness recognition [31] consists in a hybrid, parallel BiLSTM-CNN. The input text representations (computed after preprocessing), fed to this neural architecture during both the training and test stages, are derived by using 200-dimensional embedding vectors from a pretrained GloVe model [34]. The CNN applies convolution of kernel sizes 3, 4 and 5, thereby learning fixed length features of 3-grams, 4-grams, and 5-grams, respectively. The convolution is followed by a ReLU activation function. These convolutional features are then downsampled by using a 1-D max pooling function. The CNN outputs are concatenated, combined with the BiLSTM output and jointly fed into a fully connected output neural layer, activated by a sigmoid function to produce the final semantic score: a real number in the range [0, 1].

The pretrained neural model employed for polarity recognition is based on a BiLSTM architecture. A fully-connected embedding layer is used in the front-end of the network, that has learnt to map each input word to a 200-dimensional vector representation. These embeddings are fed to a BiLSTM layer followed by a max-pooling operation. Then, multiple ReLU-activated fully-connected neural layers with dropout are employed to further reduce the dimensionality of the output. The produced dense feature representation is fed to the final sigmoid-activated fully-connected layer that gives us the final real-valued sentiment score for polarity in the range [0, 1].

Table 1 summarizes the achieved recognition accuracy (%) of each of the four opinion classifiers on the test set of the respective training dataset.

Table 1 Achieved accuracy of each of the four opinion classifiers on the test set of the respective training dataset.

Model	***Accuracy***
Bias	75.64%
Figurativeness	84.45%
Polarity	88.00%
Offensiveness	84.00%

During the test stage for all four DNN models, each pretrained DNN is actually employed for computing a different part of the individual 4D real-valued tweet descriptor for each incoming tweet. The output semantic score denotes how offensive/biased/figurative/negative the message is judged to be. An output score of 0 means very high possibility of it being non-offensive, non-biased, literal or negative, respectively. An output score of 1 means very high possibility of it being offensive, biased, figurative or positive, respectively.

A score near 0.5 would imply that the tweet is judged to be neutral in the corresponding attribute, or it simply cannot be classified.

3.2.4 Hyperparameters

The optimal hyperparameters used for training the DNN classifiers were obtained by manual tuning and are presented in Table 2. The pretrained polarity classifier used the hyperparameters found in the relevant software repository[7].

N_layers denotes the number of hidden layers in the LSTM and N_hidden is the size (nodes) of these layers. Weight_decay denotes the L2 regularization factor. Lr_decay denotes the learning rate multiplying factor. Wd_multiplier denotes the weight decay multiplying factor. Batch_size denotes the number of tweets processed at each step of the optimization. Dropout, dropout_enc and dropout_op denote the dropouts used after the LSTM, Embedding and Output layer respectively.

Table 2 Hyperparameters used for training the sentiment classifiers.

Hyperparameter	***Values***
n_hidden	200
n_layers	2
dropout	0.5
weight_decay	1e-7
dropout_enc	0.2
dropout_op	0.5
lr_decay	0.7
wd_multiplier	6
learning_rate	5e-3
batch_size	52

3.3 Step 3: Aggregation

Having obtained an individual 4D, real-valued semantic descriptor per tweet, the next step is to aggregate the derived vectors into the desired public opinion descriptor. To do this, the user has to select the **aggregation strategy** and the **aggregation target**.

The target refers to the granularity of the aggregation process and directly influences the final number of aggregate public opinion description vectors. Two possible choices are the most straightforward:

- *Complete aggregation.* All individual tweet descriptors are merged into a single aggregate public opinion description vector, representing the entire message pool extracted in Step 1.
- *Temporally segmented aggregation.* The overall range of dates out of which the entire message pool was extracted in Step 1 is partitioned in isochronous,

[7] https://github.com/DheerajKumar97/US-2020-Election-Campaign-Youtube-Comments-Sentiment-Analysis

non-overlapping and consecutive time periods. All individual tweet descriptors falling under each period are merged into a single aggregate public opinion description vector. This is separately performed for each period.

With complete aggregation, the outcome is a single 4D vector. With temporally segmented aggregation, the outcome is a 4D timeseries. Examples of temporally segmented aggregation targets would be day-by-day or week-by-week aggregation. Depending on the application, different additional aggregation targets may also be envisioned.

The aggregation strategy refers to how a set of individual 4D tweet descriptors are combined into a single aggregate 4D descriptor. Three possible choices are the most straightforward:

- *Element-wise vector mean.*
- *Element-wise vector median.*
- *Element-wise vector trimmed mean.*

All three of these choices may be implemented simply by performing computations separately along each of the four descriptor dimensions. As before, different additional aggregation strategies may also be envisioned, depending on the application.

4 Evaluation

The proposed mechanism was evaluated on the well-known 2016 and 2020 United States Presidential Election Tweets datasets, using a day-by-day temporally segmented aggregation target. All three aggregation strategies described in Section 3 were separately followed and assessed. Details and results follow in the next Subsections.

4.1 Datasets

The 2016 US Presidential Election tweet dataset from Kaggle[8] contains 61 million rows. Their overall time range is from 2016-08-30 to 2017-02-28, with 20 days missing, leading to a total of 163 days. From this initial dataset, we retained approximately 32 million tweets after applying a common cleaning process: removal of empty rows, of non-English text, of duplicate tweets and of messages that contained less than 5 words after text preprocessing. This is important for proper semantic analysis, since an adequate number of words per tweet is essential to achieving high opinion mining accuracy. Subsequently, the keywords “Clinton”, “Obama” and “Trump” were exploited for partitioning the messages into ones referring to Democrats and ones referring to Republicans.

Overall, this entire manual process was equivalent to performing Step 1 of the proposed mechanism. It was not needed in its entirety for the smaller second dataset that we employed, i.e., the US Election 2020 Tweets from

[8] https://www.kaggle.com/paulrohan2020/2016-usa-presidential-election-tweets61m-rows

Kaggle[9], since its tweets are preseparated in two partisan groups (Democrats and Republicans). It contains 1.72 million rows in total, with an overall time range from 2020-10-15 to 2020-11-08, meaning 25 days in total. From this initial dataset, approximately 720 thousand tweets were kept after applying the cleaning process described above.

Subsequently, the proposed mechanism was separately applied to the two message pools of each dataset. As a result, two day-by-day 4D descriptor timeseries were derived, covering overall the exact same time range: one for the Republicans and one for the Democrats (separately for each dataset). Indicatively, for the 2016 US Presidential Election tweet dataset, generating the descriptors for all relevant tweets required 24 hours. Day-by-day aggregation required 10 minutes for each aggregation strategy. Experiments were performed on a desktop computer with an AMD Ryzen 5@3.2GHz CPU, 16GB of DDR4 RAM and an nVidia GeForce GTX1060 (6GB RAM) GP-GPU.

4.2 Analysis 1: Timeseries Forecasting

The first type of evaluation performed on the derived timeseries was to assess their predictability using AI-enabled forecasting. Since the two constructed timeseries compactly capture public opinion about the two respective parties during a heated pre/post-election period, forecasting has obvious political usefulness: it may allow an interested organization to predict near-future changes in its public image, using only Twitter data. Of course, in this context, "near-future" implies a forecasting horizon of a few weeks at the most.

4.2.1 Implementation

The two timeseries were day-by-day 4D descriptions of evolving public opinion about the Democrats and the Republicans, respectively. However, since three different aggregation strategies were employed (mean, median, trimmed mean), in fact six 4D timeseries were derived overall, with all of them covering the same period. Since forecasting is typically performed on univariate timeseries, each descriptor channel was then handled separately, leading to a total of 24 different timeseries.

