IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received May 2, 2020, accepted May 17, 2020, date of publication May 25, 2020, date of current version June 10, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2997330

Light Gradient Boosting Machine for General Sentiment Classification on Short Texts: A Comparative Evaluation

FATIMAH ALZAMZAMI[1], MOHAMAD HODA[2], AND ABDULMOTALEB EL SADDIK[1], (Fellow, IEEE)

[1]Multimedia Communication Research Laboratory, School of Electrical Engineering and Computer Science, University of Ottawa, Ottawa, ON K1N 6N5, Canada
[2]Department of Surgery, Faculty of Medicine, Ottawa Hospital Research Institute, Ottawa, ON K1H 8L6, Canada

Corresponding author: Fatimah Alzamzami (falza094@uottawa.ca)

ABSTRACT Recently, the focus on sentiment analysis has been domain dependent even though the expressions used by the public are unsophisticatedly familiar regardless of the topics or domains. Online social media (OSNs) has been a daily venue for informal conversational contents from various domains ranging from sports and cooking to politics and human rights. Generating specific resources for every domain independently requires high cost and extensive efforts. In response, we propose to build a general multi-class sentiment classifier using our Domain-Free Sentiment Multimedia Dataset (DFSMD). Based on the proven capabilities of Light Gradient Boosting Machine (LGBM) in dealing with high dimensional and imbalance data, we have trained an LGBM model to recognize one of three sentiments of tweets: positive, negative, or neutral. We have conducted extensive comparisons and evaluations for six other standard sentiment classification algorithms and different sets of features including OSNs-specific ones. Our results have shown that LGBM model is the winner among the other six algorithms. It has been also shown that our dataset contains distinguishing characteristics in the three classes. Moreover, hashtag words are shown to be significantly important in capturing the sentiments of tweets. In addition, our findings have revealed the effectiveness of our approach in adapting general-domain sentiment to domain-specific sentiment analysis.

INDEX TERMS Domain-free, datasets, sentiment analysis, gradient boosting, LGBM, XGB, SVM, Naïve Bayes, random forest, logistic regression, machine learning, social media, hashtag, slang.

I. INTRODUCTION

As has become evident, online social media (OSNs) platforms have proliferated in recent years and immense amount of data from different domains is being publicly published online. This data unquestionably contains rich opinion information that could be leveraged in opinion analysis research and applications. However, it is nearly impossible to manually monitor the huge volume of data published online. Consequently, machine intelligence is inevitably necessary to automate the monitoring of online stream of conversations and talks that express various opinions regardless of the aspects involved; let it be preferences, agreements, refutations or even neutrality over a discussed topic. These opinion-rich conversations fall under the umbrella of sentiment analysis, which has been proven to be effective in recognizing opinions in many real-world scenarios [1].

Many OSNs platforms have imposed restrictions on text length. Although Twitter has doubled its character count from 140 to 280, text limitation raises the challenge of extracting useful sentimental clues from such short and unstructured inputs in comparison with long texts [2]. Nonetheless, restricting users to a limited writing space has compelled them to disseminate messages with concise expressions. The usage of hashtag is one example of summarizing and emphasizing an opinion or an emotion of the overall context of a message. With the input limit restriction, we have also observed the emergence of online cultural language that includes slangs, short forms, emojis, etc. This has resulted in contents with a mix of spoken and online language. Another linguistic representational challenge, therefore, has been raised to utilize the online language along with the existing formal sentiment resources. It should be noted that OSNs

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

platforms (Twitter in our work case) is an open environment for different domains. Sport fans use Twitter to cheer for their teams. Political conversations take place on Twitter on daily basis and especially during critical events like elections. Also, people use Twitter to search for and exchange reviews about products, movies, or events. The trend among sentiment researchers is to build sentiment classifiers for each domain independently [1], [3]. This actually is very costly due to the following reasons: (1) Data collection needs to be customized to the target domains. This requires extensive efforts to search and match the required data. (2) Data annotation needs domain experts. It is very hard to have experts agree to annotate a large volume of data, and even if they do, the cost in terms of time and expenses is high, let alone the tediousness of the task. (3) Individual sentiment models share the base knowledge of sentiments (e.g. like, dislike, love, hate) regardless of the domains they fall under. "Ronaldo was a disappointment in today's match" and "Trump is such a disappointment" reflect negative sentiment even though both sentences are from different domains. Therefore, there is redundancy in preprocessing and training sentiment models for different domains. The challenges of the domain-specific sentiment modeling shed the light on the importance of generalizing the sentiment learning independently of any domains. This general knowledge of models would, in turn, help in speeding up the learning process of specific domains or the process of domains adaptation. Instead of learning a model from scratch, the prior knowledge learnt through general sentiment would act as a base knowledge to start from there. Another advantage for building general sentiment classifiers is that reasonable performance in terms of resources and training can be obtained at a lower cost than building a model for every domain. In this work, we focus on building a general sentiment three-class classifier to sort tweets as positive, negative, or neutral. The nature of data shared on Twitter is different from the nature of reviews which tend to be predominantly negative or positive. The first step towards solving the domain-free problem is to create a domain-free sentiment dataset. Our dataset (DFSMD) [4] was collected and annotated using high quality techniques to meet the purpose of this study.

It is a common phenomenon to have data imbalance on sentiment datasets collected through OSNs [5]. This is the case with our DFSMD dataset. Literature [6] suggests that having balanced dataset would improve the learning process. However, it is too expensive and time consuming to balance the data while preserving the natural distribution to avoid biases. Many traditional classification algorithms assume that the training datasets are evenly distributed; SVM is one example [7]. On the other hand, the newly introduced Light Gradient Boosting Machine (LGBM) [8] makes it possible to deal with uneven data distributions. Its natural design allows it to deal with data imbalance through the Gradient-based One-Side Sampling (GOSS) technique. GOSS gives more consideration to the data samples with high error (i.e. difficult cases or minor class in our case) as it assumes that samples with low error are already well-trained and do not need more training. Another advantage of LGBM is that it has the ability to deal with high dimensional data as it is the case in sentiment analysis problems. This is done through using "Exclusive Feature Bundling" technique that reduces the number of features while keeping the effective ones. Finally, it has been proven that LGBM converges the fastest among its siblings in the gradient boosting family. When dealing with sentiment classification, data sparsity and data imbalance are the most common problems [7]. Hence, we believe that LGBM is a great fit to learn a multi-class sentiment using our DFSMD dataset. To the best of our knowledge, our paper presents one of the first works to build LGBM classifier for a general purpose multi-class sentiment classification on short informal texts.

Hybrid approach of lexical and statistical machine learning methods have been popular in recent sentiment classification works [9], [10]. In this work, we follow the same hybrid approach which will be explained in the related work section. We train an LGBM classifier to give the sentiment (i.e. positive, negative, or neutral) of tweets based on the five types of features. It is well known that the quality of classifiers is highly dependent on the chosen features they are trained on. We carefully exploit features that represent social short text (i.e. tweets) and that might contain sentiment signals. Further, we train other six classifiers for evaluation purposes.

In this work, we aim to achieve the following four goals. (1) Build a classification model based on LGBM for general sentiment analysis on short informal texts (i.e. tweets in this study). (2) Investigate how our proposed features would perform on our Domain-Free Sentiment Multimedia Dataset (DFSMD). (3) Investigate the correlation between positive, negative, and neutral terms extracted from our DFSMD dataset. (4) Compare LGBM with six other classifiers (i.e. classic, ensemble algorithms, deep learning) in identifying sentiment of short informal texts. (5) Investigate the effectiveness of adapting general-domain sentiment to domain-specific sentiment analysis. We summarize the contributions of this work as follows:

- Develop a state-of-the-art LGBM-based model for general sentiment analysis on short informal texts.
- Conduct extensive comparisons and evaluations of different sentiment classification algorithms with different feature subsets through a comprehensive set of experiments performed on our dataset (DFSMD).
- Conduct cross-domain sentiment experiments through adapting general-domain to specific-domain sentiment and generalizing specific-domain to general-domain sentiment.

The rest of the paper is organized as follows. Section II explains in details the related work. Section III introduces our dataset (DFSMD). The proposed sentiment framework is presented in Section IV. Sections V-IX describe our sentiment classification approach. In Section X, the analysis of correlation between the three classes is presented. The experiment design and set up are explained in Section XI.

The results and analysis are discussed in details in Section XII. Finally, in Section XIII we conclude our proposed work and findings and discuss future work.

II. RELATED WORK

The interest in sentiment analysis keeps increasing among researchers as it represents a seed for many further research domains [2], [11] such as fine-grained emotion analysis, psychological human needs analysis, and smart cities. Many sentiment analysis works have been done on both long texts (i.e. document-level) [12] and short-texts (i.e. message-level) [13]. The average length of a document-level text is 241 tokens in IMDB dataset [14]. Unlike a long text, the short text has an average length of $\approx$ 81 tokens which is the average length that we have observed in our dataset.

In this work, we focus on analysing short tweets. We believe that short texts provide concise expression and require lower features space than long texts.

Many researchers have focused on domain-specific sentiment analysis [3], [15]. Studies on product reviews and political voting forecasts are examples on domain-specific sentiment analysis [2]. This extends to the methodology that previous studies adopted to collect their datasets. Authors in [16] used emotion keywords to collect tweets while others [9] used domain-related keywords (e.g. event-related or topic-related) to build their dataset.

Furthermore, we have found that these datasets suffer from at least one of four main limitations that contradict with the purpose of this study [4]: (1) ignoring the objectivity part of texts, (2) texts were automatically or noisy annotated, (3) if a dataset is manually annotated, the number of annotators is small, (4) the size of a dataset is small. Studies have repeatedly reported that training simple models using large datasets yields better results than sophisticated models trained over small datasets [17], [18].

For text classification, the performance of classifiers is highly dependent on selected features. The right features will guarantee good learning output. In text classification, BOW using TF or TF-IDF is the most popular feature. Its effectiveness directly depends on the quality of the dataset it was derived from. Most of the sentiment classification studies use BOW as one of the features to build their models [12], [19], [20]. Since BOW ignores the order of words which in turn ignores the context of texts, *n*-gram techniques provide a partial solution to the lack-of-context problem [21], [22]. It has been shown that using BOW and *n*-grams features is insufficient for sentiment learning [9], [23]. Considering specific features containing or representing opinion information has proven to better improve the sentiment learning than when only BOW is used. Authors in the study [10] showed that linguistic feature has enhanced the learning performance over BOW feature on MVSA dataset. Similarly, frequency of POS feature has demonstrated a better classification performance than BOW feature when trained on Latent Dirichlet Allocation (LDA) algorithm [23].

Although individual features such as sentiment lexicon or BOW are necessary for sentiment learning, they are far from enough to yield good results [24]. Integrating BOW with sentiment-rich clue features has shown to be more effective in the sentiment analysis [25]. The integration of emoticons with BOW as proposed in the study [26] has boosted the sentiment learning performance by 13% than when using BOW feature only. Aloufi and El Saddik [9] showed that combining sentiment lexicons and POS features with the BOW feature also improves the sentiment learning process. The use of sentiment lexicons is shown to be necessarily informative for the sentiment classification [27] especially for the minor classes in cases where the classes are imbalanced. The results provided by Niu *et al.* [20] showed that using SentiStrenght sentiment lexicon yielded better performance than BOW-TF for the minor class. When using sentiment lexicons for training sentiment models, we actually combine two learning approaches as suggested in literatures [3], [10], [28], [29]: (1) statistical machine learning approach and (2) lexicon-based approach. When using features other than BOW, the occurrence of feature and frequency of occurrences [27], [30] are the two popular approaches to use in sentiment analysis. We adopt the two approaches in our proposed features.

Previous works on sentiment analysis have focused mostly on support vector machine (SVM), naïve -bayes (NB), logistic regression (LR), random forests (RF), and decision trees (DS) to build sentiment classifiers [2]. The reason that they are the most applied classifiers is due to the better performance they provide in comparison to other classifiers such as k-nearest neighbour (KNN). Niu *et al.* [20] used NB, maximum entropy (ME), and SVM to learn sentiment from texts. The result showed that SVM was the winner among the other classifiers. Bilal *et al.* [31] conducted a similar sentiment analysis research using NB, DT, and KNN algorithms. NB classifier has shown better performance than DT and KNN methods. Another sentiment analysis work done by Wan and Gao [13] on Airline Service twitter dataset, showed that RF outperforms NB, SVM, Bayesian Network, and DT when conducting binary classification while DT outperforms the others when training on three classes. Also, four classifiers were used in training a binary sentiment model in the work [22] and the results showed that SVM was the winner among NB, ME, and stochastic gradient descent (SGD). Recently, deep learning algorithms have achieved very good results in sentiment analysis domain [32]. It differs from machine learning in its ability to learn features directly from data. However, explainability of features and learning can be heuristically understood [33]. In contrary, machine learning along with feature engineering, would easily offer such explainablity and interpretability of learning and feature importance especially for unstructured texts. The availability of sentiment resources created by domain experts, makes it easier and faster to craft features and hence reduce the computational complexity of deep learning. In this work, we propose to use machine learning along with feature engineering in order to understand our dataset (DFSMD) and to provide explainable evaluation of its quality on the sentiment learning.

