Received January 30, 2019, accepted February 11, 2019, date of publication February 13, 2019, date of current version March 5, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2899260

Sarcasm Detection Using Soft Attention-Based Bidirectional Long Short-Term Memory Model With Convolution Network

LE HOANG SON[1,2], AKSHI KUMAR[3], SAURABH RAJ SANGWAN[3], ANSHIKA ARORA[3], ANAND NAYYAR[4], AND MOHAMED ABDEL-BASSET[5]

[1]Division of Data Science, Ton Duc Thang University, Ho Chi Minh 700000, Vietnam
[2]Faculty of Information Technology, Ton Duc Thang University, Ho Chi Minh 700000, Vietnam
[3]Department of Computer Science & Engineering, Delhi Technological University, Delhi 110042, India
[4]Graduate School, Duy Tan University, Da Nang 550000, Vietnam
[5]Department of Operations Research, Faculty of Computers and Informatics, Zagazig University, Zagazig 44519, Egypt

Corresponding author: Mohamed Abdel-Basset (analyst_mohamed@zu.edu.eg)

ABSTRACT A large community of research has been developed in recent years to analyze social media and social networks, with the aim of understanding, discovering insights, and exploiting the available information. The focus has shifted from conventional polarity classification to contemporary application-oriented fine-grained aspects such as, emotions, sarcasm, stance, rumor, and hate speech detection in the user-generated content. Detecting a sarcastic tone in natural language hinders the performance of sentiment analysis tasks. The majority of the studies on automatic sarcasm detection emphasize on the use of lexical, syntactic, or pragmatic features that are often unequivocally expressed through figurative literary devices such as words, emoticons, and exclamation marks. In this paper, we propose a deep learning model called sAtt-BLSTM convNet that is based on the hybrid of soft attention-based bidirectional long short-term memory (sAtt-BLSTM) and convolution neural network (convNet) applying global vectors for word representation (GLoVe) for building semantic word embeddings. In addition to the feature maps generated by the sAtt-BLSTM, punctuation-based auxiliary features are also merged into the convNet. The robustness of the proposed model is investigated using balanced (tweets from benchmark SemEval 2015 Task 11) and unbalanced (approximately 40000 random tweets using the Sarcasm Detector tool with 15000 sarcastic and 25000 non-sarcastic messages) datasets. An experimental study using the training- and test-set accuracy metrics is performed to compare the proposed deep neural model with convNet, LSTM, and bidirectional LSTM with/without attention and it is observed that the novel sAtt-BLSTM convNet model outperforms others with a superior sarcasm-classification accuracy of 97.87% for the Twitter dataset and 93.71% for the random-tweet dataset.

INDEX TERMS Sarcasm, deep learning, attention, social data.

I. INTRODUCTION

The biggest challenge in sentiment analysis tasks is to accurately determine the veracity of the statement in literal sense so as to classify text on the basis of polarity (positive or negative). Sentiment analysis achieves decent results in the case of literal language as it conveys the expected interpretation. However, the use of figurative language that is inherently emblematic represents something other than the concrete meaning, thereby making sentiment analysis a non-trivial problem. ''Sarcasm is defined as a specific type of sentiment where people express their negative feelings using positive or intensified positive words in the text'' [1]. It is a rhetoric narrative that presents a communicatively significant discord between the actual situation and the utterance content. For example, a tweet/post, ''*It is wonderful feeling to waste hours in traffic jams!*'' clearly indicates this discord between the actual situation of ''being stuck in traffic jam'' and the utterance content ''wonderful.'' This contrast and shift of sentiments in sarcastic expressions validates sarcasm as a special instance of sentiment analysis. Therefore, the detection of sarcastic expressions will enhance

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

the automatic sentiment analysis of huge and diverse social web data.

Automatic sarcasm detection is typically a text-classification problem, which relies on a variety of feature-extraction and learning techniques. In general, the Twitter post size characterizes short text with a character limit of 280, which is insufficient in providing the desired word co-occurrence data. This data sparseness is challenging for conventional learning models based on statistical features, such as term frequency–inverse document frequency (TF-IDF) or co-occurrence. Deep learning architectures have proven capabilities when dealing with natural language, as they can learn by example, offer a flexible set of modeling alternatives while curtailing feature engineering, scale up to big data, exploit unlabeled data to detect trends and identify patterns too complex for humans to notice, and be used for multitask learning [2]. The **primary contribution** of this work includes:

- Improving the performance of the sentiment analysis task by building, training and evaluating a deep learning model, *s*Att-BLSTM convNet, a hybrid of *soft* attention-based bidirectional long short-term memory (*s*Att-BLSTM) and convolution neural network (convNet)
- Feature engineering for automatic sarcasm detection using figurative literary devices such as words, emoticons, and exclamation marks.
- Performance benchmarking by testing different kinds of models and datasets for sarcastic tone classification in real-time.

The proposed deep learning model, *s*Att-BLSTM convNet, has eight layers:

- *Input Layer:* The input tweet to the model.
- *Embedding Layer:* Each word within tweet is mapped into a low dimension vector using GloVe. This look-up encodes the input into real-valued vectors called embedding vectors.
- *BLSTM layer:* The output of the word-embedding layer is fed into the BLSTM layer. The purpose of this layer is to learn high-level features from the previous step.
- *Attention Layer:* A *soft* attention mechanism (based on *softmax*) is used, which generates semantic context vectors. These are merged with the BLSTM output features and auxiliary features to train the convNet.
- *Convolution Layer:* The convolution operation is performed to obtain a convolved feature vector.
- *Activation Layer:* A **Re**ctified **L**inear **U**nit (ReLU) activation function is applied on the feature vectors (output of convNet layer).
- *Down-sampling Layer:* The max-pooling operation is used as a down-sampling strategy in convNets.
- *Representation Layer:* A fully connected layer that consists of a linear transformation layer and *softmax* activation function to generate the output predictions.

To build word embeddings, GloVe [3], which generates a word vector table, is used. GloVe is a count-based model of representing words by feature vectors. This model maps all the tokenized words in each tweet to its respective word vector table. Proper padding is performed to unify the feature vector matrix. This matrix is given as an input to the bidirectional long-short-term-memory (BLSTM) layer. The BLSTM networks set up an additional layer to the unidirectional LSTM networks, where the hidden-to-hidden connections flow in opposite temporal order. The network thus exploits information from the past and the future. The output vector representation from the BLSTM is transformed into an attention-weighted sum of these output vectors. This word-level attention hypothesis relies on valuable words within text that contribute to the meaning in context, and thus computes attention probabilities over the input stimuli from the BLSTM. Consequently, the output feature representations from BLSTM are concatenated together as the output of the attention layer, which, combined with auxiliary features, is provided as input to the convNet. The convNet consists of multiple filters with a variable-size window connected to a small part of the input to generate a convolved feature map.

Several filters are used to extract features in a convolution layer and each filter gives an activation map. The next layer is the activation layer, or the ReLU layer, which introduces nonlinearity to the network and generates a rectified feature map. This rectified feature map is fed to the pooling layer to reduce the dimensionality of the feature map. A *k-max* pooling operation is employed, which selects the top k features with respect to the various hidden layers and generates a pooled feature map. This pooled feature map is input to the fully connected *softmax* layer, which calculates the probability of any output word and classifies the tweet as sarcastic or non-sarcastic.

