IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received October 15, 2021, accepted October 29, 2021, date of publication November 8, 2021, date of current version November 11, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3125768

Short Text Classification Using Contextual Analysis

SAMI AL SULAIMANI AND ANDREW STARKEY
School of Engineering, University of Aberdeen, Aberdeen AB24 3UE, U.K.
Corresponding author: Sami Al Sulaimani (s.alsulaimani1.19@abdn.ac.uk)

This work was supported by the Ministry of Higher Education, Research and Innovation-Sultanate of Oman.

ABSTRACT Micro blogging tools provide a real time service for the public to express opinions, to broadcast news and information and offer an opportunity to comment and respond to such output. Word usage in social media is continually evolving. Micro bloggers may use different sets of words to describe a specific event and they may use new words (i.e. neither exist in the training dataset nor in informal or formal dictionaries) or use words in new contexts. Dynamically capturing new words and their potential meaning from their context can help to reflect the words relationship in social media, which then can be useful for solving various problems, like the event classification task. Different approaches have been proposed in this regard, one of them is Contextual Analysis. This paper focuses on examining the potential of this approach for grouping short texts (tweets) talking about the same event into the same category. A new transparent method for text multi-class categorization is presented. It uses the Contextual Analysis approach to capture the most important words in the context of an event and to detect the usage of similar words in different contexts. In order to test the efficacy in these areas, this study evaluates the performance of the proposed method and other well known methods, such as Naïve Bayes, Support Vector Machines, K-Nearest Neighbors and Convolutional Neural Networks. On average, the experiments' results show that the proposed multi-class classification method can effectively categorize tweets into various event groups, with a high f1-measure score f1>97.09% and f1>95.27%, in the imbalanced classes and high number of classes experiments, respectively. However, similar to the baseline methods, the performance is negatively influenced by the imbalanced dataset. The Convolutional Neural Networks method produces the best performance among the other algorithms with f1>97.74% in all experiments, which is 1.73% and 2.72% higher than the lowest performance of Naive Bayes and K-Nearest Neighbors, respectively, but does not meet the requirements of transparency of results.

INDEX TERMS Text analysis, event classification, contextual analysis, supervised machine learning.

I. INTRODUCTION

Micro blogging tools have evolved recently to offer a real time service for the public. Micro blogging is a form of social media that facilitates communication by offering people a platform to express opinions, to broadcast news and information and provide an opportunity to comment and respond to such output. People tend to use these services as a medium to publish various types of mostly useful content (e.g. texts, images and short video clips of events as soon as they occur). Most blogs refer to real-life events, such as social events (e.g. weddings parties, graduation ceremonies, etc.), political events (e.g. presidential campaigns) and emergency events (e.g. terror attack, earthquake, tsunami, etc.).

One of the most popular micro blogging services is Twitter. The number of Twitter posts has increased rapidly since the service was launched in 2006: on average the number of posts on Twitter every second is 6000 [1], [2], but it is not clear whether this number applies to new tweets only or includes replies. In addition to supporting tweets Twitter provides a developer platform with Application Programming Interface (API) services. This enables researchers to access real-time and historical social data. As a result, many scientists and researchers have come to use the information available through Twitter in a variety of ways. While some have studied its structure and characteristics, others have helped develop applications linked to its API. Some results

The associate editor coordinating the review of this manuscript and approving it for publication was Wei Liu.

were quite unexpected. It is found, for example, that monitoring tweets could help detect earthquakes [3].

This makes the process of automatically categorizing the vast amount of collected short texts into various events' groups a very useful task. Machine learning algorithms offers some capabilities in such analysis. For example, work described in [4] successfully applied Naïve Bayes algorithm to infer the sentiment of hotels' reviews. However, the classification decisions by some of these methods, like Support Vector Machines, are generally less interpretable [5].

Although the importance of interpretability is clear for various critical real world applications, like medical diagnosis, there is no general agreement on its definition. Lipton [6] suggests that this term is not a monolithic concept, in which different properties for interpretability are proposed. These properties are categorized in two groups, which represents the two main notions of interpretability: transparency (i.e., how does the model work?) and post-hoc interpretability (i.e., what else can the model tell me?). The discussion of the topic is out of the scope of this paper, readers interested in this can find more details in prior work described in [6] and [7]. This paper adopts the evaluation approach conducted by Mori and Uchihira [5], which is motivated by the theoretical outcomes of the efforts in [6], to assess the interpretability of the proposed method. The three properties of the transparency, i.e. Simulatability, De-composability and Algorithmic Transparency, that are suggested by [6] are used to provide a qualitative assessment, as the following:

- Question 1: ''Is the entire model simple enough to be fully understood by a user?''
- Question 2. ''Is each part of the model (each input, parameter, and calculation) intuitively explainable?''
- Question 3. ''Is the algorithm deterministic (non-stochastic) without using any random numbers?''

To contribute in this emerging field, i.e. interpretable machine learning, this paper proposes a new approach for short text multi-class classification problems that can be easy to interpret. It uses a method, called Contextual Analysis [8], to build a tree-like structure for the words that appear in a similar set of sources, and then creates a model for the classification purpose.

This rest of this paper is organized as follows. Section II introduces, in general, some of the related works on the classification of short texts. Section III describes briefly the Contextual Analysis approach. Section IV presents details of our proposed multi-class classification approach using the Contextual Analysis method. Section V discusses the experiments and results obtained by comparing the new approach and the baseline methods. Finally, Section VI gives the conclusion remarks and our future work.

II. RELATED WORK

A number of works have attempted to employ various approaches to classify micro blogging posts, like tweets, into two categories (binary classification) or more. The majority of these efforts use existing machine learning techniques, such as Support Vector Machines (SVM), Naïve Bayes (NB), K-Nearest Neighbors (KNN), etc.

The authors in [9] proposed an approach to classify tweets that are related to ''news'', ''events'', ''opinions'', ''deals'' and ''private messages'' using Naïve Bayes algorithm. Seven binary features were suggested (e.g. whether a tweet contains time event phrases or not) as well as one nominal feature for the authors information. The main findings of this study are: the author feature show a discriminative ability; and their selected features produced higher classification accuracy compared to the traditional bag of words strategy.

The work conducted in [10] classified tweets' texts into 18 predefined set of generic categories, such as ''technology'', ''science'', ''politics'', etc. They examined various machine learning algorithms in their text-based approach, in which the best accuracy, 65%, was achieved using Naïve Bayes Multinomial classifier.

The efforts in [11] adopted the supervised learning methods in their proposed system to automatically classify citizen complaint tweets into general topics (such as ''department of transportation'', ''education'', etc.) and specific topics (such as, ''flood'', ''damaged roads'', etc.). They evaluated two different scenarios to accomplish this task. The first one starts by classifying the general topics and then, based on them, the specific topics are further classified. The second scenario is that the specific topics are directly classified. Their results show that the former scenario achieved better accuracy than the later, and the best result was obtained by using Support Vector Machines with Sequential Minimum Optimization.

In [12] the authors classified tweets into 14 categories (sensitive topics), such as ''racism'', ''sexual orientation'', ''family & personal'', etc., in order to develop a privacy protection approach. Naïve Bayes algorithm was selected for the classification purpose. The authors found that the topic classification performance improved by 3.4% by adding user' topic preferences along with tweets' texts that were processed by Term Frequency–Inverse Document Frequency method.