A moderate 7-day forecasting horizon was selected, since this allows for rather reliable predictions while still being practically useful. A recent stacked LSTM architecture was adopted [35], with each LSTM cell being followed by a fully connected neural layer. The overall DNN was trained separately for each timeseries, using Back-Propagation Through Time (BPTT) and a Continuous Coin Betting (COCOB) optimizer [36]. This training approach was selected over alternative optimizers (Adagrad and Adam) because it displayed superior performance in [35]. The fact that COCOB attempts to minimize the loss function by self-tuning its learning rate, accelerates convergence in comparison to other gradient descent-based algorithms with a constant or decaying learning rate, where the convergence becomes slower close to the optimum.

[9] https://www.kaggle.com/datasets/manchunhui/us-election-2020-tweets

The use of BPTT for updating model parameters during training is necessary when employing LSTMs.

Optimal hyperparameters for training the forecasting DNN model were obtained by using Sequential Model-based Algorithm Configuration (SMAC) [37] and are presented in Table 3. minibatch_size denotes the number of timeseries considered for each full backpropagation in the LSTM. Epoch_size denotes how many times the dataset is traversed within each epoch. L2_regularization and Gaussian noise added to the input are used to reduce overfitting. LSTM unit weights were initialized using a random_normal_initializer.

Table 3 Hyperparameters used for training the forecasting model.

Hyperparameter	***Value***
cell_dimension	20
gaussian_noise_stdev	1e-4
l2_regularization	1e-4
max_epoch_size	1
max_num_epochs	2
minibatch_size	4
num_hidden_layers	1
random_normal_initializer_stdev	1e-4

Out of the two evaluation datasets, only the 2016 one was used for training the forecasting model. A segment was withheld for test purposes from the end of each timeseries, with a length equal to the forecasting horizon; the remaining data constituted the training dataset. Moreover, this pretrained model was also separately tested on the 2020 dataset. The results from testing on both datasets are presented in the sequel.

Deseasonalisation was applied as a common preprocessing step, since DNNs are weak at modelling seasonality [38]. That was achieved by decomposing each timeseries into seasonal, trend, and remainder components, in order to subsequently remove the seasonality component, by employing STL decomposition. If a timeseries exhibited no seasonality, this step simply returned zero seasonality.

Sliding window schemes were adopted for feeding inputs to the DNN and deriving the outputs, with the output window size n set to be equal to the size of the forecasting horizon $H = 7$. The input window size m was empirically set to $9 = n1.25$. Each training timeseries was broken down into blocks of size $m + n$, thus forming the input–output pairs for each LSTM cell instance.

4.2.2 Metrics and Results

A set of common timeseries forecasting evaluation quantitative metrics were employed for assessing the predictability of the computed timeseries.

First, the *Symmetric Mean Absolute Percentage Error* (SMAPE) is defined as follows:

Table 4 Forecasting results on the US 2016 Presidential Election Tweets dataset for the six constructed timeseries. "Dem" denotes the Democrats, "Rep" denotes the Republicans, while "mean", "med" and "trim" imply the three respective aggregation strategies: mean, median and trimmed mean. In each case, the SMAPE/MASE metrics have been independently averaged across the four descriptor channels using both the mean and the median operator. A lower value is better for both metrics, while SMAPE is a percentage.

Timeseries	*Mean SMAPE*	*Median SMAPE*	*Mean MASE*	*Median MASE*
Dem-Mean	0.1096	0.0563	1.2396	1.0799
Dem-Med	0.1380	0.0913	1.0620	1.1711
Dem-Trim	0.1798	0.0676	1.3364	1.0885
Rep-Mean	0.0529	0.0280	0.7689	0.6937
Rep-Med	0.0492	0.0330	0.5158	0.5303
Rep-Trim	0.0737	0.0314	0.6931	0.6549

Table 5 Forecasting results on the US 2020 Presidential Election Tweets dataset for the six constructed timeseries. "Dem" denotes the Democrats, "Rep" denotes the Republicans, while "mean", "med" and "trim" imply the three respective aggregation strategies: mean, median and trimmed mean. In each case, the SMAPE/MASE metrics have been independently averaged across the four descriptor channels using both the mean and the median operator. A lower value is better for both metrics, while SMAPE is a percentage.

Timeseries	*Mean SMAPE*	*Median SMAPE*	*Mean MASE*	*Median MASE*
Dem-Mean	0.1793	0.1490	1.6975	1.6943
Dem-Med	0.3431	0.2088	1.7526	1.6620
Dem-Trim	0.2847	0.1903	1.7535	1.7434
Rep-Mean	0.0961	0.0813	1.5233	1.4572
Rep-Med	0.1867	0.1303	1.7228	1.6949
Rep-Trim	0.1472	0.0999	1.5961	1.5637

$$SMAPE = \frac{100\%}{H} \sum_{k=1}^{H} \frac{|F_k - Y_k|}{(|Y_k| + |F_k|)/2}, \tag{1}$$

where H, F_k, and Y_k indicate the size of the horizon, the forecast of the DNN and the ground-truth forecast, respectively.

Due to the low interpretability and high skewness of SMAPE [39], the scale-independent *Mean Absolute Scaled Error* (MASE) metric was also employed. For non-seasonal timeseries, it is defined as follows:

$$MASE = \frac{\frac{1}{H}\sum_{k=1}^{H}|F_k - Y_k|}{\frac{1}{T-1}\sum_{k=2}^{T}|Y_k - Y_{k-1}|}, \tag{2}$$

In Eq. (2), the numerator is the same as in SMAPE, but normalised by the average in-sample one-step naive forecast error. A MASE value greater than 1 indicates that the performance of the tested model is worse on average than the naive benchmark, while a value less than 1 denotes the opposite. Therefore, this error metric provides a direct indication of forecasting accuracy relatively to the naive benchmark.

Since these metrics are computed for univariate timeseries forecasting, we employed mean and median aggregation across the four descriptor channels for each of the six timeseries. The results obtained for the 2016 and 2020 datasets

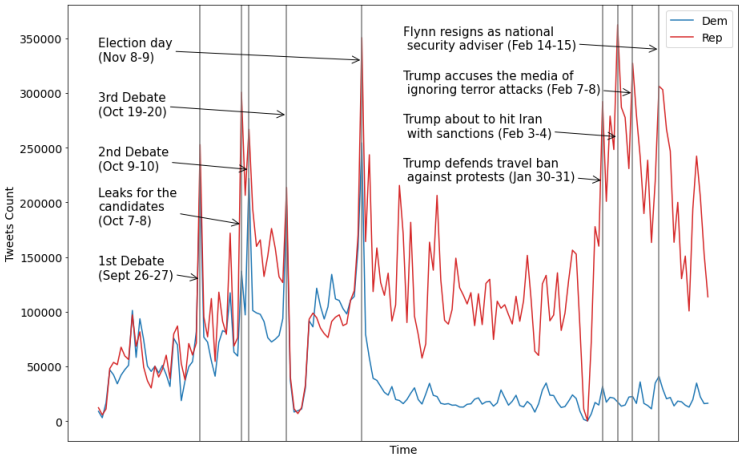

Fig. 2 Daily number of tweets for Democrats (Dems) and Republicans (Reps) in 2016 dataset. The two dates given per event are the date of that event (first) and the date of the respective reaction in Twitter (second).