- Basic sentiment analysis tools have difficulty distinguishing between positive and negative statements, especially when sarcasm, irony, and mixed feelings are included discretely or in combination. Practical models to predict sarcastic text are thus imperative, and the proposed model combines semantics from the attention-based bidirectional LSTM network with auxiliary pragmatic features to a deep convolution network for enhanced prediction performance.
- The multiple layers involve nonlinearity that aids learning complex representations.

The rest of the paper is structured as follows. Section 2 briefly discusses the pertinent work within the domain of sentiment analysis and deep learning based automatic sarcasm detection followed by Section 3, which illustrates the proposed *s*Att-BLSTM convNet model and its working details. Section 4 explicates the results and provides an analysis of the same. The final section, Section 5, concludes the study and expounds upon the scope of future work.

II. RELATED WORK

The growing pursuit of Internet users in all forms of social media has intensified researchers' interest to intelligently mine the content available, both quantitatively and qualitatively. Twitter, currently the most famous micro-blog, connects people across the globe and has a high level of user involvement. It has gradually emerged as a huge source of sentiment-rich data [4]. The keyword "sentiment analysis" was initially witnessed in the published work [5] in 2003, and since then, both primary [4], [6], [7] and secondary studies have been reported across pertinent literature [8]–[11]. Furthermore, the literature is well-equipped with studies pertaining to sentiment analysis using machine learning paradigms on specifically textual user-generated online content on social media. Aloufi and El Saddik [12] proposed a model for sentiment analysis of football specific tweets using three classifiers, namely, support vector machine, multinomial Navies Bayes and random forest. Pai and Liu [13] put forward a model for prediction of vehicle sales by sentiment analysis of twitter data and stock market values using least squares support vector regression. Research using deep learning models for sentiment analysis has also been reported. Tseng *et al.* [14] analyzed textual opinions found in teaching evaluation questionnaires and applied the analysis results to assisting the selection of outstanding teaching faculty members using attention-based LSTM. Wu and Chi [15] proposed a model having quadratic connections of LSTM capable of capturing complex semantic representations of natural language texts and evaluated on the benchmark dataset, the Stanford Sentiment Treebank. Bouazizi and Ohtsuki [6] extended the concept of binary or ternary classification and proposed an approach to classify text collected from Twitter into seven sentiment classes. The authors further proposed [16] multi-class sentiment analysis which addresses the identification of the exact sentiment conveyed by the user using the task of quantification that identifies all the existing sentiments within a tweet instead of attributing a single sentiment label to it. Zhang *et al.* [17] proposed a model combining convolution neural network and bidirectional gated recurrent unit for sentence semantic classification. Han *et al.* [18] used CNN and BLSTM to design four kinds of memory network models for text sentiment classification. Fu *et al.* [19] proposed a lexicon-enhanced LSTM model that uses sentiment lexicon as extra information pre-training a word sentiment classifier. Wang *et al.* [20] proposed a sophisticated algorithm based on deep learning and information geometry for sentiment classification.

Significant studies exemplify sarcasm as a contradiction in polarity [21], [22]. Sarcasm detection has been reported as a non-trivial challenge for sentiment analysis across literature studies. There have been expansive efforts by researchers to investigate sarcasm-detection techniques on various social media channels as a sub-task of sentiment analysis [23]. Bouazizi and Ohtsuki [24] propose a pattern-based approach to detect sarcasm on Twitter considering four sets of features that cover the different types of sarcasms.

Architectures based on deep learning techniques have gained popularity in natural language programming applications as they are capable of handling data sparseness in unbalanced datasets. Studies on bullying and verbal aggression have been reported deep neural network for short-text sentiment analysis [25]. A few such approaches have been reported for automatic sarcasm detection as well. Felbo *et al.* [26] proposed a DeepMoji model, which is based on the occurrences of emoji, for detecting the emotional content on Twitter. It uses a variant of LSTM, a 6-layer model that is a hybrid of BLSTM and the attention mechanism for the detection of sarcastic tweets. A new convolution network-based architecture is presented in [27] that learn user embeddings in addition to utterance-based embeddings, which allows the authors to learn user-specific context. The similarity between word embeddings is used by Joshi *et al.* [28] as features for sarcasm detection. The authors have concatenated features from previous works to these word embedding-based features. Ghosh and Veale [29] have built a model combining a convNet with a recurrent neural network (RNN, long short-term memory) followed by a deep neural network. It is observed that this approach shows an improvement for the deep learning architecture when compared with recursive SVM. In [30], models based on a pre-trained convNet have been developed that are used to extract emotion, sentiment, and even personality features for sarcasm detection. In another study, [31], bidirectional LSTM is used to capture syntactic and semantic information over tweets, along with a pooling neural network to extract contextual features automatically from history tweets. Performance using neural features is compared with discrete manual features and an improvement is observed. Attention mechanism has been previously used by researchers in the domain of natural language understanding as it selectively focuses on information-rich phrases within the sequence of sentence. Recently, authors Seo *et al.* [32] and Wang *et al.* [33] demonstrated the effective use of attention for compression and question-answering.

III. AUTOMATED SARCASM DETECTION

The proposed framework for automatic sarcasm detection has four key components, namely, (*A*) data acquisition, (*B*) preprocessing, (*C*) implementation of the *s*Att-BLSTM convNet model, and (*D*) evaluation using performance measures. These are explained in detail in the following subsections.

A. DATASET ACQUISITION

To evaluate the performance of the proposed model, the following datasets have been used for empirical analysis. The purpose of using two varied datasets was to further validate the results achieved and determine the robustness of the selected features and the model trained on these features.

1) BALANCED DATASET: SEMEVAL 2015 TASK 11

The original SemEval dataset consists of 20000 hashtag-annotated tweets, which were created for the Twitter sentiment analysis task. SemEval 2015 task 11 [34] defined the challenge as a "dataset for sentiment analysis of figurative language on Twitter." The objective of the task is specifically to evaluate the use of figurative language, where the word or phrase described as "figurative" meant that it is being used in a way that was symbolic and represented something other than its concrete meaning. Although figurative language encompasses an extensive variety of literary tools and techniques, such as similes, emoticons, metaphor, personification, idiom, and hyperbole, this benchmark dataset contains tweets with specific figurative types, namely, sarcasm, irony, and metaphor. The sarcasm corpus is thus constructed by collecting self-annotated tweets with the hashtags #irony, #sarcasm, and #not. Although there is a subtle difference between the three, we consider all of them in the same category for performing this empirical analysis. Thus, this study is also indicative of how this generalization influences the obtained results. The experiments used the SemEval 2013 task 2 dataset as the negative class data [35]. Manual perusal of the two datasets revealed that both contain tweets from general topics, thus avoiding any bias in our experimentation. To collect data for the experiments, we tried to retrieve 15961 tweets using a list of the dataset's tweet ids provided, out of which 7994 are sarcastic and 7324 are non-sarcastic.

2) IMBLANCED DATASET: RANDOM TWEETS

In real time, the presence of sarcasm in tweets is less frequent. A randomly sampled dataset containing 15,000 sarcastic and 25,000 non-sarcastic tweets is built using the Sarcasm Detector tool.[1]

B. PRE-PROCESSING

After the data acquisition, to intelligently mine the text, the tweets are pre-processed to clean and transform the data for feature extraction [36]. These include the following processes.