In [13] an attempt is made to utilize machine learning algorithms, include Support Vector Machines, Niave Bayes and Adaboost, in order to build a classifier that can detect Islamic State of Iraq and Syria (ISIS) related tweets. The classifier was trained by a number of 619 features from three different types, such as stylometric features (e.g. frequent words, hashtags, word bigrams, etc.), temporal features (e.g. hour of day, types of day, day, etc.) and sentiment features (very negative, negative, neutral, positive and very positive). Adaboost produced the best performance.

The work presented in [14] where a real time event detection framework is proposed to identify large-scale (global) and related small-scale (local) events from micro blogging posts. In the classification part of their framework, they employed Naive Bayes algorithm in order to distinguish between ''event'' and ''non-event'' tweets. They found that this method produced the best f1 score (85.43%) compared to Support Vector Machines (83.86%) and Logistic Regression (80.22%).

Other researchers have paid considerable attention to employ deep learning techniques, such as Convolutional Neural Networks (CNN) [15], Recurrent Neural Network (RNN) [16], etc., and their various architectures for short text classification in Twitter posts. For example, [17] proposed an approach that combines SVM and CNN for short text sentiment analysis, in which CNN was used for feature extraction and SVM for the classification task.

Another interesting example is the work in [18] where the authors experimented with various techniques, such as Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), CNN-LSTM, CNN-GRU and SVM, in order to detect hate speech in Twitter. Their effort tend to focus on classifying tweets (in Arabic) into five classes, include ''racial'', ''sexism'', ''general hate'' and ''not hate''. The best performance was achieved by using CNN-LSTM approach, with an f1 score of 73%.

The work in [19] focused on using machine learning methods to provide an early warning for depressions symptoms among Arab women. A model is created for Arabic language by applying LSTM in order to classify tweets into two categories, namely ''depression'' or ''not depression''. They found that this method gives the best performance, f1 score (69%) compared to other approaches, such as CNN, SVM, etc.

It is well recognized in literature [20], [21], that despite these successes, that focus needs to be given to the availability of transparent methods to solve classification problems in general, and is especially important for detecting critical events via social media. A number of techniques, like the deep learning methods, have been shown for their effectiveness to solve various text classification problems, however, their opaque nature hindered their usage in some critical domains that require skeptical users trust. Offering algorithmic decisions that are transparent in nature is an urgent need for the analysts in these fields and for complying with the introduced regulations that adopt the ''Right to Explanation'' [22].

III. CONTEXTUAL ANALYSIS

Abdul Aziz and Starkey [8] proposed a novel approach known as Contextual Analysis which builds a tree-like structure, called Hierarchical Knowledge Tree (HKT), in order to capture the relationship between the words based on their appearance in the same context. This relationship can be articulated by grouping the words that appear in a similar set of sources (tweets for example) in a node in the tree, and in its child nodes as a parent-child relationship, depending on the strength of this relationship. For detailed description about this approach, the readers are referred to the original work in [8].

It is important to note that in a related research line to the Contextual Analysis work, efforts have been conducted to use graph properties in order to capture the contextual information of short texts [23]–[26]. For example, in [25] the authors used posts in Twitter to build a language graph based on words (or hashtags) co-occurrence, in which a node represents a word and an edge represents a link connecting two words co-occurring in the given text. By examining seven link prediction methods (such as Weighted Common Neighbors, Weighted Adamic-Adar, etc.), it is found that the links between the words (or hashtags) can be predicted in spite of the incomplete graph structure.

Similar to the link prediction approaches, the Contextual Analysis method can establish a link between two words even if they do not co-occur within any specific text (source) in the dataset. A node that encapsulates two words (or more) can be linked to one or more child-nodes, which may contain words that do not necessarily exist in all words' sources in the parent node. However, unlike the link prediction approaches, the algorithm develops this latent link during the construction of the tree and without using any external methods.

In general, and differently from the traditional graph approaches, Contextual Analysis is capable of determining the various senses of the words (i.e. different meanings of words) based on their context automatically. For example, the word ''beat'' may refer to a defeat in a football game (Liverpool beat Everton), or to an act of stirring cooking ingredients (beat the fat with the sugar). This lexical ambiguity is addressed by creating a simple hierarchical structure to capture the various topics and their sub-topics, for example ''beat'' as a sub-topic for ''Liverpool, Everton'' and ''beat'' as a sub-topic for ''fat, sugar'' depending on the sources. More details about this representation are clearly presented in later section (Section V).

The work conducted in [8] applied Contextual Analysis to predict the performance of supervised machine learning models and to give an indication when these models start to degrade. The experiments described in [27] employed Contextual Analysis in the classification task for sentiment analysis. By using a training dataset, it creates a Hierarchical Knowledge Tree (HKT). This allows analysis of the nodes against the labeled sources that map to them and can then be determined to have mostly positive or mostly negative sources (using a pre-defined threshold value). These nodes, also called influential nodes, are then used to classify any new dataset.

Although Contextual Analysis received the lowest, on average, performance figures in in-domain and cross-domain sentiment analysis in comparison with state-of-art machine learning models, the difference is not significant. It is important to note that this algorithm introduced a new measure, called unclassified results, which are caused by either the new words in the testing dataset or the equivalence between the number of positive and negative words in the sources. Also, the words relationships in the tree can hold important information about their context which can be used for further analysis. However, there are no experiments in this previous work to show how this method performs in multi-class classification tasks. Also, the performance for this algorithm has been assessed using one corpus, Amazon reviews dataset, which contains longer texts' sources compared to other important domains for

this type of analysis, like Twitter. Further investigation is required.

Abdul Aziz and Starkey [8] suggested that the important words for the sentiment classification purpose can be identified via the 'influential nodes' in the tree. Using the labeled samples, two techniques were proposed in order to highlight these nodes: via calculating the node accuracy (i.e. the accuracy of sources in each class) or using the Term Frequency-based ratio (i.e. dividing the total number of sources in each node for every class, positive or negative, over the total number of samples). Then, according to a certain threshold against the output of one of these two processes, the influential nodes, which encapsulate the important words, are triggered. However, it is not clearly described how the threshold value should be selected. Also, there is no guidance on how to fire the influential nodes in a tree that contains more than two classes (i.e. multi classes). In other words, how can the contextual analysis tree be employed to capture the important words for a multi-class problem?

Up to now, there have been no attempts to examine how Contextual Analysis approach can be employed to solve multi-class classification tasks and whether it is capable of doing so. Also, there has been no systematic analysis of Contextual Analysis in addressing the problems when a training sample comprises of imbalanced classes or a high number of classes.

Although these problems, i.e. imbalanced classes or a high number of classes, have recently gained extensive attention [28]–[31], we believe that there is a lack of experimental evaluation of various machine learning algorithms in the context of short text multi-class classification domains, especially in micro blogging posts (where the imbalanced distribution of various classes within these posts is highly skewed in nature), and in particular the applicability of the Contextual Analysis algorithm to these two areas has also not been determined. In addition, the solutions for imbalanced data cannot be applied to real time analysis of unlabeled short text data. Thus, this work provides empirical assessments of Contextual Analysis and the well known machine learning approaches for comparison purposes, such as Naïve Bayes (NB), Support Vector Machines (SVM), K-Nearest Neighbors (KNN) and Convolutional Neural Networks (CNN), in this domain.

In this paper, a new method for the Contextual Analysis based on the average precision and recall is presented, which can help to employ the constructed nodes in order to capture the important words in multi-class problems. It converts every word in the tree to a vector, which can then be utilized for multi-class classification tasks. This is the first study to empirically employ Contextual Analysis for this type of problem.