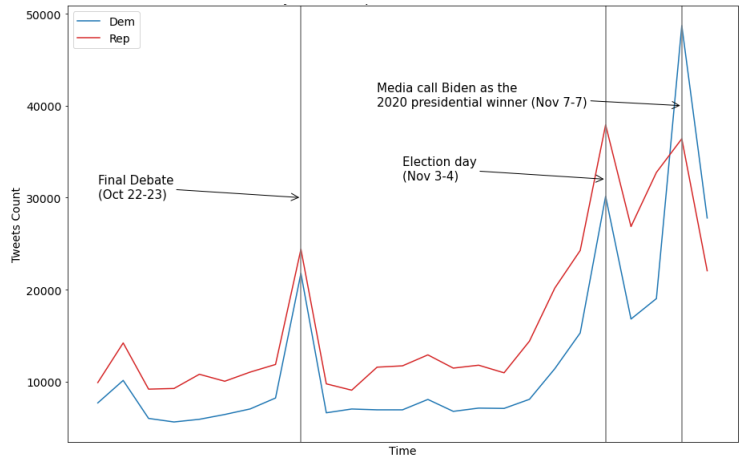

Fig. 3 Daily number of tweets for Democrats (Dems) and Republicans (Reps) in the 2020 dataset. The two dates given per event are the date of that event (first) and the date of the respective reaction in Twitter (second).

are shown in Tables 4 and 5, respectively, where larger SMAPE or MASE values indicate worse forecasting accuracy.

In general, the timeseries constructed using the proposed mechanism seem to be predictable to an acceptable degree by using the employed DNN model. Moreover, forecasting behaves similarly for both datasets, leading us to draw a common set of conclusions. First, the mean aggregation strategy resulted in the timeseries with the best overall forecasting behaviour. Second, based on both metrics, it is clear that forecasting performs worse for the Democrats than for the Republicans, implying that public opinion concerning them (as

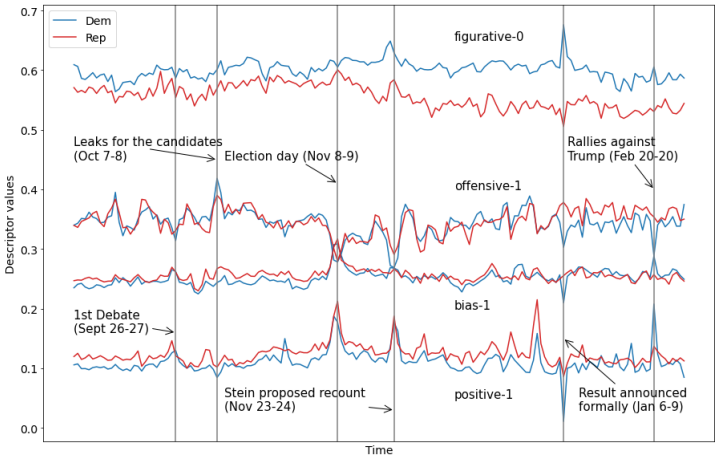

Fig. 4 Per-channel day-by-day values of the 4D timeseries constructed from the 2016 data using the proposed descriptor and the mean aggregation strategy, separately for Democrats (Dems) and Republicans (Reps). The two dates given per event are the date of that event (first) and the date of the respective reaction in Twitter (second).

expressed in Twitter) was less stable and predictable during the examined period. Finally, we notice a drop in accuracy when testing on the 2020 dataset (small in absolute terms), compared to the 2016 one. This is to be expected, since the forecasting DNN was pretrained only on the training set of the 2016 dataset.

4.3 Analysis 2: Visualizations and Qualitative Evaluation

A set of visualizations were computed from the 4D timeseries constructed using the proposed mechanism, in order to facilitate manual inspection of the outcome. Given the conclusions of Subsection 4.2, only the timeseries derived by mean aggregation were exploited here. This Subsection presents this qualitative evaluation process and its results, along with auxiliary information about the original 2016/2020 US Presidential Elections datasets.

First, Figure 2 depicts the number of tweets posted every day in the complete dataset's time range (from 2016-08-30 to 2017-02-28), separately for the Democrats and the Republicans. For the most important events like the three presidential debates and the election day, increased Twitter traffic is observed for both parties. Leaks for both Clinton and Trump that took place in 2016-10-07 seem to have affected more the latter candidate, as the majority of posts expressed an opinion about him. Equal traffic is observed for both parties just before the election day (2016-11-08), since in that stage Twitter plays a significant role in the campaign of both candidates. However, the number of tweets concerning Democrats drops significantly after the election day and their defeat. In contrast, people kept tweeting frequently about the winner and center of attention Donald Trump regarding his actions as a president of the US, like the travel ban, Iran sanctions, etc.

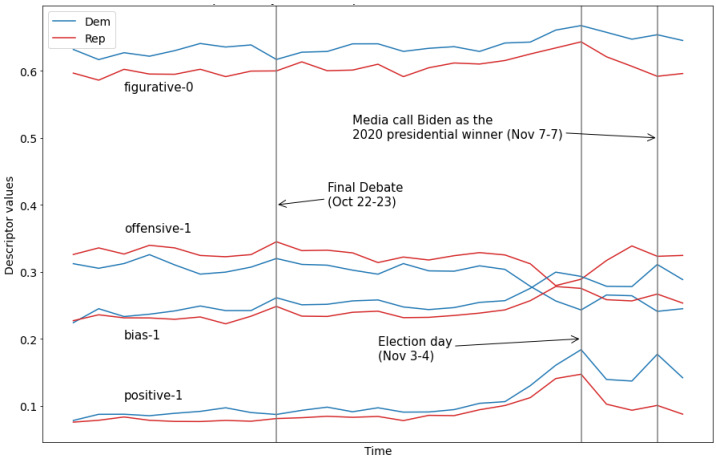

Fig. 5 Per-channel day-by-day values of the 4D timeseries constructed from the 2020 data using the proposed descriptor and the mean aggregation strategy, separately for Democrats (Dems) and Republicans (Reps). The two dates given per event are the date of that event (first) and the date of the respective reaction in Twitter (second).

Figure 3 depicts the number of tweets posted from 2020-10-15 to 2020-11-08, separately for the Democrats and the Republicans. Again, increased Twitter traffic is observed for both parties during the most important events. However less events are observed in the 2020 plot, which is due to the smaller size of the dataset as a whole, in comparison compared to the 2016 dataset. In general, there were more tweets posted about Trump up until November 7. On that day, the media called Biden as the 2020 presidential winner and Twitter traffic exploded for Biden, surpassing Trump posts by a significant margin. This obviously makes sense as Biden's victory is officially announced for the first time.

Having established original Twitter traffic patterns, concurrent daily values of the 4D timeseries constructed using the proposed descriptor and the mean aggregation strategy are depicted in Figure 4, separately for each party of the 2016 US election (party affiliation is color-coded). In this Figure, as in many following ones, we label each timeseries by one of the two opposite class labels (e.g., "positive" or "figurative") followed by the tweet classifier output value which implies this label (e.g., a tweet fully and undoubtedly classified as figurative/positive, has been assigned a value of 0/1 by the figurativeness/polarity classifier, respectively). It is evident that the timeseries maintain a stable class along all four dimensions and across the entire time range for both parties: their class is negative (in polarity), unbiased, non-offensive and literal. This indicates a general public stance towards the competing politicians, which reflects a judgemental and indignant (negative + literal) but simultaneously educated (unbiased + non-offensive) public.

Figure 5 depicts the mean daily 4D descriptor for each party of the 2020 US election. Again, the timeseries maintain the same stable classes with the

2016 data along all four dimensions and across the entire time range for both parties. An interesting conclusion that can be drawn by comparing the plots of the 2016 and the 2020 elections is that the public seems to have a rather fixed stance towards the competing parties, carried over from one election period to the next one, with the winner determined by a small margin/difference.