- URLs, mentions, hashtags, and numbers in the tweets are replaced with placeholders
- All tweets are tokenized by the Natural Language Toolkit (NLTK) [37]
- All tokens are sampled in lowercase
- All tweets are stemmed to the root word using Porter's Stemmer [38]
- All non-ASCII-English characters were removed, to limit the domain of data to the English language
- All words are replaced by pre-trained GloVe word vectors. Words that do not exist in the pre-trained embeddings are set to a zero vector

Several auxiliary features are used to enhance the classification model. These are primarily pragmatic markers concerned with language use. We consider the following five punctuation-based features that help understand the subtleties of sarcasm in text, as these indicate figurative text and symbolic clues within the text:

- Number of exclamation marks (!) in the tweet
- Number of question marks (?) in the tweet
- Number of periods (.) in the tweet
- Number of capital letters in the tweet
- Number of uses of "or" in the tweet

[1] www.thesarcasmdetector.com

C. THE PROPOSED sATT-BLSTM CONVNET MODEL

The proposed deep learning model uses 8 layers: the input layer, embedding layer, BLSTM layer, attention layer, convolution layer, activation and ReLU layer, max pooling layer, and representation layer. Fig. 1 depicts the architecture of the proposed multi-layer model. The details of each layer are provided in the subsequent subsections.

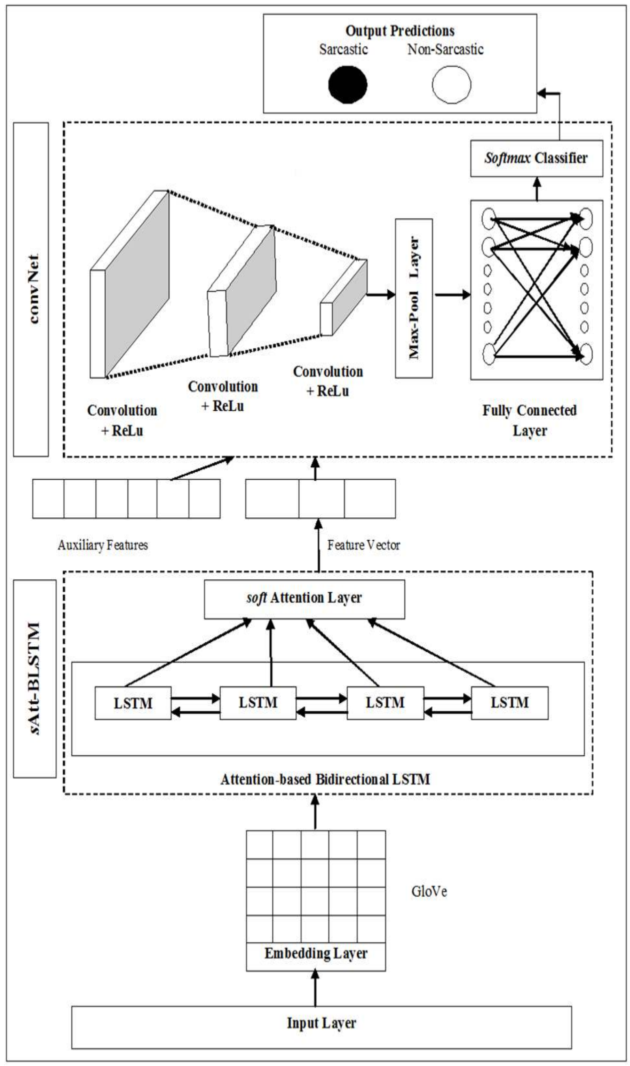

FIGURE 1. System architecture of the proposed sAtt-BLSTM ConvNet.

1) INPUT LAYER

The tweets after pre-processing are fed to the input layer. The input layer is connected to the embedding layer, which builds word embeddings using GloVe.

2) EMBEDDING LAYER

The embedding layer maps the input into real-valued vectors using encoding from look-up tables. Word embeddings facilitate learned word representations. The benefits of extracting features based on word embedding to detect sarcasm have been recently reported [28]. In this study, to build word embeddings, GloVe, which generates a word vector table, is used. GloVe is a count-based model of representing words by feature vectors. This log-bilinear model studies the relationship of words by counting the number of times they co-occur. Thus, this model aids in mapping all the tokenized words in each tweet to their respective word vector tables. Proper padding is performed to unify the feature vector matrix. That is, if the total number of given tweets is Z and there is a tweet X with t tokens, generation of a word vector Table 5 with dimension d of the word vectors is completed using GloVe. Thus, for all Z, each t in X is mapped to its respective V. After this mapping, each X is expressed as a vector of the word embeddings concatenation (E). Thus, the feature vector matrix is obtained as shown in (1).

$$\mathrm{F} = \mathrm{U} + \mathrm{B} + \mathrm{T} + \mathrm{P} + \mathrm{E}, \tag{1}$$

where + is the concatenation operator of the vector. The tweets are of varying length, so to unify the feature vector matrix representation of tweets, the tweets with the maximum length in the given corpus are used as a threshold value. This is done basically to fix the length of the tweet matrix. Hence, for all the tweets that were shorter than this threshold, zero padding was performed. This matrix was finally fed as input (i.e., F) to the BLSTM layer.

3) BLSTM LAYER

LSTM is an RNN that contains special units in the recurrent hidden layer called memory blocks [39]. Self-connections exist within the memory cells. There is an input, output, and forget gate for each memory cell. The hidden layer of LSTM is also called the LSTM cell. LSTM has the capability to plot long-term dependencies by defining each memory cell with a set of gates $\Re^d$, where d is the memory dimension of the hidden state of LSTM [28]. At each iteration t, the LSTM cell has the layer input x_t and the layer output h_t. LSTM contains three gates, which are functions of the current input $\mathrm{x_t}$ and hidden state $\mathrm{h_{t-1}}$: input gate $\mathrm{i_t}$, forget gate $\mathrm{f_t}$, and output gate $\mathrm{o_t}$. The cell also takes the cell input state s_t, the cell output state c_t, and the previous cell output state c_{t-1} into account while training and updating parameters.

The input gate, forget gate, output gate, and input cell state can be calculated using equations (2)–(7).

$$i_t = \sigma(w_{i_x}x_t + w_{i_h}h_{t-1} + b_i) \tag{2}$$
$$f_t = \sigma(w_{f_x}x_t + w_{f_h}h_{t-1} + b_f) \tag{3}$$
$$o_t = \sigma(w_{o_x}x_t + w_{o_h}h_{t-1} + b_o) \tag{4}$$
$$s_t = tanh\left(w_{s_x}x_t + w_{s_h}h_{t-1} + b_s\right) \tag{5}$$
$$c_t = f_t \odot c_{t-1} + i_t \odot s_t \tag{6}$$
$$h_t = tanh\,(c_t) \odot o_t \tag{7}$$

In equations (2)–(7),

- $\odot$ is the element-wise product;
- $w_i\, w_f\, w_o\, w_s$ are weighing factors used for mapping the hidden layer input to the three gates and the input cell state;
- $b_i\, b_f\, b_o\, b_s$ are bias vectors;
- σ is the gate activation function, which is normally the sigmoid function;
- tanh is the hyperbolic tangent function.

