Received February 28, 2021, accepted March 17, 2021, date of publication March 23, 2021, date of current version April 2, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3068323

Context-Based Feature Technique for Sarcasm Identification in Benchmark Datasets Using Deep Learning and BERT Model

CHRISTOPHER IFEANYI EKE[1,2], AZAH ANIR NORMAN[1], AND LIYANA SHUIB[1]
[1]Department of Information Systems, Faculty of Computer Science and Information Technology, University of Malaya, Kuala Lumpur 50603, Malaysia
[2]Department of Computer Science, Faculty of Computing, Federal University, P.M.B 046, Lafia, Nigeria

Corresponding authors: Azah Anir Norman (azahnorman@um.edu.my) and Liyana Shuib (liyanashuib@um.edu.my)

ABSTRACT Sarcasm is a complicated linguistic term commonly found in e-commerce and social media sites. Failure to identify sarcastic utterances in Natural Language Processing applications such as sentiment analysis and opinion mining will confuse classification algorithms and generate false results. Several studies on sarcasm detection have utilised different learning algorithms. However, most of these learning models have always focused on the contents of expression only, leaving the contextual information in isolation. As a result, they failed to capture the contextual information in the sarcastic expression. Secondly, many deep learning methods in NLP uses a word embedding learning algorithm as a standard approach for feature vector representation, which ignores the sentiment polarity of the words in the sarcastic expression. This study proposes a context-based feature technique for sarcasm Identification using the deep learning model, BERT model, and conventional machine learning to address the issues mentioned above. Two Twitter and Internet Argument Corpus, version two (IAC-v2) benchmark datasets were utilised for the classification using the three learning models. The first model uses embedding-based representation via deep learning model with bidirectional long short term memory (Bi-LSTM), a variant of Recurrent Neural Network (RNN), by applying Global Vector representation (GloVe) for the construction of word embedding and context learning. The second model is based on Transformer using a pre-trained Bidirectional Encoder representation and Transformer (BERT). In contrast, the third model is based on feature fusion that comprised BERT feature, sentiment related, syntactic, and GloVe embedding feature with conventional machine learning. The effectiveness of this technique is tested with various evaluation experiments. However, the technique's evaluation on two Twitter benchmark datasets attained 98.5% and 98.0% highest precision, respectively. The IAC-v2 dataset, on the other hand, achieved the highest precision of 81.2%, which shows the significance of the proposed technique over the baseline approaches for sarcasm analysis.

INDEX TERMS Natural language processing, sarcasm identification, Bi-LSTM, GloVe embedding, BERT.

I. INTRODUCTION

Recently, affective computing and sentiment analysis research has gained much recognition [1]. The notion behind sentiment analysis is to determine the polarity of the emotion word in an expression. Analysis of people's sentiment (also referred to as opinion mining) identifies subjective information in source documents. The process of identifying people's opinions (sentiments) about products, politics, services, or individuals brings a lot of benefits to the organisations [2], [3]. The possibility of identifying subjective information is essential. It helps generate structured knowledge that serves as a piece of important knowledge for decision support systems and individual decision-making [4]. For instance, affective computing and sentiment analysis can improve recommendation systems and customer relationship management by revealing customers' likes and dislikes or eliminating the item recommendations that got negative feedback from the customers [5]. Most of the social content found on the Web consists of figurative words such as sarcasm and irony. For example, the Internet Argumentation Corpus obtained from *4forums.com* consists of 12% sarcastic utterances [6]. In social media, various people usually employ sarcasm or irony to show their emotions, making it difficult to analyse

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

people's sentiments. Sarcastic languages can shift the sentiment polarity in the textual document, which might reduce the predictive accuracy of sentiment analysis. In sarcastic statements, there is a contradiction between the expressed textual utterances and the individual's aim in making such sarcastic utterances.

According to [7], Sarcasm is defined as ''a verbal device, with the intention of putting someone down or an act of saying one thing while the meaning is opposite.'' People use sarcastic statements that correspond to the reverse of what they speak to injure one's emotion. Sarcasm identification studies have gained attention in recent years. Maynard and Greenwood [7], in their research, demonstrated that sarcasm identification from sarcastic utterances might enhance the accuracy of sentiment analysis. Sarcasm identification has become an essential step in analysing people's sentiments [8], [9]. A sarcastic statement represents a conflict between the individual's motive making the utterance and the actual composition. For instance, a sarcastic expression, ''I love to work on holidays!'' shows a conflict between the clear statement ''on holidays'' and the expression ''love.'' The contradiction and the sentiment polarity shift prove that Sarcasm is a unique case of sentiment analysis. Sarcasm is extremely contextual and topic reliant, and as a result, some contextual clues and shifts in polarity sentiment can assist in sarcasm identification in the text. Moreover, it resolves the obscurity of the meaning and improves the overall sentiment classification of a huge amount of user's textual data obtained from social media. The insufficient knowledge of the situation ''Context,'' the environment, and the specific topic will result in difficulty detecting sarcastic utterances [10]. Context understanding is one of the main challenging phases of moderation content. The term ''Context'' in sentiment analysis refers to supplementary support that may either increase or change the content polarity. Thus, the context vector determines the accuracy of the sentiment analysis, and the predictive model will guarantee the reliability of the overall prediction.

Various studies on sarcasm identification have relied on content and pattern-based features. For instance, Mukherjee and Bala [11] employed content-based linguistic features for sarcasm classification. The study relied solely on the emoticon, word use, and the sentence, in general, to differentiate sarcastic from non-sarcastic sentences. The technique produced reasonable performance results based on the data set that was used. However, the predictive model performance relied intensely on the linguistic feature content, which is likely to degrade when applied to other data sets due to its dependence on word use. Hence, the obtained result cannot be generalised to a satisfactory extent.

Similarly, Rajadesingan, *et al.* [12] investigated the behavioural method for sarcasm analysis by utilising psychological and behavioural theories to build the behavioural model. Different features that include emotion, complexity, expression, and contrast were extracted to train the model. The study's empirical analysis showed enhanced performance on the proposed method compared with the conventional approaches. Similarly, Bouazizi and Ohtsuki [13] defined a pattern method for sarcasm classification on tweets. In their proposed technique, the author separated the sarcasm identification algorithm into four different analyses: syntactic-related, sentiment-related, punctuation-related, and pattern-related features for the analysis. They proposed an effective and reliable pattern for sarcasm identification by grouping words into two separate categories: ''CI'' and ''GFI.'' While the CI emphasises the importance of the word's contents in the expression, the GFI class concentrates more on the grammatical function of the word. The Random Forest classifier was employed for prediction purposes, and an accuracy of 83.1% with a precision of 91.1% was obtained. Thus, the comparative analysis indicated that the pattern-based technique outperformed other methods. However, the study relied more on the words' patterns in the expression, which are not sufficient in capturing all the sarcastic sentiments. The approaches mentioned above performed optimally well in sarcasm analysis. However, they failed to recognise the importance of context information in sarcastic utterances to address the ambiguity associated with a sarcastic expression.

In this research, the context embedding that considers both local and global context information has been employed to construct the deep learning and BERT model features by considering feature fusion techniques for sarcasm classification using textual data. Three benchmark datasets provided by Riloff *et al.* [14], Ghosh and Vale [15], and IAC-v2 [16] have been utilised to test the model. A deep learning architecture consisting of Bi-LSTM has been employed with GloVe embedding to construct context vectors that represent a semantic word as features. GloVe embedding integrates 'local context window' and 'Global statistics of matrix factorisation' methods, which are the two key model families [17]. The GloVe embedding can construct a word representation that learns grammatical and semantic information and captures the word's context and global corpus information. Four major performance metrics, such as precision, recall, accuracy, and f-measure, have been utilised to evaluate the model's empirical analysis. The main contributions of this study are summarised below.

- ➢ Understand sarcasm as a unique instance of context-based sentiment analysis.
- ➢ Define and extract local content and the global context by employing GloVe embedding features.
- ➢ Build a deep learning model based on Bi-LSTM to automatically identify sarcasm using context information to address the feature engineering problem.
- ➢ Proposes a feature fusion technique, which comprised BERT feature, hashtag feature, sentiment related feature, syntactic features, and GloVe embedding feature for sarcasm classification. To the best of our knowledge, this is the first study that integrates BERT feature and word embedding with linguistic and sentiment related feature to improve the classification performance for sarcasm identification to address the context and sentiment polarity issue in sarcastic utterances

- The proposed technique was evaluated via various extensive experiments on the two benchmark Twitter datasets, and the results demonstrated that the proposed technique slightly outperformed the baseline methods for sarcasm classification.

The remainder of the sections are organised; thus: In section 2, the literature survey is discussed. Section 3 provides the proposed technique. In section 4, the experimental settings and procedures are discussed. Section 5 presents the experimental design. In section five, empirical results and discussion are presented. Section 6 finally concludes the article with suggestions for future work.

II. LITERATURE SURVEY

Sarcasm identification task has been studied by employing different methods, including lexicon-based, conventional machine learning, deep learning, or even a hybrid approach. Besides, several reviews on sarcasm detection have also been conducted. For instance, Eke, *et al.* [18] performed SLR on sarcasm identification in Textual data. The study was carried out by considering 'dataset collection, preprocessing techniques, feature engineering techniques (feature extraction, feature selection, and feature representation), classification algorithms, and performance measures.' The study revealed that content-based features are the most employed features for sarcasm classification. The study also revealed that the standard evaluation metrics such as precision, recall, accuracy, f-measure, and Area under the curve (AUC) are the most used parameters for evaluating classifiers' performance. Moreover, the study also revealed that when there is an imbalance in the class distribution of the dataset, the AUC performance measure is the right choice due to its robustness in resisting the skewness in the dataset. The review concluded by identifying recent challenges and proposing the open research direction to provide a solution to the sarcasm identification studies issue.

Various scholars have studied sarcasm identification tasks Joshi, *et al.* [19], stated two methods, namely, the ''Incongruent words-only'' method and ''all-words: method'' in ''Expect the Unexpected: Harnessing Sentence Completion for Sarcasm Detection'' research by employing ''Sentence completion'' for sarcastic analysis: For evaluation purpose, two sets of data were used, which includes (i) Twitter data collected by Riloff, *et al.* [14], consisting 2278 tweets ('506 sarcastic, and 1772 non-sarcastic). (ii) Discussion forum data collected by Walker, *et al.* [6] that contain manually labelled balanced tweets ('752 sarcastic and 752 non-sarcastic). However, 'WordNet similarities and word2vec' were employed to measure the similarities in the performance. Two-fold cross-validation was used for evaluation purposes. Thus, the overall predictive results attained an F-score of 54% by employing the Word2Vec similarity for the all-words method, but 80.24% of F-score was obtained with the WordNet incongruous words-only method. On the other hand, an 80.28% F-score is obtained using the WordNet similarity and Incongruous words-only method with 2-fold cross-validation.