IV. MULTI-CLASS CLASSIFICATION APPROACH USING CONTEXTUAL ANALYSIS

As is clearly explained in the original paper on describing the generation process of the Contextual Analysis tree [8], every node encapsulates information about the words and their sources. Some of these nodes are believed to be considered as important pillars in understanding the data, whereas other nodes may be discarded in the analysis. This verdict is highly dependent on the predefined threshold setting.

In this paper, inspired by word2vec [32], it is suggested that the understandability of the data can be improved by considering all nodes, with various influence, in the process of analyzing the training dataset and in detecting the important words in the tree. Also, the nodes should trigger their importance in any class without any external intervention (i.e. pre-defined threshold value). Therefore, this study presents a new method, based on the original work, in order to involve all tree nodes in the analysis.

Suppose that a training dataset (D) consists of (n) number of records and (m) number of classes ($Class$), it is hypothesized that the strength of any single node ($Node_a$) in a set of the constructed tree nodes $Node_1, \ldots, Node_x$ in every class $Class_1, \ldots, Class_z$ can be captured by calculating the average precision and recall (f1 score) (see equation (1), (2), (3)) of every class in this node (i.e. $Node_a$), where x and z are the number of nodes in the tree and the number of the classes in the training samples, respectively.

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive} \tag{1}$$

$$Recall = \frac{TruePositive}{TruePositive + FalseNegative} \tag{2}$$

$$f1Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{3}$$

The output of this process is a vector $Node_aVec = Node_af1Class_1, \ldots., Node_af1Class_x$, where each numerical element is made up of the f1 scores and represents the strength of each node in a specific class. This process should be repeated for every node $[Node_1, \ldots, Node_x]$ in the tree. The final output of this phase is a set of vectors for all nodes in the tree $T = Node_1Vec, \ldots, Node_xVec$. To give an illustration, Fig. 1 shows three nodes that are encapsulated in the first level of a tree, and where each node presents varying strengths for the four different classes. According to the figure, class (Class1) is the dominant class in the node (Node1), in which all other classes have less apparent influence in this node. Also, Class3 is the weakest class in this level of the tree, however, it will show its strength in other levels of the tree and whenever its f1 score for any node is high.

The next phase is to create a vector ($WordVec$) for every word $Word_1, \ldots., Word_y$ that is present in the whole tree. This vector demonstrates the word strength in every class in the tree. By looping through all the words, any word vector ($Word_aVec$) accumulates the nodes vectors values $Node_1Vec, \ldots, Node_xVec$ wherever this word is encountered in any node. The result of this phase is a set of words' vectors $Word_1Vec, \ldots, Word_yVec$ that represent the degree of influence of each word in every class. These vectors are then transformed using SoftPlus function (see equation (4)).

$$f(x) = ln(1 + e^x) \tag{4}$$

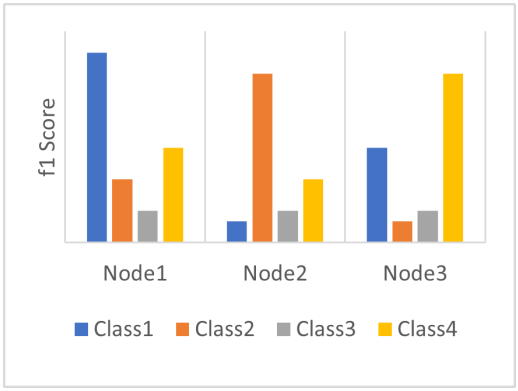

FIGURE 1. Example of a hierarchical knowledge tree container that encapsulates three nodes. The influence of each class (Class1, Class2, Class3 and Class4) in every node is represented by f1 scores. The class with highest score in the node is the dominant class. While Class1 is the dominant class in Node1, Class2 and Class4 have the greatest influence on Node2 and Node3, respectively.

TABLE 1. Hardware and software configurations.

Operating System	Windows 10 64-bit operating system, x64-based processor
Running Memory	8 GB (7.88 usable)
Processor	Intel(R) Core(TM) i7-3632QM @2.20 GHZ 2.2 GHZ
Hard Disk	917 GB (601 GB free)
Software	Visual Studio 2019 (community), SQL Server Express 2017, SQL Management Studio (V18.2).

V. EXPERIMENTS

A. EXPERIMENT ENVIRONMENT

All the programming works on this paper's experiments were carried out using C# and Structured Query Language (SQL). The hardware and software configuration of the experiments is shown in Table 1.

B. EXPERIMENT DATASET

The experimental data was downloaded from the University of Glasgow website on December 2019 [33]. Work described in [34] created an event detection corpus from Twitter which contains 120 million tweets, collected in 2012. They managed to identify 506 events linked to more than 150 thousand tweets, which are manually annotated using crowd sourcing. However, this corpus only contains tweet ids. We managed to collect 70 thousands tweets out of 150 thousands through Twitter API.

C. EXPERIMENT DETAILS

1) TEXT PRE-PROCESSING

The first phase of the experiments in this study is to pre-process the texts in the given corpus. A simple pre-processing is undertaken for every tweet in the training and testing datasets. All hyperlinks and any non-alphabetic or non-numeric characters, except ''#'' and space characters, are removed from the text. Also, by using Microsoft.ML library [35], tweets are tokenized based on the space between any set of characters and stop words are removed. To achieve a fair evaluation, all sub-datasets, either for training or testing purposes, from the main corpus went through the same pre-processing phase for Contextual Analysis and the baseline algorithms i.e. Naïve Bayes, Support Vector Machines (SVM), K-Nearest Neighbors (KNN) and Convolutional Neural Networks (CNN) (see V-C3), before conducting the experiments.

2) CONTEXTUAL ANALYSIS IMPLEMENTATION

Contextual Analysis algorithm starts by creating a lookup table for the words. The aim of this process is to create a unique numeric representation for every word in order to speed up the computations required by this algorithm.

This is followed by the core steps in the Contextual Analysis approach which are implemented by creating nodes and their Hierarchical Knowledge Tree (HKT) containers. Every node contains two different sets: a set of words and a set of sources. By starting with the word ($Word_1$) with the highest number of sources, the first node ($Node_1$) is created. This node is then encapsulated in the first (or seed) HKT container ($Seed\_HKT$) which can be comprised of more than one node.

All words in the nodes that are included in this container ($Seed\_HKT$) must have number of sources above the threshold value (α) which is calculated against the ($Word_1$) number of sources. Also, the words that share similar set of sources, above a threshold value (β), are grouped in a single node. In this paper, these parameters are set to 0.7 and 0.5 for (α) and (β), respectively. This selection is based on preliminary experiments on the selected corpus for this paper.

After creating the first HKT container ($Seed\_HKT$), a set of remaining words for every node (i.e. not used in the creation of pre-assessor node) is used to build sub-level HKT. Every sub-level HKT must be linked to a parent node.

The last two steps are focused on creating nodes and words vectors. Using the labeled events in the training dataset, all nodes and words in the tree are vectorized according to the method that is explained in section IV.