Recently, researchers have proposed to use ensemble classifiers (a combination of multiple classifiers) to build more accurate sentiment classifiers for textual contents on OSNs [34]. It has been found that ensemble methods are effectively capable of scaling out as data volume increases. In Lin and Kolcz research [35], individual models are trained independently and then evidences from each model are combined for the final prediction. Predictions from the ensemble method have been shown to be better than predictions from individual classifiers. This type of ensemble uses bagging approach and is based on taking the majority votes of all the classifiers [36]. Another type of ensemble uses boosting approach that utilizes the weighted average to build a strong learner from weak ones [37]. Adaptive Boosting (AdaBoost) and Gradient Boosting (GBM) are the most common techniques of the boosting ensemble. In this paper, we have used the GBM method, as it proves to well handle the high dimensionality and high sparseness problems [5], which is an advantage in the case of sentiment analysis. We have experimented with two powerful GBM algorithms: (1) Extreme Gradient Boosting (XGB) [38] and Light Gradient Boosting Machine (LGBM) [8]. Authors in [39] proposed to use XGB with lexical and embedding features for emotional analysis of tweets. Combining the XGB model with the convolutional neural network model has shown an improvement in the overall performance of the proposed system. The capabilities of XGB to cope with large-scale data has allowed the ensemble model to improve its overall performance. Another work [1] proposed to build an XGB sentiment classifier for financial news and headlines. When training on combination of unigram and bi-gram feature, the XGB model has shown to be more effective than when training on other features of TF-IDF and paragraph vector features. Again, a sentiment model learnt using XGB algorithm has shown to outperform other algorithms (i.e. SVM and Gradient Boosting Trees (GBT)) when evaluating Telugu news collected from news websites [40]. The model was trained to recognize the polarity (i.e. positive and negative) of news texts in Telugu language. An LGBM model was trained on telephone conversations data for the purpose of finding the sentiment intensity of the conversations. The LGBM model showed a powerful advantage with 4% better performance than LR model on a combination of text and audio data. TF-IDF was the only textual feature used to train the LGBM. Fan *et al.* [41] built a sentiment model for recognizing the opinions (i.e. positive, neutral, and negative) of English national team fans during FIFA World Cup 2018. They trained LR, XGB, and LGBM models independently on tweets using word-based and character-based TF-IDF features. Then they calculated the weighted average of all the three predictions from the three proposed models as the final predicted result. The results were promising and showed that the sentiment peaked when the England team were scoring victorious. However, the work did not report the performance of individual classifiers; instead, it reported the result after combing all the three classifiers in terms of weighted performance average. In our work, we propose to train both XGB and LGBT using five types of features. To the best of our knowledge, this paper presents one of the first works to build a general sentiment (i.e. domain free) model based on LGBM using our domain-free dataset (DFSMD).

III. DATASET

In order to train our sentiment classifier, we have used the Domain-Free Sentiment Multimedia Dataset (DFSMD) [4]. DFSMD was collected using Twitter Stream API. DFSMD is distinguished from other datasets in the way it was collected and annotated. The data collection was not restricted to any domains, keywords, locations, or any predefined filters. The questions and annotators of the dataset were selected carefully to limit potential biases during the annotation. Furthermore, the annotators of the dataset were selected on the basis of providing sentiment agreement with three expert psychologists. The DFSMD contains 11941 (56%) tweets; 6683 of which are positive, 2275 (19%) are negative, and 2983 are neutral (25%). The dataset was published in an earlier study and is publicly available upon request.

IV. SENTIMENT CLASSIFICATION FRAMEWORK

Figure 1. illustrates our framework for the domain-free sentiment classification for short texts. We followthe same framework used in general classification problems. The data acquisition component is responsible for collecting data based on filter-free criteria. We used Twitter Stream API for this job. Then, the retrieved data underwent a cleaning process to finally generate the Domain-Free Sentiment Multimedia Dataset (DFSMD). Further details on the data collection can be found in our earlier work [4]. Before the data is passed to the sentiment engine, it is split into training and testing sets in order to facilitate the learning and evaluation processes. The sentiment engine consists of four components: data preprocessing, feature engineering, sentiment learning, and sentiment model. In the preprocessing phase, the data is prepared based on criteria to keep the important parts of the data that utilize the sentiment learning (explained in details in Section VIII.). The preprocessed data is then used to extract meaningful features in the feature engineering component, and represented them in a vector format. Both data preprocessing and feature engineering process the data in three phases in order. This is because extracting some features is dependent on the existence of pieces of information that will be removed eventually before the data is fed into the sentiment learning component. A classification algorithm is selected in the sentiment learning component and its parameters are tuned as a prior step to the learning process. While only the training data is fed into the sentiment learning component, both training and testing sets are evaluated by the learnt sentiment model. Finally, the result is expressed as a one class of positive, neutral, or negative. The details of the components are presented in the following sections.

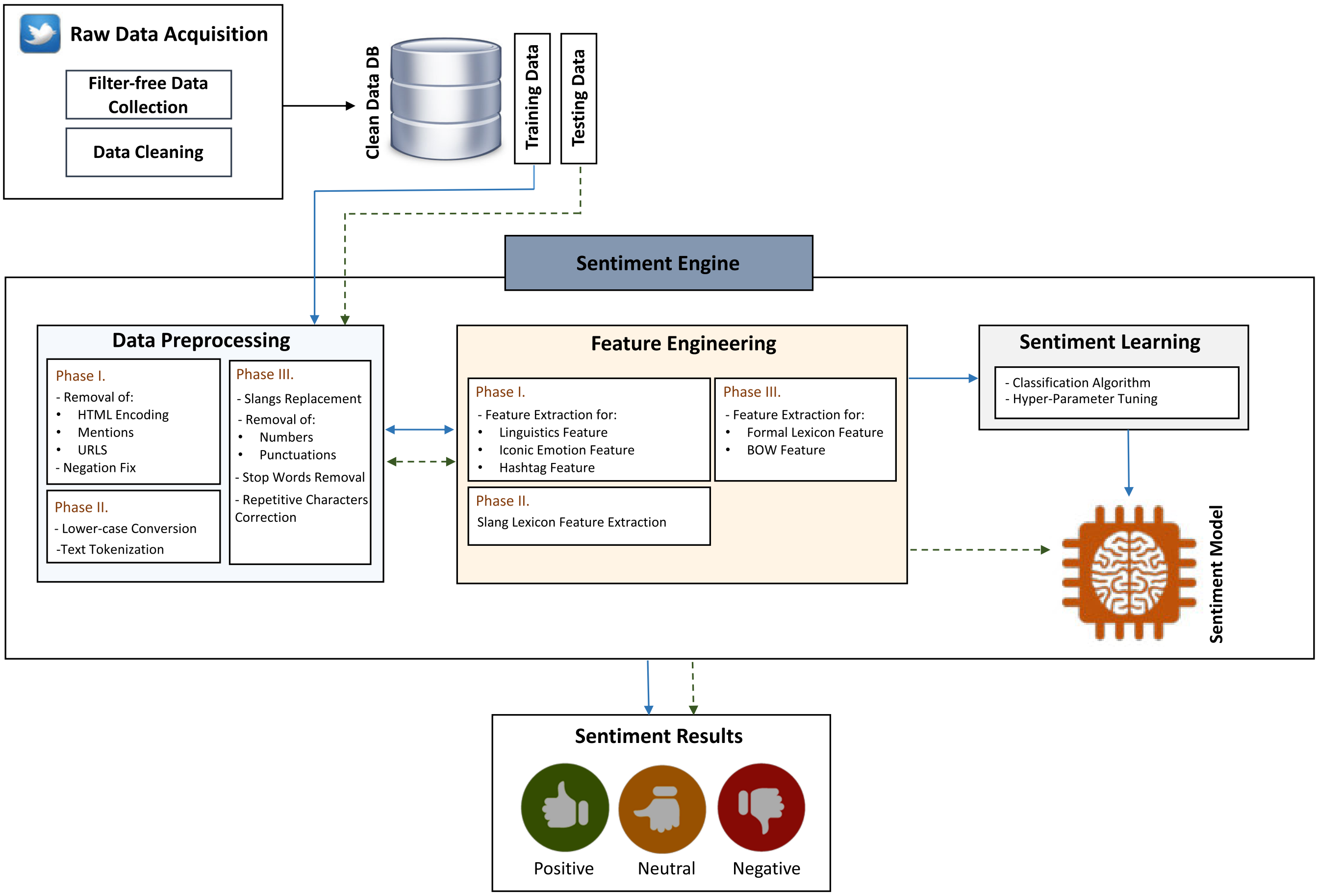


FIGURE 1. Sentiment classification framework.

V. GRADIENT BOOSTING MACHINES

Gradient boosting is one type of ensemble learning. Unlike classic learning approach, ensemble learning approach combine a set of weak learners to construct one strong learner [34]. In contrast to the bagging technique where the models are made independently, the models in the the ensemble boosting technique are made sequentially by iteratively minimizing the error of earlier learnt models [38]. It learns a predictive model by combining the M additive tree models $(f_0, f_1, f_2, \ldots, f_M)$ to predict the results (Eq.1).

$$f(x) = \sum_{m=0}^{M} f_m(x) \tag{1}$$

The tree ensemble model is optimized by reducing the expected generalization error L according to Eq.2:

$$L = \sum_{i}^{n} (y_i - \hat{y}_i)^2 \tag{2}$$

L is a loss function that measures the delta loss between the target y_i and the prediction $\hat{y}_i$ of a data point.

There are three fundamental reasons listed by Dietterich [36] to use an ensemble-based methods:

- **Statistical**: combining and averaging multiple learners provide a better generalization on the learning of data which in turns, reduce the risk of choosing inadequate classifiers.
- **Computational**: during learning, it is computationally difficult for an algorithm to search for a local optima in order to learn the best representation (i.e. decision boundaries) of the data. Neural network algorithms, for instance, utilize gradient descent to minimize the loss function during the training to learn the best model. In this case, there is only one starting point for the local search. With the ensemble algorithms, we have an advantage of having multiple starting points for the local search. This may provide a better approximation of the true function (i.e. decision boundary) than an individual classier does.
- **Representational**: there are cases where a single classifier is not able to learn a decision boundary that separate different classes, or the decision boundary is very complex. Here comes the advantage of the ensemble-based learning where it provides different decision boundaries learnt from different classifiers.

For the reasons mentioned earlier, we believe that using ensemble gradient boosting helps to increase the robustness of classifiers while decreasing their variances and biases. The nature of the boosting technique could decrease errors as it reduces the failures of individual classifiers while optimising their advantages at the same time. Hence, a more reliable model could be produced. In this work, we utilize two powerful gradient boosting algorithms: Extreme Gradient Boosting (XGB) and Light Gradient Boosting Machine (LGBM). They are the state-of-the-art algorithms from the gradient boosting family. XGB was introduced in 2016 [38] and LGBM was introduced by Microsoft in 2017 [8]. We train XGB and LGBM classifiers to give the sentiment class (i.e. positive, neutral, negative) based on the five types of features explained later on in Section IX.

A. EXTREME GRADIENT BOOSTING (XGB)

XGB [38] is an ensemble tree-based method that implements a gradient boosting machine learning framework for regression and classification problems. XGB grows trees using level-wise algorithms. It differs from RF in the way it grows, orders, and combines the results. XGB uses different algorithms for splits finding. Exact Greedy and Approximate algorithms were introduced first in [38]. Histogram-based algorithm was then proposed to be used for the splits finding after LGBT algorithm was invented. When histogram is used, trees grow in leaf-wise manner.

The method works by bucketing features values into group of bins to construct features histogram. The splitting is performed on the bins instead of on the features. The bucket bins are constructed before each tree is built, hence, it speeds up the training which in turns reduces the computation complexity. In this work, we use the histogram method for deciding the best split. During the parameters turning, we found out that the three algorithms yield similar results. Therefore, we decided to go with histogram method since it takes faster training time on large sparse datasets, than the other splitting algorithms.

The decision to make a split is based the loss value that the split produces. The split will happen if the loss value exceeds a certain threshold, otherwise, the split will be ignored. This shows the advantage of leaf-wise gradient boosting methods over the RFs in reducing the number of splits while keeping the quality of the splits.

Furthermore, XGB uses sparsity-aware split algorithm that works on sparse vectorized textual data (i.e. the case in our work). When computing the split, the sparsity-aware split algorithm proposes to ignore the zero features, and then allocates all the data with zero values to the side of the split that reduces the loss the most.

B. LIGHT GRADIENT BOOSTING MACHINE (LGBM)

LGBT [8] is another gradient boosting algorithm that uses a leaf-wise algorithm to grow trees vertically.