In addition to the handcrafted feature proposed by various studies for sarcasm classification, the word polarity disambiguation approach has also gained recognition by many scholars in recent years. For instance, Wu and Wen [20] studied a Knowledge-based (unsupervised) approach for automatic disambiguation of dynamic sentiment-ambiguous adjectives using a search engine. Remarkably, the author exploited character-based and pattern-based approaches to extrapolate nouns' sentiment expectation and find adjectives' polarity. In another study, Xia, *et al.* [21] proposed an approach that utilises opinion-level features to resolve words' polarity ambiguity. In this approach, the author examined the inter-opinion (e.g., Discourse, correlative words in the sentence) and intra-opinion (e.g., Indicative words, opinion target) feature. They employed the probability approach to resolving the word polarity disambiguation by adopting the Bayesian model. However, they experimented on pinion corpus, and the results of the experiment showed a substantial impact in the disambiguation of word polarity using the opinion-level feature. In a recent study, Wang, *et al.* [22] proposed the word sense disambiguation deep learning method. In the proposed approach, the sense path in a target context is modelled by exploiting the domain-specific background knowledge from WordNet by employing the word embedding feature extracted from an external corpus. The method revealed the hidden semantic relationship within word sense by utilising the 'PageRank algorithm' to exploit sense path via WordNet structure while representing the text context target with latent semantic analysis. In a related study,.Abdalgader and Al Shibli [23] proposed a variant of the 'graph-based word sense disambiguation approach by exploiting all the occurrence of semantic information acquired using the WordNet to facilitate graph semantics connection for finding the anticipated meanings of words in a specified context. In this proposed approach, the similarity between the graph nodes that comprised all related word semantic information for sentence-level disambiguation is measured. Next, the real meanings are concurrently allocated to every target word by applying a graph centrality measure that provides the important degree between the graph nodes. However, the experimental evaluation and comparison results of the approach with the benchmark dataset outperformed the state-of-the-arts WSD approaches.

The idea of the ensemble learning approach was initially proposed by Fersini, *et al.* [24] while studying ''Detecting Irony and Sarcasm in Microblog: the role of Expressive Signals and Ensemble Classifiers.'' The author considers the 'Bayesian Model averaging' and various classification algorithms based on their reliabilities and marginal probability predictions. However, they considered Bayesian Model Averaging and Majority Voting as the main ensemble approach in the classification phase. For evaluation purposes, the author selected the baseline model that attained the best predictive performance and four configurations that include BoW, PoS,

PP, PP & PoS. However, the predictive analysis indicates that the proposed majority voting ensemble model performed better than the single classifier. The author also showed that positives could enhance Sarcasm and that pragmatic features are discriminative in capturing ironic utterances. In another study, ONAN, *et al.* [25] proposed a Turkish news article's satire detection method. The authors employed linguistic Inquiry and word count software for feature extraction by considering the linguistic and psychology feature sets. In this study, ensemble learning, five deep learning architecture, and word embeddings scheme were considered. The experimental analysis of the proposed approach showed that the deep learning approach outperformed other approaches, which showed the significance of the proposed methods. Ptáček, *et al.* [26] made the first attempt to study a multilingual approach for sarcasm identification. In their study, two different languages were considered (English and Czech). The authors utilised both English and Czech datasets to compare the sarcastic occurrence in both languages. The dataset consists of 140,000 tweets composed in Czech and 780,000 tweets composed in English. Twitter API was utilised to stream the tweets. During the classification phase, two classifiers (Support vector machine and Maximum entropy classifier) were employed to evaluate the models' predictive performance. In the testing phase, a 5-fold cross-validation approach was used in each classifier. Thus, 0.947 and 0.924 F-measure was achieved on a balanced and imbalanced English dataset with the RF classifier. However, SVM produced a better result on the Czech dataset, attaining an F-measure of 0.582 by enhancing features with different patterns.

In relation to affective computing and sentiment classification, Esuli, *et al.* [27] proposed a cross-lingual sentiment qualification approach whereby the training data are present in a source language but absent in the target language required for performing sentiment qualification. Thus, this method addresses those application contexts whereby there is a presence of the training document for different source languages and the absence of the training document on the interested target language. The author utilised the distributional correspondence indexing (DCI) and structural correspondence learning (SCL) approach for cross-lingual text classification. However, the experimental analysis using the benchmark datasets on cross-lingual sentiment classification yielded promising prediction results on cross-lingual sentiment qualification. In another study, Yang, *et al.* [28] proposed a new method tagged ''Segment-level joint topic-sentiment model (STSM)'' sentiment classification. The proposed approach aimed to determine the document sentiment polarity by capturing the correlation of the topic sentiment. The author modelled the joint topic-sentiment's correlation by inserting the sentiment layer between the segment and topic layers. However, the sentiment classification's predictive performance shows that the proposed approach can enhance complex and compound sentences' performance. Moreover, the alignment of sentiment and topics also indicates the significance of the proposed method. Agrawal, *et al.* [29], in their study on sarcasm identification, explored emotion categories features such as sadness, happiness, surprise, etc.; the authors went deeper by considering the sequential information encoding among the effective features state. The comparative analysis of the proposed approach demonstrates the effectiveness of the method. In a recent study, Onan, *et al.* [30] examined the classification performance of conventional machine learning and deep learning models for sentiment analysis on product review, and the predictive performance indicates the effectiveness of the proposed method on sentiment classification.

Recently, the application of multi-tasks learning has gained recognition and has been demonstrated in various NLP tasks, including key phrase boundary detection [31] and implicit discourse relationship detection [32]. In a related study, Majumder, *et al.* [33] proposed a 'multitask learning framework using DNN' for sentiment and sarcasm identification study. In their research, they demonstrated that the two tasks are related and, as a result, modelled the two tasks using a single neural network. However, the experimental results slightly performed better than the existing approach, which reveals that the multi-task network improves sarcasm classification in sarcasm and a polarity shift in the sentence. In a related study, Mishra, *et al.* [34] proposed a method for automatic cognitive feature extraction using a CNN variation for sentiment and sarcasm identification tasks. The author utilised the gaze data present in the dataset. In the modelling phase, the author modelled the two tasks separately. The experimental analysis of the learned features shows that the hybrid of automatically learned features produces a promising result, indicating that it can represent deep linguistic subtleties in the textual data, which has remained an issue in sentiment and sarcasm classification studies.

Riloff, *et al.* [14] presented an approach for detecting a specific form of Sarcasm whereby a contradiction exists between positive sentiment and negative situations. They proposed a 'bootstrapping algorithm' that utilises a single seed word that automatically identifies and learns a phrase that shows positive sentiment and negative situations from the sarcastic tweets. The authors created two baseline approaches and employed the LIBSVM library to model SVM classifiers, using 10-fold cross-validation for model evaluation. However, a precision of 64% and 39% recall were obtained by employing SVM on unigram and bigram features. Thus, this method performed optimally well, but many sarcastic tweets were not captured in the classes mentioned above of Sarcasm. Also, the method depends on the occurrence of every possible ''Negative situation'' on the training data, which is inefficient on a new tweet data Mukherjee and Bala [11] presented a method that provides knowledge to a system, which interprets the author's linguistic style by considering different sets of features for sarcasm identification in a microblog. They used authorial style-based features for their study, and in the classification phase, Naïve Bayes and fuzzy clustering algorithms were used. The experimental analysis indicates that the use of supervised and unsupervised learning, and the inclusion of

features that are independent of text produced better accuracy in sarcasm detection. However, the approach is only limited to authorial style-based features and may not work well with other feature sets.

In the existing methods, word-level approaches require a couple of times in training big social data analysis. To overcome the limitation, Hussain and Cambria [35] proposed a novel 'Semi-supervised learning model' by combining 'random projection scaling' as part of the vector space model (VSM) and a SVM to implement cognitive on a knowledge-based of affective common sense. However, the experimental analysis results revealed an important enhancement in both polarity identification and emotion detection over the classification rule because both labelled data and unlabelled data are employed for classification learning compared to the formal method that utilises only the labelled data. Thus, it opened the opportunity for further research on semi-supervised learning methods on big social data analysis. In a related study, Duan, *et al.* [36] proposed a new semi-supervised learning method that considers both training and testing sets for sentiment classification of stock text messages. The method was proposed to resolve the issue common in short message modelling, such as data sparsity in mathematical representation. Moreover, the author constructed a Generative Emotion Model with categorised words (GEM-CW) to extract sentiment features from both training and testing sets. However, the extracted features were more discriminative for sentiment classification than those derived using the conventional approach that only considers training sets. The analysis results indicate that the proposed learning approach and the model are significant for sentiment classification in short text and can attain better results than the traditional methods.

Rajadesingan, *et al.* [12] went deeper and looked into the psychology involved in the sarcastic expression. Their study presented behavioural modelling for sarcasm detection by identifying the various forms of Sarcasm and their existence in tweets. The study shows the significance of historical information acquired from the past tweet in identification sarcasm. Though the approach looks very effective in such an instance, it cannot perform well in the absence of past knowledge about the user. This is because most of the features employed in the study were extracted from the data obtained from the past tweet to make a decision. Thus, it is difficult to apply the approach for a real-time stream of tweets, where users are randomly posting tweets due to the fast growth in the size of the knowledge base, which requires the repetition of training on data each time new tweet data is acquired.

The paradigm of the deep learning approach has recently attracted various researchers to combine it with the conventional machine learning approach for sarcasm identification. For instance, Mehndiratta, *et al.* [37] presented a method of automatic sarcasm identification in textual data using a DCNN. Their study used sentiment polarity as a feature set and extracted feature vector using the skip-gram word2vec model technique. The authors further fed the feature into the convolutional neural network. Their study performed optimally well but has a limitation of word sense not being captured separately. Ghosh and Veale [15] proposed a DNN model for sarcasm classification in tweets. The study integrated machine learning with a deep learning model (a hybrid of CNN, DNN, and LSTM). However, the proposed model's predictive results outperformed the baseline approach for sarcasm detection by attaining an F-score of 92% [38]. Similarly, Onan [39] conducted a study on ''Topic-Enriched word embedding for sarcasm Identification.'' The study employed a deep learning method by comparing Topic-enriched word-embedding models with traditional word embedding variations: GloVe, Word2vec, LDA2vec, and FastText. Besides, the author also experimented with conventional features, including pragmatic, incongruity (implicit & explicit), and lexical features. The experimental analysis was performed on a dataset by considering various subsets, ranging from 5,000 to 30,000. However, the aforementioned model's performance showed that LDA2vec produced a better result compared with other word embedding schemes. Besides, the fusion of conventional pragmatic features, lexical, explicit, and implicit incongruity with the word embedding scheme enhance the model's predictive performance. Recently,

In a recent study, [40] employed multimodal features that consist of textual, speech, and video features to recognise Sarcasm. The textual features in the data sets were represented using BERT (Bidirectional Encoder Representation from Transformer) [41], a specification for sentence representation. On the other hand, speech feature extraction was extracted using Libnsa, a well-known library for speech extraction [42], by considering only the low-level feature for audio data to exploit the audio modality information. Also, pool five layers of an ImageNet [43] were utilised on each frame for visual feature extraction in video pronouncement. However, the experimental analysis indicated that multimodal features produced a better predictive performance than the unimodal features with about a 12.9% reduction in error rate. Recently, [44] presented an effective sarcasm identification framework on social media data by considering a deep learning approach with neural language models such as FastText, GloVe, and word2vec. The authors introduced inverse gravity moment based on weighted word embedding with trigram. The empirical analysis of the proposed framework attained an accuracy of 95.3%, which indicates the effectiveness of the proposed framework.