3) BASELINE METHODS IMPLEMENTATION

Before applying the developed system with the new method, Naïve Bayes, Support Vector Machines, K-Nearest Neighbors algorithms and Convolutional Neural Networks are selected in order to accurately compare the performance of the results. The implementations published in [36] and [37] are used for the experiments in this paper.

a: Naïve BAYES

Naive Bayes is a common supervised machine learning algorithm for classification tasks. It is a probabilistic algorithm that is based on Bayes Theorem. This technique has been widely studied to solve various machine learning classification problems, more commonly in the domain of text classification [38]. By representing a document (D) as a bag

of words, Naïve Bayes algorithm starts by estimating the posterior probability of each class (using the training dataset) via Bayes rule [39]:

$$P(C|D) = \frac{P(C) \times P(D|C)}{\sum_{c \in C} P(C = c) \times P(D|C = c)} \quad (5)$$

where $P(C|D)$ is the posterior probability that a given set of a document's terms (D) belongs to a class (C), $P(C)$ is the prior probability of the occurrence of the class (C) in the corpus, $P(D|C)$ is the conditional probability that a randomly chosen set of document's terms from documents in the class (C) is in the document (D), and $P(D)$ is the probability that a randomly chosen document from the corpus is the document (D). Then, the algorithm gives an output of the highest probable class for the query document (D), as follows:

$$Class(D) = argmax_{C \in AllClasses} P(C|D) \quad (6)$$

It is important to mention that Naïve Bayes makes the assumption that the documents' terms are independent from each other.

b: SUPPORT VECTOR MACHINES

Support Vector Machines (SVM) is a supervised machine learning algorithm. The goal of SVM is to learn an optimal hyperplane that separates the samples, like tweets, according to classes. It is designed to find the greatest possible margin between the hyperplane and the training samples [40]. This is achieved by identifying two other parallel hyperplanes that passes one or more of the instances, called support vectors, and with an optimal distance from the central hyperplane. The unseen samples are then classified according to which side of the hyperplane they falls on.

c: K-NEAREST NEIGHBORS

K-Nearest Neighbour (KNN) is one of the frequently used algorithms for text classification. It categorizes documents into one of the predefined categories in the training dataset. It is based on the assumption that nearby points should be classified to the same class [41]. Given a document (D), this algorithm begins by finding the K closest instances to (D) by comparing to all samples in the training set. Then, it uses the categories of the k top closest neighbors to identify the category of the input document (D). It is noteworthy that the only task accomplished during the K-NN training phase is storing training documents. The core process is triggered when a new query document is fired during the categorization phase. Thus, it is referred to as a lazy learner.

d: CONVOLUTIONAL NEURAL NETWORKS

Convolutional Neural Networks (CNN) [15], also known as ConvNet, is a popular method that falls under the deep learning umbrella. It was originally implemented in the realm of image-based applications, for instance, to solve image classification problem. Also, studies have shown the potential of the CNN based architectures to solve various text classification tasks [42], [43]. For example, in [43] the author empirically demonstrates the effectiveness of their CNN based approach in the sentiment analysis and question classification.

Although there are various CNN architectures for the text classification purpose, they mainly comprise of two main components, namely: the feature extraction stage; and the classification stage. Typically, these components consist of various types of layers, in which the output of each layer is fed as an input for the next layer, as in the following:

- Embedding layer: For text classification tasks using CNN, the input documents need to be transformed to matrices. Each word in the document is mapped to a low-dimensional vector. This layer conducts this mapping operation. Although these embeddings can be randomly initialized and then learned during the training phase, they can also be selected from pre-defined models.
- Convolutional layer: The main purpose of this layer is to automatically learn features' representations of the inputs. It contains a number of kernels (or filters) in order to perform the convolution operation on the input data. This process computes feature maps for each kernel, which is followed by an activation function, such as RelU.
- Pooling layer (also known as sub-sampling layer): This layer receives the output of the convolutional operation in order to help in reducing the dimensions of the input features. By using the advantages of pooling techniques, most commonly max pooling operation, higher-level features are obtained.
- Fully-connected layer: Following the process of extracting high-level features in the previous layers (i.e. feature extractions component), the output is fed to one or more fully connected layers for the classification phase. This is a classical feed-forward neural network hidden layer, in which it delivers the results to an output layer (i.e. the last layer of CNN).

It is important to note that the number of the stacked convolutional and pooling layers in the architecture varies according to the problem in hand. In this paper, as a baseline method, we apply the approach that is presented in [43], in which three kernel settings (3,4,5) are implemented followed by the max pooling technique for each, and similar initial parameters are used. However, in this work we train the model without predefined word embeddings (i.e. random initialization). Details about this approach can be found in the original work.

D. EXPERIMENTAL DESIGN

In order to examine the effectiveness of the new method to solve multi-class classification tasks against problems found in real world real time data, the experiments are carefully designed to measure the performance from two different perspectives:

- Imbalanced Classes: when there is an unequal distribution of samples for the classes in the dataset.

- High Number of Classes: when the number of classes in the dataset is high.

1) IMBALANCED CLASSES EXPERIMENTAL SETUPS

A dataset is considered imbalanced when the difference between the number of samples in the majority class (a class with the highest number of samples) and the minority class (a class with the lowest number of samples) is significant [31]. This can be measured by calculating the imbalance ratio, as the following:

$$Imbalance\_Ratio = \frac{\#of\ SamplesMajorityClass}{\#of\ SamplesMinorityClass} \quad (7)$$

As a rule of thumb, if *Imbalance_Ratio*>1.5, the dataset is deemed imbalanced.

To test whether Contextual Analysis can effectively classify imbalanced classes, 11 different sub-datasets from the main corpus are selected. Each subset (group) contains a different set of events, as described in Table 2. There are three different themes for the eleven groups: Theme A, Theme B and Theme C. Theme A contains groups that have balanced classes (*Imbalance_Ratio*<1.5), each of them consisting of a high number of tweets. In Theme B, the groups contain classes that are balanced but contain low number of tweets. Groups in theme C comprise of imbalanced classes (*Imbalance_Ratio*>1.5) with varying tweet counts.

a: EVALUATION METHODOLOGY

To compare Contextual Analysis and the baseline methods, the algorithms are fed with these series of themes, after every input precision, recall and f1 score are calculated. To estimate the overall performance, the macro averaged f1 measure is selected (see equation (8), (9), (10), where q is the number of classes.). In order to increase the reliability of the measures, Contextual Analysis and the baseline method are fed with each group five times, giving a total of 55 different trials. In every attempt the training and testing tweets are selected randomly.

$$Macro\_Precision = \frac{\sum_{i=1}^{q} Precision_i}{q} \quad (8)$$

$$Macro\_Recall = \frac{\sum_{i=1}^{q} Recall_i}{q} \quad (9)$$

$$Macro\_f1 = 2 \times \frac{Macro\_Precision \times Macro\_Recall}{Macro\_Precision + Macro\_Recall} \quad (10)$$

b: RESULTS

All the results on the imbalanced classes dataset are shown in Table 3 (with the best results for each group highlighted in the table) and in Fig. 2.

On average, as can be seen from Table 3 and Fig. 2, Convolutional Neural Networks approach outperforms all methods. Also, Support Vector Machines produces the second best results, however when it is compared to the other three algorithms in Theme B, the difference is negligible (p>0.05 using paired t-test). Interestingly, the results of the experiments in all themes show no significant difference between the f1 measures of Contextual Analysis, Naïve Bayes and KNN, p>0.05, using One-way analysis of variance. However, the performance of Naïve Bayes in Theme C is the worst, where the difference between the number of tweets in each group's classes is high. This is mostly due to the low recall value, see table 3 and figure 3.

TABLE 2. Dataset setup for theme A, theme B, and theme C. In total, there are 11 sub-datasets, each contains different set of events. Bold value indicates the event ID. Each them is designed with different settings. For example, the datasets in theme C comprise of imbalanced classes, with Imbalance_Ratio>1.5.