Moreover, by visually inspecting Figures 2, 4 and 3, 5 for the 2016 and 2020 datasets, respectively, a correlation can be identified between the occurrence of crucial events and abrupt changes (spikes) in the number of tweets or public opinion. As expected, this reaction in Twitter takes place the day after the event.

By comparing the timeseries of the two parties in figure 4, one can observe that tweets about Republicans are less negative, less unbiased and less literal, while there is no clearly distinguishable difference between the two parties concerning offensiveness. These observations shed new light to the election results of November 8th. A less negative opinion is clearly an advantage in itself for Republicans, but combining it with a more biased opinion reflects the possibility that there were more Trump's partisans active in Twitter. Given that partisans are decided voters that do not easily change their opinion, these conclusions drawn from analysing the constructed timeseries paint the picture of a significant Republican advantage, by only using public Twitter data.

Interestingly, posts about Trump appear to be less literal. Despite common perceptions that figurative language is most often used to express negative opinions, tweets about Republicans are on average less negative. A possible explanation is that figurativeness doesn't reflect on the voters' decision and is mainly being used by Twitter users to attract higher attention. Therefore, our analysis indicates that if a voter is clearly against a candidate, it is more probable for them to be straightforward in their comments.

Corresponding conclusions can be drawn from Figure 5 regarding the 2020 US elections. One can observe that the tweets about Democrats are less negative, less unbiased, more literal and less offensive. The less negative and less unbiased attributes can be interpreted as beneficial factors for the Democratic party (like in 2016). The main difference in the 2020 data are the less offensive and more literal tweets that seemingly further contribute to Democratic dominance, as figurative language usually implies negativity [40].

Principal Component Analysis (PCA) was exploited for applying dimensionality reduction to the 4D mean Republican/Democrat timeseries, so that they can be visualized in 2D plots. The 2D descriptor points per party are presented in Figures 6, 7 for the 2016 dataset and 8, 9 for the 2020 dataset. Here, outlying data points correspond to the spikes of the original time-domain plots of Figures 2, 4 and 3, 5. Thus, outliers in PCA Figures indicate the occurrence of crucial events. This visualization can help us identify incidents that significantly affect public opinion, but does not immediately provide us with corresponding information about the semantic descriptor values.

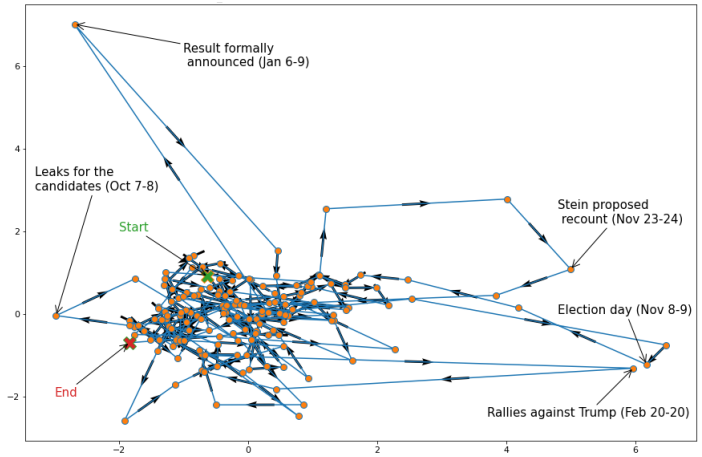

Fig. 6 PCA-based 2D visualization of the constructed 4D timeseries for the Democrats, using a mean aggregation strategy, across the entire 2016 dataset time range (163 days). The two dates given per event are the date of that event (first) and the date of the respective reaction in Twitter (second).

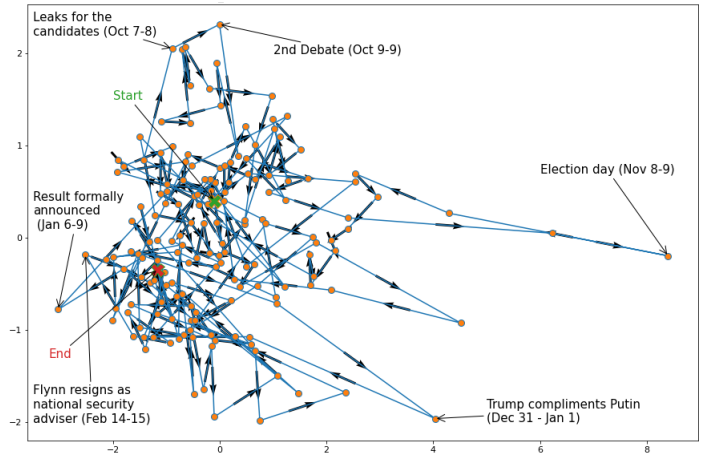

Fig. 7 PCA-based 2D visualization of the constructed 4D timeseries for the Republicans, using a mean aggregation strategy, across the entire 2016 dataset time range (163 days). The two dates given per event are the date of that event (first) and the date of the respective reaction in Twitter (second).

However, the following observations can be made based on the outliers. Regarding the 2016 data we can tell from Figures 6, 7 that the events influencing public opinion about the Democrats the most were the elections themselves, the formal announcement of the results and the leaks about the candidates. The respective events for the Republicans are identical, with the addition of the second candidate debate and the compliments made by Donald Trump on

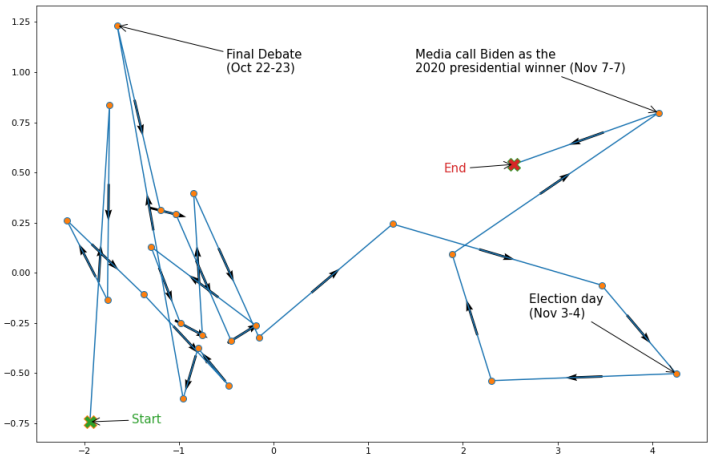

Fig. 8 PCA-based 2D visualization of the constructed 4D timeseries for the Democrats, using a mean aggregation strategy, across the entire 2020 dataset time range (25 days). The two dates given per event are the date of that event (first) and the date of the respective reaction in Twitter (second).

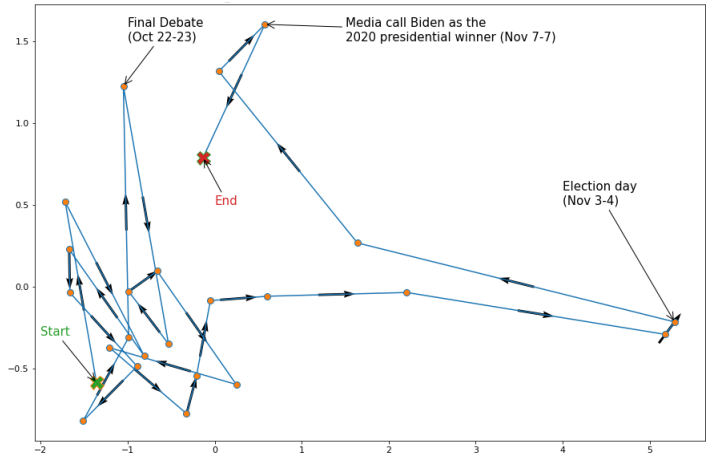

Fig. 9 PCA-based 2D visualization of the constructed 4D timeseries for the Republicans, using a mean aggregation strategy, across the entire 2020 dataset time range (25 days). The two dates given per event are the date of that event (first) and the date of the respective reaction in Twitter (second).