The final output of the LSTM layer is a vector of all the outputs, represented by (8).

$$Y_t = [h_{t-n}, \ldots\ldots, h_{t-1}] \tag{8}$$

The bidirectional LSTM connects two hidden layers to a single output layer. The forward-layer output sequence $h^{\rightarrow}$ is iteratively calculated using inputs in a positive sequence from time t-n to time t-1, while the backward-layer output sequence, $h^{\leftrightarrow}$ is calculated using the reversed inputs from time t-n to t-1 [40]. Both the forward- and backward-layer outputs are calculated by using the standard LSTM updating equations. This creates two copies of the hidden layer, one fit in the input sequences as is and one as a reversed copy of the input sequence [40]. These representations are concatenated together using the attention mechanism. That is, the BLSTM layer generates an output vector in which each element is calculated by using (9).

$$y_t = \sigma\left(h^{\rightarrow}t, h^{\leftrightarrow}t\right) \tag{9}$$

where σ is a soft attention function to combine the two output sequences. By this means, our model can capture as many salient words from both directions as possible. Thus, similar to that of the LSTM layer, the final output of the BLSTM layer can be represented by a vector (10).

$$\mathrm{Y_t} = [\mathrm{y_{t-n}}, \ldots\ldots, \mathrm{y_{t-1}}] \tag{10}$$

4) ATTENTION LAYER

The idea of attention [41] has originated by the observation that each word contributes to the meaning of tweet (sentence) differently that is; all words do not contribute equally to the representation of the sentence meaning. The users reading text can intuitively identify important, meaningful parts of the sentence. The attention mechanism adds an additional structure to the network, which is learned as part of the standard model training via back-propagation. A deterministic, differentiable attention mechanism, known as soft attention [42], is used, where the weights on each location are usually given by a *softmax*, and the output of the attention module is a weighted sum of representations at each location. This whole process can be described by a differentiable function, so that training can be performed jointly with the rest of the network using back-propagation.

We use a word-level attention mechanism to focus on the words that have a closer semantic relationship to the sentence meaning. That is, the word-level attention is applied to the feature representations of BLSTM. The output is thus a weighted combination of all the input states, not just the last state. The attention mechanism gives an attention score $e_{i,t}$ to each word t in the sentence i, as given by (11).

$$e_{i,t} = g(Wh_t c), \tag{11}$$

where g is an activation function.

Then the weight probability $a_{i,t}$ of each h_t is computed by (12).

$$a_{i,t} = \frac{\exp(e_{i,t})}{\sum_{k=1}^{T} \exp(e_{i,k})} \tag{12}$$

- Auxilary Features

Pragmatic features, such as punctuation or repetition of words, aid in understanding the subtleties of sarcasm in text, as these indicate figurative text and symbolic clues within the text. Punctuation is considered ideal for the representation of syntactic features, as the punctuation features contribute to a substantial boost in classification quality [42]. These can be exemplified by the use of exclamation points/marks, repeated letters (especially vowels), or uppercase letters (ALL CAPS) in textual messages such as tweets/posts. A few examples where these punctuation features impact the emotion quotient in text are given below:

➢ An exclamation point or exclamation mark (!) intensifies the emotional content preserving the semantic orientation [43]. For example: *"The location is great!!!!"* is more intense than *"The location is great."*

➢ Repeated letters (especially vowels) added to elongate the spelling of a word (e.g., "haaaaaaapppyyy") is used to emphasize the word or to express enthusiasm in computer-mediated communication [44].

➢ Capitalization, such as explicitly using ALL CAPS to emphasize a sentiment-relevant word with other non-capitalized words in the document, increases the intensity of the sentiment preserving semantic orientation [45]. For example, "Simply LOVED the place!" is more intense than "Simply loved the place!"

Thus, sarcasm detection can be made easy by observing the syntactic pattern of tweet to identify punctuation such as:

➢ Number of contiguous question marks, exclamation points, and combinations thereof

➢ Whether the last token contains a question mark or exclamation point

➢ Occurrence of punctuation sequences such as "!!!" and "?!"

In this study, the set of punctuation-based features shown in Table1 are extracted for each tweet.

TABLE 1. Punctuation-based features used.

excl	No. of exclamation marks in the post/tweet
ques	No. of question marks in the post/ tweet
dots	No. of periods in the post/tweet
caps	No. of capital letters in the post/tweet
quotes	No. of quotation marks (' or ") in the post/tweet

5) CONVOLUTION LAYER

ConvNet has two primary parts, namely, feature learning (Conv, ReLU, and Pool) and classification (fully connected and softmax). The output features of the sAtt BLSTM are combined with auxiliary features and provided as input to the convNet. Each convolving filter has a variable window size (s), height, and width. This window slides over to the input matrix and calculates localized feature vector (v_j) for each possible word (window size) using a nonlinear activation function (f) and a bias (b_a).

A transition matrix (T) is generated for each filter. This filter comprises of a layer of weights (weight matrix, W) that are applied to parts of the input matrix to obtain a single unit as the output. This is performed for the bias and weight during the network-learning adjustment, as illustrated in (13).

$$v_j = f\left(T.F_{j:j+s-1} + b_a\right) \tag{13}$$

Here, $F_{j:j+s-1}$shows the local vector from position j to $j + s - 1$ in F. Hence, formation of this v_j together with the completion of the convolution operation generates a new vector, n, as in (14).

$$n = [v_1, v_2, v_3, \ldots, v_{d-s+1}] \tag{14}$$

After the sliding operation, the output from each part is then combined in order to obtain the resultant output, called the transformed feature map, that serves as the input to the next layer: the ReLU Layer.

6) RELU LAYER

The activation or ReLU Layer [46] is applied for dealing with the nonlinearity in the convNet model. It generates a rectified feature map, which is fed to the pooling layer to reduce the dimensionality of the feature map.

7) POOLING LAYER

In this work, max pooling [47] is used, which takes the largest element from the rectified feature map. It reduces the dimensionality of the feature map and helps retain the most significant features. The k-max pooling operation (p) is performed for feature selection that chooses the top k features conforming to various hidden layers, where p is expressed as (15).

$$p = \max\{v_1, v_2, v_3, \ldots, v_{d-s+1}\} \tag{15}$$

8) REPRESNTATION LAYER

The output layer is a fully connected layer that consists of the softmax activation function. The pooled feature map is input to the *fully connected* softmax layer, which calculates the probability of any output word and classifies the tweet as sarcastic or non-sarcastic as an output. The output vector of the *softmax* layer ($O^{(z)}$) is as (16).

$$O^{(z)} = T^{(z)} O^{(z-1)} + b_a^{(z)}, \tag{16}$$

where

$O^{(z-1)}$ denotes the output vector the pooling layer
$T^{(z)}$ is the transition matrix
$b_a^{(z)}$ represents the bias factor of softmax layer.

Thus, using a fully connected *softmax* layer yields the probability-distribution [48] of the sarcasm detection (P_d), which is denoted using (17).

$$P_d(j|X,\theta) = \frac{[\exp(O_{j[Z]})]}{[\sum_{k=1}^{n} \exp(O_{j[Z]})]}, \tag{17}$$

where θ is the parameter of the model and j is the class label.

D. RESULTS AND ANALYSIS

For discussing the results, the empirical analysis has been broadly divided into two parts: (i) parameter setting for the proposed model and (ii) comparison with multiple baselines on the basis of classification accuracy.