Theme	Group ID	Training Dataset Count	Testing Dataset Count	Imbalance Ratio	**'EventID'**: No. Tweets
A	1	2645	1138	1.12	**'55'**:660, **'6'**:648, **'12'**:636, **'31'**:628, **'46'**:624, **'15'**:587
	2	2025	874	1.33	**'7'**:564, **'5'**:527, **'2'**:501, **'18'**:454, **'20'**:428, **'47'**:425
	3	1680	723	1.12	**'57'**:420, **'146'**:413, **'32'**:404, **'4'**:398, **'35'**:392, **'36'**:376
B	4	462	203	1.01	**'279'**:111, **'275'**:111, **'341'**:111, **'241'**:111, **'493'**:111, **'447'**:110
	5	444	194	1.06	**'211'**:110, **'183'**:108, **'365'**:106, **'83'**:105, **'69'**:105, **'181'**:104
	6	428	186	1.02	**'481'**:103, **'98'**:103, **'45'**:103, **'65'**:102, **'487'**:102, **'467'**:101
C	7	1591	686	5.95	**'55'**:660, **'6'**:648, **'12'**:636, **'279'**:111, **'275'**:111, **'341'**:111
	8	1338	578	5.32	**'7'**:564, **'5'**:527, **'2'**:501, **'211'**:110, **'183'**:108, **'365'**:106
	9	1081	465	4.08	**'57'**:420, **'146'**:413, **'32'**:404, **'481'**:103, **'98'**:103, **'54'**:103
	10	8138	3496	52.36	**'8'**:7225, **'157'**:1580,**'14'**:957, **'6'**:648, **'4'**:398, **'254'**:241, **'482'**:165, **'422'**:144, **'72'**:138, **'153'**:138
	11	6700	2877	25.90	**'1'**:3419, **'11'**:2898,**'22'**:1430, **'12'**:636, **'35'**:392, **'383'**:241,**'349'**:165, **'13'**:132, **'70'**:132, **'439'**:132

2) HIGH NUMBER OF CLASSES EXPERIMENTAL SETUPS

In order to investigate the effects of an input dataset that contains a high number of classes on the performance of the method on the classification task, 30 different balanced sub-datasets (with the identified Group ID) as described in Table 4 are selected. Although there are various techniques to alleviate the skewed distribution in the classes, the well known Random Under Sampling (RUS) mechanism [29] is

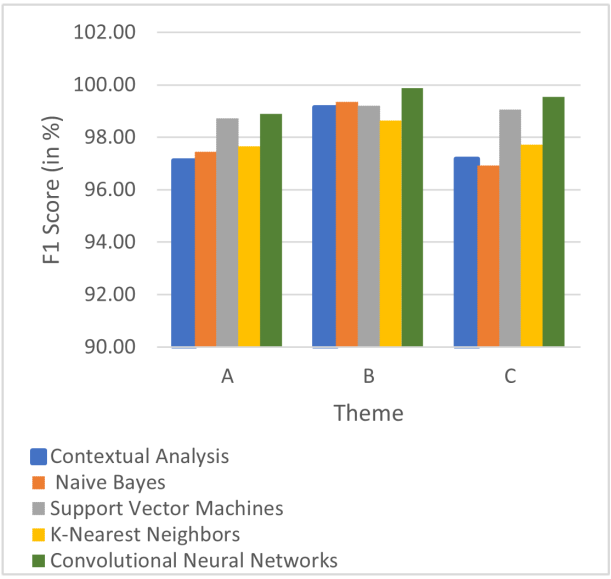

FIGURE 2. Performance comparison between different text categorization methods (Contextual Analysis, Naive Bayes, Support Vector Machines, K-Nearest Neighbors and Convolutional Neural Networks) on imbalanced classes' datasets (see Table 2). Macro averaged f1 scores are presented. On average, Convolutional Neural Networks method gives the best f1 score (99.55%) in the imbalanced dataset (Theme C), which is 2.63% higher than the performance of Naive Bayes classifier (96.92%). Contextual Analysis produces 97.18% in this Theme.

TABLE 3. Performance comparison between different text categorization methods (Contextual analysis, Naive bayes, Support vector machines, K-nearest neighbors and convolutional neural networks method) on imbalanced classes' datasets(%) (see Table 2). Macro averaged metrics, i.e. macro precision, macro recall and macro f1, scores are presented. The best f1 results are highlighted in green, the worst in red.

	Theme	Contextual Analysis	Naive Bayes	Support Vector Machines	K-Nearest Neighbors	Convolu. Neural Networks
Avg. Precision	A	97.22	97.50	98.75	97.72	98.90
	B	99.18	99.36	99.25	98.73	99.87
	C	97.87	98.14	99.58	98.98	99.76
Avg. Recall	A	97.15	97.45	98.71	97.70	98.91
	B	99.15	99.34	99.18	98.63	99.87
	C	96.60	95.89	98.57	96.62	99.36
Avg. f1	A	97.09	97.45	98.71	97.65	98.89
	B	99.14	99.34	99.19	98.63	99.87
	C	97.18	96.92	99.05	97.71	99.55

selected, in which the distribution of the classes is adjusted by randomly removing samples from the majority classes. Thus, each group in the sub-datasets contains the same proportion of randomly selected tweets from randomly selected events from the corpus. The number of tweets in each event is constrained to 200. For example, the experiments in Group 1,2 and 3 contains 5 different events, each event contributes, randomly, by exactly 140 tweets and 60 tweets to form the training and the testing sub-dataset, respectively.

a: EVALUATION METHODOLOGY

Similar to the previous experiment, the Contextual Analysis algorithm is compared with the four baseline methods and the macro averaged f1 measure is chosen. Three random trials are conducted to construct the training and testing dataset, giving a total of 90 different trials.

TABLE 4. Dataset setup for high number of classes experiment. Each row indicates that there are three different balanced sub-datasets selected with the same settings (i.e. same number of training samples, testing samples and events). However, each group contains randomly selected tweets from randomly selected events from the corpus.

Group ID	Training Dataset Count	Testing Dataset Count	Number of Events
1, 2, 3	700	300	5
4, 5, 6	1400	600	10
7, 8, 9	2100	900	15
10, 11, 12	2800	1200	20
13, 14, 15	3500	1500	25
16, 17, 18	4200	1800	30
19, 20, 21	4900	2100	35
22, 23, 24	5600	2400	40
25, 26, 27	6300	2700	45
28, 29, 30	7000	3000	50

b: RESULTS

The results that are summarized in Fig. 3 compare the performance of the algorithms when the input dataset contains higher number of classes. Overall, the figure reveals that there is a clear trend of decreasing performance of the five algorithms when the number of events increases. Similar to the results of the previous experiment, Convolutional Neural Networks method achieves the best performance among all approaches. On the other hand, as is apparent from Fig. 3, KNN achieved the lowest performance, $p>0.05$ (paired t-test), in most of the datasets. Interestingly, when the number of the events in the sample is less than 30, no significant difference is found among Contextual Analysis, Naive Bays and SVM, $p>0.05$ using One-way analysis of variance.

3) INTERPRETABILITY

a: EVALUATION METHODOLOGY

To assess the interpretability for the five algorithms in the context of this paper, the evaluation approach given in [5] is chosen. Reference [5] employs three properties of transparency, i.e. Simulatability, Decomposability and Algorithmic Transparency, that are suggested by [6] to form a qualitative approach to assess the interpretability of a given method (see section I).

Yet, in the context of this paper, none of these algorithms can satisfy the requirements for the first question, i.e. "Is the entire model simple enough to be fully understood by a user?". Lipton [6] claims that a model is considered simple if a human, in reasonable time, can follow its generation procedure that involves every calculation required to process the input data with the parameter settings. Therefore, the assessment in this part is constrained to 10 input samples.