A leaf that reduces the loss the most is chosen to split and grow the tree. LGBM uses histogram-based method to find best splits candidates. To improve the training, LGBM uses a sampling algorithm Gradient-based One-Side Sampling (GOSS) to indicate the importance of data instances. Its main function is to concentrate on data samples with larger gradients and ignore the data with small gradients. The assumption is that the data with small gradients have lower errors; thus they are already well trained. Therefore, GOSS proposed to ignore these less-informative data points and use the rest to compute the information gain when finding the best splits. However, this will result in a bias problem towards the sample with larger gradients, and will change the original distribution of the data. To solve this issue, GOSS performs a random sampling on the data with small gradients while keeping all the samples with large gradients. Because the sample would still be biased towards the data with large gradients, GOSS increases the weights (i.e. adding a constant multiplier) of the data instances with small gradients when computing the information gain.

In addition, LGBM uses Exclusive Feature Bundling algorithm to handle sparsity in datasets. It combines mutually exclusive features in a nearly lossless way resulting in reducing the number of features while keeping the most informative ones.

VI. TRADITIONAL CLASSIFICATION ALGORITHMS

In this section, we introduce different learning algorithms, including the Support Vector Machine (SVM), Logistic Regression (LR), Multinomial Naïve Bayes (MNB), Random Forest (RF), all of which are widely used in text classification.

A. SUPPORT VECTOR MACHINE (SVM)

SVM algorithm has shown a robust performance in text classification [42]. The goal of SVM is to select a hyperplane that maximizes the margin between the closest instances of the two classes.

Sentiment analysis in this work is a multi-class classification. SVM is by default a binary-class classifier. We follow "one-vs-all" approach to solve the multi-class classification problem using SVM. Note that we attempt to use linear kernel in our experiments. The initial experiments with non-linear SVM have shown a decreased learning performance.

B. LOGISTIC REGRESSION

Logistic regression is considered one of the best discriminative models. It learns the posterior class probability directly from the training data. Its goal is to find a decision boundaries between classes in the feature space. The posterior probability, in binary classification, is given by applying the *sigmoid* function on a linear combination of given inputs. Logistic regression can be generalized to work on multi-class classification problems by using *softmax* function to derive the posterior probabilities by normalizing a given feature vector to probability values values between [0, 1].

C. MULTINOMIAL NAÏVE BAYS

Naïve Bayes is a probabilistic classifier. Though it is called naive, it performs well in text categorization [43]. The core of the algorithm is based on Bayes theorem. The Multinomial Naïve Bayes (MNB) classifier assumes multinomial distribution so that it can be used with discrete features like words counts in text classification. In our work, we attempt to use the Multinomial Naïve Bayes classifier for our three-class sentiment analysis problem.

D. RANDOM FOREST (RF)

Random Forest (RF) is an ensemble tree-based classification algorithm. It uses bagging techniques in which trees are fully grown to their maximum extent. The trees, in RF, are trained independently using a random sample of data. Every tree in RF is generated based on bootstrapped training instances and a random set of features. Each learnt tree is a weak learner. By combining all the weak learners, we have one final strong model. The overall prediction of the RF is computed based on the majority votes from all the individual weak learners (i.e. individual trees). RF has shown a robust performance to noise and overfitting problems that would affect a single decision tree [44]. Moreover, RF can efficiently handle large size of data and is inherently suited for multi-class problems.

VII. DEEP LEARNING

Deep learning is deep neural networks that use multiple layers of processing units for feature extraction in order to learn performing tasks directly from data. Every layer in the neural network consists of a specific number of neuron units in which their outputs are inputs to the next layer. The links between neurons in each layer are weights. The neural network learns by repeatedly updating the weights after data forwarding through the network. These weights are adjusted in order to minimize the difference (i.e. error) between predicted and actual outputs.

Deep learning has shown robust capabilities for sentiment multi-class sentiment analysis [32], [45]. In our work, we attempt to use Multi-layer perception (MLP) and Long-Short-Term-Memory (LSTM) [46] architectures for our three-class sentiment analysis.

VIII. DATA PREPROCESSING

In this work, we use text to build a sentiment classifier. Therefore, we refer to data preprocessing as text preprocessing. Data preprocessing is the process of converting data to a format that a computer understands. Preprocessing data is a very important step in machine learning in general and in sentiment classification in particular. It builds the resource knowledge for the machine models to learn from. High quality preprocessing ensures the quality of features that the model will learn from. Thus, the quality of the model highly depends on the quality of the dataset and selected features set.

Since we are working on sentiment analysis in this study, we are interested in the features that give a hint of opinion or emotion. For example, verb words like "support" or "hate" provide an opinion unlike pronoun words like "I" or "him" that does not provide any opinion. Therefore, it is very crucial to preprocess the data, to remove excess noise and retain useful information. The existence of noise in the data could affect the learning performance. In this work, the data preprocessing consists of five steps, in order, as follows:

- **Removing encoding symbols**: Since we use Twitter data, we might encounter cases where HTML encodings have not been converted into text. Hence, there was a need to clean this noise.
- **Removing user mention**: User mentions do not provide any opinion hints; therefore, we decided to remove them.
- **Removing URLs**: Even though URLs provide information, they might contain long texts, images, or videos. This requires different preprocessing steps. Therefore, we decided to remove them.
- **Converting text to lower case**: Treating a word that appears capitalized in the beginning of a sentence and the same word appearing lowered in the middle of a sentence would result in redundancy, hence declining performance accuracy. In addition, keeping words with upper case initials is not useful in building our sentiment model since we do not do Entity Recognition or any related tasks. We generally look for words that capture opinion, sentiment, or emotion. However, we exploit the sentiment clues that might exist in all-caps [10] words before we convert texts into lower case.
- **Fixing slang, negation, and repetition**: In OSNs, users tend to use abbreviations to express opinions due to the limited space. Also, they tend to use their daily language (i.e. informal or slang) and probably with the same tone they speak in real life. For example, when a strong opinion occurs, users tend to intensify words that express what they feel. "I LOOOOVE this pic" and "It isn't easy to 4gv" are example of how OSNs people tend to use short form in posts. Human eyes understand the abbreviations when encountering them because they know the origins of these abbreviations. The same analogy applies to learn our models. In our work, we apply three types of cleaners to put the words in our corpus to their original forms: (1) fixing slang: to replace slang words used in OSNs to its original terms like "bff " will be replaced by "best friend forever". (2) fixing negation: to replace abbreviated negation words to its original components. In "she isn't worried" example, we see a problem of the negation word "isn't" after tokenization; the word "t" (i.e. "not") will be meaningless. In "she isnt worried" example, the model will treat "isnt" differently than "isn't" and eventually they will be learnt as different tokens even though they are the same. Hence, this would harm our model learning. As a result, we replace "isn't" and "isnt" by "is not". (3) fixing repetition: to remove character repetition and replace it with a single character. For example, word "niiice" will be replaced by "nice". Note that the repetition is fixed only if the repetition of consecutive characters is greater than 2. This is to ensure the originality of words that inherently consist of two consecutive characters. We believe in the importance of this step as it ensures the generality of learning sentiments of words (i.e. especially when using sentiment lexicons) regardless of their positions in sentences. It is worth to mention that we take a note of all-cap words

and character repetition before the cleaning process in order to use them in feature engineering later on.

- **Removing special characters and numbers**: Numbers and most special characters do not contain sentiment insights; as a result, we decided to ignore them in this paper. However, punctuation-based emoticons and emojis will be extracted and the presence of Twitter special symbols like hashtag(s) will be noted before removing them to be used later in the feature engineering. We believe that the hashtagged text, emoticons, and emojis provide useful information related to opinions or sentiments [4], [47]. Hence, they would improve the performance of the learning.
- **Tokenizing text**: Text tokenization is the process of segmenting the text into meaningful words called tokens. The meaning of the whole text depends on these words. Therefore, it is an essential step in text classification as it helps capture the relations of words in text.
- **Removing stop words**: Many works in the literature proposed to drop stop words when training a textual classifier. This is because some stop words such as "is" and "be" will not drive the sentiment learning [5], [22]. However, removing stop words in the context of sentiment analysis could be problematic especially if the context is affected. For example, if "I, she, is, not" are stop words, then the sentence "I thought she is not happy" would be learnt as a positive sentiment which is not true at all.

  Therefore, we decided to include this cleaning step in this work to investigate whether stop words removal will cause sensitivity to the sentiment performance or not.

Note that all the cleaning steps were implemented using regular expressions. We tried to cover as many cases as possible when designing our REs such as covering all the cases of URLs and usernames. NLTK toolkit was used for tokenization and stop words removal.

IX. FEATURE ENGINEERING

As seen in Section VIII, sentiment classification on large textual datasets requires a lot of preparation work on the back end. This step is important in order to transform a text into a format that an algorithm can use. The transformation process, which involves representing textual data numerically, is called "feature extraction".

Words and other attributes of text represent either discrete (i.e. frequency of words) or categorical (i.e. presence of words) features. In feature engineering, we aim at mapping these words and attributes into real-valued vectors. We have used different techniques to choose the numerical representations of the textual features.

In this work, we have used different types of features, including Bag-Of-Words (BOW), *n*-gram, sentiment lexicons, and linguistic hints. We have also used features representing OSNs culture such as iconic emotion, and hashtag. Detailed description of the features is presented in the following subsections.

A. BAG-OF-WORDS (BOW)

BOW is a well-known technique in text processing. It generates a list of words, called vocabulary, from a dataset. Each tweet is represented as vector with each word represented with a numerical value depending on the used numerical representation method. For example, a word is given 1 value if it is present in the vocabulary or 0 if otherwise. Another technique is the frequency of occurrences of the words of a text in the vocabulary. The two most common approaches to numerically represent a text are: (1) Term Frequency (TF) which represents the number of time a word occurs in a tweet with respect to its total number of occurrences in the whole dataset, (2) Frequency-Inverse Document Frequency (TF-IDF) which represents the level of importance of a word in the whole dataset. In this work, we have adapted the TF approach since it shows better performance over TF-IDF during our initial experiments.

B. n-GRAM

BOW ignores the word order, which results in ignoring the context of texts. To solve this, *n*-gram technique is incorporated to extend the BOW model where a document is represented as n consecutive words [21]. The literature suggests the order of $n <= 3$ consecutive words. In our work, we investigate the impact of using uni-gram, bi-gram, and tri-gram. We use TF approach for our features vector representation.

The *n*-gram feature vector consists of each n consecutive words in a tweet, as seen below:

- **Uni-gram feature**: the vector will be of m dimension where m is the size of our constructed vocabulary. Each item in the uni-gram vector represents a word from the vocabulary list.
- **Bi-gram feature**: the vector will be of $m - 1$ dimension where m is the size of our constructed vocabulary. Each item in the bi-gram vector represents a two-consecutive words from the constructed vocabulary.
- **Tri-gram feature**: the vector will be of $m-2$ dimension where m is the size of our constructed vocabulary. Each item in the bi-gram vector represents a three-consecutive words from the constructed vocabulary.

C. FORMAL-WORD SENTIMENT LEXICONS

Various resources of sentiment lexicons have been developed so that sentiment learning benefit from textual sources [48]. Each lexicon was built based on a philosophy including but not limited to coarse grained or fine-grained sentiment classification: what part of text to annotate, single-word level or *n*-gram level. As a result, we propose to use two different lexicon resources: (1) AFINN-111 Lexicon (AFINN) [49]: it contains 2,477 words which were built based on Affective Norms for English Words (ANEW). Each word is annotated with score ranging from 1 to 5 or from -5 to -1 for positive and negative words, respectively, (2) NRC Hashtag Sentiment Lexicon (NRC) [50]: It contains 54,129 words extracted from 775,000 tweets. The tweets are automatically labelled based on the polarity of hashtag such as "amazing", and "terrible".

We have extracted two features, based on the presence of words from our tweets, in the used lexicons. Each feature represents the frequency of positive and negative words, respectively, for each individual lexicon.

For AFINN lexicon, the features extracted are:

- **Affin-positive-feature**: contains the frequency of positively scored words (i.e. in a tweet) which exist in Affin.
- **Affin-negative-feature**: contains the frequency of negatively scored words (i.e. in a tweet) which exist in Affin.

For NRC lexicon, the features extracted are:

- **NRC-positive-feature**: contains the frequency of positively scored words (i.e. in a tweet) which exist in NRC.
- **NRC-negative-feature**: contains the frequency of negatively scored words (i.e. in a tweet) which exist in NRC.

D. SLANG SENTIMENT LEXICONS

OSNs users tend to use their daily informal language when interacting online [25]. Furthermore, they use many abbreviations for faster communication and due to limited space provided for writing. Hence, the daily informal language used provides a more convenient tool for communication than a formal language does. As a result, we propose to use a sentiment lexicon designed for online slang language to collect useful information related to expressing opinions or emotions which might be missed in the formal-word lexicons. In this paper, we use SlangSDlexicon [51]. SlangSD contains 96462 slang words labelled as positive, neutral, or negative. The three features created, based on the presence of words from our tweets in the SlangSD lexicons, are as follows:

- **SlangSD-positive-feature**: contains the frequency of positively scored words (i.e. in a tweet) which exist in SlangSD.
- **SlangSD-neutral-feature**: contains the frequency of neutral scored words (i.e. in a tweet) which exist in SlangSD.
- **SlangSD-negative-feature**: contains the frequency of negatively scored words (i.e. in a tweet) which exist in SlangSD.