President Putin. This possibly reflects the increased relevance of national security concerns in the US public discourse. Regarding the 2020 data, Figures 8, 9 confirm the three crucial events indicated by the spikes in Figures 3, 5.

Given the explanatory power of specific dates shown to be semantic outliers in the constructed timeseries, a different way to exploit the proposed public opinion description mechanism was also investigated: to focus on individual salient dates. In the context of this paper and given the previously discussed

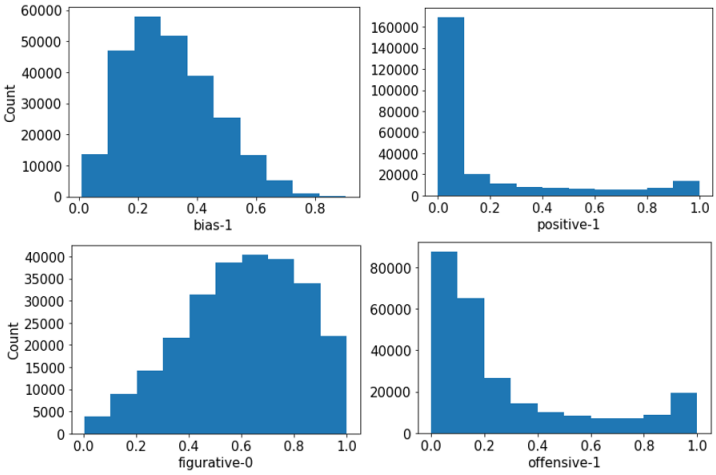

Fig. 10 Histograms of the four descriptor dimensions, depicting how the number of tweets is distributed over the DNN classifier outputs. These histograms concern the Democrats on Nov 9, 2016 (the day after election).

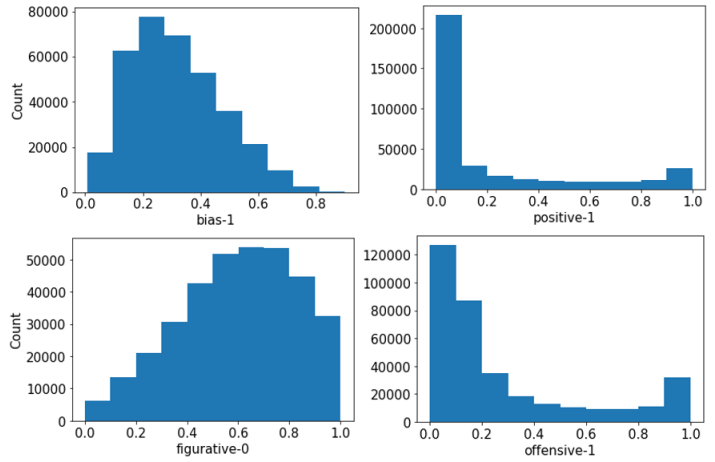

Fig. 11 Histograms of the four descriptor dimensions, depicting how the number of tweets is distributed over the DNN classifier outputs. These histograms concern the Republicans on Nov 9, 2016 (the day after election).

observations, the day after the elections (November 9, 2016 and November 4, 2020) was selected as the target date.

Figures 10 and 11 show how the tweets posted on November 9, 2016 were distributed along the four descriptor dimensions, separately for the two parties and before any aggregation strategy was applied. These 10-bin histograms show the distribution of the number of tweets (vertical axis) over the semantic values of each of the four descriptor dimensions (horizontal axis). The employed semantic values (outputted by the 4 pretrained DNN classifiers) were

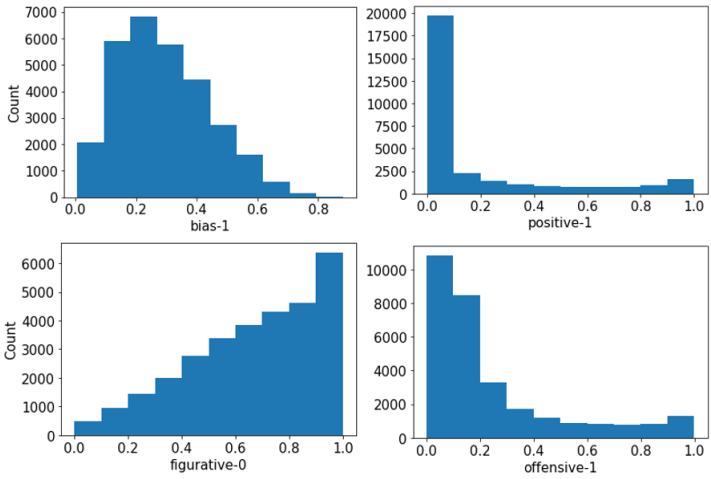

Fig. 12 Histograms of the four descriptor dimensions, depicting how the number of tweets is distributed over the DNN classifier outputs. These histograms concern the Democrats on Nov 4, 2020 (the day after the election).

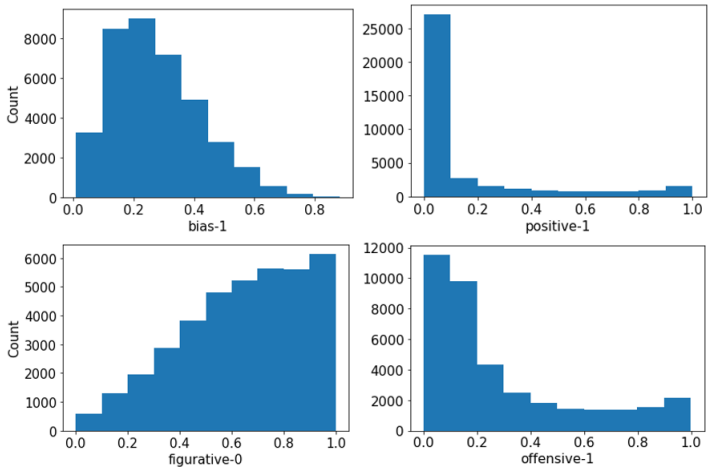

Fig. 13 Histograms of the four descriptor dimensions, depicting how the number of tweets is distributed over the DNN classifier outputs. These histograms concern the Republicans on Nov 4, 2020 (the day after the election).

real numbers in the interval [0,1]. Therefore, the horizontal axis has not been normalized in range: each of the 10 bins corresponds to a subrange of length 0.1.

The histograms are almost identical for Democrats and Republicans, an observation compatible with the behaviour captured in Figure 4. Moreover, similar histogram shapes can be discerned for the bias-figurativeness and for the polarity-offensiveness features. Bias and figurativeness have approximately shifted normal distributions, implying that the mean aggregation strategy is

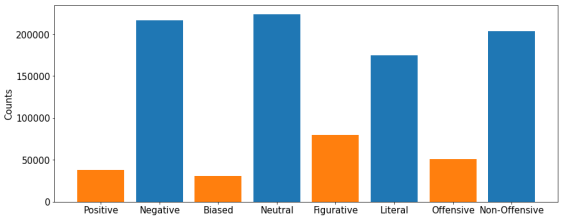

Fig. 14 Number of tweets concerning Democrats, separately for each class of the four descriptor dimensions, on November 9, 2016. The two colors distinguish between the opposite classes of each semantic dimension.

indeed a good choice during timeseries construction. In contrast, polarity and offensiveness histograms are significantly more polarized in shape, rendering the mean aggregation strategy less reliable in their case for that particular date.