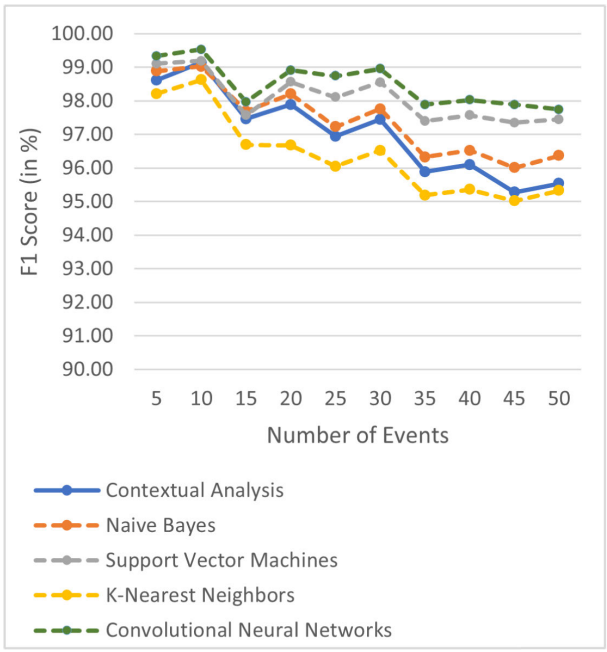

FIGURE 3. Performance comparison between different text categorization methods (Contextual analysis, Naive bayes, support vector machines, K-nearest neighbours and convolutional neural networks method) on high number of classes' datasets (see Table 4). Macro averaged f1 scores are presented. On average, Convolutional Neural Networks method gives the best classification performance with the lowest of (f1 = 97.35%) for the 45 events' dataset, which is 2.72% higher than the performance of K-Nearest Neighbours classifier (95.02%).

b: RESULTS

As a result of the deterministic behavior of the four algorithms (Contextual Analysis, Naïve Bayes, Support Vector Machines and K-Nearest Neighbors), they all fulfill the transparency requirement of the third component, i.e. question 3. With a particular input dataset, each algorithm will always generate the same model. On the other hand, Convolutional Neural Networks approach is non-deterministic due to the use of a stochastic optimization method in their training phase.

With the specified constraint above, Contextual Analysis, Naïve Bayes and KNN are assessed as simple methods and their parts allows an intuitive explanation. This is due to the nature of these approaches, where each step can be easily followed to generate the prediction model. On the other hand, SVM and CNN are regarded as a non-transparent method in component 1 and 2. SVM generates a separating hyperplane for classification tasks and the support vectors with the parameter tuning optimization inside the algorithm that can lead to a model that can be difficult to comprehend. Like the other methods in the deep learning family, the nested structure and the underlying complicated interactions among the various layers as well as the optimizations processes, contribute negatively to the transparency assessment of CNN.

In this context, it is observed that Contextual Analysis may offer a valuable insight of the most important events' words and their relationships in a simple representation. It is noticed that there is a logical link between the words that appear in the nodes in the upper levels of the created tree and the events in the annotated dataset (represented by its description). Also,

TABLE 5. Descriptions for the events in Table 2 - Group 5 as they appear in the main corpus [33].

Event Id	Description
69	The death toll in the U.S. attributed to Hurricane Sandy rises to at least 90
83	Buster Posey grand slam leads SF Giants to historic Division Series
181	Football match between USC and Oregon
183	Mike Wallace scoring a touchdown during the Giants and Pittsburgh football game.
211	A woman is killed and 12 other people injured in a series of hit and run incidents in Cardiff, south Wales. A 31-year-old van driver is arrested by police.
365	Ontario premier Dalton McGuinty announces his resignation.

by navigating through the child HKTs of these nodes, other words that are strongly related to the event context emerge. To illustrate, Fig 4 shows a screenshot of a tree representation of the first level (or seed) HKT's nodes (such as Node1, Node2, etc.) and their child HKTs' nodes (such as Node1.1, Node1.2 and Node1.3 as child nodes for Node1) that are produced using the Contextual Analysis application and the events dataset in Table 2 - Group 5. The Seed-HKT encapsulates the most important words in the corpus, such as ''wallace'', ''mike'', ''slam'', ''cardiff'', etc.. The words that are found in a similar set of sources are grouped in the same node, such the words ''dalton'',''mcgunity'', ''premier'' and ''ontario'' in Node4 and ''hurricane'' and ''sandy'' in Node5. The nodes with the symbol <>, such as Node7, represents data not matching the words shown at that level and ensures that every sample (tweet) matches at least one node. Each node is linked to at most one container (Child-HKT) for its child nodes which represents other important words found in the parent node's tweets, although at a lower frequency in the corpus. For example, the words ''speed'' and ''touchdown'' are the other important words found in the tweets that mention the words in Node1. Also, the word ''resigns'' in Node4.1 appears to be an important word when people tweeted about the words in Node4.

According to the annotated description of these events, as summarized in Table 5 and found in the original work of the used corpus [33], most of them can be linked to a certain node with its descendant HKTs. For example, Node4 captures the words in the context of the resignation of the Ontario premier Dalton McGuinty (event id 365). The words that are displayed in this node and in its child HKT, such as ''mcguinty'', ''dalton'', ''premier'', ''ontario'', ''resigns'' are related to this event. Also, by investigating the child HKTs for Node3 it is found that the words ''cardiff'', ''injured'', ''hit'', ''run'', ''people'' and ''11'' are mainly used in this context which are strongly related to the description of the event id 211. Given the HKT is developed in an unsupervised manner without reference to the target class of the tweets, this is an important result demonstrating how the process automatically analyses the underlying structure in the dataset in an easily understandable output.