E. OSNs LINGUISTIC HINTS

Beside using slang language in OSNs, the use of intensifiers like all-caps and character repetition would indicate a strong sentiment. Studies have shown that intensifiers are widely used in online conversations [10]. Therefore, we believe that they will be useful to build our classifiers. We extract four types of linguistic-hint features:

- **All-caps presence feature**: contains the presence of all-caped words, like "HORRIBLE", in a tweet.
- **All-caps frequency feature**: contains the occurrences frequency of all-caped words in a tweet.
- **Letter-repetition presence feature**: contains the presence of words with consecutive repetitive letters, like "beeest", in a tweet.
- **Letter-repetition frequency feature**: contains the occurrences frequency of f words with consecutive repetitive letters in a tweet.
- **Exclamation-mark presence feature**: contains the presence or absence of exclamation mark in a tweet. Literature states that the use of exclamation mark could indicate a strong feeling [52]. In addition, sentences ending with exclamation mark would convey emotion rather that stating a fact.
- **Exclamation-mark frequency feature**: contains the frequency of exclamation mark in a tweet. Previous works claim that consecutive use of exclamation mark would increase the attention to the feel of the opinion expressed [52].
- **Question-mark frequency feature**: contains the frequency of question mark in a tweet. Consecutive question marks are a sign of opinion intensification [50].

F. ICONIC EMOTION

Punctuation-based emoticons and emojis have become a ubiquitous part of OSNs culture. Users intensively use them when communicating online. Sometimes, users tend to emphasize them as they express their feelings more than words do, as well as to save space for more information to share. The latter case is especially for Twitter since it limits posts to 140-280 words. Emoticons and emojis have proven to have an important communicative role in areas like opinion expression and conversation ambiguity clarification [53]. In this paper, we follow two approaches to extract the features related to the use of emoticons and emojis: (1) their presence/absence, (2) the sentiment they provide. For the sentiment approach, we utilized AFFIN-emoticons lexicons [49] and emoji sentiment lexicon [54].

For emoticons, the features extracted are:

- **Emoticon presence feature**: contains the presence or absence of emoticons in a tweet.
- **Emoticon-positive frequency feature**: contains the frequency of positively scored emoticons (i.e. in a tweet) which exist in AFFIN-emoticon lexicon.
- **Emoticon-negative frequency feature**: contains the frequency of negatively scored emoticons (i.e. in a tweet) which exist in AFFIN-emoticon lexicon.

For emojis, the features extracted are:

- **Emoji presence feature**: contains the presence or absence of emojis in a tweet.
- **Emoji-positive frequency feature**: contains the frequency of positively scored Emojis (i.e. in a tweet) which exist in emoji sentiment lexicon.
- **Emoji-negative frequency feature**: contains the frequency of negatively scored Emojis (i.e. in a tweet) which exist in emoji sentiment lexicon.

G. HASHTAG

The presence of a hashtag in online posts gives a weight to the aspect it represents whether it is a topic, event, or emotion. It is specific to Twitter and its popularity has expanded to cover all social media arenas such as Facebook, Instagram, and Flicker. A hashtag summarizes the overall opinion of texts. Its short length nature makes the choice of its word(s) reflect stronger feeling or opinion. In the example "players

TABLE 1. Top 10 term frequencies from our dataset, with and without stop words for positive, negative, and neutral classes.

Top 10 term frequencies with stop words for DFSMD analysis

Word	Positive Frequency	Negative Frequency	Neutral Frequency
the	2453	2891	1114
to	2170	1948	931
and	1286	1203	478
you	1403	902	508
in	1085	1056	473
of	1041	1045	520
is	971	1062	445
for	1167	809	478
my	1143	767	338
it	796	896	317

Top 10 Term frequencies without stop words for DFSMD analysis

Word	Positive Frequency	Negative Frequency	Neutral Frequency
exam	442	341	259
just	301	423	171
day	470	266	77
like	270	344	179
love	564	51	38
today	338	181	76
time	208	219	92
tomorrow	107	357	37
new	254	133	101
happy	442	25	9

didn't show their best today. # shame'', the hashtagged word ''shame'' emphasizes a negative sentiment more than the message of the tweet itself. In this paper, we explore hashtag as an individual type of feature. We extract two features from this section:

- **Hashtag presence feature**: contains the presence or absence of hashtags in a tweet.
- **Hashtag frequency feature**: contains the frequency of hashtags in a tweet.

X. CLASSES FEATURES RELATION

The first part of Table 1. shows the frequency of the top 10 words from our dataset in positive, neutral, and negative classes. We have observed that most of the top 10 words are stop words. We have also observed that the frequency of their use appear to be nearly equal among the positive, negative, and neutral classes. For example, the word ''my'' is used ≈ 17% and 16%, and 11% times in positive, negative, neutral classes respectively. Note that the frequency of the word ''my'' for the neutral class appear to be far less than the positive and negative classes. This is due to the class imbalance. Besides, the neutral class has the minority number of instances in comparison to the positive and negative classes. However, the percentage of its term frequency is close to the other classes. This observation follows Zipf's law that states that words with low usage frequency rank are used more often while words with high usage frequency rank are used rarely [55]. From this observation, we claim that the use of stop words to find a relationship between the features of our classes, will not be of help. So, we decided to remove the stop words. In this analysis, we have used 21641 terms, after removing the stop words. The second part of Table 1 shows the top 10 words among the three classes after removing the stop words. Now we can see that some of the words started to give useful information about positive class like ''happy'' and ''love''. They have much high frequency in positive class than in other classes. There are still some high frequent words that provide neutral sentiment (e.g. ''day'' and ''just''); however, those words will not impose an importance in learning the positive class characteristics.

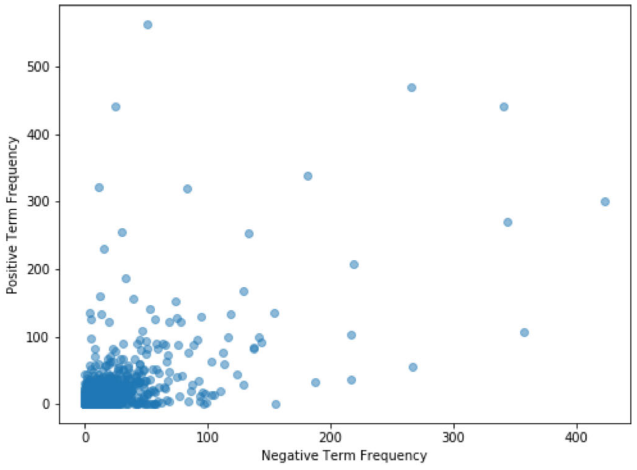


FIGURE 2. Relationship between positive and negative classes in term of term frequency metric.

In order to find a relationship between the features in different classes, we need to decide on a metric that can capture the characteristics of words belonging to each class. By using the frequency metric only, as seen in Figure 2. for the positive and negative classes, we are not able to infer any meaningful relationships between the features of the classes. We have observed that most of the words fall below 600 usage frequency which makes it difficult to infer a meaningful correlation. On the other hand, very few words have high frequency from which we can infer an inverse relation between the words in two different classes. For example, high frequent words in positive classes have low frequency usage in negative class. It is important to mention that stop words were removed for the purpose of analysing our dataset (DFSMD) and were kept for the learning process.

So, the assumption here is that high frequent words that appear in a class more than in another will be useful features to learn that class. Accordingly, we have used Eq.3 to compute the ratio of a word belonging to a class with respect to the total frequency of the same word in all the classes. Also, we have used Eq.4 to calculate the ratio of a word belonging to a class with respect to the overall frequency of the same class.

$$f_{k_i}(w) = \frac{frequency_{k_i}(w)}{\sum_{c=1}^{C} frequency_{k_c}(w)} \tag{3}$$

$$g_{k_i}(w) = \frac{frequency_{k_i}(w)}{\sum frequency_{k_i}} \tag{4}$$

Eq.3 yields good results in cases where the frequency of words belonging to a class is very high compared to the other classes. For example, the word "fabulous" has 7 occurrences in positive class where it appears 0 times in negative and neutral classes. However, the frequency of occurrences of these words is too low to consider as features to learn the class. Therefore, it is not possible to generalize a relationship from this equation. From Eq.4 we could not capture useful characteristics of words to be used as a useful measure to learn distinct classes since the ratio reflects the same information as the word frequencies. Besides, we already have seen the limitation of using only frequency to find a relationship. To overcome these limitations, we use Commulative Distributed Function (CDF) as a metric to reflect the meaning of both equations and, hence, to recognize the characteristics of important words for individual classes. *CDF* at value w is defined as follows:

$$CDF(w) = P(X \leq w) \tag{5}$$

where X is a real random variable and P is the probability that X takes a value $\leq w$.

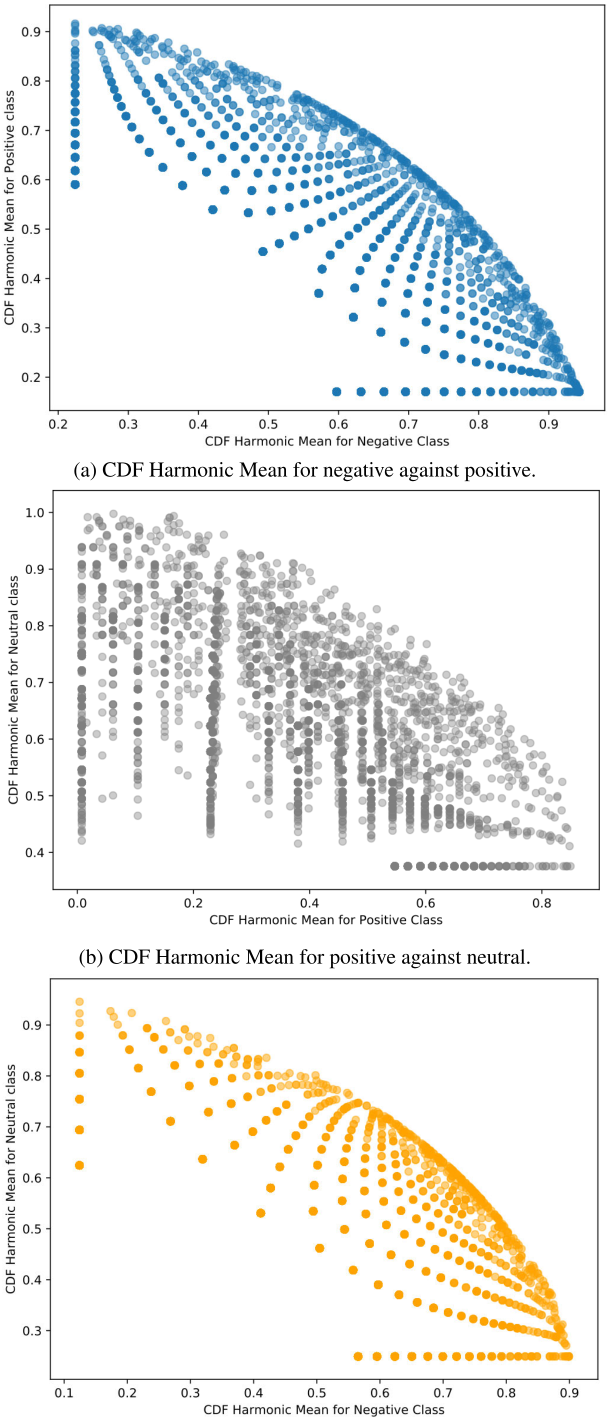
We compute *CDF* for both $f_k(w_i)$ and $g_k(w_i)$ in order to reflect their meanings over the words distributed among individual classes in term of accumulative manner.

Finally, we combine the CDF results for both $f_k(w_i)$ and $g_k(w_i)$ in a hope to provide a better capture of the characteristics of important words for each class. We use Harmonic Mean (Eq.6) due to its nature of equalizing weights given to all data points to avoid any bias towards high data points.

$$H = \frac{n}{\sum_{i=1}^{n} \frac{1}{x_i}} \tag{6}$$

where n is the size of data points and x_i is a data point.

From Figure 3a. we can infer that there is a relationship between words in positive and negative classes. Points with positive/negative high frequency have low negative/positive frequency. This can be seen in the data points close to the upper left corner and points close to the bottom right corner, respectively. Data points with positive CDF harmonic mean greater than 0.5 and less than 0.5 for negative CDF harmonic mean represent the important words for the positive class. The same applies to negative and neutral class as illustrated in Figure 3b. From the same figure, we can see that the number of important words for the neutral class is less than that of the negative class. This is due to the class imbalance for the neutral class. On the other hand, we are not able to draw a clear relationship in the case of positive and neutral classes as seen in Figure 3c. They seem to share many words with close usage frequencies. This is not surprising since neutral words tend to be closer to the positive side than to the negative one. In other words, the negative expressions tend to be strongly subjective. If we have three centroids, one for each class, the centroid of neutral class will be closer to positive centroid than to the negative one.



(a) CDF Harmonic Mean for negative against positive.