In this study, a novel context-based features technique is proposed for sarcasm identification in three benchmark datasets. Two learning models were constructed for the proposed technique. The first model uses semantic features based on word embedding using a global vector (GloVe). Word embedding helps in learning the representations and relationships among words. The GloVe is a count-based model that captures the relationship between words in a sentence (relatedness) and constructs the learned representation of a real value words vector. Moreover, it helps map all the tokenised

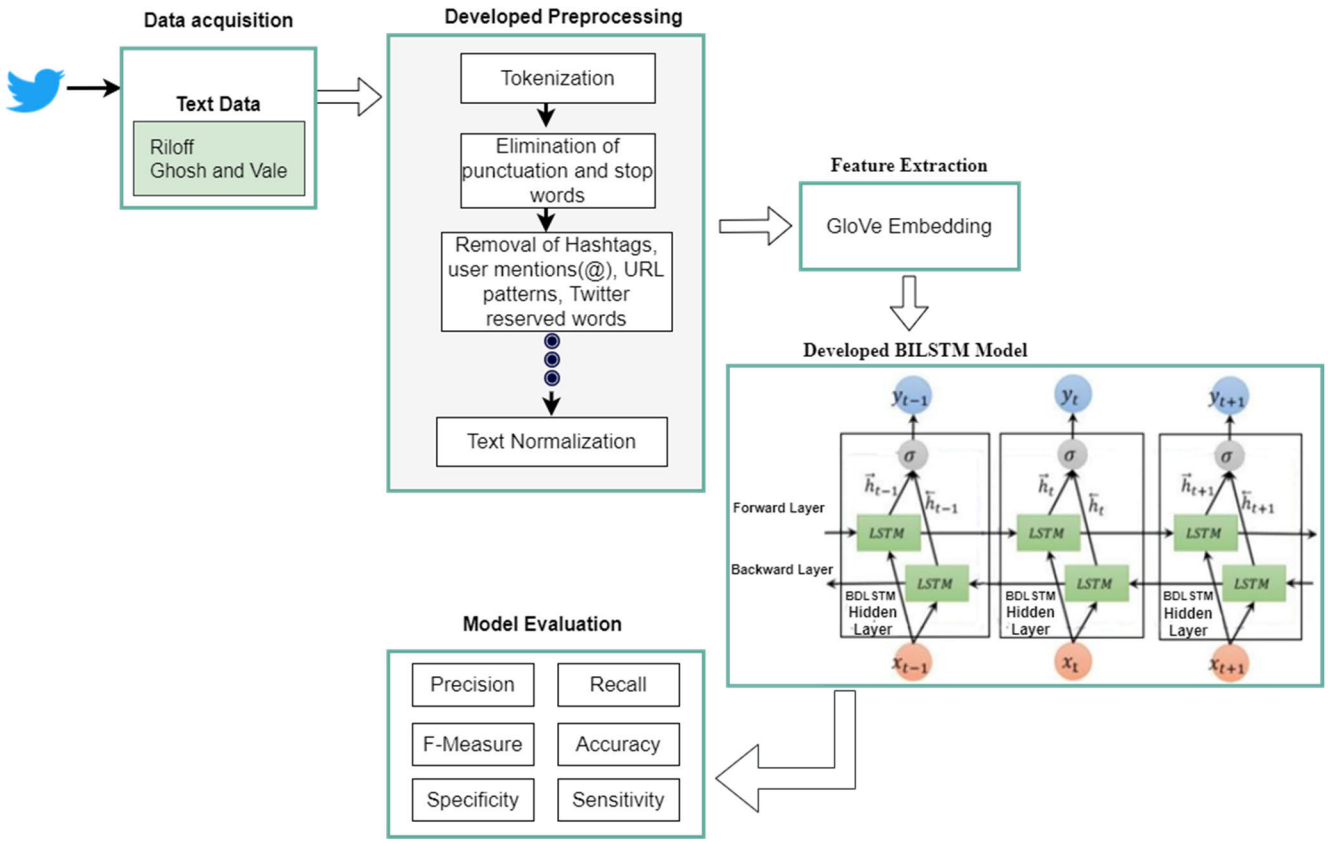

FIGURE 1. Systematic flow of a deep learning approach.

words in every tweet to its corresponding word table vector. Thus, the generated features are employed to train and test the model for sarcastic and non-sarcastic categories. The second model is constructed based on transformer learning (BERT) and feature fusion that comprised BERT feature, hashtag feature, sentiment related feature, syntactic feature, and Glove embedding feature. BERT is the state-of-the-art model that captures the context in sarcastic expression.

III. AUTOMATIC SARCASM IDENTIFICATION

The automatic sarcasm identification model primarily consists of five major components: (i) Data acquisition (ii) Data preprocessing. (iii) Feature Extraction (iv) Construction of classification model, and (v) Evaluation of the constructed model. In this study, two learning model approaches have been considered to automatically identify sarcasm in textual data. They include a deep learning-based approach and a transformer-based approach. Each of the approaches is explained in the sections below.

A. DEEP LEARNING BASED APPROACH

This section provides a methodological description of the deep learning methodology. The framework for the deep learning process is depicted in FIGURE 1. Each of the framework segments is described below.

1) DATA ACQUISITION

The sarcasm identification tasks begin with the collection of datasets for the study. Dataset is very crucial in any data mining studies. In this approach, two benchmarks Twitter dataset were utilised to construct sarcastic and non-sarcastic datasets. Twitter, social media platform, enables users to exchange their ideas, news, and emotion with their co-users. With the help of Twitter API, a connection between the Twitter server and users is provided to make the tweet archive easily accessible. One of the major advantages of Twitter data is that one can collect as many tweets as possible since people posts messages daily. In this approach, two benchmark datasets that natural language processing researchers popularly use were employed, including the Riloff dataset [14] and Ghosh and Veale [15]. Riloff dataset is the first public tweet dataset for sarcasm identification collected by [14]. The dataset is manually annotated and validated by experts. Ghosh and Veale obtained their tweet dataset with a hashtag labelled as #sarcastic, #sarcasm, #ironic [26]. The authors employed a feedback-based approach that enabled them to validate the sarcasm label by consulting the authors. The statistical description of the two datasets is given in Table 1.

2) DATA PRE-PROCESSING

The first step in the preprocessing of tweets is tokenisation. Each text data (tweet) is divided into smaller parts in the tokenisation step, either into words or sentences. Tokenisation tasks can be performed using the NLTK library. Next is the elimination of unwanted information. Most of the social media data, especially Twitter, come along with some noise and, as a result, requires a preprocessing step to eliminate

TABLE 1. Statistical description of the dataset.

Dataset	Train	Test	Dev	Total
Tweets [14]	1564	274	118	1,956
Tweets [15]	43,943	7,690	3296	54,929

those unwanted items. For instance, tweets may contain stop words, punctuations, special characters (such as @, #, etc.), and URL links. Thus, all the items that do not contribute to the classification task are eliminated before the feature extraction process. Data set can be prepared by removing those unwanted items from the tweet contents, such as special characters, numbers, hashtags, tweets composed in other languages than English, stop words, tweets shorter than three words length, and URL links [45], [46]. Also, the other basic preprocessing methods, including text normalisation (stemming, lemmatising, lower case conversion, word equivalent number conversion, and handling of the misspellings), and POS (parts of speech) tagging are also performed in this stage, which can be implemented using the Natural Language Processing (NLP) toolkit. The processed data is required to be transformed into an array representation of the features to simplify the model training.

3) FEATURE EXTRACTION

In this approach, the word embedding feature (glove) feature has been utilised. Textual word is usually regarded as discrete and categorical features. Thus, it is required to be represented in a vector format. The vector representation can be carried out by converting the text to the vector space model (VSM). This process can be performed in two different stages. In the first stage, a dictionary of the dataset's term is created (tweets dataset in this study). In so doing, each unique dataset term is defined in the vector space with a unique identity.

On the other hand, in the second stage, the numerical representation of terms' is obtained and added to the vector space. The representation can be done by employing some methods such as TFIDF, TF, and word embedding. However, word embedding representation is used in this study. Word embedding can be considered the state-of-the-art approach for word representation in low dimensional vector space without compromising the contextual similarity. Also, an almost similar representation can be obtained on the words with the same meaning. In contrast, optimum performance can be attained by training the embedding with a large amount of textual data. Most popular embedding methods include GloVe, BERT, XLNet, Word2Vec, FastText, and ElMo [47]. However, we employed pre-trained GloVe in this study due to its outstanding performance with BILSTM compared with other pre-trained embedding based on our related studies' analysis. In summary, the preprocessed tweets produce a two-dimensional vector space model (GloVe embedding). However, each word in a processed tweet is a representation of each row in this vector.

4) CLASSIFICATION

Deep learning consists of the learning of deep representation of data that helps in building an optimised solution from algorithm to solve conventional machine learning problem. Deep learning is a powerful learning algorithm that surpasses finding a word representation of data in any particular task. It can automatically extract novel features from the varying sets of features in the training data without human effort. In other words, it extracts more features in the absence of labels on the dataset [48]. In this study, a deep learning model approach that uses the Bidirectional LSTM model (a subset of RNN) was validated on the benchmark dataset to enhance the performance results on sarcasm analysis has been validated on the benchmark dataset to improve performance results of sarcasm analysis. The model selection motivation is that the model has obtained promising results in many NLP applications since it runs both forward and backward operations on the input clause information. As a result, it has a better understanding of the context in sarcastic expression. The semantic feature, also known as word embedding features in this study, has been proved important in any deep learning approach for NLP tasks.

In this study, GloVe, a word embedding scheme that automatically captures contextual features from the text, was utilised [49]. GloVe is a word embedding scheme that relies on the weighted least-square model and trains not only on the local context information of the word (usually used by word2vec) but also on the global word-to-word co-occurrence count in a corpus to obtain a word vector. This process is referred to as parallel implementation, and it facilitates the GloVe in modelling on a large dataset. Thus, it integrates the discriminative features obtained from two model families: 'Global matrix factorisation' and 'Local content window' to construct a new one [38], [50]. In this pre-trained word embedding approach, the preprocessed data will be employed to extract word vectors using word embedding (GloVe), and these features will be utilised as a feature for modelling. In this case, the text is usually represented in numerical form. The deep learning model comprises machine learning and an artificial neural network that characterises the core of the network as it contains multiple hidden layers. The deep learning model consists of neural networks with various layers that contain a wide range of parameters. The network layers are situated in one of the fundamental network architectures such as convolutional neural network, recurrent neural network, and recursive network. Convolutional neural network architecture consists of the input layers, convolutional layers, pooling layers, and output layers. The input data are fed through the input layer and then pass to the convolutional layer. In the convolutional layer, feature maps are extracted, bypassing the convolutional filter on input data. However, multiple filters are utilised to input data for multiple feature extraction. The final decision is made by the fully connected layer, connected to the output layer and the previous layer. A recurrent neural network is a standard network that uses an edge to feed into the next time slides instead of feeding into

Algorithm 1 :Algorithm Representation for Deep Learning Approach for Sarcasm Identification Training Process