1) PARAMETER SETTING

Optimal selection of parameters is imperative to achieve superlative performance results. We use the validation data to tune the hyper-parameters so as to obtain the best results. Table 2 lists the values used in this work.

TABLE 2. Hyperparameter values.

Hyperparameter	Value
Dimension of GloVe vectors	200
Hidden units of LSTMs (Forward, Backward)	500 each
Mini-batch size	10
Number of convNet layers	3
Number of convolution filters	100
Window size	(3,3)
Activation function of convNet	ReLU
Regularization	Dropout Operation
Drop-out rate	0.5: word embedding; 0.2: BLSTM; 0.4: convNet
Learning rate	0.2

The weights of the network are randomly initialized from a Gaussian distribution with variance scaled to 0.01, and the biases are initialized to zero except the forget-gate bias, which is initialized to 1.

2) PERFORMANCE RESULT

The proposed model is evaluated to predict sarcasm in tweets using two datasets, namely SemEval 2015 task 11 and random tweets containing a total of 40,000 tweets. The results have been assessed using key performance indicators (accuracy, recall, precision, and F-measure [49, 50]). Table 3 lists the results of the proposed *s*Att-BLSTM convNet model implemented on the two datasets and Fig. 2 depicts the results graphically.

3) COMPARISON WITH BASELINE DEEP LEARNING MODELS RESULT

We compare the results of the proposed model with three other deep learning architectures, namely, LSTM, BLSTM without attention, and convNet. The word embedding was performed using GloVe for each baseline model and the evaluation was made for both datasets using the four key performance indicators. The results obtained for LSTM, BLSTM without attention, BLSTM with attention and convNet are shown in Tables 4, 5, 6 and 7, respectively.

TABLE 3. Performance of the proposed *S*Att-BLSTM convNet model.

*s*Att-BLSTM convNet	SemEval Dataset	Random Tweets
Accuracy	97.87	93.71
Recall	96.83	92.67
Precision	92.14	90.49
F-measure	93.57	88.29

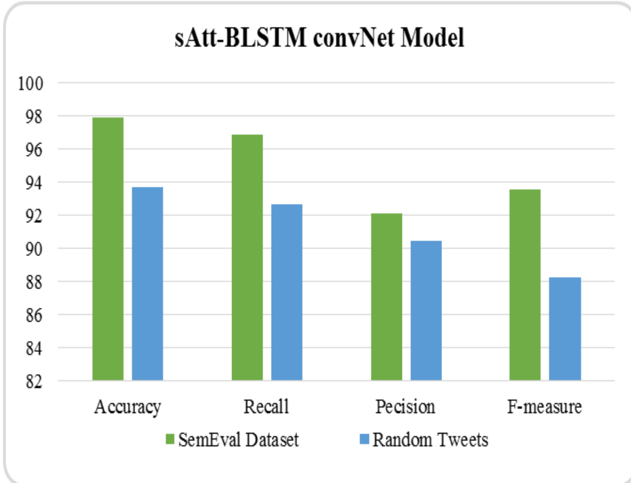

FIGURE 2. sAtt-BLSTM convNet Model Performance on two Datasets.

TABLE 4. Performance OF LSTM.

LSTM model	SemEval Dataset	Random Tweets
Accuracy	84.89	79.75
Recall	83.51	78.01
Precision	86.78	74.98
F-measure	80.39	76.91

TABLE 5. Performance of BLSTM without attention.

BLSTM without Attention	SemEval Dataset	Random Tweets
Accuracy	86.32	81.03
Recall	84.39	80.27
Precision	81.61	78.76
F-measure	85.25	79.45

TABLE 6. Performance of BLSTM with attention.

BLSTM without Attention	SemEval Dataset	Random Tweets
Accuracy	89.03	85.65
Recall	85.51	82.34
Precision	87.76	85.55
F-measure	87.25	84.04

TABLE 7. Performance of convNet.

ConvNet model	SemEval Dataset	Random Tweets
Accuracy	91.60	88.28
Recall	90.53	87.61
Precision	90.19	86.01
F-measure	88.57	84.72

It can be clearly observed that the proposed *s*Att- BLSTM convNet outperforms the other models with an accuracy of 91.60% achieved for the SemEval dataset and 88.28% for random tweets. LSTM shows the least accuracy of 84.89% and 79.75% for the SemEval and random tweet datasets, respectively. The models, in order from lowest to highest accuracy, are LSTM < BLSTM without Attention < BLSTM with Attention < convNet < *s*Att-BLSTM convNet. The best recall is also observed for the proposed *s*Att-BLSTM convNet for both the datasets. However, the convNet model demonstrates the best precision value of 92.19% for the SemEval dataset, whereas the proposed sAtt-BLSTM convNet model shows the best precision for the random tweets. Table 8 summarizes the comparison of the accuracy results obtained by the above five models. Fig. 3 graphically illustrates these comparative results.

TABLE 8. Accuracy comparison of models.

Dataset → / Models ↓	SemEval Dataset	Random Tweets
LSTM	84.89	79.75
BLSTM without Attention	86.32	81.03
BLSTM with Attention	89.03	85.65
convNet	91.60	88.28
*s*Att-BLSTM convNet	97.87	93.71

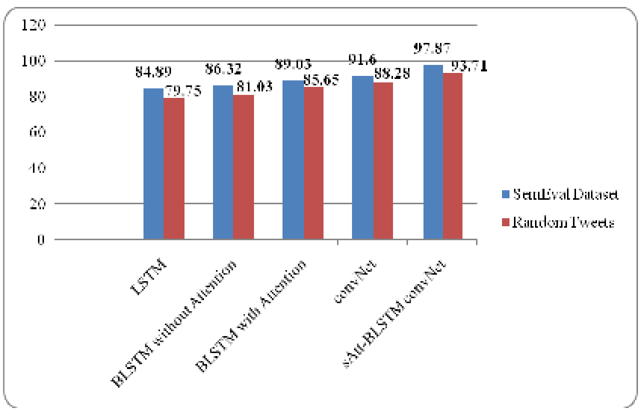

FIGURE 3. Accuracy Comparison of the Proposed Model with Baseline Models.

IV. CONCLUSION

Social media is an informal means of communication that includes considerable use of slang, malformed words, short forms, colloquial expressions, mash-up words, etc. All these amplify the ambiguity, vagueness, and imprecision in the available social web content and makes the task of analyzing generic and specific sentiment computationally difficult. Sarcasm is a specific type of sentiment and its automatic detection is a dynamic area of research. In this work, we proposed *s*Att-BLSTM convNet, a hybrid of *soft* attention-based bidirectional long short-term memory (*s*Att-BLSTM) and convolution neural network (convNet), to detect sarcasm in short text (tweets). Semantic word embeddings and pragmatic auxiliary features were used to train the network. The proposed model demonstrates the highest classification accuracy for both the datasets as compared to the baseline models. The use of mash-up languages and novelty in vocabulary with advanced architectures [51]–[81] add to the challenges of automated detection of sarcasm and characterize some open problems for future research within the domain.