(b) CDF Harmonic Mean for positive against neutral.

(c) CDF Harmonic Mean for negative against neutral.

FIGURE 3. Commutative Distribution Function (CDF) Harmonic Mean for class rate and class frequency for every pair of classes: (positive, negative), (positive, neutral), and (negative, neutral).

The findings from this section show the quality of the dataset we proposed to use since we were able to determine the important words for the classes. Therefore, we claim the effectiveness of our word features, as they reflect an obvious association to the sentiment classes, in contributing to sentiment learning process.

XI. EXPERIMENTAL SETUP AND DESIGN

In this section, we will investigate the capabilities of the state-of the art LGBM algorithm in learning a three-class sentiment using five types of features on the DFSMD dataset. We will then conduct evaluation comparisons of LGBM with SVM, Logistic Regression, Multinomial Naïve Bays, Random Forest, and Extreme Gradient Boost (XGB) using the same dataset. The objective of our experiments is to compare different algorithms in order to find the best player with the best settings of features. To achieve this, we propose to conduct four types of experiments: (1) to explore the optimal size of word features, (2) to study the sensitivity of keeping/removing stop words on the sentiment learning, (3) to study the effect of different subsets of features, (4) to investigate the role of hashtagged words and slang words in the sentiment learning from OSNs texts. The DFSMD dataset is used for training and testing and it is randomly split into 60% for training and 40% for testing.

During initial experiments, we observed that our classifiers had produced some wrong predictions on negative and neutral data samples more than on the positive samples. An interpretation of this behavior is related to the fact that the models encountered class imbalance. Previous studies [6] state that relatively balanced class distribution yields better results. By following the same approach of collecting the dataset, we did data augmentation to partially balance the negative class since it has the lowest ratio (19%). The class distributions have become 46%, 33%,21% for positive, negative, neutral classes respectively. It is valid to do data augmentations to fix the imbalance problem [1]. We did not use the sampling technique since we wanted to keep the data natural as much as possible, and to avoid creating biases in our data, as well.

For the evaluation metrics, we use accuracy, precision, recall and F-score as they are commonly used in classification evaluation. Since our problem is a multi-class classification and our dataset is imbalanced, we attempt to use micro average F-score for the evaluation. Micro average F-score is computed by aggregating the contributions from all the classes instead of averaging individual contribution for each class like in macro-average F-score. The accuracy is defined as the ratio of the correct predicted samplers to the total number of samples in a test set. Precision, recall and F-score give a better view of model performance than accuracy alone does. They are calculated as illustrated in the following equations:

$$\begin{aligned} Precision &= \frac{TP}{TP+FP} \\ Recall &= \frac{TP}{TP+FN} \\ F-score &= \frac{2\times P\times R}{P+R} \end{aligned} \tag{7}$$

XII. RESULTS AND ANALYSIS

In this section, we present empirical results of the experiments we have conducted based on the design explained in Section XI. Subsection XII-A. reports the results of the words feature sizes and the effect of stop words using uni-gram, bi-gram, and tri-gram models. Subsection XII-B. presents the results and analysis of the effect of using different feature subsets on the sentiment learning. The results of hashtag and slang words contributions are discussed in Subsection XII-D.

A. SIZE OF WORD FEATURES

In this experiment we will investigate the optimal number of word features to be used in building our sentiment classifier. Based on the findings from Section X, we claim that the words extracted from our dataset have distinguishing characteristics for classes especially for the positive and negative classes. From our dataset, we have extracted 21310 vocabulary words including stop words and 21030 words excluding stop words, which is a large number. Therefore, we need to find out a reasonable number of words to use as features to train our sentiment classifiers. We choose to examine word sizes of 1000, 2000, 3000, 4000, and 5000 on six classifiers using BOW and n-gram features. Note that the sizes represent the maximum number of words based on their term frequency TF. For n-gram models, we examine uni-gram, uni-bi gram, and uni-bi-tri gram models. While investigating the number of features, we explore the impact of removing and keeping stop words on the sentiment learning process. As a result, our experiments are conducted on six different combinations of BOW and n-gram models with and without stop words. The results are illustrated in Figure 4.

From the results, we see that the best performance occurs with uni-gram model when keeping stop words. This finding should not be surprising since we use short texts (i.e. tweets) of length average of ≈ 76 tokens. In other words, stop words seem to be of importance in learning sentiment in this work. On the other hand, uni-bi gram and uni-bi-tri gram models work better when removing stop words. However, their overall performance did not exceed uni-gram model when stop words were kept. As a result, we consider uni-gram model with keeping stop words in our sentiment classification.

According to the results depicted in Figure 4., the optimal size window seems to differ between *uni*, *bi*, *tri* grams when keeping or removing stop words, among the six classifiers.

For example, the logistic regression (LR) and Multinomial Naïve Bays (MNB) models have the best performance at words size of 5000 when using *uni*-gram with stop words. Words size of 1000 shows to be the best with Support Vector Machine (SVM) model when using *uni*-gram and keeping stop words. Random forest (RF) classifier shows best similar results when using 4000 and 5000 words with and without stop words on *uni*-gram model and with stop words on *uni*-bi gram, respectively. For Gradient Boost (XGB) model, the best performance appears to be similar among all the size windows on uni-gram including and excluding stop words and $(uni-bi-tri)$-gram including stop words. Even though $(uni-bi-tri)$-gram model with stop words seems to have the best performance, the difference in performance accuracies is almost negligible. Finally, *uni*-gram with stop words wins

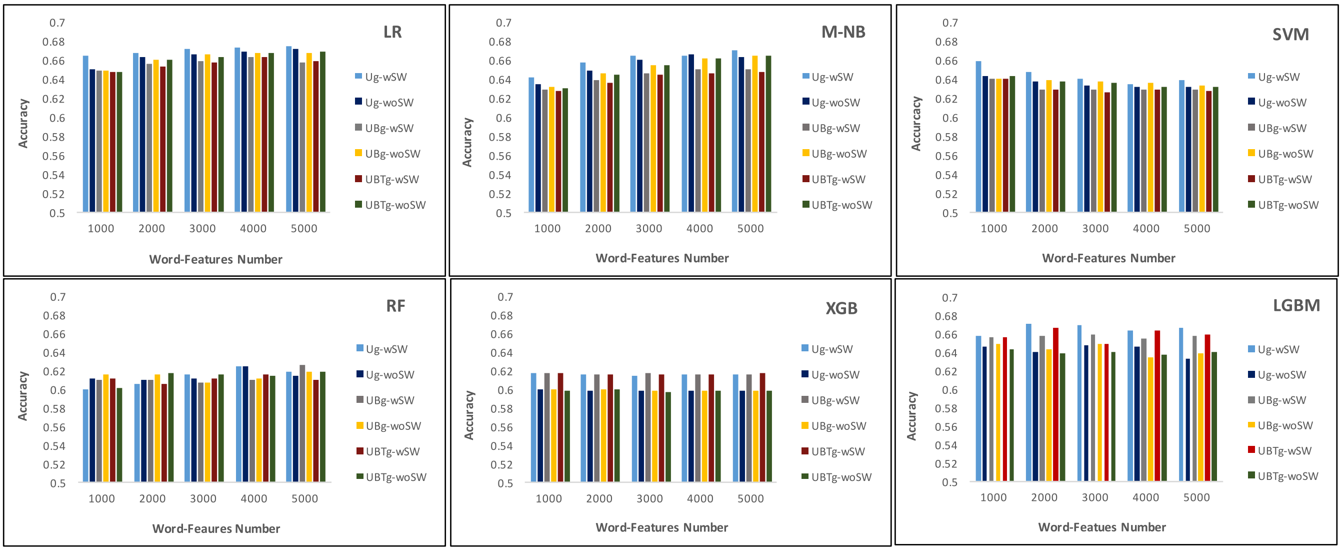


FIGURE 4. Performance of TF-BOW and *n*-gram with/without stop words using different vocabulary sizes evaluated by accuracy. Ug, Bg, and Tg stand for uni-gram, bi-gram, and tri-gram, respectively. wSW, woSW stand for with stop words and without stop words, respectively.

TABLE 2. Performance of fifteen cross features sets on six classifiers evaluated by accuracy and F-score.

	LR			M-NB			SVM			RF			XGB			LGBM		
Features	**Accuracy**	**F-MacAve**	**F-MicAve**	**Accuracy**	**F-MacAve**	**F-MicAve**	**Accuracy**	**F-MacAve**	**F-MicAve**	**Accuracy**	**F-MacAve**	**F-MicAve**	**Accuracy**	**F-MacAve**	**F-MicAve**	**Accuracy**	**F-MacAve**	**F-MicAve**
BOW	67.65	0.611	0.68	66.18	0.611	0.672	64.68	0.6	0.654	61.68	0.531	0.604	60.51	0.465	0.605	65.86	0.573	0.656
Formal Lexicon (FL)	62.32	0.455	0.62	61.7	0.448	0.614	62.3	0.454	0.62	61.89	0.48	0.625	62.46	0.48	0.624	61.96	0.479	0.624
Slang Lexicon (SL)	46.26	0.21	0.46	46.26	0.21	0.46	46.26	0.21	0.46	48.53	0.342	0.485	50.07	0.35	0.492	48.41	0.348	0.492
Linguistic (Lngs)	48.84	0.355	0.499	47.84	0.274	0.483	48.84	0.355	0.499	48.78	0.355	0.5	50.28	0.355	0.498	48.83	0.355	0.499
Emoticon-Emoji (Emt-Emj)	47.2	0.246	0.475	46.62	0.25	0.47	47.22	0.246	0.475	47.2	0.246	0.475	47.58	0.247	0.475	47.2	0.247	0.475
Hashtag	45.88	0.246	0.475	45.88	0.25	0.47	45.88	0.246	0.475	45.88	0.246	0.475	46.3	0.247	0.475	45.88	0.247	0.46
FL+Lngs	62.27	0.478	0.622	61.16	0.453	0.612	62.35	0.477	0.622	60.96	0.486	0.605	64.55	0.502	0.622	62.49	0.505	0.619
FL+Lngs+SL	63.23	0.497	0.634	61.87	0.468	0.625	63.11	0.491	0.632	58.76	0.519	0.594	65.09	0.512	0.636	63.04	0.527	0.634
FL+Lngs+SL+Emt-Emj	64.8	0.541	0.648	63.85	0.472	0.637	64.94	0.512	0.647	60.47	0.546	0.604	66.19	0.561	0.651	64.79	0.57	0.648
FL+Lngs+SL+Emt-Emj+Hashtag	65.79	0.57	0.653	64.68	0.495	0.649	65.29	0.54	0.655	61.78	0.543	0.605	66.58	0.584	0.655	65.68	0.584	0.648
BOW+FL	69.01	0.582	0.682	68.5	0.636	0.666	65.03	0.63	0.685	65.22	0.571	0.646	68.53	0.591	0.66	69.1	0.63	0.688
BOW+FL+SL	68.51	0.59	0.687	68.5	0.517	0.669	65.49	0.617	0.686	64.96	0.54	0.643	68.5	0.542	0.661	68.65	0.603	0.68
BOW+FL+SL+Lngs	69.03	0.626	0.69	68.75	0.627	0.688	66.1	0.611	0.661	65.23	0.559	0.652	69.81	0.566	0.673	69.15	0.616	0.692
BOW+FL+SL+Lngs+Emt-Emj	70.13	**0.647**	0.704	69.19	**0.637**	**0.697**	66.25	0.629	0.674	66.2	**0.573**	**0.665**	70.46	0.591	0.684	70.96	0.647	0.707
All	**71.1**	0.635	**0.707**	**69.74**	0.552	0.69	**67.86**	**0.648**	**0.707**	**66.25**	0.551	0.656	**71.01**	**0.612**	**0.687**	**71.79**	**0.654**	**0.712**

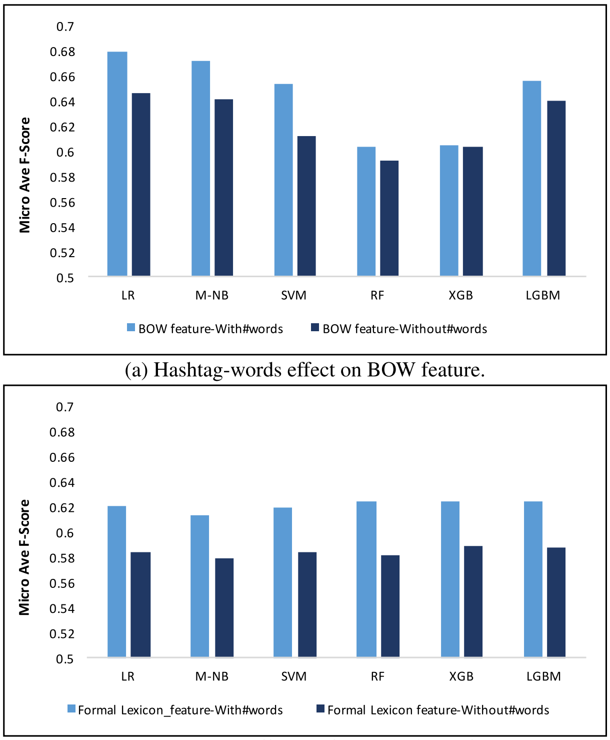
at 2000 size window when training Light Gradient Boosting Machine (LGBM) classifier.