Input: Training on processed data using Bi-LSTM with pre-trained GloVe

Output: Sarcasm prediction report

1. Data collection and preprocessing
   - Obtain benchmark dataset
   - Read the dataset
   - Preprocess the data
   - Apply tokenisation on data using the NLTK technique
   - Split data into train, test and validation set
2. Feature Extraction
   - Apply pre-trained word embedding
   - Load GloVe embedding
   - Create embedding matrix by assigning vocabulary with pre-trained word embedding
3. Set the parameter for the network
   - Set the value of the hidden unit
   - Minimum batch size
   - Dimension of GloVe vector
   - Max epoch value
   - Mini batch size
   - Regularisation value Optimise the Bi-LSTM network parameter
4. Add callback
6. Obtain the network output
7. Train the model
   - Train the Softmax layer using the supervised approach
   - Stack the Bi-LSTM and Softmax layer
8. Apply the fine-tuning strategy
9. Test the model using the pre-trained network and test data
10. Obtain the prediction
11. Output the prediction report

the next layer in a similar time slide. It contains a cycle that signifies the existence of short memory in the network. On the other hand, the recurrent neural network operates similarly to a hierarchical network that does not require time slides allocation to the input sequence but rather processes the input in a hierarchical tree structure. As stated above, this study will employ Bi-LSTM to construct a deep learning model for sarcasm identification.

a: LONG-SHORT TERM MEMORY (LSTM)

LSTM was created as an enhanced form of the standard recurrent neural network [51], [52] to modify its state to verify what to retain and what to discard. LSTM is created by increasing the memory capability of RNNs [53]. The core aim of creating LSTM is to address the exploding and vanishing gradient problem found in the standard RNN. During the training process, LSTM maintains the error to back-propagate using deeper layers in which learning continues over various steps. LSTM is created to learn long-distance dependencies within the sequential data. It keeps the contextual semantic information for dependencies in a long-range context using special memory cells. In each LSTM unit, which consists of the input, forget, and output gate is employed to coordinate and decide on the fraction of information to hold, discard, and move to the next step. It also decides when to issue read, write, and delete permission through gates that either pass or block information flow through the LSTM unit. LSTM architecture is depicted in FIGURE 2. To compute the input, forget, and output gate together with the input cell state, equations 1-6 below can be employed.

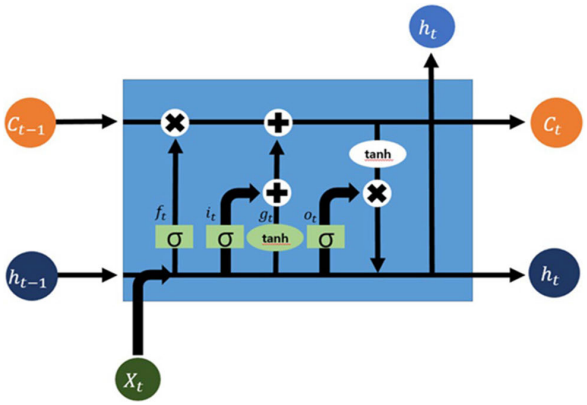

FIGURE 2. LSTM representation.

$$i_t = \sigma(W_{i_y} x_t + W_{i_z} h_{t-1} + b_i) \quad (1)$$

$$\boldsymbol{f_t} = \boldsymbol{\sigma}(\boldsymbol{W_{f_y} x_t} + \boldsymbol{W_{f_z} h_{t-1}} + \boldsymbol{b_f}) \quad (2)$$

$$\boldsymbol{o_t} = \boldsymbol{\sigma}(\boldsymbol{W_{o_y} x_t} + \boldsymbol{W_{o_z} h_{t-1}} + \boldsymbol{b_o}) \quad (3)$$

$$\boldsymbol{d_t} = (\boldsymbol{W_{d_y} x_t} + \boldsymbol{W_{d_z} h_{t-1}} + \boldsymbol{b_d}) \quad (4)$$

$$\boldsymbol{c_t} = \boldsymbol{f_t} \otimes \boldsymbol{c_{t-1}} + \boldsymbol{i_t} \otimes \boldsymbol{d_t} \quad (5)$$

$$\boldsymbol{h_t} = \boldsymbol{tanh}(\boldsymbol{C_t}) \otimes \boldsymbol{O_t} \quad (6)$$

where $\otimes$ represents element products; $b_i b_f b_o b_d$ represents bias vectors. tanh represents a hyperbolic tangent function,

σ = sigmoid function that represents gate activation function. $W_i W_f W_o W_d$ represents the weighing factors utilised for mapping input cell state and three gates with the input hidden layers.

$rh_t = [h_{t-n} \ldots\ldots\ldots\ldots h_{t-1}]$ epresents the final LSTM layer output (i.e., a vector of all output)

b: BI-DIRECTIONAL LSTM (BI-LSTM)

As indicated by [54], Bi-LSTM can capture compositional information in a sentence (for each input sentence). Bi-directional LSTM is made up of the forward operation network that reads the clause information in the forward direction between word 1 and n, and the backward operation network that reads the clause information in the backward direction. Thus, the generated hidden states from both directions (forward and backwards) are joined to form hidden

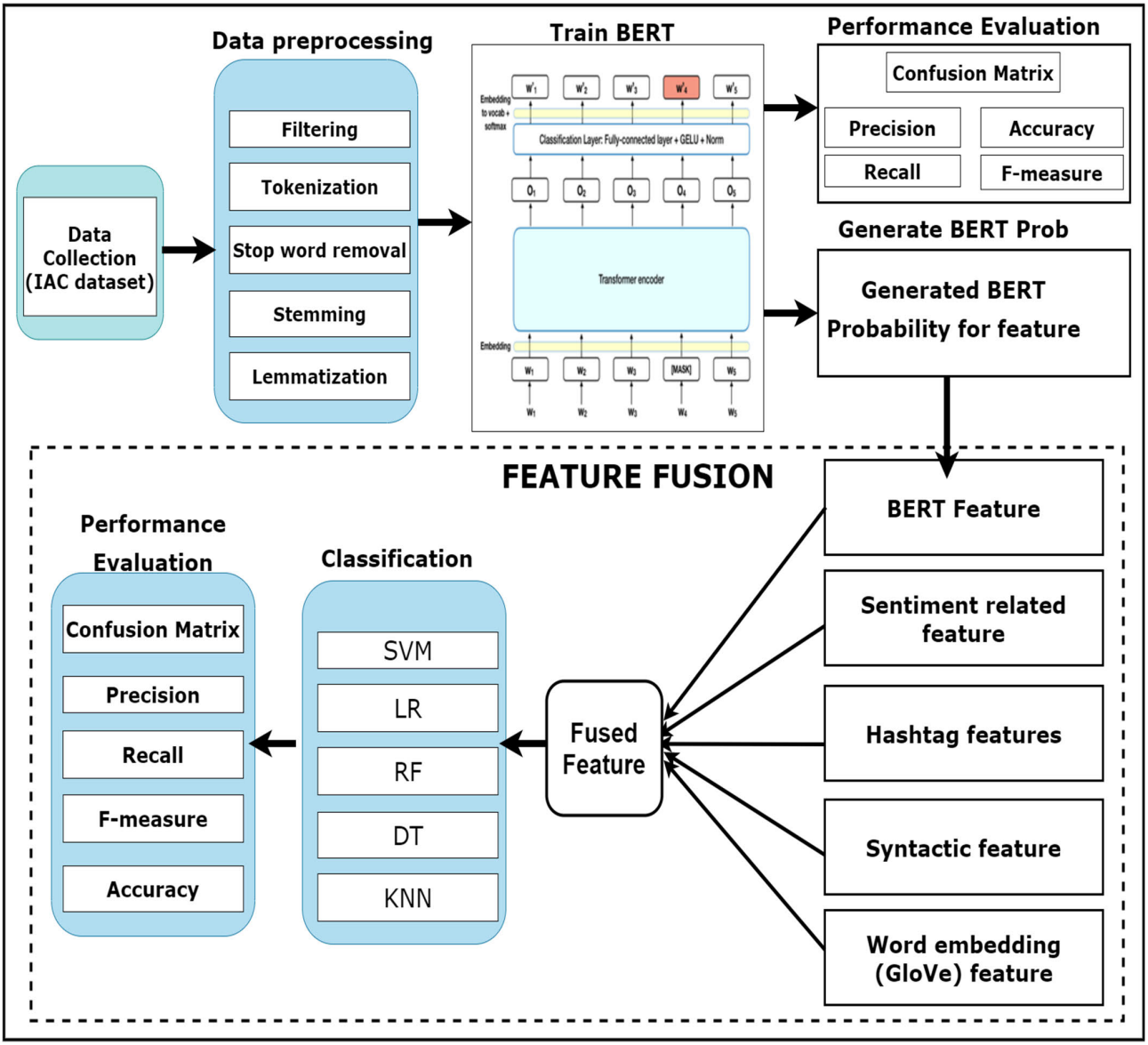

FIGURE 3. Systematic flow of BERT model and feature fusion approach.

states for Bi-LSTM. The output of the network generates both future and past contexts. Thus, each output vector element obtained by Bi-LSTM is computed by applying equation 7 [55]

$$y_t = \sigma(h^{\rightarrow}, h^{\leftrightarrow}t) \tag{7}$$

where σ is a function that outputs two sequences, the function can be used for summation, multiplication, average, and concatenation function. However, a vector representation can be employed to represent the final output of a Bi-LSTM layer, as shown in the equation below.

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

Thus, concatenating the Bi-directional layer and LSTM layer constructs Bi-LSTM, and the LSTM results will be automatically concatenated.

B. TRANSFORMER APPROACH AND FEATURE FUSION

This section provides a methodological description of the BERT approach and features a fusion technique for sarcasm identification. The framework for the transformer and feature fusion process is depicted in FIGURE 3. Each of the framework segments is described below.

1) DATASET

In order to provide a complete evaluation of the proposed approach, the internet argument corpus version (IAC-v2) dataset has been utilised. The IAC-v2 dataset was made publically available by [16]. It contains three sub-corpora, in which the highest one is referred to as ''generic,'' containing 3260 posts per class obtained from iacv2 prepared for the sarcasm detection dataset. Many scholars on sarcasm detec-

tion study tasks have utilised the dataset. The distribution of train, test, and validation sets are provided in table1.

Dataset	Train	Test	Dev	Total
IAC-v2 [16]	4546	1364	585	6495

2) DATA PREPROCESSING

In this section, a similar preprocessing technique used for the deep learning approach was employed in this section.