REFERENCES

[1] S. K. Bharti, B. Vachha, R. K. Pradhan, K. S. Babu, and S. K. Jena, "Sarcastic sentiment detection in tweets streamed in real time: A big data approach," *Digit. Commun. Netw.*, vol. 2, no. 3, pp. 108–121, 2016.
[2] S. N. Sivanandam and S. N. Deepa, *Principles of Soft Computing (With CD)*. Hoboken, NJ, USA: Wiley, 2007.
[3] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2014, pp. 1532–1543.
[4] A. Kumar and A. Jaiswal, "Empirical study of Twitter and Tumblr for sentiment analysis using soft computing techniques," in *Proc. World Congr. Eng. Comput. Sci.*, 2017, pp. 1–5.
[5] K. Dave, S. Lawrence, and D. M. Pennock, "Mining the peanut gallery: Opinion extraction and semantic classification of product reviews," in *Proc. 12th Int. Conf. World Wide Web*, 2003, pp. 519–528.
[6] M. Bouazizi and T. Ohtsuki, "A pattern-based approach for multi-class sentiment analysis in Twitter," *IEEE Access*, vol. 5, pp. 20617–20639, 2017.
[7] S. Shayaa *et al.*, "Sentiment analysis of big data: Methods, applications, and open challenges," *IEEE Access*, vol. 6, pp. 37807–37827, 2018.
[8] A. Kumar and T. M. Sebastian, "Sentiment analysis: A perspective on its past, present and future," *Int. J. Intell. Syst. Appl.*, vol. 4, no. 10, pp. 1–14, 2012.
[9] A. Kumar and T. M. Sebastian, "Sentiment analysis on Twitter," *Int. J. Comput. Sci. Issues*, vol. 9, no. 3, pp. 372–378, 2012.
[10] T. Al-Moslmi, N. Omar, S. Abdullah, and M. Albared, "Approaches to cross-domain sentiment analysis: A systematic literature review," *IEEE Access*, vol. 5, pp. 16173–16192, 2017.
[11] A. Kumar and A. Jaiswal, "Systematic literature review of sentiment analysis on Twitter using soft computing techniques," *Concurrency Comput., Pract. Exper.*, p. e5107, 2019. doi: 10.1002/cpe.5107.
[12] S. Aloufi and A. El Saddik, "Sentiment identification in football-specific tweets," *IEEE Access*, vol. 6, pp. 78609–78621, 2018.
[13] P. F. Pai and C. H. Liu, "Predicting vehicle sales by sentiment analysis of Twitter data and stock market values," *IEEE Access*, vol. 6, pp. 57655–57662, 2018.
[14] C. W. Tseng, J. J. Chou, and Y. C. Tsai, "Text mining analysis of teaching evaluation questionnaires for the selection of outstanding teaching faculty members," *IEEE Access*, vol. 6, pp. 72870–72879, 2018.
[15] D. Wu and M. Chi, "Long short-term memory with quadratic connections in recursive neural networks for representing compositional semantics," *IEEE Access*, vol. 5, pp. 16077–16083, 2017.
[16] M. Bouazizi and T. Ohtsuki, "Multi-class sentiment analysis in Twitter: What if classification is not the answer," *IEEE Access*, vol. 6, pp. 64486–64502, 2018.
[17] D. Zhang, L. Tian, M. Hong, F. Han, Y. Ren, and Y. Chen, "Combining convolution neural network and bidirectional gated recurrent unit for sentence semantic classification," *IEEE Access*, vol. 6, pp. 73750–73759, 2018.
[18] H. Han, J. Liu, and G. Liu, "Attention-based memory network for text sentiment classification," *IEEE Access*, vol. 6, pp. 68302–68310, 2018.
[19] X. Fu, J. Yang, J. Li, M. Fang, and H. Wang, "Lexicon-enhanced LSTM with attention for general sentiment analysis," *IEEE Access*, vol. 6, pp. 71884–71891, 2018.

[20] M. Wang, Z. H. Ning, C. Xiao, and T. Li, "Sentiment classification based on information geometry and deep belief networks," *IEEE Access*, vol. 6, pp. 35206–35213, 2018.

[21] E. Camp, "Sarcasm, pretense, and the semantics/pragmatics distinction," *Noûs*, vol. 46, no. 4, pp. 587–634, Dec. 2012.

[22] E. Riloff, A. Qadir, P. Surve, L. D. Silva, N. Gilbert, and R. Huang, "Sarcasm as contrast between a positive sentiment and negative situation," in *Proc. EMNLP*, vol. 13, 2013, pp. 704–714.

[23] E. Cambria, S. Poria, F. Bisio, R. Bajpai, and I. Chaturvedi, "The CLSA model: A novel framework for concept-level sentiment analysis," in *Proc. Int. Conf. Intell. Text Process. Comput. Linguistics*. London, U.K.: Springer, 2015, pp. 3–22.

[24] M. Bouazizi and T. O. Ohtsuki, "A pattern-based approach for sarcasm detection on Twitter," *IEEE Access*, vol. 4, pp. 5477–5488, 2016.

[25] C. Junyi, S. Yan, and K.-C. Wong, "Verbal aggression detection on Twitter comments: Convolutional neural network for short-text sentiment analysis," *Neural Comput. Appl.*, pp. 1–10, Mar. 2018. doi: 10.1007/s00521-018-3442-0.

[26] B. Felbo, A. Mislove, A. Søgaard, I. Rahwan, and S. Lehmann. (2017). "Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm." [Online]. Available: https://arxiv.org/abs/1708.00524.

[27] S. Amir, B. C. Wallace, H. Lyu, and P. C. M. J. Silva. (2016). "Modelling context with user embeddings for sarcasm detection in social media." [Online]. Available: https://arxiv.org/abs/1607.00976

[28] A. Joshi, P. Bhattacharyya, M. Carman, J. Saraswati, and R. Shukla, "How do cultural differences impact the quality of sarcasm annotation?: A case study of indian annotators and american text," in *Proc. 10th SIGHUM Workshop Lang. Technol. Cultural Heritage, Social Sci., Hum.*, 2016, pp. 95–99.

[29] A. Ghosh and T. Veale, "Fracking sarcasm using neural network," in *Proc. 7th Workshop Comput. Approaches Subjectivity, Sentiment Social Media Anal.*, 2016, pp. 161–169.

[30] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: Continual prediction with LSTM," in *Proc. 9th Int. Conf. Artif. Neural Netw. (ICANN)*, 1999, pp. 1–20.

[31] M. Zhang, Y. Zhang, and G. Fu, "Tweet sarcasm detection using deep neural network," in *Proc. 26th Int. Conf. Comput. Linguistics, Tech. Papers (COLING)*, 2016, pp. 2449–2460.

[32] M. Seo, A. Kembhavi, A. Farhadi, and H. Hajishirzi. (2016). "Bidirectional attention flow for machine comprehension." [Online]. Available: https://arxiv.org/abs/1611.01603

[33] W. Wang, N. Yang, F. Wei, B. Chang, and M. Zhou, "R-NET: Machine reading comprehension with self-matching networks," Natural Lang. Comput. Group, Microsoft Res. Asia, Beijing, China, Tech. Rep. 05, 2017.

[34] A. Ghosh *et al.*, "Semeval-2015 task 11: Sentiment analysis of figurative language in twitte," in *Proc. 9th Int. Workshop Semantic Eval. (SemEval)*, 2015, pp. 470–478.

[35] J. Hltcoe, "Semeval-2013 task 2: Sentiment analysis in Twitter," Assoc. Comput. Linguistics, Atlanta, GA, USA, Tech. Rep. 2, 2013, p. 312.