Since we consider uni-gram with stop words model as it yields the best performance for all the classifiers, we consider the size windows where each classifier works the best. Then, we average all the sizes and use the average as the maximum number of word features to train our classifiers for the rest of the experiments. The average of maximum number of features used in this work is 3000 words.

B. CROSS FEATURES SUBSETS

Table 2. shows the results of six sentiment classifiers of fifteen cross combinations of our proposed feature types explained in Section.IX. We split the experiments into four stages: (1) examining individual feature types, (2) examining formal lexicon (FL) feature with different combinations of the rest of the feature types excluding bag-of-words (BOW) feature, (3) examining BOW feature with different combinations of the rest of the feature types. We decided to examine FL and BOW as main features in stage 2 and 3 due to their highest contributions in sentiment learning among the other feature types. From the same table, we can see that using BOW features only yielded the highest performance accuracy and F-score among all the classifiers, followed by FL feature. While their accuracy scored above 60% and average F-score (i.e. micro) is above 0.60, the rest of feature types scored less than 50% accuracy and 0.50 micro average F-score when used individually. Linguistic hint features (Lngs) seem to perform better than iconic emotion (Emt-Emj), slang lexicon (SL), and hashtag that comes at the lowest performance rank. This finding shows an evidence of the association between our word features and sentiment classes. Hence, our classifiers are able to properly learn the distinguishing characteristics for each class. In addition, the result of using FL feature shows that a high percentage of our dataset vocabulary is contained in the lexicons, which also shows the distinguishing characteristics of our word features. Moreover, we have observed that the more formal lexicons we add, the better vocabulary coverage we obtain, which in turn improves the sentiment learning. In this work, we have decided to use two formal lexicons as discussed in Section IX.

Despite the low performance that slang, linguistic, iconic emotion, and hashtag features show when used individually, they still provide some sentiment signs that would enrich



(a) Hashtag-words effect on BOW feature.

(b) Hashtag -words effect on formal lexicons feature.

FIGURE 5. The effect of hashtag words on the performance of six classifiers when training on BOW and formal lexicons individually.

the sentiment learning. This can be seen from the results of stages 2 and 3 in two observations. First, their performance accuracy is approximately close to 50% and micro average F-score approximately close to 0.50. Even though this percentage is not high enough, it still shows that these features might carry some sentiment information that could be used as supplementary features along with BOW and formal lexicon features. Second, the more features we combine the better learning performance we obtain. In stage 2, LGBM model trained solely on formal lexicon features yields performance accuracy of $\approx$ 62% (micro average F-score of 0.62). When combining formal lexicon with linguistic feature the performance stays the same; however, when adding the slang feature, the accuracy and F-score slightly improved to 63% and 0.63, respectively. The performance accuracy improves to $\approx$ 64.8% (micro average F-scoreof $\approx$ 0.65) with the addition of the iconic emotion feature. When combining formal lexicon features with the rest of the features, the performance improved by $\approx$ 3% more than when the model was trained on formal lexicon alone. The other five classifiers show similar results.

In stage 3, we consider BOW as a base feature and combine it with different subsets of features. Combining the two most contributing features has shown to boost the performance by $\approx$ 7% more than when LGBM, LR, MNB, XGB models were trained on only FL feature. The same classifiers trained on BOW only improved by $\approx$ 2-3% when combining BOW and FL features. As illustrated in Table 2., combining BOW with all the feature types yields the best performance among all the classifiers and all different feature subsets. Compared to the performance of models when trained only using BOW, accuracy of LGBM model has improved from 65.86% to 71.79% when combining all the features together. The same results apply to LR, MNB, SVM, RF, and XGB when all the features are combined. Even though slang lexicon feature and linguistic feature show to have sentiment signals, they do not seem to add value when they are not combined together in a feature subset. This can be seen in two cases: (1) when linguistic feature is combined with formal lexicon feature, (2) when slang lexicon feature is combined with BOW and formal lexicon features. However, when slang lexicon and linguistic features are combined together, they seem to add a little value to the learning process. This shows that the tweets might contain slang words written with intensifiers such as ''WTH'' and ''looool''. Moreover, precision and recall of the linguistic feature for neutral class is zero among all the classifiers. This is not surprising because intensifiers like all-caps and repetitive-characters are usually used when there is an opinion that needs to be stressed on. On the other hand, the precision and recall for positive and negative class are quite reasonable and reflect the performance of the classifiers and the distribution of the positive and negative classes. In addition, internet language of emoticons, emojis, and hashtags have shown to have an impact on the sentiment learning among all the classifiers. Emoticons and emojis have proven to enhance the classification accuracy from 63% to $\approx$ 64.8% in LGBM model when combined with formal lexicon, linguistic, and slang lexicon features. Furthermore, adding the hashtag feature to the previous combination has shown to increase the LGBM model accuracy to $\approx$ 65.7%. This result is consistent when combining emoticons and emojis with BOW, formal lexicon, slang, and linguistic feature while training LGBM model. The performance accuracy has improved from 69% (i.e. using BOW and FL) to $\approx$ 71%. It further improved after adding the hashtag feature to reach $\approx$ 71.8%. In terms of precision and recall for neutral class when training using only emoticons and emojis features, they are shown to have zero values. Again this is not surprising; emoticons and emojis are usually used for emotional expression such as ''smiley face'' and ''angry face''. Based on this finding, we have observed that people use emoticons and emojis to express subjective opinion rather than objective ones.

LGBM model shows the best learning performances in term of accuracy and F-score, followed by LR. Further, MNB and XGB models are shown to be strong competitors followed by SVM model. However, the only problem with XGB is that it is too slow to converge in comparison with the other five algorithms. This result is consistent with the findings of LGBM's authors [8] where XGB has shown a noticeably slower convergence rate and less learning performance than LGBM. RF classifier is shown to have the weakest performance among the rest. This shows the efficiency of gradient boosting algorithms compared to the bagging technique

TABLE 3. F-score, precision, and recall for positive, negative, and neutral classes when using all proposed features.

	LR			MNB			SVM			RF			XGB			LGBM		
	F-Score	**Precision**	**Recall**	**F-Score**	**Precision**	**Recall**	**F-Score**	**Precision**	**Recall**	**F-Score**	**Precision**	**Recall**	**F-Score**	**Precision**	**Recall**	**F-Score**	**Precision**	**Recall**
Positive	**0.79**	0.75	0.83	0.78	0.71	0.88	0.78	0.75	0.81	0.75	0.67	0.84	0.77	0.73	0.82	**0.79**	0.77	0.81
Negative	0.76	0.71	0.8	0.73	0.67	0.82	0.76	0.73	0.8	0.7	0.68	0.72	0.74	0.71	0.77	**0.77**	0.74	0.8
Neural	0.36	0.5	0.28	0.14	0.59	0.08	0.4	0.48	0.34	0.2	0.39	0.14	0.33	0.45	0.26	**0.41**	0.48	0.35

TABLE 4. F-score, precision, and recall for positive, negative, and neutral classes when using slang lexicon feature only.

	LR			M-NB			SVM			RF			XGB			LGBM		
	F-Score	**Precision**	**Recall**	**F-Score**	**Precision**	**Recall**	**F-Score**	**Precision**	**Recall**	**F-Score**	**Precision**	**Recall**	**F-Score**	**Precision**	**Recall**	**F-Score**	**Precision**	**Recall**
Positive	**0.63**	0.46	**1**	**0.63**	0.46	**1**	**0.63**	0.46	**1**	0.61	0.497	0.789	0.608	**0.502**	0.77	0.609	0.501	0.776
Negative	0	0	0	0	0	0	0	0	0	0.407	0.463	0.364	**0.44**	0.47	**0.413**	0.435	**0.472**	0.404
Neutral	0	0	0	0	0	0	0	0	0	**0.008**	**0.106**	**0.004**	0.002	0.091	0.001	0	0	0

adopted by RF. Gradient boosting techniques used in LGBM and XGB make use of trees with fewer yet better quality splits instead of growing trees to their maximum extent. Also, the techniques used in LGBM shows their strength in dealing with difficult cases (i.e. neutral class in our case since it is the minority class). LGBM has learnt neutral class with F-score value of ≈ 0.41 in comparison with 0.4, 0.36, 0.32, 0.20, 0.13, for SVM, LR, XGB, RF, MNB (shown in Table 3.). As a result, we attempt to use LGBM, with all the proposed features to build our general sentiment classifier using our dataset.

The main limitation of this work is the data imbalance, especially for the neutral class. Despite this fact, our models give good performance results even on neutral class for some classifiers. Positive and negative classes scored the highest learning performance values. For some of the models, an acceptable performance was yielded for the neutral class. Table 3. illustrates our models performances across the three classes, positive, negative, neutral. From this result, we have observed that our dataset contains separating characteristics for the three classes in which the classifiers could successfully learn from. Further, the proposed OSN-specific features have proven their supplementary effect on the sentiment learning. Another limitation is that this study focused on predicting explicit sentiments from social media texts but was not designed to predict implicit sentiments contained in sarcastic texts. Despite this limitation, the iconic features (i.e. emojis and emoticons) could assist in recognizing sentiments in this case.

C. COMPARISON OF DEEP LEARNING AND TRADITIONAL LEARNING

In this section, we will compare the performance of deep learning with that of traditional learning on the same proposed features (i.e. all features combined), in order to obtain fair results.

Two experiments were conducted on two types of deep learning architectures:

- **MLP**: the input to the MLP network is our proposed feature vector. Two dense (i.e. hidden) layers were used to construct the network. We added one fully-connected softmax layer which outputs the probability distribution over the three classes.
- **LSTM**: the input to the LSTM network is our proposed feature vector. One LSTM layer was used to construct the network. We added one fully-connected softmax layer on top of the LSTM, which outputs the probability distribution over the three classes.

TABLE 5. Comparison between Deep learning and traditional learning on all-features combined in term of learning performance and training time.

	Accuracy	F-MacAve	F-MicAve	TrainingTime (sec)
LR	71.1	0.63	0.707	0.21
MN-NB	69.74	0.552	0.69	**0.07**
SVM	67.86	0.648	0.707	0.30
RF	66.25	0.551	0.656	0.96
XGB	71.01	0.612	0.687	39
LGBM	**71.79**	**0.654**	**0.712**	0.98
LSTM	69.42	0.65	0.69	43.46
MLP	68.42	0.65	0.68	11.75

Table.5 shows the results of deep learning in comparison with those of traditional learning in terms of learning performance and training time. From Table. 5, it can be seen that gradient boosting machines and logistic regression classifiers outperform deep learning classifiers in both learning performance and training time. LGBM model yields the best learning results with accuracy score of ≈ 72% and micro F-score value of 0.71 in comparison to MLP, LSTM models which yield ≈ 68%, 69% accuracy and 0.68, 0.69 F-score. In addition, LGBM requires ≈ 1 second to train the classifier, whereas deep learning models take at least 12 (i.e. up to 43 more times in our case) more times to train (≈ 12 seconds). This highlights the requirements of high-end hardware resources for deep learning in comparison with traditional machine learning that requires low-end hardware to perform the learning process.

These results show the effectiveness of traditional learning (i.e. with LGBM as the winner) compared to deep learning when features are well engineered in a way they provide visible patterns to learning algorithms. Contrary to machine learning, deep learning reduces the complexity of extracting

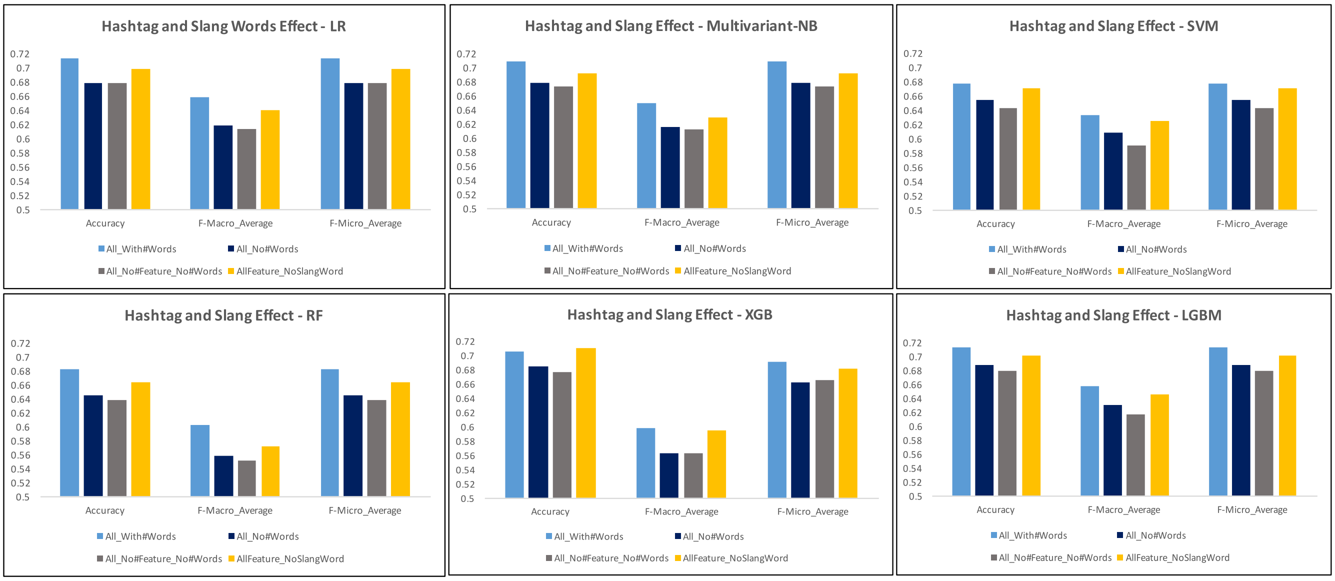


FIGURE 6. Effect of hashtag and slang words on the sentiment learning for six classifiers. All_With#Words refers to all features with hashtag word. All_No#Words refers to all features without hashtag words. All_No#Feature_No#Words refers to all features except the hashtag one without hashtags words. AllFeature_NoSlangWord refers to all features without slang words.

features for every problem; it learns high and low level features directly from data [32]. Based on these results, we conclude that machine learning is more effective than deep learning in terms of learning and complexity when features are well identified and extracted per domain. Compared to deep learning in this case, machine learning algorithms require less training time and less computational complexity.