3) FEATURE EXTRACTION

In addition to the BERT feature, three other handcrafted features that comprised hashtag features (positive and negative hashtag) were proposed and extracted for feature fusion classification. These features are described below.

a: HASHTAG FEATURE

A hashtag is sometimes used to express some emotional content. The hashtag is used to disambiguate the actual meaning by the Twitter user to pass a message. For example, in a tweet, ''Thank you for always sending me money, #i hate you.'' In this expression, the hashtag ''#i hate you'' shows that the user is not really expressing thanks to the intended but tremendously hating him for not sending him money. We call the utterance mentioned above a negative hashtag token. Hashtag features can be represented as a positive or negative hashtag. In this study, three hashtag features are defined: a positive hashtag, a negative hashtag, and the positive and negative hashtag co-existence. The hashtag features are extracted by using a sentiment lexicon that consists of a list of negative hashtag words such as ''#hate, #pity, #waste, #discrimination, etc.,'' and a list of a positive hashtag such as ''#happy, #perfect, #great, #goodness, etc.'' However, using this lexicon, the number of positive hashtags and negative hashtags present in the tweet text is computed and added as a feature. The third feature is extracted by checking the co-occurrence of positive hashtags and negative hashtags in the same token. However, if there is co-occurrence, one is measured; otherwise, zero (0) is measured.

b: SENTIMENT RELATED FEATURE

Sentiment-related feature: the most common form of Sarcasm that occurs in social media is a whimper. In whimper, the composer of sarcastic utterance uses positive sentiment to describe a negative situation. In this regard, the expression of Sarcasm makes use of contradicting sentiment that can be observed in the expression of the negative situation using positive sentiment as found in the study conducted on sarcasm analysis by Riloff, *et al.* [14]. For example, 'I love being always cheated.' In this study, we investigated the contradiction between the word's sentiment and other tweets' components to recognise such sarcastic statements. To this end, sentiment-related features are extracted from each tweet and counted. A SentiStrength [56] lexicon was utilised to obtain the sentiment scores of the words. SentiStrength is a sentiment lexicon that utilises linguistic rules and information to detect English text sentiment. The lexicon usually provides the polarity sentiment (positive and negative) of words like question words, emotion words, booster words, negation words, idioms, slangs, and emoticons. The score uses an integer that ranges from -5 to $+5$, in which the larger absolute value represents the stronger sentiment. In addition, we also extracted more features that show the contrast between the sentiment components that include; positive words, highly emotional positive words, negative words, and highly emotional negative words. Finally, we defined two more features that check the contrast between different components (co-occurrence between negative and positive components) within the same tweet. Therefore, the sentiment-related feature vector contains 8 features.

c: SYNTACTIC FEATURE

The syntactic feature plays a significant role in offering information about the syntactic structure of tweets. This research defines Parts of speech feature, laughing expression, and interjection word as syntactic features extracted from the tweet's content. However, the NLTK tokeniser library was employed to perform tokenisation on the processed tweets. Firstly, we extracted the POS feature using the parts of the speech library, and the count of its presence in the sarcastic text is taken. This study only concentrated on POS with some sentimental contents such as nouns, adverbs, and adjectives. Furthermore, POS tags are mapped with each corresponding POS group, and only the tokenised words that correspond with the chosen three parts of speech groups as aforementioned were preserved in the text. This research utilised a similar approach used in [57] and extracted adverbs, adjectives, and nouns. Secondly, to extract the second feature, we identified laughter words used to express pleasures or joy. Thus, we added laughing features: the sum of internet laughs, represented with lol, hahaha, hehe, rofl, and imao. The feature is extracted by creating a list containing the most common laughing words, and it was employed to find the frequency of such words. Then, the frequency of such words present in the text was computed and added as a feature. The third feature is extracted by identifying interjection words such as woo, oh, wow, etc. in the tweets and the frequency of interjection words is computed and added as a feature.

4) CLASSIFICATION

BERT is a multi-layer bidirectional transformer encoder trained on BooksCorpus [58] and English Wikipedia that contain 800M and 2,500M tokens, respectively, which can learn deep bi-directional representations and, in the future, can be fine-tuned to perform various tasks like NER. However, Data is tokenised before pre-training using WordPiece embeddings. In the pre-training phase, two unsupervised methods are used, Masked Language Model (MLM) and Next Sentence Prediction (NSP). In MLM, also known as masked language prediction task, 15% of the word in the

input sequence are marked out; thus, the whole sequence is provided to a deep bidirectional transformer [59] encoder, and the masked words are predicted by model learning. In sentence prediction, on the other hand, BERT takes input sentences A and B for learning the relationship between sentences. Empirically, its established bidirectional nature allows the model to understand the features from data efficiently. Unlike the traditional sequential or recurrent models, the entire input sequence is being preprocessed by attention architecture once, enabling the parallel processing of input tokens. FIGURE 3 provides the visualisation of BERT architectural layers. The fine-tuning of the pre-trained BERT model by adding an extra layer could produce state-of-the-art results in various natural language processing tasks [60]. BERT model is comprised of two separate models called the BERTbase model and BERTlarge model. On the one hand, the BERTbase model consists of an encoder with 12 layers (also known as transformer blocks), 110 million parameters, and 12 self-attention heads. On the other hand, the BERT-large model is comprised of 340 million parameters, 16 attention heads, and 12 layers. In the BERTbase model, the number of hidden dimensions extracted from embedding is 768 [41].

a: INPUT AND OUTPUT REPRESENTATION

BERT accepts the utmost length of 512 sequences of tokens as input and represents the token sequences of a 768-dimensional vector as an output. In BERT, the maximum of two-segment insertion can be made in each input sequence, including [SEP] and [CLS]. Special classification token ([CLS]) embedding is usually the initial input sequence token. It holds a special classification embedding chosen as the first token in the last hidden layer to represent the full sequence in a sarcasm classification task. However, the last hidden state that correlates to this token is employed as the aggregate sequence to represent sarcasm classification tasks. In addition, pairs of a sentence are crowded together in one sequence. These sentences can be differentiated into two methods. Firstly, a special token ([SEP]) embedding is employed to separate the sentences. In our classification task, we shall employ only [CLS] embedding input sequence tokens. Secondly, a learner embedding is added in every token with an indication that shows the sentence that each sentence belongs to (either A or B).

In our sarcasm detection task, we use social media data, which requires first to carry out an essential step of analysing the contextual information obtained from the pre-trained BERT layer and fine-tune the model using annotated dataset. This is carried out because the BERT model is pre-trained on general corpora. During fine-tuning, the weight is updated by utilising a labelled dataset that is new to the previously trained model.

5) FEATURE FUSION APPROACH

Word embedding features is not enough in capturing all the sarcasm clue in sarcastic utterances, owing to the drawback in word embedding feature. One of the major limitations of word embedding is that it ignores the sentiment polarity of words [61], [62]. Though word embedding based word vector captures the word's context, words having opposite polarity are mapped into close vectors. For example, the two different words "like" and "unlike" can occur in the same context as illustrated in sentences below:

"I like that footballer" and "I dislike that footballer." Thus, the word embedding (word vector) feature lacks enough sentiment information in performing sarcasm classification, and it does not precisely capture the overall sentiment of the sarcastic expression.

Thus, a feature fusion approach is proposed that comprised of BERT feature, hashtag feature, sentiment related feature, syntactic features, and GloVe embedding feature for sarcasm classification to address the sentiment polarity problem in sarcasm utterances. To the best of our knowledge, this is the first study that integrates the BERT feature and Word embedding with linguistic and sentiment-related features to improve the classification performance for sarcasm identification to address the context and sentiment polarity issue in sarcastic utterances.

IV. EXPERIMENTAL DESIGNS

In this section, various empirical analysis is performed to implement the proposed contextual feature-based technique for sarcasm identification using a deep learning model. Anaconda framework was utilised to implement the proposed approach for the sarcasm detection model. The system configuration is window ten (10) pro (64-bit operating system), running on Intel core i7 processor with 12 GB RAM. A detailed explanation of the parameter settings and their impact on the model performance and how the fine-tuning of the parameter was carried out to enhance the proposed deep learning technique's performance are provided in the subsequent sections.

A. EXPERIMENTAL SETTINGS

1) DEEP LEARNING APPROACH

The first step before experimenting is data preparation. The detail of data preparation is described in section 3. In this study, pre-trained word embedding (GloVe) was adopted for the deep learning model [49]. Here, the word embedding serves as an input layer. The study employed Twitter GloVe word embedding obtained from 2B tweets and 27B tokens, containing 1,2M vocabulary with 200d vectors with the word data level. GloVe word embedding scheme with word feature vector dimension k is fixed at 200. However, some fine-tuning of the embedding was performed during the training process, and a dropout of 0.2 rates was utilised to prevent the over-fitting problem in the model. The hyper-parameter dimension d for Bi-LSTM hidden units' layers is fixed at 100 for the forward direction and 100 for the backward direction (total of 200). Conversely, 50 epoch maximum was trained for the model. The minimum training batch size for both datasets is tuned to 128 mini-batch sizes.

TABLE 2. Hyperparameter settings.

Hyper-parameter	Value
Dimension of GloVe vector	200
Hidden units of LSTM (Forward and backwards)	100
Minimum batch size	128
Regularisation	Dropout operation
Drop-out Rate	0.2
Learning rate	0.2
Word embedding	GloVe
Activation function	Softmax
Train max epoch	50
Optimisation algorithm	Adam algorithm

The initialisation of all weight matrices was performed by sampling in a similar direction and setting biases to zero. Adam algorithm optimisation [63] was adopted for model parameter optimisation with the learning rate set to 0.001 rates. Tensorflow and Keras were used for the implementation of the architecture. It is crucial to select the optimal parameter to obtain the best predictive results. Table 2 depicts the summary of the Hyper-parameter setting for this study.

2) TRANSFORMER (BERT) APPROACH

In this setting, the dataset was preprocessed by filtering some unwanted items that could reduce the classification performance, such as user mention, URL links, hashtags, foreign language characters, stop words removal, and non-English ASCII character. The preprocessed training set is randomly split into two training (70%) and validation (30%). The experiment was carried out on a system running on window 10 with 64-bit operating systems. The system uses an Intel Core^{TM} i7-4770 CPU @ 3.400GHz with 16GB Random Access Memory (RAM) capacity.

The proposed model uses $\text{BERT}_{\text{Base}}$ for encoding the input preprocessing text and produces a 768-dimensional hidden state of classification token [CLS]. The study utilised Adam optimiser [64] for optimisation, using the learning rate of 0.001. The model is trained for 35 epochs with a batch size of 16. Moreover, the maximum sequence length is set to 128, the reprocess input data and overwrite output directory is set to true.

B. EVALUATION MEASURE

Evaluation measures are the performance indicators that have been established to measure the output of the classification algorithms. In the evaluation phase, the constructed model predicts the class of unlabelled text (sarcastic or non-sarcastic) using the training data sets. The predictive performance of the constructed model can be evaluated by employing the following parameters:

Accuracy (ACC) provides the percentage ratio of the predicted instances. It measures the overall correctly classified instances. It is computed by dividing the overall number of true instances (that consists of true positive and true negative) by all the instances. **Precision (PRE)** provides model accuracy in the existence of false positive instances. Thus, the model accuracy provides the overall occurrence of false positive instances with the rejection of positive instances. Precision is computed by finding the ratio of true positive over a positive result. **Recall (REC)** is used to measure accuracy, which shows the model performance in the existence of a false negative instance. It is the proportion of actual positives, which are predicted positive. Thus, the false negative shows the wrongly predicted instance on the data. It mathematically denotes the true positive ratio against all the true results. **F-measure (F-M)** is a cumulative factor to test the overall effect of the recall and precision in order to find the overall impact of false negative instances and false positive instance over the whole accuracy. It denotes the harmonic mean of precision and recall in the presence of critical equality of false positive and false negative. The standard F-measure is F1-score, which provides equal importance of recall and precision. **The true positive (TP)**result is noticed when the predicted tweet is found to be sarcastic, and the result of the classification shows exactly sarcastic after the experimental evaluation. **True negative (TN)** The true negative result is obtained when the predicted tweet is not sarcastic, and the classification result also validates it as not sarcastic. **False positive (FP)** occurs in a situation where a true negative result is obtained when the predicted tweet is not sarcastic, but the classification result indicates that the tweet is sarcastic. **False negative (FN)** occurs when the true positive result is obtained when the predicted tweet is sarcastic, but the classification result shows that tweet is not sarcastic. **Sensitivity (SEN)** determines the model's appropriateness in detecting the positive class outcome (detection probability). **Specificity (SP)** determines the exactness of positive class assignment (True negative rate) **Confusion matrix:** Also known as the error matrix, is a unique table representation that gives the picture of the classifier's execution, especially the supervised learning classification. The confusion matrix consists of two instances (''predicted'' and ''actual'') of the same sets of classes. Basically, the negative is discarded, whereas the positive is identified. Thus, after the classification, true positive is the instance that is accurately classified, whereas false positive is not correctly classified. False positive instances symbolise type 1 error, indicating that the number of instances is not correctly indicated as positive. On the other hand, true negatives are those instances that are correctly discarded, and false negatives denote the incorrectly classified instances. False negative symbolises type 2 error, indicating that the number of instances is incorrectly classified as negative. The pictorial diagram of the confusion matrix is depicted in Table 3

$$Acc = \frac{TP + TN}{TP + TN + FP + FN} \tag{9}$$

$$PRE = \frac{TP}{TP + FP} \tag{10}$$

$$REC = \frac{TP}{TP+FN} \tag{11}$$

TABLE 3. Confusion matrix.