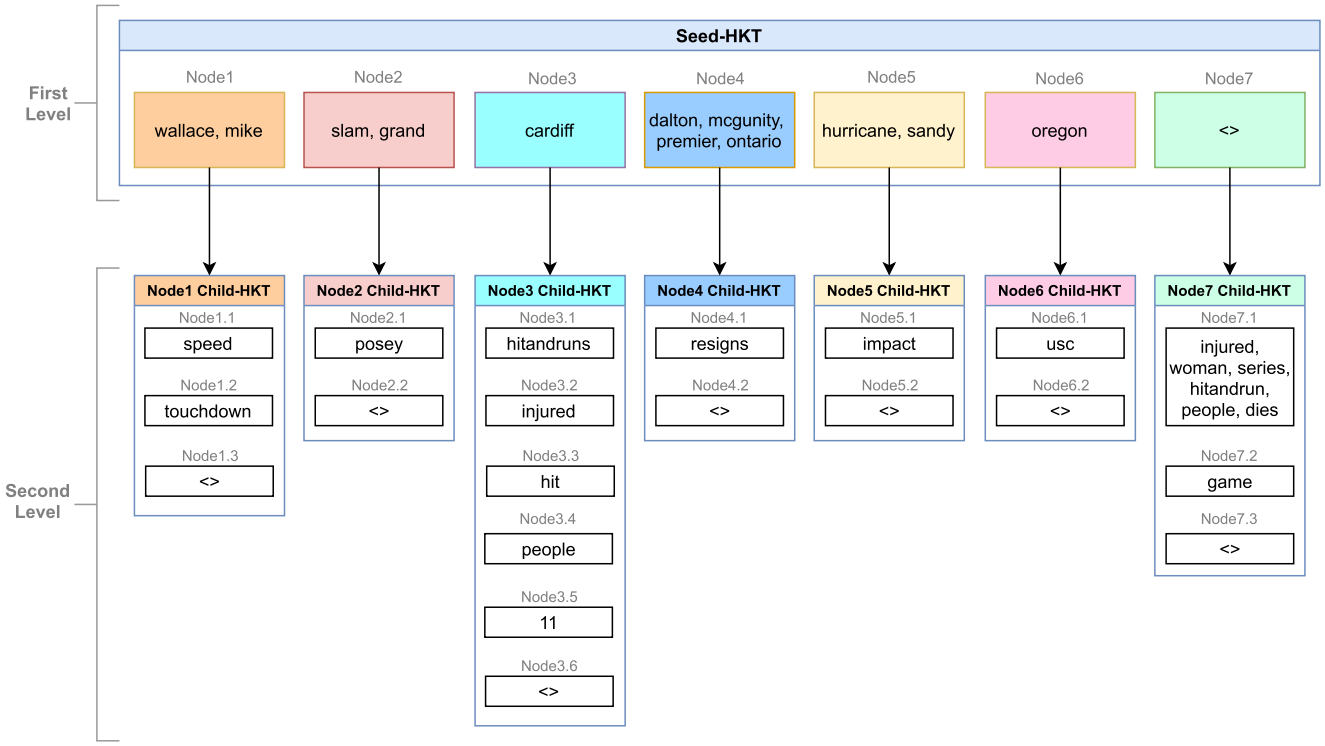

FIGURE 4. A hierarchical tree representation showing first and second level HKTs and their corresponding nodes that are produced using the Contextual Analysis application and the events dataset in Table 2 - Group 5. The first level contains six nodes (Node1 to Node6) that encapsulate the most important words in the corpus and one special node (Node7) with the symbol "<>". This node is treated differently because it represents data not matching the words shown at the same level. Each node in the first level, for this example, is linked to its Child-HKT, which is a container for the other important words in the tweets that contain the words in their parent node. For example, the word "impact" in Node5.1 is the most occurrence word in the tweets that mentioned the words "hurricane" and "sandy" in Node5. Note: this figure only shows the two levels of the created tree which contains other granular details.

VI. CONCLUSION & FUTURE WORKS

In this paper a new approach, based on Contextual Analysis, for text multi-class classification is proposed. Various experiments were carefully designed to measure the performance of the proposed method from two different perspectives: Imbalanced Classes and High Number of Classes. In order to evaluate the performance, a comparative study is conducted, using well known classification techniques such as Naïve Bayes, Support Vector Machines, K-Nearest Neighbors and Convolutional Neural Networks over real-world event corpus form Twitter. On average, the result shows that the proposed method performs well in categorizing short texts (tweets) into various groups (events), with f1>97.09% and f1>95.27% in the imbalanced classes and high number of classes experiments, respectively. For most tasks, this level of performance would be considered to be acceptable. Also, the interpretability assessment reveals this approach is simple and transparent, unlike the other methods used in this study.

There are several issues reserved for future work. While Support Vector Machines and Convolutional Neural Networks fail to satisfy the transparency requirements for interpretability, they outperform the other methods in most of the experiments, with f1>97.35% and f1>97.74%, respectively. Converting Support Vector Machines type methods to transparent methods has proven extremely difficult [7]. Further studies are needed to improve the performance of the proposed Contextual Analysis method without sacrificing the transparency, and to define transparency in a manner which will allow comparison with other methods more directly. The results show that performance of the proposed approach is disadvantaged by the imbalanced training data, careful attention should be devoted in this regard. Future work needs to focus on how new words can be identified dynamically and their potential meaning from their context in real time, in which the data are unlabeled and highly skewed. The Contextual Analysis approach has been shown to give competitive performance in classification tasks without being best in class but due to its transparency could be used for real time problems such as detecting new events as they occur in real time.

REFERENCES

[1] *Internet Live Stats, Twitter Usage Statistics*. Accessed: Mar. 29, 2020. [Online]. Available: http://www.internetlivestats.com/twitter-statistics/
[2] *Twitter by the Numbers: Stats, Demographics & Fun Facts*. Accessed: Dec. 2, 2020. [Online]. Available: https://www.omnicoreagency.com/twitter-statistics/
[3] T. Sakaki, M. Okazaki, and Y. Matsuo, "Earthquake shakes Twitter users: Real-time event detection by social sensors," in *Proc. 19th Int. Conf. World Wide Web (WWW)*, 2010, pp. 851–860, doi: 10.1145/1772690.1772777.
[4] M. J. Sánchez-Franco, A. Navarro-García, and F. J. Rondán-Cataluña, "A naive Bayes strategy for classifying customer satisfaction: A study based on online reviews of hospitality services," *J. Bus. Res.*, vol. 101, pp. 499–506, Aug. 2019, doi: 10.1016/j.jbusres.2018.12.051.