D. HASHTAG AND SLANG WORDS

Even though using hashtag features did not contribute much to the sentiment learning for all the six classifiers, the existence of hashtagged words shows to be important. Removing the hashtagged words has shown a negative impact on the classification performance. The classification performance, in terms of accuracy and F-score, for all the six classifiers has decreased when the hashtagged words were removed as illustrated in Figure 6. In LGBM classifier; the accuracy dropped from 71.8% to 68.8% and the same results are shown for the corresponding F-score.

This result highlights the weight of hashtagged words as they contain valuable information directly related to the overall sentiment of tweets. The overall opinion or feeling of a tweet can be highlighted in these hashtags. From this we can conclude that hashtags are used as effective feature-keywords that describe the overall opinion of a message. Also, the results show that the number of hashtags and whether they exist or not in messages, are not as important as the hashtag words themselves. The existence of the hashtag words extends the vocabulary size and enriches its value with sentiment-related additional words. Hence, it provides models with more words to learn from especially, for BOW and lexicon features as seen in Figure 5. Our finding is supported by the results obtained by Mohammad and Bravo-Marquez [47]. They found that removing emotional-word hashtags has caused the emotion intensity of tweets to drop. This indicates that the hashtagged words are not redundant within texts in terms of the overall opinion recognition.

In lexicon features, it is clear that the number of matched words between our vocabulary and the lexicons increased when the hashtagged words are kept. Figure 6. implies that the hashtag words contain opinion information and this is shown in the results of when the hashtag words are kept in the vocabulary in comparison to when they are removed. In other words, hashtagged words proved to be important as they contain sentimental insights that could enhance the classification learning. Although the hashtag features used in this work (i.e. number of hashtag words, boolean existence of hashtag(s)) are not as important as the hashtagged words themselves, they are still shown to be of assistance and could be used as supplementary elements to learn sentiments. This is illustrated in Figure 6. in all the classifiers except the LR. The positive difference in the learning performance is very small; however, it increases the learning capabilities. As a result, we have decided to keep considering them as features for our models.

We have also investigated the impact of replacing slang words with their original formats on the sentiment learning in this work. Compared to removing hashtagged words, the impact of ignoring slang words is shown to be less harmful on the sentiment learning for all the classifiers. This implies two possibilities: (1) the number of slang words after replacing them with their original forms, has less coverage in the formal lexicons, (2) the slang words do not contain as much sentiment information as hashtagged words do. This explanation is consistent with our results in term of F-score when using the slang lexicon feature only for training our models. When using the slang lexicon feature

TABLE 6. Details of three sentiment datasets used in cross-domain experiments. DFSMD is used for general domain sentiment, IMDB is used for movie reviews domain sentiment, and CL'16-'17 is used for sport domain (Champions League) sentiment.

	DFSMD [3]	**IMDB** [56]	**CL-'16-'17** [9]
Source	Twitter	IMDB	Twitter
Text Length Ave	81	1270	82
Instances	14,488	25,000	14,000
No. of Positive	6683	11500	5150
No. of Negative	4822	11500	4275
No. of Neutral	2983	-	5442

TABLE 7. Sentiment performance of our general sentiment model LGBM on two domain-specific datasets: IMDB movie reviews and CL'16-'17 tweets.

	Positive			**Negative**		
	Precision	**Recall**	**F-Score**	**Precision**	**Recall**	**F-Score**
General DFSMD -> IMDB	0.62	0.51	0.56	0.6	**0.67**	0.63
General DFSMD -> CL'16-'17	**0.74**	**0.71**	**0.73**	**0.71**	0.64	**0.67**

TABLE 8. Sentiment performance of two domain-specific LGBM models trained individually on IMDB and CL'16-'17 datasets. The sentiment performance was evaluated on our general sentiment dataset DFSMD.

	Positive			**Negative**		
	Precision	**Recall**	**F-Score**	**Precision**	**Recall**	**F-Score**
IMDB -> General DFSMD	0.69	**0.94**	**0.79**	**0.82**	0.41	0.54
CL'16-'17 -> General DFSMD	**0.79**	0.58	0.67	0.81	**0.43**	**0.56**

only, the micro average F-score for all the six classifiers is below 50%, as seen in Table 4. The precision and recall are above 60% for the positive class, while they are either zero or below 45% for the negative and neutral class. For example, the classifiers LR, MNB, and SVM have zero values of precision and recall for both negative and neutral class while LGBM have precision and recall of zero for the neutral class only. This means that most of the slang words with feeling are found to be positive. The reason for this might be because the positive class have the majority of tweets in our dataset. Another reason that would explain the slight effect of slang words on sentiment learning could be due to the fact that the slang lexicon SlangSD [51] was not manually annotated. The SlangSD lexicon was automatically created and labelled based on the overall sentiment of tweets/text where the slang words appeared. Furthermore, the main purpose for using slang in OSNs is to shorten the words to fit the limited spaces like the case in Twitter. "tbh", "id", "b4", and "ttyl" are examples of slang words used by social media users. These slang words have no sentiment meaning in them; instead, they are abbreviations for "to be honest", "I don't know", "before", and "talk to you later", respectively. Unlike hashtagged words that seem to be used to emphasize the important aspects of texts including feelings and opinions, slang words are shown to be used for a more convenient and fast communication and when there is space constrains on messages. However, it is important to shed the light on the importance of considering slang words in sentiment learning as they represent a cultural language of social media and, hence, it contains some opinionated words such as "BFF means Best Friend Forever" and "lol means laughing".

E. CROSS-DOMAIN SENTIMENTS

In this section, we will present two experimental scenarios for cross-domain sentiment prediction: (1) examining our general sentiment classifier (i.e LGBM) on datasets of two domains: movie reviews and sports, (2) examining domain-specific sentiment models (i.e. movie reviews and sports) on our general sentiment dataset (DFSMD).

We trained two domain-specific LGBM models; one for IBDM movie reviews and another for sports (CL'16-'17) tweets. We used the same exact experimental settings and features (i.e. all proposed features combined) that we used to train our general LGBM model. We split the data into 60% for training and 40% for testing. The LGBM sentiment model trained on IMDB two-class dataset performs well at accuracy score of ≈ 86% (micro Ave. F-score of 0.86), whereas LGBM classifier trained on three-class sports CL'16-'17 dataset yields an acceptable learning performance at ≈ 57% of accuracy (micro Ave. F-score of 0.57)).

We conducted our experiments on three datasets as shown in Table.6. For IMDB and CL datasets, we used a subset of their instances for the purpose of our evaluation.

Table.7 presents the results of adapting our general domain sentiment model (LGBM)to domain-specific sentiment datasets. It can be seen that it is effective to adapt general sentiment modelling to domain-specific sentiment analysis. Our LGBM general model shows a good sentiment prediction performance on both movie reviews and sports tweets, for positive and negative classes. Our LGBM model was able to recall 51%, 67% (precision of 0.60, 0.71) of the positive, negative instances of IMDB dataset. We observed that the positive recall is lower than the negative one. This could be due to the fact that review texts have special dictionary and sentence patterns that do not necessarily exist in general conversations. However, our general LGBM model could correctly recognize > 50% of the positive class with 62% precision. The result is actually promising since our LGBM was trained on short texts to learn three classes, while IMDB reviews are of long texts (see Table.6) and only consist of two classes. This finding indicates that people generally use the common words and phrases to express opinions, regradless of the texts lenght. Again, this shows the quality of our dataset (DFSMD), in terms of data contents and annotation, for sentiment classification. In addition, our LGBM classifier performs even better on the sports CL'16-'17 dataset; it could successfully recall 71%, 64% (precision of 0.74, 0.71) of positive, negative instances. This is not surprising as the CL'16-'17 dataset consists of short text tweets. We can see that the recall of our general LGBM model is slightly lower in the sports domain than that of the movies reviews. This is due to the fact that sports domain reserves a special language where many terms and phrases indicate sentiments contrary to those of general original sentiments [3].

Generalizing domain-specific sentiment modelling is shown to be less effective than adapting general sentiment modelling to domain-specific analysis. From Table.8, we can observe that domain-specific LGBM models could not properly learn the negative instances of the general dataset even though they yield good learning performance on the positive instances. This implies that domain-specific data introduces learning confusion to general sentiment [3] as some terms might indicate negative sentiments in a specific domain but positive sentiments in general domain.

XIII. CONCLUSION AND FUTURE WORK

This work proposes to build a LGBM-based (Light Gradient Boosting Machine) classier to learn general sentiment using our domain-free dataset (DFSMD). The results show that the LGBM sentiment classifier is the winner in terms of accuracy and F-score, among the other six well-known classifiers for sentiment analysis: Logistic Regression, SVM, Random Forest, Multinomial Naïve Bayes, Extreme Gradient Boosting, and Deep Learning. LGBM has demonstrated its strength in handling class imbalance problem by yielding the best F-score for the minority class. Also, it has shown to converge faster than XGB and deep learning classifiers. The learning of LGBM and the other classifiers are shown to improve when more sentimental OSN-related features are combined with the base features of BOW and formal sentiment lexicons. Further, sentimental features such as linguistic hints, emoticons and emojis have not shown contribution for neutral class; rather, they have shown an excellent contribution for the subjective classes. Moreover, our findings have revealed the effectiveness of traditional machine learning in comparison to deep learning when features are well engineered and extracted. Having such well defined features provides explainability to both learning and analysis. In addition, the experiments have shown that our dataset contains distinguishing characteristics among positive, negative, and neutral classes. Further, stop words are shown to have a positive impact on the sentiment learning for short texts. Our findings suggest that hashtag words are significantly important for sentiment learning. On the other hand, slang words are shown to have little sentiment information and are used more for a convenient communication on OSNs. In addition, our approach has been proven effective in adapting general-domain sentiment to domain-specific sentiment, in comparison to generalizing domain-specific sentiment to general-domain sentiment problems.

For future directions, high quality slang sentiment lexicons should be given more attention in order to investigate their role on OSNs and to improve the sentiment learning. Also, we are interested in using our models for general purpose sentiment analysis as a prior knowledge to adapt and learn different domains such as learning sentiment intensity for sports fans, emotion and emotion intensity. We are also interested in conducting more extensive investigations to study the transferability of texts lengths for sentiment learning. Finally, we plan to combine gradient boosting machines with deep neural networks in a wider study and to investigate the capabilities of both algorithms strengths in sentiment classification.

REFERENCES

[1] V. John and O. Vechtomova, "Sentiment analysis on financial news headlines using training dataset augmentation," 2017, *arXiv:1707.09448*. [Online]. Available: http://arxiv.org/abs/1707.09448

[2] Z. Li, Y. Fan, B. Jiang, T. Lei, and W. Liu, "A survey on sentiment analysis and opinion mining for social multimedia," *Multimedia Tools Appl.*, vol. 78, no. 6, pp. 6939–6967, 2018.

[3] S. Aloufi, F. Alzamzami, M. Hoda, and A. El Saddik, "Soccer fans sentiment through the eye of big data: The UEFA champions league as a case study," in *Proc. IEEE Conf. Multimedia Inf. Process. Retr. (MIPR)*, Apr. 2018, pp. 244–250.

[4] R. Abaalkhail, F. Alzamzami, S. Aloufi, R. Alharthi, and A. El Saddik, "Affectional ontology and multimedia dataset for sentiment analysis," in *Proc. Int. Conf. Smart Multimedia*. Cham, Switzerland: Springer, 2018, pp. 15–28.

[5] V. Athanasiou and M. Maragoudakis, "A novel, gradient boosting framework for sentiment analysis in languages where NLP resources are not plentiful: A case study for modern Greek," *Algorithms*, vol. 10, no. 1, p. 34, Mar. 2017.