	True condition	
Predicted condition	TP	FP (type 1 error)
	FN (type 2 error)	TN

TABLE 4. Performance results of the deep learning model.

Performance measure (%)	Riloff dataset	Ghosh dataset
Precision	98.00	98.50
Recall	99.50	98.50
F-measure	99.00	98.00
Accuracy	99.00	98.21
Sensitivity	95.56	97.90
Specificity	100	98.63

$$F - M = 2 * (\frac{PRE * REC}{PRE + REC}) \quad (12)$$

$$SP = \frac{TN}{TN + FP} \quad (13)$$

$$SEN = \frac{TP}{TP + FN} \quad (14)$$

Where TP represents a true positive number, TN denotes a true negative number, FN represents a false negative number, and FP represents a false positive number.

V. EXPERIMENTAL RESULTS AND DISCUSSIONS

This section discusses the results obtained in the experimental analysis of the proposed technique. The experimental results on the three benchmark datasets are reported in Table 4 and Table 7, whereas the performance comparison of deep learning based on two Twitter datasets is represented in FIGURE 6. Also, both datasets' confusion matrices are represented in FIGURE 4 and FIGURE 5. Precision has been utilised as the key performance metric. However, other standard evaluation metrics that include accuracy, F-measure, recall, sensitivity, and specificity were also employed as supplementals to evaluate the proposed model with the baseline approach. The performance evaluation has been established with three benchmark datasets for sarcasm classification to evaluate the proposed technique's effectiveness. The evaluation of the proposed approach is done based on the baseline study and dataset.

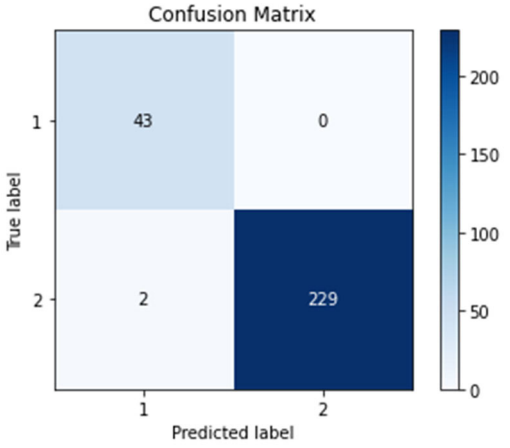

FIGURE 4. Confusion matrix for Riloff dataset.

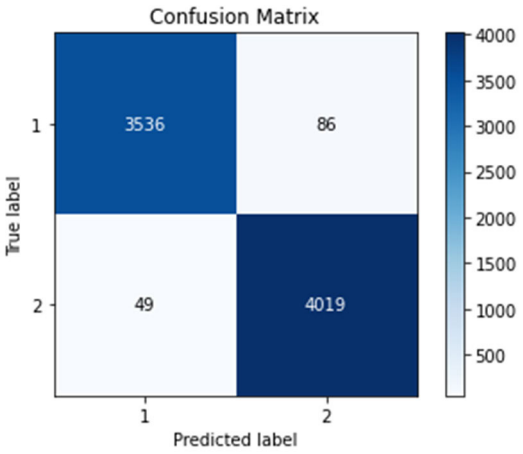

FIGURE 5. Confusion matrix for ghosh and vale dataset.

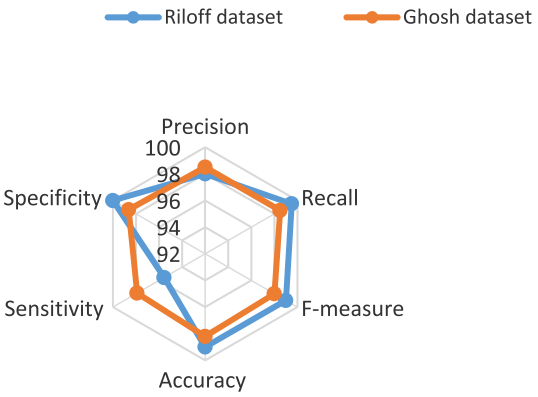

FIGURE 6. Performance results of the deep model on two datasets.

A. DEEP MODEL PERFORMANCE RESULTS ON RILOFF DATASET

In Table 4, the predictive results obtained on the Rillof data are presented. It can be observed from the table that

TABLE 5. Comparison results of Riloff dataset with baseline.

Baseline	PRE (%)	REC (%)	F-M (%)	ACC (%)
Riloff, et al. [14]	65.00	40.80	50.1	59.40
Tay, et al. [54]	63.94	63.45	62.52	62.69
Ghosh, et al. [65]	69.76	66.62	67.81	78.72
Zhang, et al. [66]	*96.77*	56.21	55.96	64.32
Our proposed approach	**98.00**	99.50	99.00	99.00

the predictive performance attained a precision of 98%. We can also observe that the model achieved almost perfect prediction with other performance measures such as recall, f-measure, and accuracy. The result also shows a 100% prediction with specificity. However, the sensitivity measure recorded the least result with a predictive result of 95.56%. Our deep learning approach on the Riloff dataset was also evaluated with four baseline studies on sarcasm analysis. Table 5 and FIGURE 7 present the comparison results. It can be observed from Table 5 that our model improved the baseline performance substantially. Our proposed approach's highest result is represented in bold, whereas the highest performance on the baseline approaches is indicated in italic. The results show that our approach outperformed the baseline in terms of precision, recall, f-measure, and accuracy by attaining a precision of 98%. Our proposed technique on this dataset nearly attains the perfect model performance with an accuracy of 99%, recall of 99.5%, precision of 98%, and f-measure of 99% of our model. It also shows that the Zhang approach outperformed other baseline approaches in terms of precision only, but Ghosh and Tay's approach outperformed in terms of accuracy, recall, and F-measure. It should also be noted that the Riloff approach performed least in terms of precision, f-measure, and recall when compared to other baseline approaches but outperformed the Tay approach in terms of precision. Thus, the proposed context-based feature technique using the deep learning approach attained average detection precision between 1.23% to 33% compared to the existing method using the Riloff dataset.

B. DEEP MODEL PERFORMANCE RESULTS ON GHOSH AND VALE DATASET

Table 4 also depicts the predictive results obtained on the Ghost dataset. It can be observed from the table that the predictive performance attained a precision of 98.5%. We can also observe that the model achieved high-performance results with other performance measures such as recall, f-measure, and accuracy. The result also shows specificity attained the prediction of 98.63% when compared with other performance measures. However, the sensitivity measure

TABLE 6. Comparison results of ghosh and vale dataset with baselines.

Baseline	PRE (%)	REC (%)	F-M (%)	ACC (%)
Ghosh, et al. [65]	73.30	71.70	72.50	-
Xiong, et al. [67]	*76.39*	72.56	74.42	80.09
Our proposed approach	**98.50**	98.00	98.50	98.21

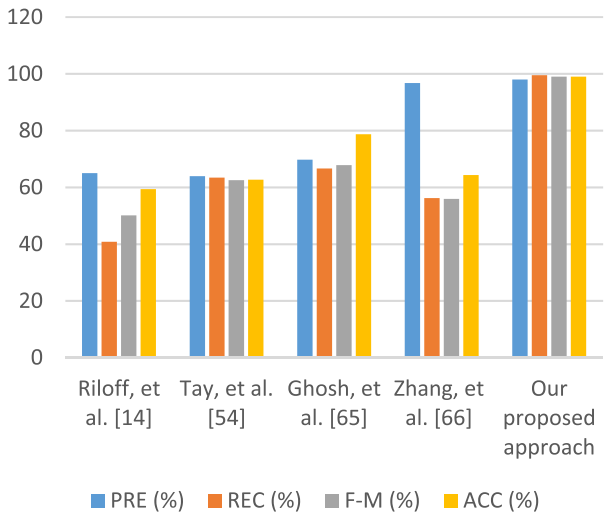

FIGURE 7. Performance evaluation of the deep learning model compared to the baseline on Riloff dataset.

recorded the least result with a prediction of 97.9%. Thence, to evaluate the proposed technique on this dataset, two baseline studies have been utilised. The results of the evaluation are depicted in Table 6 and FIGURE 8. Our proposed approach's highest result is indicated in bold, whereas the highest result obtained from the baseline is in italic. It can be observed that our proposed deep learning approach outperformed all baseline approaches by attaining accuracy of 98.21%, precision of 98.5%, recall of 98%, and f-measure of 98%. Referring to the baseline studies, it can be noticed that Ghosh's study has the least performance in terms of precision, recall, and F-measure. However, the author did not determine the accuracy performance in their study. The evaluation of our proposed approach with the baseline shows that our proposed deep learning approach outperformed the Xiong approach in terms of accuracy with 18.12%, precision with 22.11%, recall with 14.44%, and f-measure with 24.08%. Also, it supersedes the Ghosh and vale approach in precision with 25.2%, recall with 14.44%, and f-measure with 24.08%. Thus, the proposed context-based technique using deep learning attained average detection precision between 22.11% to 25.2% compared to the existing method using the Ghosh dataset. The result shows that models involving Bi-LSTM can learn the contextual information from the

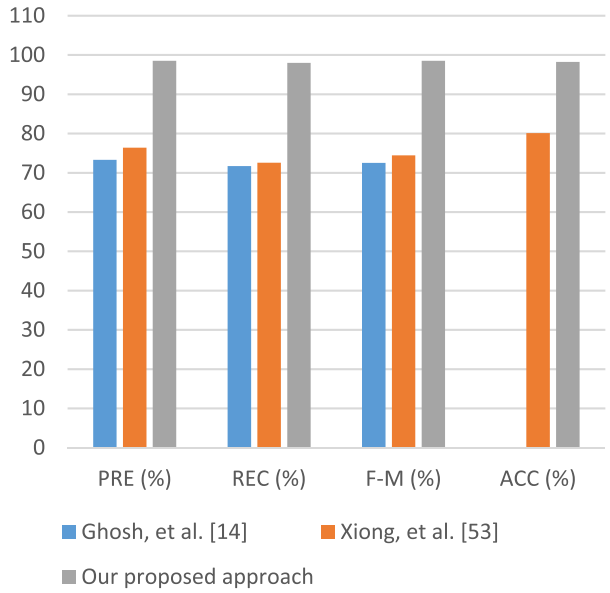

FIGURE 8. Performance evaluation of the deep learning model compared to the baseline on ghosh dataset.