[36] I. Guyon and A. Elisseeff, "An introduction to feature extraction," in *Feature Extraction*. Berlin, Germany: Springer, 2006, pp. 1–25.

[37] E. Loper and S. Bird, "NLTK: The natural language toolkit," in *Proc. ACL Workshop Effective Tools Methodol. Teach. Natural Lang. Process. Comput. Linguistics, Assoc. Comput. Linguistics*, vol. 1, 2002, pp. 63–70.

[38] M. F. Porter, "An algorithm for suffix stripping," *Program*, vol. 14, no. 3, pp. 130–137, 1980. [Online]. Available: https://tartarus.org/martin/PorterStemmer/

[39] H. Salehinejad, J. Baarbe, S. Sankar, J. Barfett, E. Colak, and S. Valaee. (2017). "Recent advances in recurrent neural networks." [Online]. Available: https://arxiv.org/abs/1801.01078

[40] A. Graves, N. Jaitly, and A.-R. Mohamed, "Hybrid speech recognition with deep bidirectional LSTM," in *Proc. IEEE Workshop Autom. Speech Recognit. Understand. (ASRU)*, Dec. 2013, pp. 273–278.

[41] M. T. Luong, H. Pham, and C. D. Manning. (2015). "Effective approaches to attention-based neural machine translation." [Online]. Available: https://arxiv.org/abs/1508.04025

[42] K. Xu *et al.*, "Show, attend and tell: Neural image caption generation with visual attention," in *Proc. Int. Conf. Mach. Learn.*, Jun. 2015, pp. 2048–2057.

[43] D. Davidov, O. Tsur, and A. Rappoport, "Enhanced sentiment learning using Twitter hashtags and smileys," in *Proc. 23rd Int. Conf. Comput. Linguistics, Posters, Assoc. Comput. Linguistics*, 2010, pp. 241–249.

[44] M. Thelwall and K. Buckley, "Topic-based sentiment analysis for the social Web: The role of mood and issue-related words," *J. Amer. Soc. Inf. Sci. Technol.*, vol. 64, no. 8, pp. 1608–1617, 2013.

[45] C. H. E. Gilbert and A. Vader, "VADER: A parsimonious rule-based model for sentiment analysis of social media text," in *Proc. 8th Int. Conf. Weblogs Social Media (ICWSM)*, 2014, pp. 1–10.

[46] Y. Li and Y. Yuan, "Convergence analysis of two-layer neural networks with relu activation," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 597–607.

[47] Y. Kim. (2014). "Convolutional neural networks for sentence classification." [Online]. Available: https://arxiv.org/abs/1408.5882.

[48] Z. Jianqiang, G. Xiaolin, and Z. Xuejun, "Deep convolution neural networks for Twitter sentiment analysis," *IEEE Access*, vol. 6, pp. 23253–23260, 2018.

[49] D. M. Powers, "Evaluation: From precision, recall and F-measure to ROC, informedness, markedness and correlation," *J. Mach. Learn. Technol.*, vol. 2, no. 1, pp. 37–63, 2011. [Online]. Available: http://hdl.handle.net/2328/27165

[50] M. P. S. Bhatia and A. Kumar, "A primer on the Web information retrieval paradigm," *J. Theor. Appl. Inf. Technol.*, vol. 4, no. 7, pp. 657–662, 2008.

[51] D. J. Hemanth, J. Anitha, A. Naaji, O. Geman, D. E. Popescu, and L. H. Son, "A modified deep convolutional neural network for abnormal brain image classification," *IEEE Access*, vol. 7, pp. 4275–4283, 2018.

[52] L. H. Son and H. Fujita, "Neural-fuzzy with representative sets for prediction of student performance," *Appl. Intell.*, vol. 49, no. 1, pp. 172–187, 2019.

[53] D. J. Hemanth, J. Anitha, L. H. Son, and M. Mittal, "Diabetic retinopathy diagnosis from retinal images using modified hopfield neural network," *J. Med. Syst.*, vol. 42, no. 12, p. 247, 2018.

[54] D. J. Hemanth, J. Anitha, and L. H. Son, "Brain signal based human emotion analysis by circular back propagation and deep Kohonen neural networks," *Comput. Elect. Eng.*, vol. 68, pp. 170–180, May 2018.

[55] C. N. Giap, L. H. Son, and F. Chiclana, "Dynamic structural neural network," *J. Intell. Fuzzy Syst.*, vol. 34, pp. 2479–2490, Jan. 2018.

[56] S. Doss *et al.*, "APD-JFAD: Accurate prevention and detection of Jelly Fish attack in MANET," *IEEE Access*, vol. 6, pp. 56954–56965, 2018.

[57] D. T. Hai, L. H. Son, and T. Le Vinh, "Novel fuzzy clustering scheme for 3D wireless sensor networks," *Appl. Soft Comput.*, vol. 54, pp. 141–149, May 2017.

[58] R. Kapoor, R. Gupta, R. Kumar, L. H. Son, and S. Jha, "New scheme for underwater acoustically wireless transmission using direct sequence code division multiple access in MIMO systems," *Wireless Netw.*, pp. 1–13, May 2018. doi: 10.1007/s11276-018-1750-z.

[59] R. Kapoor, R. Gupta, L. H. Son, S. Jha, and R. Kumar, "Detection of power quality event using histogram of oriented gradients and support vector machine," *Measurement*, vol. 120, pp. 52–75, May 2018.

[60] R. Kapoor, R. Gupta, L. H. Son, S. Jha, and R. Kumar, "Boosting performance of power quality event identification with KL divergence measure and standard deviation," *Measurement*, vol. 126, pp. 134–142, Oct. 2018.

[61] H. V. Long, M. Ali, L. H. Son, M. Khan, and D. N. Tu, "A novel approach for fuzzy clustering based on neutrosophic association matrix," *Comput. Ind. Eng.*, to be published. doi: 10.1016/j.cie.2018.11.007.

[62] P. T. M. Phuong, P. H. Thong, and L. H. Son, "Theoretical analysis of picture fuzzy clustering: Convergence and property," *J. Comput. Sci. Cybern.*, vol. 34, no. 1, pp. 17–32, 2018.

[63] Y. H. Robinson, E. G. Julie, K. Saravanan, R. Kumar, and L. H. Son, "FD-AOMDV: Fault-tolerant disjoint ad-hoc on-demand multipath distance vector routing algorithm in mobile ad-hoc networks," *J. Ambient Intell. Hum. Comput.*, pp. 1–18, Nov. 2019. doi: 10.1007/s12652-018-1126-3.

[64] K. Saravanan, E. Anusuya, R. Kumar, and L. H. Son, "Real-time water quality monitoring using Internet of Things in SCADA," *Environ. Monitor. Assessment*, vol. 190, no. 9, p. 556, 2018.

[65] K. Saravanan, S. Aswini, R. Kumar, and L. H. Son, "How to prevent maritime border collision for Fisheries?—A design of real-time automatic identification system," *Earth Sci. Inform.*, pp. 1–12, Nov. 2019. doi: 10.1007/s12145-018-0371-5.

[66] K. Singh, K. Singh, L. H. Son, and A. Aziz, "Congestion control in wireless sensor networks by hybrid multi-objective optimization algorithm," *Comput. Netw.*, vol. 138, pp. 90–107, Jun. 2018.