The fact that the majority of tweets fall within the interval [0,0.2] for the polarity and offensiveness dimensions, means that they have been clearly classified as rather negative and non-offensive: a fully negative/positive and absolutely non-offensive/offensive tweet would be characterized by a value of 0/1 in both dimensions, respectively. These histograms paint the picture of a public that is carping and complaining in the aftermath of the elections, yet avoids the use of offensive language. Concerning the relative absence of intermediate values lying within the range [0.2,0.8], we can say that classification regarding polarity and offensiveness was straightforward and the respective models pretty confident. This implies that indeed most tweets were clearly negative or positive, as well as clearly non-offensive or offensive, without many users being neutral in these respects. In contrast, classification regarding bias and figurative attributes does not lead to such polarized results. This is because the DNN models have trouble classifying these tweets as pure instances of a specific class (e.g., the "figurative" or the "literal" class), leading to intermediate values near 0.5. This implies that most users were rather neutral with regard to these semantic dimensions. Still, we can clearly see that the majority of tweets tend to be non-biased and literal.

Similar histogram shapes are observed for the respective tweet distributions of November 4, 2020 for both Democrats 12 and Republicans 13. This was no surprise given the similarity of Figures 4 and 5. However, there is a noticeable difference in the figurative elements of 2016 and 2020, where the distribution is shifted right, indicating more literal language used on that day's tweets. This can be confirmed by comparing Figures 4 and 5.

Finally, Figures 14, 15 and 16, 17 depict the number of tweets per party, separately for each class of the four descriptor dimensions, on the election days November 9, 2016 and November 4, 2020. This visualization provides a glimpse to the non-dominant classes that disappeared when constructing the timeseries using the mean aggregation strategy.

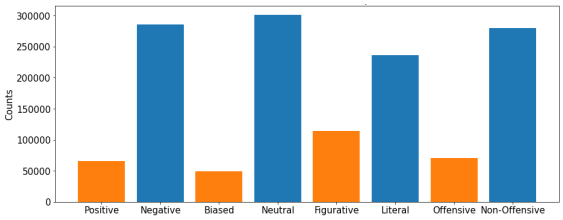

Fig. 15 Number of tweets concerning Republicans, separately for each class of the four descriptor dimensions, on November 9, 2016. The two colors distinguish between the opposite classes of each semantic dimension.

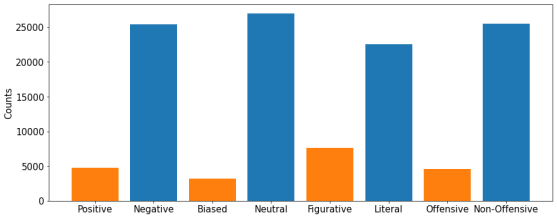

Fig. 16 Number of tweets concerning Democrats, separately for each class of the four descriptor dimensions, on November 4, 2020. The two colors distinguish between the opposite classes of each semantic dimension.

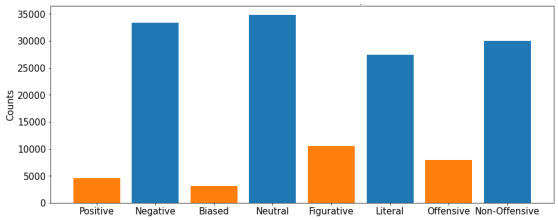

Fig. 17 Number of tweets concerning Republicans, separately for each class of the four descriptor dimensions, on November 4, 2020. The two colors distinguish between the opposite classes of each semantic dimension.

4.4 Analysis 3: Poll/election results prediction

To validate the correlation of the proposed mechanism/descriptor with public opinion actually captured in political polls and election results, an additional set of experiments were conducted using: a) the 2016 USA Presidential Elections dataset, and b) actual, national-level poll results from that pre-election period. The goal was to assess said correlation through estimating the predictive ability of the timeseries that are constructed by using the proposed mechanism.

The following two assumptions were made: a) the final result of a multi-day poll is considered valid for all days during which the census was being conducted, and b) the mean value per day was obtained for different polls covering overlapping periods. Thus, a dataset of mean daily national poll results for the

Democratic and Republican parties was constructed, covering the period from 2016-08-30 up to the election day.

The day-by-day timeseries constructed using the proposed mechanism (under the element-wise vector mean aggregation strategy) and the poll results were temporally aligned for the three-month period prior to the elections. A 7-day sliding window was shifted through these 3 months with a 1-day step, so as to derive the following **x**-**y** pair for each such window ($\mathbf{x} \in \mathbb{R}^{56}$, $\mathbf{y} \in \mathbb{R}^2$). The dimensionality of **x** is given by the number of parties (2) times the descriptor timeseries dimensionality (4) times the days covered by the window (7). The 2 entries of the respective vector **y** are the poll results (one vote percentage per party) of the day following the current temporal window. Notably, *the actual election results were employed instead of polls for the last window.* The outcome of this process was a regression dataset for learning to map public opinion descriptors constructed according to the proposed mechanism to poll/election results.

Overall, 55 **x**-**y** pairs were contained in this dataset. A single-hidden-layer MultiLayer Perceptron (MLP) was trained as the regression model, using a random 80%/20% training/test split. The temporally last data point (for the time window leading to the election day) was manually selected to be in the test set. 5-fold cross-validation in the training set was employed for manual hyperparameter tuning. Data point sampling was randomized during training, while preprocessing included only min-max normalization. The model was implemented in PyTorch, using an Adam optimizer and a Mean Square Error loss function. Optimal batch size, number of hidden neurons and learning rate were found to be 16, 16 and 0.04, respectively, while training proceeded for 40 epochs.

The exact same process was then repeated from scratch, but using only the polarity dimension from the timeseries derived through the proposed mechanism. This was done in order to emulate previous Twitter/NLP-based public opinion quantification methods from the existing literature, which only consider polarity, and compare against them. Thus, this reduced dataset contained modified $\tilde{\mathbf{x}}$-**y** pairs, where $\tilde{\mathbf{x}} \in \mathbb{R}^{14}$. Prediction results are shown in Table 6, using the MAPE metric for evaluating test accuracy. As it can be seen, the error achieved by using the proposed mechanism is very low (under 5%) and significantly lower compared to the case where only the polarity dimension is exploited.

Two conclusions can be drawn from this: a) during its training, the employed MLP successfully discovered strong correlations between the outcomes of the proposed mechanism and the poll/election results, and b) a reduced version of the proposed mechanism that stands for previous methods found in the literature (taking only polarity into account) performs worse than our method.

To further validate these results, we separately computed the normalized Pearson correlation (with values within the real range $[-1, 1]$) between each of the 8 timeseries for this 3-month period (4 timeseries per party) and the

Table 6 Poll/election results prediction accuracy, evaluated using the MAPE metric (percentage, lower is better).

Method	MAPE
Full proposed mechanism	4.17 %
Polarity-only mechanism	6.53 %

respective poll timeseries (one per party). As it can be seen in Fig. 18, the majority of the derived timeseries are strongly correlated with polls, either positively or negatively, with correlation values away from 0 at least for one of the two parties (in most cases for both). Note that in this Figure, as in many previous ones, we label each timeseries by one of the two opposite class labels it encodes (e.g., "positive" or "figurative") followed by the tweet classifier output value which implies this label. E.g., a tweet fully and undoubtedly classified as figurative/positive, has been assigned a value of 0/1 by the figurativeness/polarity classifier, respectively. In contrast, a tweet fully and undoubtedly classified as literal/negative, would have been assigned a value of 1/0 by the figurativeness/polarity classifier, respectively. Party affiliation is color-coded.