TABLE 7. Comparative results of the deep learning model, Transformer (BERT) model, and feature fusion on IAC-v2 dataset.

Models	PRE (%)	REC (%)	F-M (%)	ACC (%)
BLSTM	68.4	68.9	68.5	62.1
BERT	80.0	77.0	79.2	75.5
Proposed Feature Fusion	81.2	78.5	79.8	76.3

sarcastic expression and uses the information to a large magnitude by enhancing the model performance.

C. BERT AND FEATURE FUSION PERFORMANCE RESULTS ON IAC-v2 DATASET

In Table 7, the predictive results obtained on the IAC-v2 data is presented. It can be observed from the table that the comparative results of the deep learning model (Bi-LSTM), BERT model, and feature fusion are presented. The deep learning model attained the highest precision of 68.4%, transformer approach 80%, and feature fusion 81.2%. Therefore the proposed feature fusion slightly outperformed the BERT approach. We can also observe that the feature fusion approach achieved outperformed other performance measures such as recall, f-measure, and accuracy. The feature fusion approach with BERT on the IAC-v2 dataset was also evaluated with three baseline studies on sarcasm analysis to show the significance of the proposed approach. Table 8 and FIGURE 9 present the comparison results. It can be observed from Table 5 that the proposed feature fusion with the BERT approach improved the baseline performance substantially. The proposed approach's highest result is represented in bold, whereas the highest performance on the baseline approaches

TABLE 8. Comparison results of the proposed feature fusion with the baselines.

Baseline	PRE (%)	REC (%)	F-M (%)	ACC (%)
Oraby, et al. [16]	71.0	*77.0*	*74.0*	-
Poria, et al. [69]	*77.5*	68.2	72.6	-
Gangi, et al. [68]	73.0	70.0	72.0	-
Proposed Feature Fusion	**81.2**	**78.5**	**79.8**	**76.3**

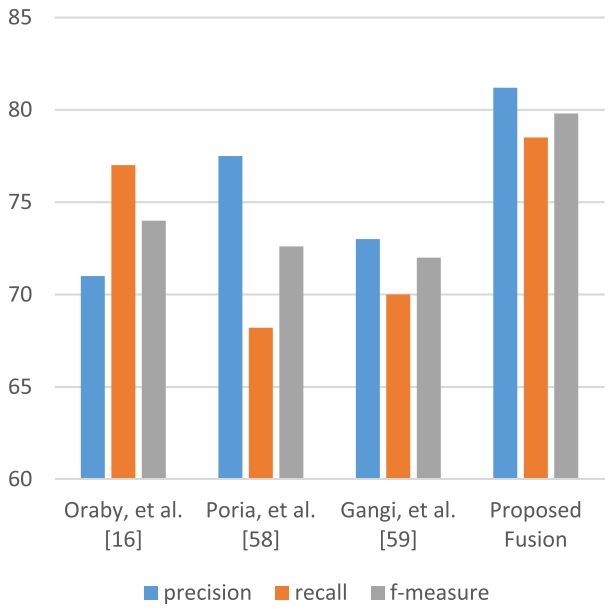

FIGURE 9. The comparative performance of the feature fusion with BERT and baselines.

is indicated in italic. The results show that our approach outperformed the three baselines in terms of precision, recall, and f-measure by attaining a precision of 81.2%. The proposed feature fusion on this dataset outperformed baselines in terms of recall and f-measure by attaining a recall of 78.5% and an f-measure of 79.8%. It also shows that out of the three baselines evaluated on the proposed approach, the Poria, *et al.* [68] approach outperformed other baselines in terms of precision only by attaining a precision of 77.5%. However, Oraby, *et al.* [16] approaches outperformed the Poria, *et al.* [69] and Gangi, *et al.* [68] in terms of recall and f-measure. It should also be noted that the Gangi, *et al.* [68] approach performed least in terms of f-measure and recall when compared to other baseline approaches but outperformed the Poria approach in terms of precision. Thus, the proposed context-based feature technique using the feature fusion approach with BERT attained average detection precision between 3.7% to 10.2% when compared to the existing method using the IAC-v2 dataset.

Based on the two datasets' overall results, it can be observed that the proposed approach can address the sarcasm identification on the tweet domain. It indicates the efficiency of the deep neural model that uses a Bidirectional long short term memory network. The result shows that models involving Bi-LSTM can learn the contextual information from the sarcastic expression and uses the information to a large magnitude by enhancing the model performance. Also, BERT features can capture some contextual information and word sense in sarcasm expression, and feature fusion with BERT features can address word embedding-based features commonly found in deep learning approaches.

VI. CONCLUSION

Sarcasm identification has been a crucial challenge in natural language processing. The sarcasm identification task is a classification problem aimed at distinguishing sarcastic utterances from the non-sarcastic counterparts. Accurate identification of sarcasm can enhance the sentiment analysis and opinion mining study. This study has focused on the context-based feature technique for sarcasm identification using deep learning, transformer learning, and conventional machine learning models. Two Twitter and Internet Argument Corpus, version two (IAC-v2) benchmark datasets were utilised for classification using the three learning models. The first model uses embedding-based representation via deep learning model with bidirectional long short term memory (Bi-LSTM), a variant of recurrent neural network (RNN), by applying Global vector representation (GloVe) for the construction of word embedding and context learning. The second model is based on Transformer using a pre-trained Bidirectional Encoder representation and Transformer (BERT). In contrast, the third model is based on feature fusion that comprised BERT feature, sentiment related, syntactic, and GloVe embedding feature with conventional machine learning. The effectiveness of this technique is tested with various evaluation experiments. However, the technique's evaluation on the two Twitter benchmark datasets attained 98.5% and 98.0% highest precision, and the IAC-v2 dataset attained the highest precision of 81.2%, respectively. When evaluated with the baselines, the obtained results on the three benchmark datasets outperformed all the baselines approaches, which shows the significance of the proposed technique for sarcasm analysis. Though this technique has been experimented with on the benchmark dataset, it can also be employed on the randomly generated live tweet dataset to test the model's predictive performance. Besides, other deep learning models have produced promising results as reported in several NLP tasks, especially the augmented and attention-based neural networks. The architecture can be considered in further study. Accordingly, the advancement text-image feature engineering approach, in which text is represented as an image, also shows the effectiveness of social media utterances is another promising research direction for sarcasm classification.

REFERENCES

[1] A. Pak and P. Paroubek, "Twitter as a corpus for sentiment analysis and opinion mining," in *Proc. LREc*, vol. 10, 2010, pp. 1320–1326.
[2] G. Wang, J. Sun, J. Ma, K. Xu, and J. Gu, "Sentiment classification: The contribution of ensemble learning," *Decis. Support Syst.*, vol. 57, pp. 77–93, Jan. 2014.
[3] V. Vyas and V. Uma, "Approaches to sentiment analysis on product reviews," in *Sentiment Analysis and Knowledge Discovery in Contemporary Business*. Hershey, PA, USA: IGI Global, 2019, pp. 15–30.
[4] E. Fersini, E. Messina, and F. A. Pozzi, "Sentiment analysis: Bayesian ensemble learning," *Decis. Support Syst.*, vol. 68, pp. 26–38, Dec. 2014.
[5] E. Cambria, "Affective computing and sentiment analysis," *IEEE Intell. Syst.*, vol. 31, no. 2, pp. 102–107, Mar. 2016.
[6] M. A. Walker, J. E. F. Tree, P. Anand, R. Abbott, and J. King, "A corpus for research on deliberation and debate," in *Proc. LREC*, Istanbul, Turkey vol. 12, 2012, pp. 812–817.
[7] D. G. Maynard and M. A. Greenwood, "Who cares about sarcastic tweets? Investigating the impact of sarcasm on sentiment analysis," in *Proc. LREC*, 2014, pp. 1–7.
[8] A. Joshi, P. Bhattacharyya, and J. M. Carman, "Automatic sarcasm detection: A survey," *ACM Comput. Surv.*, vol. 50, no. 5, p. 73, 2017.
[9] N. Parde and R. Nielsen, "Detecting sarcasm is extremely easy," in *Proc. Workshop Comput. Semantics Beyond Events Roles*, 2018, pp. 21–26.
[10] Y. Karuna and G. R. Reddy, "Broadband subspace decomposition of convoluted speech data using polynomial EVD algorithms," *Multimedia Tools Appl.*, vol. 79, no. 7, pp. 5281–5299, 2018.
[11] S. Mukherjee and P. K. Bala, "Sarcasm detection in microblogs using Naïve Bayes and fuzzy clustering," *Technol. Soc.*, vol. 48, pp. 19–27, Feb. 2017.
[12] A. Rajadesingan, R. Zafarani, and H. Liu, "Sarcasm detection on Twitter," presented at the 8th ACM Int. Conf. Web Search Data Mining, 2015.
[13] M. Bouazizi and T. O. Ohtsuki, "A pattern-based approach for sarcasm detection on Twitter," *IEEE Access*, vol. 4, pp. 5477–5488, 2016.
[14] 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. Conf. Empirical Methods Natural Lang. Process.*, 2013, pp. 704–714.
[15] A. Ghosh and D. T. Veale, "Fracking sarcasm using neural network," in *Proc. 7th Workshop Comput. Approaches Subjectivity, Sentiment Social Media Anal.*, 2016, pp. 161–169.
[16] S. Oraby, V. Harrison, L. Reed, E. Hernandez, E. Riloff, and M. Walker, "Creating and characterizing a diverse corpus of sarcasm in dialogue," Tech. Rep., 2017.
[17] B. G. Hb, M. A. Kumar, and K. Soman, "Distributional semantic representation in health care text classification," in *Proc. FIRE (Work. Notes)*, 2016, pp. 201–204.
[18] C. I. Eke, A. A. Norman, L. Shuib, and H. F. Nweke, "Sarcasm identification in textual data: Systematic review, research challenges and open directions," *Artif. Intell. Rev.*, vol. 53, pp. 4215–4258, Nov. 2019.
[19] A. Joshi, S. Agrawal, P. Bhattacharyya, and M. J. Carman, "Expect the unexpected: Harnessing sentence completion for sarcasm detection," in *Proc. Int. Conf. Pacific Assoc. Comput. Linguistics*. Singapore: Springer, 2017, pp. 275–287.
[20] Y. Wu, M. Wang, and P. Jin, "Disambiguating sentiment ambiguous adjectives," in *Proc. Int. Conf. Natural Lang. Process. Knowl. Eng.*, Oct. 2008, pp. 1191–1199.
[21] Y. Xia, E. Cambria, A. Hussain, and H. Zhao, "Word polarity disambiguation using Bayesian model and opinion-level features," *Cognit. Comput.*, vol. 7, no. 3, pp. 369–380, Jun. 2015.
[22] Y. Wang, M. Wang, and H. Fujita, "Word sense disambiguation: A comprehensive knowledge exploitation framework," *Knowl.-Based Syst.*, vol. 190, Feb. 2020, Art. no. 105030.
[23] K. Abdalgader and A. Al Shibli, "Context expansion approach for graph-based word sense disambiguation," *Expert Syst. Appl.*, vol. 168, Apr. 2021, Art. no. 114313.
[24] E. Fersini, F. A. Pozzi, and E. Messina, "Detecting irony and sarcasm in microblogs: The role of expressive signals and ensemble classifiers," in *Proc. IEEE Int. Conf. Data Sci. Adv. Analytics (DSAA)*, Oct. 2015, pp. 1–8.
[25] A. Onan and M. A. Tocoglu, "Satire identification in Turkish news articles based on ensemble of classifiers," *Turkish J. Electr. Eng. Comput. Sci.*, vol. 28, no. 2, pp. 1086–1106, Mar. 2020.