[6] G. M. Weiss and F. Provost, "Learning when training data are costly: The effect of class distribution on tree induction," *J. Artif. Intell. Res.*, vol. 19, pp. 315–354, Oct. 2003.

[7] A. Ali, S. M. Shamsuddin, and A. L. Ralescu, "Classification with class imbalance problem: A review," *Int. J. Adv. Soft Comput. Appl.*, vol. 7, no. 3, pp. 176–204, 2015.

[8] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, "Lightgbm: A highly efficient gradient boosting decision tree," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 3146–3154.

[9] S. Aloufi and A. El Saddik, "Sentiment identification in football-specific tweets," *IEEE Access*, vol. 6, pp. 78609–78621, 2018.

[10] Z. Li, S. Zhu, H. Hong, Y. Li, and A. El Saddik, "City digital pulse: A cloud based heterogeneous data analysis platform," *Multimedia Tools Appl.*, vol. 76, no. 8, pp. 10893–10916, 2017.

[11] M. Bouazizi and T. Ohtsuki, "A pattern-based approach for multi-class sentiment analysis in twitter," *IEEE Access*, vol. 5, pp. 20617–20639, 2017.

[12] K. Ravi and V. Ravi, "A survey on opinion mining and sentiment analysis: Tasks, approaches and applications," *Knowl.-Based Syst.*, vol. 89, pp. 14–46, Nov. 2015.

[13] Y. Wan and Q. Gao, "An ensemble sentiment classification system of Twitter data for airline services analysis," in *Proc. IEEE Int. Conf. Data Mining Workshop (ICDMW)*, Nov. 2015, pp. 1318–1325.

[14] A. Wei Yu, H. Lee, and Q. V. Le, "Learning to skim text," 2017, *arXiv:1704.06877*. [Online]. Available: http://arxiv.org/abs/1704.06877

[15] M. Jiang, M. Lan, and Y. Wu, "ECNU at SemEval-2017 task 5: An ensemble of regression algorithms with effective features for fine-grained sentiment analysis in financial domain," in *Proc. 11th Int. Workshop Semantic Eval. (SemEval-)*, Vancouver, BC, Canada: Association for Computational Linguistics, Aug. 2017, pp. 888–893.

[16] R. Alharthi, B. Guthier, C. Guertin, and A. El Saddik, "A dataset for psychological human needs detection from social networks," *IEEE Access*, vol. 5, pp. 9109–9117, 2017.

[17] T. Brants, A. C. Popat, P. Xu, F. J. Och, and J. Dean, "Large language models in machine translation," in *Proc. Joint Conf. Empirical Methods Natural Lang. Process. Comput. Natural Lang. Learn. (EMNLP-CoNLL)*, 2007, p. 858.

[18] C. Dyer, A. Cordova, A. Mont, and J. Lin, "Fast, easy, and cheap: Construction of statistical machine translation models with mapreduce," in *Proc. 3rd Workshop Stat. Mach. Transl. (StatMT)*, 2008, pp. 199–207.

[19] A. Sharma and S. Dey, "A comparative study of feature selection and machine learning techniques for sentiment analysis," in *Proc. ACM Res. Appl. Comput. Symp. (RACS)*, 2012, pp. 1–7.

[20] T. Niu, S. Zhu, L. Pang, and A. El Saddik, "Sentiment analysis on multi-view social data," in *Int. Conf. Multimedia Modeling*. Cham, Switzerland: Springer, 2016, pp. 15–27.

[21] N. Liu, B. Zhang, J. Yan, Z. Chen, W. Liu, F. Bai, and L. Chien, "Text representation: From vector to tensor," in *Proc. 5th IEEE Int. Conf. Data Mining (ICDM)*, Nov. 2005, p. 4.

[22] A. Tripathy, A. Agrawal, and S. K. Rath, "Classification of sentiment reviews using n-gram machine learning approach," *Expert Syst. Appl.*, vol. 57, pp. 117–126, Sep. 2016.

[23] E. S. Usop, R. R. Isnanto, and R. Kusumaningrum, "Part of speech features for sentiment classification based on latent Dirichlet allocation," in *Proc. 4th Int. Conf. Inf. Technol., Comput., Electr. Eng. (ICITACEE)*, Oct. 2017, pp. 31–34.

[24] B. Liu, "Sentiment analysis and subjectivity," in *Handbook of Natural Language Processing*, 2nd ed. Boca Raton, FL, USA: Taylor & Francis, 2010, pp. 627–666.

[25] M. Z. Asghar, F. M. Kundi, S. Ahmad, A. Khan, and F. Khan, "T-SAF: Twitter sentiment analysis framework using a hybrid classification scheme," *Expert Syst.*, vol. 35, no. 1, Feb. 2018, Art. no. e12233.

[26] H. Wang and J. A. Castanon, "Sentiment expression via emoticons on social media," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Oct. 2015, pp. 2404–2408.

[27] A. Muhammad, N. Wiratunga, R. Lothian, and R. Glassey, "Domain-based lexicon enhancement for sentiment analysis," in *Proc. SMA BCS-SGAI*, 2013, pp. 7–18.

[28] Y. Dang, Y. Zhang, and H. Chen, "A lexicon-enhanced method for sentiment classification: An experiment on online product reviews," *IEEE Intell. Syst.*, vol. 25, no. 4, pp. 46–53, Jul. 2010.

[29] O. Kolchyna, T. T. P. Souza, P. Treleaven, and T. Aste, "Twitter sentiment analysis: Lexicon method, machine learning method and their combination," 2015, *arXiv:1507.00955*. [Online]. Available: http://arxiv.org/abs/1507.00955

[30] A. Giachanou and F. Crestani, "Like it or not: A survey of twitter sentiment analysis methods," *ACM Comput. Surv.*, vol. 49, no. 2, pp. 1–41, 2016.

[31] M. Bilal, H. Israr, M. Shahid, and A. Khan, "Sentiment classification of Roman-Urdu opinions using Naïve Bayesian, decision tree and KNN classification techniques," *J. King Saud Univ.-Comput. Inf. Sci.*, vol. 28, no. 3, pp. 330–344, 2016.

[32] A. Yadav and D. K. Vishwakarma, "Sentiment analysis using deep learning architectures: A review," *Artif. Intell. Rev.*, pp. 1–51, Dec. 2019.

[33] F. K. Dosilovic, M. Brcic, and N. Hlupic, "Explainable artificial intelligence: A survey," in *Proc. 41st Int. Conv. Inf. Commun. Technol., Electron. Microelectron. (MIPRO)*, May 2018, pp. 0210–0215.

[34] M. Lango, D. Brzezinski, and J. Stefanowski, "PUT at SemEval-2016 task 4: The ABC of Twitter sentiment analysis," in *Proc. 10th Int. Workshop Semantic Eval. (SemEval)*, 2016, pp. 126–132.

[35] J. Lin and A. Kolcz, "Large-scale machine learning at Twitter," in *Proc. Int. Conf. Manage. Data SIGMOD*, 2012, pp. 793–804.

[36] T. G. Dietterich, "Ensemble methods in machine learning," in *Proc. Int. Workshop Multiple Classifier Syst.* Cham, Switzerland: Springer, 2000, pp. 1–15.

[37] R. E. Schapire, "The strength of weak learnability," *Mach. Learn.*, vol. 5, no. 2, pp. 197–227, Jun. 1990.

[38] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2016, pp. 785–794.

[39] M. Jabreel and A. Moreno, "EiTAKA at SemEval-2018 task 1: An ensemble of N-Channels ConvNet and XGboost regressors for emotion analysis of tweets," 2018, *arXiv:1802.09233*. [Online]. Available: http://arxiv.org/abs/1802.09233

[40] S. S. Mukku, S. R. Oota, and R. Mamidi, "Tag me a label with multi-arm: Active learning for Telugu sentiment analysis," in *Proc. Int. Conf. Big Data Anal. Knowl. Discovery*. Cham, Switzerland: Springer, 2017, pp. 355–367.

[41] M. Fan, A. Billings, X. Zhu, and P. Yu, "Twitter-based BIRGing: Big data analysis of English national team fans during the 2018 FIFA world cup," *Commun. Sport*, vol. 8, no. 3, pp. 317–345, 2019.

[42] Z.-Q. Wang, X. Sun, D.-X. Zhang, and X. Li, "An optimal SVM-based text classification algorithm," in *Proc. Int. Conf. Mach. Learn. Cybern.*, 2006, pp. 1378–1381.

[43] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up?: Sentiment classification using machine learning techniques," in *Proc. ACL Conf. Empirical Methods Natural Lang. Process.*, vol. 10. Stroudsburg, PA, USA: Association for Computational Linguistics, 2002, pp. 79–86.

[44] M. Robnik-Šikonja, "Improving random forests," in *Proc. Eur. Conf. Mach. Learn.* Cham, Switzerland: Springer, 2004, pp. 359–370.

[45] D. A. Alboaneen, H. Tianfield, and Y. Zhang, "Sentiment analysis via multi-layer perceptron trained by meta-heuristic optimisation," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2017, pp. 4630–4635.

[46] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[47] S. M. Mohammad and F. Bravo-Marquez, "Emotion intensities in tweets," 2017, *arXiv:1708.03696*. [Online]. Available: http://arxiv.org/abs/1708.03696

[48] M. Ghiassi and S. Lee, "A domain transferable lexicon set for Twitter sentiment analysis using a supervised machine learning approach," *Expert Syst. Appl.*, vol. 106, pp. 197–216, Sep. 2018.

[49] F. Å. Nielsen, "A new ANEW: Evaluation of a word list for sentiment analysis in microblogs," 2011, *arXiv:1103.2903*. [Online]. Available: http://arxiv.org/abs/1103.2903

[50] S. M. Mohammad, S. Kiritchenko, and X. Zhu, "NRC-canada: Building the state-of-the-art in sentiment analysis of tweets," 2013, *arXiv:1308.6242*. [Online]. Available: http://arxiv.org/abs/1308.6242

[51] L. Wu, F. Morstatter, and H. Liu, "SlangSD: Building and using a sentiment dictionary of slang words for short-text sentiment classification," 2016, *arXiv:1608.05129*. [Online]. Available: http://arxiv.org/abs/1608.05129

[52] F. Beijer, "The syntax and pragmatics of exclamations and other expressive/emotional utterances," Work. Papers Linguistics 2, 2002.

[53] J. H. Hill, "The impact of emojis and emoticons on online consumer reviews, perceived company response quality, brand relationship, and purchase intent," Graduate School Scholar Commons, Univ. South Florida, Tampa, FL, USA, Tech. Rep. 6513, 2016.

[54] P. K. Novak, J. Smailović, B. Sluban, and I. Mozetič, "Sentiment of emojis," *PLoS ONE*, vol. 10, no. 12, Dec. 2015, Art. no. e0144296.

[55] A. Pak and P. Paroubek, "Twitter as a corpus for sentiment analysis and opinion mining," in *Proc. LREc*, vol. 10, 2010, pp. 1320–1326.

[56] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts, "Learning word vectors for sentiment analysis," in *Proc. 49th Annu. Meeting Assoc. Comput. Linguistics, Hum. Lang. Technol.*, vol. 1. Stroudsburg, PA, USA: Association for Computational Linguistics, 2011, pp. 142–150.

FATIMAH ALZAMZAMI received the M.Sc. degree in computer science from the Faculty of Electrical Engineering and Computer Science, University of Ottawa, Ottawa, ON, Canada, where she is currently pursuing the Ph.D. degree in computer science, under the supervision of Prof. A. El Saddik. Her research interests include machine learning, deep learning, big data, social multimedia analysis, and mining.

MOHAMAD HODA received the B.S. and M.Sc. degrees in computer science from Arts, Science, and Technology University, Lebanon, in 2003 and 2005, respectively, and the Ph.D. degree in computer science from the University of Ottawa, in 2016. He is currently a Research Fellow with the Department of Surgery, Faculty of Medicine, Ottawa Hospital Research Institute. He is supervised by Prof. A. El Saddik. His current research interests include deep learning, multimedia information retrieval, image and video understanding, and social media analysis and mining.

ABDULMOTALEB EL SADDIK (Fellow, IEEE) is currently a Distinguished University Professor and the University Research Chair with the School of Electrical Engineering and Computer Science, University of Ottawa. His research interests include the establishment of digital twins to facilitate the wellbeing of citizens using AI, the Internet of Things (IoT), AR/VR, and 5G to allow people to interact in real-time with one another as well as with their smart digital representations. He has coauthored 10 books and more than 550 publications and chaired more than 50 conferences and workshops. He has received research grants and contracts totaling more than $20 M. He has supervised more than 120 researchers and has received several international awards, for example, a ACM Distinguished Scientist, a Fellow of the Engineering Institute of Canada and the Canadian Academy of Engineers, and the IEEE I&M Technical Achievement Award, the IEEE Canada C. C. Gotlieb (Computer) Medal, and the A. G. L. McNaughton Gold Medal for important contributions to the field of computer engineering and science.