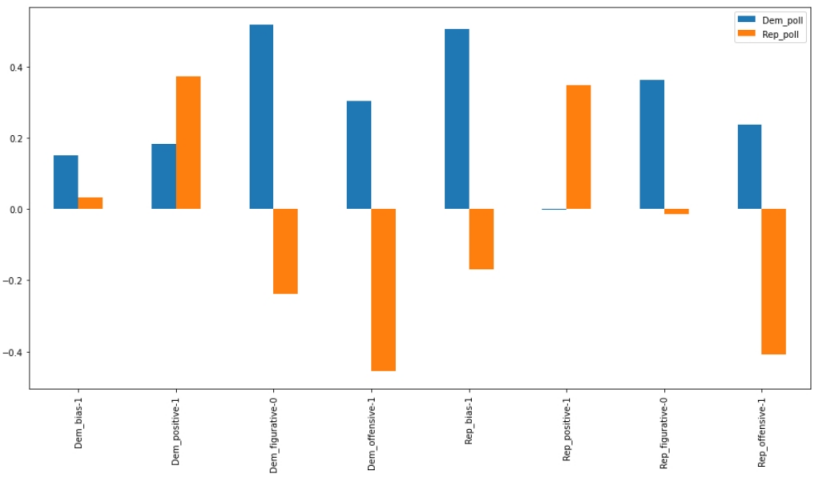

Fig. 18 Pearson correlation between each of the 8 timeseries derived by the proposed mechanism (4 per party) with the respective poll timeseries, for the 3-month period before the 2016 US presidential elections.

5 Discussion

The evaluation presented in Section 4 indicates that the proposed mechanism for automated public opinion monitoring through Twitter is a very powerful tool, able to provide valuable information for more efficient decision-making. The **multidimensional nature** of the presented descriptor conveys rich insights (analyzed in Section 4) that are not typically captured by existing

relevant methods, which only exploit sentiment (and, rarely, also bias). This is shown quantitatively in Subsection 4.4, but also through the qualitative insights extracted in Subsection 4.3. Moreover, unlike the vast majority of previously published methods, the proposed mechanism relies on state-of-the-art **DNN-based NLP tools**, a fact which guarantees enhanced accuracy in comparison to existing comparable approaches. Finally, Subsection 4.4 indicates that the proposed descriptor can indeed be exploited for **successful prediction of future poll/election results**, with its multidimensional opinion semantics giving it an advantage over previous similar approaches.

To succinctly demonstrate the usefulness of the proposed mechanism in political analysis, the most important findings extracted by applying it to the datasets of Section 4 (concerning the US presidential elections of 2016 and 2020) are summarized below:

- Public opinion concerning Democrats was less stable and less predictable, in comparison to public opinion about Republicans.
- However, in general, the public has an overall relatively stable stance towards the competing politicians: judgmental and indignant (negative + literal) but simultaneously educated (unbiased + non-offensive).
- The timeseries derived through the proposed mechanism paint a rather accurate picture of the favored candidate. This is shown both through visualizing/inspecting the timeseries and through exploiting them for learning to quantitatively predict poll/election outcomes.
- The winning party is referenced during the pre-election period in tweets that are jointly less negative + less offensive + more biased. Strong partisan presence in Twitter seems to be heavily correlated with high vote percentages.
- Crucial events do directly lead to abrupt changes in the daily number of tweets (which is to be expected), but also in public opinion. This indicates that committed/stable partisan supporters are always a minority in the American Twitter. Moreover, post hoc visualizations of the timeseries automatically derived through the proposed mechanism can actually showcase which events were the most crucial to shifts in public opinion.

A few of these findings verify similar conclusions previously drawn in the existing literature: [20] (the public has an overall relatively stable, negative stance towards the competing politicians, during a specific pre-election period, while the public sentiment timeseries paint a rather accurate picture of the favored candidate) and [13] [18] [20] (crucial events directly lead to abrupt changes in public opinion). However, the majority of our findings for the US presidential elections of 2016 and 2020, as detailed in Section 4, are original contributions of this paper. Most importantly though, the proposed mechanism is not tied to these specific elections. *It is a fully generic and almost fully automated method*, that allows interested users to easily extract similar insights *for any time period*.

6 Conclusions

Automated public opinion monitoring using social media is a very powerful tool, able to provide interested parties with valuable insights for more fruitful decision-making. Twitter has gained significant attention in this respect, since people use it to express their views and politicians use it to reach their voters.

This paper presented a novel, automated public opinion monitoring mechanism, consisting of a composite, quantitative, semantic descriptor that relies on NLP algorithms. A four-dimensional vector, i.e., an instance of the proposed descriptor, is first extracted for each tweet independently, quantifying text polarity, offensiveness, bias and figurativeness. Subsequently, the computed descriptors are summarized across multiple tweets, according to a desired aggregation strategy (e.g., arithmetic mean) and aggregation target (e.g., a specific time period). This can be exploited in various ways; for example, aggregating the tweets of each days separately allows us to construct a multivariate timeseries which can be used to train a forecasting AI algorithm, for day-by-day public opinion predictions.

In order to evaluate the usefulness of the proposed mechanism, it was applied to the large-scale 2016/2020 US Presidential Elections tweet datasets. The resulting succinct public opinion descriptions were successfully employed to train a DNN-based public opinion forecasting model with a 7-day forecasting horizon. Moreover, the constructed timeseries were thoroughly inspected in a qualitative manner in order to deduce insights about public opinion during a heated pre/post-election period. Finally, a set of regression experiments verified: a) the importance of the multidimensional opinion semantics captured in the derived timeseries, and b) the correlation of these timeseries with "ground-truth" public opinion, as captured in actual political polls and election results.

Acknowledgment

The research leading to these results has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 951911 (AI4Media). This publication reflects only the authors' views. The European Commission is not responsible for any use that may be made of the information it contains.

The authors would like to thank Georgios Chatziparaskevas for his technical aid in the evaluation of the proposed mechanism.

Competing Interests

The authors have no competing interests to declare that are relevant to the content of this article.

Data availability

The datasets generated during the current study are available from the corresponding author on reasonable request. The raw 2016 US Presidential Elections tweet dataset is publicly available at https://www.kaggle.com/paulrohan2020/2016-usa-presidential-election-tweets61m-rows.