[26] T. Ptácek, I. Habernal, and J. Hong, "Sarcasm detection on Czech and English Twitter," in *Proc. 25th Int. Conf. Comput. Linguistics, Tech. Papers*, 2014, pp. 213–223.
[27] A. Esuli, A. Moreo, F. Sebastiani, and E. Cambria, "Cross-lingual sentiment quantification," *IEEE Intell. Syst.*, vol. 35, no. 3, pp. 106–114, May 2020.
[28] Q. Yang, Y. Rao, H. Xie, J. Wang, F. L. Wang, W. H. Chan, and E. Cambria, "Segment-level joint topic-sentiment model for online review analysis," *IEEE Intell. Syst.*, vol. 34, no. 1, pp. 43–50, Jan. 2019.
[29] A. Agrawal, A. An, and M. Papagelis, "Leveraging transitions of emotions for sarcasm detection," in *Proc. 43rd Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Jul. 2020, pp. 1505–1508.
[30] A. J. C. Onan, "Sentiment analysis on product reviews based on weighted word embeddings and deep neural networks," *Concurrency Comput., Pract. Exper.* p. e5909, 2020.
[31] I. Augenstein and A. Søgaard, "Multi-task learning of keyphrase boundary classification," 2017, *arXiv:1704.00514*. [Online]. Available: http://arxiv.org/abs/1704.00514
[32] M. Lan, J. Wang, Y. Wu, Z.-Y. Niu, and H. Wang, "Multi-task attention-based neural networks for implicit discourse relationship representation and identification," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2017, pp. 1299–1308.
[33] N. Majumder, S. Poria, H. Peng, N. Chhaya, E. Cambria, A. Gelbukh, and E. Cambria, "Sentiment and sarcasm classification with multitask learning," *IEEE Intell. Syst.*, vol. 34, no. 3, pp. 38–43, May 2019.
[34] A. Mishra, K. Dey, and P. Bhattacharyya, "Learning cognitive features from gaze data for sentiment and sarcasm classification using convolutional neural network," in *Proc. 55th Annu. Meeting Assoc. Comput. Linguistics*, vol. 1, 2017, pp. 377–387.
[35] A. Hussain and E. Cambria, "Semi-supervised learning for big social data analysis," *Neurocomputing*, vol. 275, pp. 1662–1673, Jan. 2018.
[36] J. Duan, B. Luo, and J. Zeng, "Semi-supervised learning with generative model for sentiment classification of stock messages," *Expert Syst. Appl.*, vol. 158, Nov. 2020, Art. no. 113540.
[37] P. Mehndiratta, S. Sachdevai, and D. Soni, "Detection of sarcasm in text data using deep convolutional neural networks," *Scalable Comput., Pract. Exper.*, vol. 18, no. 3, pp. 219–228, Sep. 2017.
[38] R. Schifanella, P. de Juan, J. Tetreault, and L. Cao, "Detecting sarcasm in multimodal social platforms," in *Proc. 24th ACM Int. Conf. Multimedia*, Oct. 2016, pp. 1136–1145.
[39] A. Onan, "Topic-enriched word embeddings for sarcasm identification," in *Proc. Comput. Sci. Line Conf.* Cham, Switzerland: Springer, 2019, pp. 293–304.
[40] S. Castro, D. Hazarika, V. Pérez-Rosas, R. Zimmermann, R. Mihalcea, and S. Poria, "Towards multimodal sarcasm detection (an_obviously_ perfect paper)," 2019, *arXiv:1906.01815*. [Online]. Available: http://arxiv.org/abs/1906.01815
[41] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," 2018, *arXiv:1810.04805*. [Online]. Available: http://arxiv.org/abs/1810.04805
[42] C. Carr and Z. Zukowski, "Curating generative raw audio music with DOME," in *Proc. Joint ACM IUI Workshops*, 2019.
[43] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.
[44] A. Onan and M. A. Tocoglu, "A term weighted neural language model and stacked bidirectional LSTM based framework for sarcasm identification," *IEEE Access*, vol. 9, pp. 7701–7722, 2021.
[45] A. Onan, "Sarcasm identification on Twitter: A machine learning approach," in *Proc. Comput. Sci. Line Conf.* Cham, Switzerland: Springer, 2017, pp. 374–383.
[46] R. Gonzalez-Ibanez, S. Muresan, and N. Wacholder, "Identifying sarcasm in Twitter: A closer look," in *Proc. 49th Annu. Meeting Assoc. Comput. Linguistics, Hum. Lang. Technol.*, vol. 2. Stroudsburg, PA, USA: Association Computational Linguistics, 2011, pp. 581–586.
[47] Z. H. Kilimci and S. Akyokus, "The evaluation of word embedding models and deep learning algorithms for Turkish text classification," in *Proc. 4th Int. Conf. Comput. Sci. Eng. (UBMK)*, Sep. 2019, pp. 548–553.
[48] T. Young, D. Hazarika, S. Poria, and E. Cambria, "Recent trends in deep learning based natural language processing [review article]," *IEEE Comput. Intell. Mag.*, vol. 13, no. 3, pp. 55–75, Aug. 2018.
[49] 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.
[50] C. I. Eke, A. Norman, L. Shuib, F. B. Fatokun, and I. Omame, "The significance of global vectors representation in sarcasm analysis," in *Proc. Int. Conf. Math., Comput. Eng. Comput. Sci. (ICMCECS)*, Mar. 2020, pp. 1–7.
[51] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional LSTM and other neural network architectures," *Neural Netw.*, vol. 18, nos. 5–6, pp. 602–610, Jul. 2005.
[52] Y. Agiomyrgiannakis, N. Egberts, F. Henderson, H. Zen, and P. Szczepaniak, "Fast, compact, and high quality LSTM-RNN based statistical parametric speech synthesizers for mobile devices," 2016, *arXiv:1606.06061*. [Online]. Available: http://arxiv.org/abs/1606.06061
[53] H. Salehinejad, S. Sankar, J. Barfett, E. Colak, and S. Valaee, "Recent advances in recurrent neural networks," 2017, *arXiv:1801.01078*. [Online]. Available: http://arxiv.org/abs/1801.01078
[54] Y. Tay, L. Anh Tuan, S. Cheung Hui, and J. Su, "Reasoning with sarcasm by reading in-between," 2018, *arXiv:1805.02856*. [Online]. Available: http://arxiv.org/abs/1805.02856
[55] A. Graves, N. Jaitly, and A.-R. Mohamed, "Hybrid speech recognition with deep bidirectional LSTM," in *Proc. IEEE Workshop Autom. Speech Recognit. Understand.*, Dec. 2013, pp. 273–278.
[56] M. Thelwall, K. Buckley, and G. Paltoglou, "Sentiment strength detection for the social Web," *J. Amer. Soc. Inf. Sci. Technol.*, vol. 63, no. 1, pp. 163–173, Jan. 2012.
[57] M. W. Berry and M. Castellanos, "Survey of text mining," *Comput. Rev.*, vol. 45, no. 9, p. 548, 2004.
[58] S. Yu, J. Su, and D. Luo, "Improving BERT-based text classification with auxiliary sentence and domain knowledge," *IEEE Access*, vol. 7, pp. 176600–176612, 2019.
[59] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," Tech. Rep., 2017.
[60] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," Tech. Rep., 2018.
[61] O. Araque, I. Corcuera-Platas, J. F. Sánchez-Rada, and C. A. Iglesias, "Enhancing deep learning sentiment analysis with ensemble techniques in social applications," *Expert Syst. Appl.*, vol. 77, pp. 236–246, Jul. 2017.
[62] M. Giatsoglou, M. G. Vozalis, K. Diamantaras, A. Vakali, G. Sarigiannidis, and K. C. Chatzisavvas, "Sentiment analysis leveraging emotions and word embeddings," *Expert Syst. Appl.*, vol. 69, pp. 214–224, Mar. 2017.
[63] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: http://arxiv.org/abs/1412.6980
[64] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimisation," Tech. Rep., 2014.
[65] D. Ghosh, W. Guo, and S. Muresan, "Sarcastic or not: Word embeddings to predict the literal or sarcastic meaning of words," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2015, pp. 1003–1012.
[66] M. Zhang, Y. Zhang, and G. Fu, "Tweet sarcasm detection using deep neural network," in *Proc. COLING, 26th Int. Conf. Comput. Linguistics, Tech. Papers*, 2016, pp. 2449–2460.
[67] T. Xiong, P. Zhang, H. Zhu, and Y. Yang, "Sarcasm detection with self-matching networks and low-rank bilinear pooling," in *Proc. World Wide Web Conf. (WWW)*, 2019, pp. 2115–2124.
[68] M. A. Di Gangi, G. L. Bosco, and G. Pilato, "Effectiveness of data-driven induction of semantic spaces and traditional classifiers for sarcasm detection," Tech. Rep., 2019.
[69] S. Poria, E. Cambria, D. Hazarika, and P. Vij, "A deeper look into sarcastic tweets using deep convolutional neural networks," Tech. Rep., 2016.

CHRISTOPHER IFEANYI EKE received the B.Sc. degree in computer science from Ebonyi State University, Nigeria, and the M.Sc. degree in mobile computing from the University of Bedfordshire, Luton, U.K. He is currently pursuing the Ph.D. degree with the Department of Information Systems, Faculty of Computer Science and Information Technology, University of Malaya, Kuala Lumpur Malaysia. His research interests include data science, NLP, information system and security, cloud computing, machine learning, big data, and social media analytics.

AZAH ANIR NORMAN received the Ph.D. degree in information systems security. She is currently a Senior Lecturer with the Department of Information Systems, Faculty of Computer Science and Information Technology, University of Malaya, Kuala Lumpur. She had previously worked as a Security Consultant with the MSC Trustgate.com (subsidiary body of MDEC Malaysia), a certification authority in Malaysia, for more than four years. Her research interests include e-commerce security, information systems security management, security policies, and standards. She was awarded a few research grants that focused on social media security and cybersecurity practices. Her articles in selected research areas have been printed in local and international conferences and ISI/Scopus WOS journals. She actively supervises many students at all levels of study, from undergraduate (i.e., bachelor's degree) up to postgraduate (i.e., master's and Ph.D.) supervisions.

LIYANA SHUIB received the B.Comp.Sc. degree (Hons.) from Universiti, Teknologi Malaysia, Skudai, Malaysia, the master's degree in information technology from Universiti Kebangsaan Malaysia, and the Ph.D. degree from the University of Malaya, Kuala Lumpur. She graduated Ph.D. student and presently supervising several postgraduate students. She is currently a Senior Lecturer with the Department of Information System, Faculty of Computer Science and Information Technology, University of Malaya. She has more than 20 articles relevant to her research interest.