[5] T. Mori and N. Uchihira, "Balancing the trade-off between accuracy and interpretability in software defect prediction," *Empirical Softw. Eng.*, vol. 24, no. 2, pp. 779–825, Apr. 2019, doi: 10.1007/s10664-018-9638-1.
[6] Z. C. Lipton, "The mythos of model interpretability," *Commun. ACM*, vol. 61, no. 10, pp. 36–43, Sep. 2018, doi: 10.1145/3233231.
[7] V. Cherkassky and S. Dhar, "Interpretation of black-box predictive models," in *Measures of Complexity*. Cham, Switzerland: Springer, 2015, pp. 267–286, doi: 10.1007/978-3-319-21852-6_19.
[8] A. A. Aziz and A. Starkey, "Predicting supervise machine learning performances for sentiment analysis using contextual-based approaches," *IEEE Access*, vol. 8, pp. 17722–17733, 2020, doi: 10.1109/ACCESS.2019.2958702.
[9] B. Sriram, D. Fuhry, E. Demir, H. Ferhatosmanoglu, and M. Demirbas, "Short text classification in Twitter to improve information filtering," in *Proc. 33rd Int. ACM SIGIR Conf. Res. Develop. Inf. Retr. (SIGIR)*, 2010, pp. 841–842, doi: 10.1145/1835449.1835643.
[10] K. Lee, D. Palsetia, R. Narayanan, M. M. A. Patwary, A. Agrawal, and A. Choudhary, "Twitter trending topic classification," in *Proc. IEEE 11st Int. Conf. Data Mining Workshops*, Dec. 2011, pp. 251–258, doi: 10.1109/ICDMW.2011.171.
[11] T. Pratama and A. Purwarianti, "Topic classification and clustering on Indonesian complaint tweets for Bandung government using supervised and unsupervised learning," in *Proc. Int. Conf. Adv. Informat., Concepts, Theory, Appl. (ICAICTA)*, Aug. 2017, pp. 1–6, doi: 10.1109/ICAICTA.2017.8090981.
[12] Q. Wang, J. Bhandal, S. Huang, and B. Luo, "Classification of private tweets using tweet content," in *Proc. IEEE 11st Int. Conf. Semantic Comput. (ICSC)*, San Diego, CA, USA, Jan./Feb. 2017, pp. 65–68, doi: 10.1109/ICSC.2017.36.
[13] M. Ashcroft, A. Fisher, L. Kaati, E. Omer, and N. Prucha, "Detecting jihadist messages on Twitter," in *Proc. Eur. Intell. Secur. Informat. Conf.*, Sep. 2015, pp. 161–164, doi: 10.1109/EISIC.2015.27.
[14] N. Alsaedi, P. Burnap, and O. Rana, "Can we predict a riot? Disruptive event detection using Twitter," *ACM Trans. Internet Technol.*, vol. 17, no. 2, pp. 1–26, May 2017, doi: 10.1145/2996183.
[15] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural Comput.*, vol. 1, no. 4, pp. 541–551, Dec. 1989, doi: 10.1162/neco.1989.1.4.541.
[16] D. Rumelhart, G. Hinton, and R. Williams, "Learning representations by back propagating errors," *Nature*, vol. 323, pp. 533–536, Oct. 1986, doi: 10.1038/323533a0.
[17] Y. Hu, Y. Li, T. Yang, and Q. Pan, "Short text classification with a convolutional neural networks based method," in *Proc. 15th Int. Conf. Control, Automat., Robot. Vis. (ICARCV)*, Nov. 2018, pp. 1432–1435, doi: 10.1109/ICARCV.2018.8581332.
[18] A. Al-Hassan and H. Al-Dossari, "Detection of hate speech in Arabic tweets using deep learning," *Multimedia Syst.*, Jan. 2021, doi: 10.1007/s00530-020-00742-w.
[19] E. Alabdulkreem, "Prediction of depressed Arab women using their tweets," *J. Decis. Syst.*, vol. 30, pp. 102–117, Sep. 2021, doi: 10.1080/12460125.2020.1859745.
[20] A. Adadi and M. Berrada, "Peeking inside the black-box: A survey on explainable artificial intelligence (XAI)," *IEEE Access*, vol. 6, pp. 52138–52160, 2018, doi: 10.1109/ACCESS.2018.2870052.
[21] R. Binns, M. Van Kleek, M. Veale, U. Lyngs, J. Zhao, and N. Shadbolt, "'It's reducing a human being to a percentage': Perceptions of justice in algorithmic decisions," in *Proc. CHI Conf. Hum. Factors Comput. Syst.*, Apr. 2018, pp. 1–14, doi: 10.1145/3173574.3173951.
[22] L. H. Gilpin, D. Bau, B. Z. Yuan, A. Bajwa, M. Specter, and L. Kagal, "Explaining explanations: An overview of interpretability of machine learning," in *Proc. IEEE 5th Int. Conf. Data Sci. Adv. Anal. (DSAA)*, Oct. 2018, pp. 80–89, doi: 10.1109/DSAA.2018.00018.
[23] H. Sayyadi, M. Hurst, and A. Maykov, "Event detection and tracking in social streams," in *Proc. 3rd Int. Conf. Weblogs Social Media*, 2009, pp. 1–4.
[24] H. Sayyadi and L. Raschid, "A graph analytical approach for topic detection," *ACM Trans. Internet Technol.*, vol. 13, no. 2, pp. 1–23, Dec. 2013, doi: 10.1145/2542214.2542215.
[25] S. Martinčič-Ipšič, E. Močcibob, and M. Perc, "Link prediction on Twitter," *PLoS ONE*, vol. 12, no. 7, Jul. 2017, Art. no. e0181079, doi: 10.1371/journal.pone.0181079.
[26] L. Praznik, G. Srivastava, C. Mendhe, and V. Mago, "Vertex-weighted measures for link prediction in hashtag graphs," in *Proc. IEEE/ACM Int. Conf. Adv. Social Netw. Anal. Mining*, Aug. 2019, pp. 1034–1041, doi: 10.1145/3341161.3344828.
[27] A. A. Aziz, "Contextual-based approach for sentiment analysis," Ph.D. dissertation, Eng. School, Univ. Aberdeen, Aberdeen, U.K., 2020.
[28] A. Sun, E.-P. Lim, and Y. Liu, "On strategies for imbalanced text classification using SVM: A comparative study," *Decis. Support Syst.*, vol. 48, no. 1, pp. 191–201, Dec. 2009, doi: 10.1016/j.dss.2009.07.011.
[29] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Trans. Knowl. Data Eng.*, vol. 21, no. 9, pp. 1263–1284, Sep. 2009, doi: 10.1109/TKDE.2008.239.
[30] B. Krawczyk, B. Mcinnes, and A. Cano, "Sentiment classification from multi-class imbalanced Twitter data using binarization," in *Proc. Int. Conf. Hybrid Artif. Intell. Syst.*, Jun. 2017, pp. 26–37, doi: 10.1007/978-3-319-59650-1_3.
[31] M. Lango, "Tackling the problem of class imbalance in multi-class sentiment classification: An experimental study," *Found. Comput. Decis. Sci.*, vol. 44, no. 2, pp. 151–178, Jun. 2019, doi: 10.2478/fcds-2019-0009.
[32] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," Jan. 2013, *arXiv:1301.3781*.
[33] University of Glasgow. *Twitter Event Detection Dataset*. Accessed: Nov. 4, 2020. [Online]. Available: http://mir.dcs.gla.ac.U.K./resources/
[34] A. J. Mcminn, Y. Moshfeghi, and J. M. Jose, "Building a large-scale corpus for evaluating event detection on Twitter," in *Proc. 22nd ACM Int. Conf. Conf. Inf. Knowl. Manage. (CIKM)*, 2013, pp. 409–418, doi: 10.1145/2505515.2505695.
[35] *Microsoft.ML*. Accessed: Nov. 4, 2020. [Online]. Available: https://www.nuget.org/packages/Microsoft.ML
[36] C. Souza. *The Accord NET Framework*. Accessed: Nov. 4, 2020. [Online]. Available: http://accord-framework.net
[37] C. Souza. *Keras.NET*. Accessed: Aug. 15, 2021. [Online]. Available: https://scisharp.github.io/Keras.NET/
[38] B. Tang, S. Kay, and H. He, "Toward optimal feature selection in naive Bayes for text categorization," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 9, pp. 2508–2521, Sep. 2016, doi: 10.1109/TKDE.2016.2563436.
[39] W. B. Croft, D. Metzler, and T. Strohman, *Search Engines: Information Retrieval in Practice*, 1st ed. Reading, MA, USA: Addison-Wesley, 2009.
[40] V. Vapnik, *The Nature of Statistical Learning Theory*. New York, NY, USA: Springer-Verlag, 1995.
[41] Y. Yang and X. Liu, "A re-examination of text categorization methods," in *Proc. 22nd Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retr. (SIGIR)*, Aug. 1999, pp. 42–49, doi: 10.1145/312624.312647.
[42] N. Kalchbrenner, E. Grefenstette, and P. Blunsom, "A convolutional neural network for modelling sentences," Apr. 2014, *arXiv:1404.2188*.
[43] Y. Kim, "Convolutional neural networks for sentence classification," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, Oct. 2014, pp. 1746–1751.

SAMI AL SULAIMANI received the B.S. degree in computer science from Sultan Qaboos University, Oman, in 2005, and the M.S. degree in advanced software engineering from the University of Leicester, U.K., in 2014. He is currently pursuing the Ph.D. degree in computing science with the University of Aberdeen, U.K. From 2006 to 2019, he was a software engineer and a system analyst.

ANDREW STARKEY is currently a Senior Lecturer with the University of Aberdeen. He has been awarded an Enterprise Fellowship from the Royal Society of Edinburgh and Scottish Enterprise. He is also responsible for Blueflow Ltd., a spin out company from the University of Aberdeen that proposed a solution for a wide range of data analysis areas, such as financial, textual, and web data, such as blogs and discussion threads, and condition monitoring. His research interests include the area of explainable AI, automated AI, and autonomous learning methods.