References

[1] El Barachi, M., AlKhatib, M., Mathew, S., Oroumchian, F.: A novel sentiment analysis framework for monitoring the evolving public opinion in real-time: Case study on climate change. Journal of Cleaner Production, 127820 (2021)

[2] Romero, D.M., Meeder, B., Kleinberg, J.: Differences in the mechanics of information diffusion across topics: idioms, political hashtags, and complex contagion on Twitter. In: Proceedings of the International Conference on World Wide Web (2011)

[3] Baumgartner, J.C., Mackay, J.B., Morris, J.S., Otenyo, E.E., Powell, L., Smith, M.M., Snow, N., Solop, F.I., Waite, B.C.: Communicator-in-chief: How Barack Obama Used New Media Technology to Win the White House. Lexington Books, ??? (2010)

[4] Lorenzi, D., Vaidya, J., Shafiq, B., Chun, S., Vegesna, N., Alzamil, Z., Adam, N., Wainer, S., Atluri, V.: Utilizing social media to improve local government responsiveness. In: Proceedings of the Annual International Conference on Digital Government Research (2014)

[5] Ravi, K., Ravi, V.: A survey on opinion mining and sentiment analysis: tasks, approaches and applications. Knowledge-based systems **89**, 14–46 (2015)

[6] Ramteke, J., Shah, S., Godhia, D., Shaikh, A.: Election result prediction using Twitter sentiment analysis. In: Proceedings of the IEEE International Conference on Inventive Computation Technologies (ICICT) (2016)

[7] Agarwal, A., Singh, R., Toshniwal, D.: Geospatial sentiment analysis using Twitter data for UK-EU referendum. Journal of Information and Optimization Sciences **39**(1), 303–317 (2018)

[8] Chatfield, C.: Time-series Forecasting. CRC Press, ??? (2000)

[9] Bright, L.F., Sussman, K.L., Wilcox, G.B.: Facebook, trust and privacy in an election year: Balancing politics and advertising. Journal of Digital & Social Media Marketing **8**(4), 332–346 (2021)

[10] Lee, J., Lim, Y.-S.: Gendered campaign tweets: the cases of Hillary Clinton and Donald Trump. Public Relations Review **42**(5), 849–855 (2016)

[11] Buccoliero, L., Bellio, E., Crestini, G., Arkoudas, A.: Twitter and politics: Evidence from the US presidential elections 2016. Journal of Marketing Communications **26**(1), 88–114 (2020)

[12] Grimaldi, D.: Can we analyse political discourse using twitter? evidence from Spanish 2019 presidential election. Social Network Analysis and Mining **9**(1), 1–9 (2019)

[13] Cornfield, M.: Empowering the party-crasher: Donald J. Trump, the first 2016 GOP presidential debate, and the Twitter marketplace for political campaigns. Journal of Political Marketing (2017)

[14] Heredia, B., Prusa, J., Khoshgoftaar, T.: Exploring the effectiveness of Twitter at polling the United States 2016 presidential election. In: Proceedings of the IEEE International Conference on Collaboration and Internet Computing (CIC) (2017)

[15] Grimaldi, D., Cely, J.D., Arboleda, H.: Inferring the votes in a new political landscape: the case of the 2019 Spanish presidential elections. Journal of Big Data **7**(1), 1–19 (2020)

[16] Tavoschi, L., Quattrone, F., D'Andrea, E., Ducange, P., Vabanesi, M., Marcelloni, F., Lopalco, P.L.: Twitter as a sentinel tool to monitor public opinion on vaccination: an opinion mining analysis from September 2016 to August 2017 in Italy. Human Vaccines & Immunotherapeutics **16**(5), 1062–1069 (2020)

[17] Wang, M., Wu, H., Zhang, T., Zhu, S.: Identifying critical outbreak time window of controversial events based on sentiment analysis. PLOS One **15**(10), 0241355 (2020)

[18] Shi, W., Wang, H., He, S.: Sentiment analysis of Chinese microblogging based on sentiment ontology: a case study of '7.23 Wenzhou Train Collision'. Connection Science **25**(4), 161–178 (2013)

[19] Onyenwe, I., Nwagbo, S., Mbeledogu, N., Onyedinma, E.: The impact of political party/candidate on the election results from a sentiment analysis perspective using# anambradecides2017 tweets. Social Network Analysis and Mining **10**(1), 1–17 (2020)

[20] Yaqub, U., Chun, S.A., Atluri, V., Vaidya, J.: Sentiment based analysis of tweets during the US presidential elections. In: Proceedings of the Annual International Conference on Digital Government Research (2017)

[21] Kraaijeveld, O., De Smedt, J.: The predictive power of public Twitter sentiment for forecasting cryptocurrency prices. Journal of International Financial Markets, Institutions and Money **65**, 101188 (2020)

[22] Qi, F., Yang, C., Liu, Z., Dong, Q., Sun, M., Dong, Z.: Openhownet: An open sememe-based lexical knowledge base. arXiv preprint arXiv:1901.09957 (2019)

[23] Zhang, S., Wei, Z., Wang, Y., Liao, T.: Sentiment analysis of Chinese micro-blog text based on extended sentiment dictionary. Future Generation Computer Systems **81**, 395–403 (2018)

[24] Thelwall, M., Buckley, K., Paltoglou, G., Cai, D., Kappas, A.: Sentiment strength detection in short informal text. Journal of the American society for information science and technology **61**(12), 2544–2558 (2010)

[25] Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Proceedings of the Advances in Neural Information Processing Systems (NIPS) (2013)

[26] Hutto, C., Gilbert, E.: Vader: A parsimonious rule-based model for sentiment analysis of social media text. In: Proceedings of the International AAAI Conference on Web and Social Media (2014)

[27] Maqsood, H., Mehmood, I., Maqsood, M., Yasir, M., Afzal, S., Aadil, F., Selim, M.M., Muhammad, K.: A local and global event sentiment based efficient stock exchange forecasting using deep learning. International Journal of Information Management **50**, 432–451 (2020)

[28] Kordonis, J., Symeonidis, S., Arampatzis, A.: Stock price forecasting via sentiment analysis on Twitter. In: Proceedings of the Pan-Hellenic Conference on Informatics (2016)

[29] Oussalah, M., Zaidi, A.: Forecasting weekly crude oil using Twitter sentiment of US foreign policy and oil companies data. In: Proceedings of the IEEE International Conference on Information Reuse and Integration (IRI) (2018)

[30] Arias, M., Arratia, A., Xuriguera, R.: Forecasting with Twitter data. ACM Transactions on Intelligent Systems and Technology (TIST) **5**(1), 1–24 (2014)

[31] Kiran, R., Kumar, P., Bhasker, B.: OSLCFit (organic simultaneous LSTM and CNN Fit): a novel deep learning based solution for sentiment polarity classification of reviews. Expert Systems with Applications **157**, 113488 (2020)

[32] Zampieri, M., Malmasi, S., Nakov, P., Rosenthal, S., Farra, N., Kumar, R.: Semeval-2019 task 6: Identifying and categorizing offensive language in social media (offenseval). arXiv preprint arXiv:1903.08983 (2019)

[33] Ling, J., Klinger, R.: An empirical, quantitative analysis of the differences between sarcasm and irony. In: European Semantic Web Conference (2016). Springer

[34] Pennington, J., Socher, R., Manning, C.D.: Glove: Global vectors for word representation. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP) (2014)

[35] Hewamalage, H., Bergmeir, C., Bandara, K.: Recurrent Neural Networks for time series forecasting: Current status and future directions. International Journal of Forecasting **37**(1), 388–427 (2021)

[36] Orabona, F., Tommasi, T.: Training deep networks without learning rates through coin betting. Proceedings of the Advances in Neural Information Processing Systems (NIPS) **30**, 2160–2170 (2017)

[37] Hutter, F., Hoos, H.H., Leyton-Brown, K.: Sequential model-based optimization for general algorithm configuration. In: Proceedings of the International Conference on Learning and Intelligent Optimization (2011). Springer

[38] Claveria, O., Monte, E., Torra, S.: Data preprocessing for neural network-based forecasting: does it really matter? Technological and Economic Development of Economy **23**(5), 709–725 (2017)

[39] Hyndman, R.J., Koehler, A.B.: Another look at measures of forecast accuracy. International Journal of Forecasting **22**(4), 679–688 (2006)

[40] Tayal, D.K., Yadav, S., Gupta, K., Rajput, B., Kumari, K.: Polarity detection of sarcastic political tweets. In: Proceedings of the International Conference on Computing for Sustainable Global Development (INDIACom) (2014). IEEE