[67] N. Singh, L. H. Son, F. Chiclana, and J.-P. Magnot, "A new fusion of salp swarm with sine cosine for optimization of non-linear functions," *Eng. Comput.*, pp. 1–28, Jan. 2019. doi: 10.1007/s00366-018-00696-8.

[68] L. H. Son, "A novel kernel fuzzy clustering algorithm for geo-demographic analysis," *Inf. Sci.-Inform. Comput. Sci., Intell. Syst., Appl., Int. J.*, vol. 317, pp. 202–223, Oct. 2015.

[69] L. H. Son, "Generalized picture distance measure and applications to picture fuzzy clustering," *Appl. Soft Comput.*, vol. 46, pp. 284–295, Sep. 2016.

[70] L. H. Son and P. V. Hai, "A novel multiple fuzzy clustering method based on internal clustering validation measures with gradient descent," *Int. J. Fuzzy Syst.*, vol. 18, no. 5, pp. 894–903, 2016.

[71] S. Jha, R. Kumar, J. M. Chatterjee, and M. Khari, "Collaborative handshaking approaches between Internet of computing and Internet of Things towards a smart world: A review from 2009–2017," *Telecommun. Syst.*, pp. 1–18, Jun. 2018. doi: 10.1007/s11235-018-0481-x.

[72] L. H. Son and N. D. Tien, "Tune up fuzzy C-means for big data: Some novel hybrid clustering algorithms based on initial selection and incremental clustering," *Int. J. Fuzzy Syst.*, vol. 19, no. 5, pp. 1585–1602, 2017.

[73] L. H. Son and T. M. Tuan, "A cooperative semi-supervised fuzzy clustering framework for dental X-ray image segmentation," *Expert Syst. Appl.*, vol. 46, pp. 380–393, Mar. 2016.

[74] L. H. Son and P. H. Thong, "Some novel hybrid forecast methods based on picture fuzzy clustering for weather nowcasting from satellite image sequences," *Appl. Intell.*, vol. 46, no. 1, pp. 1–15, 2017.

[75] L. H. Son and T. M. Tuan, "Dental segmentation from X-ray images using semi-supervised fuzzy clustering with spatial constraints," *Eng. Appl. Artif. Intell.*, vol. 59, pp. 186–195, Mar. 2017.

[76] N. T. Tam, D. T. Hai, L. H. Son, and L. T. Vinh, "Improving lifetime and network connections of 3D wireless sensor networks based on fuzzy clustering and particle swarm optimization," *Wireless Netw.*, vol. 24, no. 5, pp. 1477–1490, 2018.

[77] N. D. Thanh, M. Ali, and L. H. Son, "A novel clustering algorithm in a neutrosophic recommender system for medical diagnosis," *Cogn. Comput.*, vol. 9, no. 4, pp. 526–544, 2017.

[78] P. H. Thong and L. H. Son, "Picture fuzzy clustering: A new computational intelligence method," *Soft Comput.*, vol. 20, no. 9, pp. 3549–3562, 2016.

[79] P. H. Thong and L. H. Son, "A novel automatic picture fuzzy clustering method based on particle swarm optimization and picture composite cardinality," *Knowl.-Based Syst.*, vol. 109, pp. 48–60, Oct. 2016.

[80] P. H. Thong and L. H. Son, "Picture fuzzy clustering for complex data," *Eng. Appl. Artif. Intell.*, vol. 56, pp. 121–130, Nov. 2016.

[81] T. M. Tuan, T. T. Ngan, and L. H. Son, "A novel semi-supervised fuzzy clustering method based on interactive fuzzy satisficing for dental X-ray image segmentation," *Appl. Intell.*, vol. 45, no. 2, pp. 402–428, 2016.

LE HOANG SON received the Ph.D. degree in mathematics–informatics from the VNU University of Science, Vietnam National University (VNU), in 2013. He was a Senior Researcher and the Vice Director with the Center for High Performance Computing, VNU University of Science, Vietnam National University, from 2007 to 2018. He has been promoted to an Associate Professor in information technology, since 2017. Since 2018, he has been the Head of the Department of Multimedia and Virtual Reality, VNU Information Technology Institute, VNU. He is also Visiting Professor with Ton Duc Thang University, Vietnam. His current research interests include artificial intelligence, data mining, soft computing, fuzzy computing, fuzzy recommender systems, and geographic information systems.

AKSHI KUMAR received the B.E. degree (Hons.) in computer science and engineering from Maharishi Dayanand University, Rohtak, in 2003, the M.Tech. (Hons.) in computer science and engineering from Guru Gobind Singh Indraprastha University, Delhi, in 2005, and the Ph.D. degree in computer engineering (Web mining) from the Faculty of Technology, University of Delhi, in 2011. She has been with the university for the past ten years. She is currently an Assistant Professor with the Department of Computer Science & Engineering, Delhi Technological University (formerly Delhi College of Engineering). Her research interests include intelligent systems, user-generated big-data, social media analytics, and soft computing.

SAURABH RAJ SANGWAN received the bachelor's degree in computer science and engineering from DCRUST, Murthal, India, and the M.Tech. degree in software engineering from the Department of Computer Science & Engineering, Delhi Technological University, Delhi, India, in 2018. He is currently a Research Scholar with the Web Research Group, Department of Computer Science & Engineering, Delhi Technological University. His research interests include intelligent systems, text mining, and social Web.

ANSHIKA ARORA received the B.Tech. degree in computer science and engineering from the University of Delhi, New Delhi, India, in 2017. She is currently pursuing the M.Tech. degree in software engineering with Delhi Technological University, Delhi, India. Her research interests include social Web, sentiment analysis, and machine learning.

ANAND NAYYAR received the M.C.A. degree (Hons.) from Punjabi University, Patiala, in 2008, the M.Phil. degree in computer science with the 1st Division from Vinayaka Missions University, in 2009, the M.Tech. degree (Hons.) in information technology in 2011, the M.B.A. degree in information systems from Sikkim Manipal University with 1st Division, and the Ph.D. degree in computer science from Desh Bhagat University, Mandi Gobindgarh, in 2017. He is currently a Professor, a Researcher, and a Scientist with the Graduate School, Duy Tan University, Da Nang, Vietnam. He has more than 300 Publications in various National, International Conferences cum International Journals (Scopus/ISI/SCI/SCIe) of high repute. He is currently in WSN, MANET, swarm intelligence, cloud computing, big data, network security, the Internet of Things (IoT), machine learning, deep learning, network simulation, and 5G communications.

MOHAMED ABDEL-BASSET received the B.Sc., M.Sc., and Ph.D. degrees in information systems and technology from the Faculty of Computers and Informatics, Zagazig University, Egypt. He is working on the application of multi-objective and robust meta-heuristic optimization techniques. He has published more than 150 articles in international journals and conference proceedings. His current research interests include optimization, operations research, data mining, computational intelligence, applied statistics, decision support systems, robust optimization, engineering optimization, multi-objective optimization, swarm intelligence, evolutionary algorithms, and artificial neural networks. He holds the program chair in many conferences in the fields of decision making analysis, big data, optimization, complexity and the Internet of Things, and editorial collaboration in some journals of high impact. He is also an/a Editor/Reviewer in different international journals and conferences.