VFAST Transactions on Software Engineering
https://vfast.org/journals/index.php/VTSE@ 2024, ISSN(e): 2309-3978, ISSN(p): 2411-6246
Volume 12, Number 3, July-September 2024 pp:177-198

BERT Model Adoption for Sarcasm Detection on Twitter Data

Tayyaba Javed [1,*], **Muhammad Asif Nauman** [2], **Rushna Zahid**[3]

[1]Department of Computer Science, Barani Institute of Information Technology, Rawalpindi, 46604, Pakistan; [2]Riphah School of Computing & Innovation, Riphah International University, Lahore, Pakistan; [3]Department of Computer Science, COMSATS University Islamabad, Vehari Campus, Pakistan

Keywords: *Deep Learning, Sentiment analysis, Twitter, LSTM, BERT*

Journal Info:
Submitted:
September 24, 2024
Accepted:
September 26, 2024
Published:
September 28, 2024

Abstract

Sarcasm is a term used to criticize someone's feelings. Sometimes, humans are not able to identify sarcastic comments, and they typically express the reverse of what they mean when they make snarky remarks. Therefore, the detection of sarcasm within a text automatically is a difficult task. Its significance in enhancing sentiment analysis has also made it an important study field. In previous studies, different approaches to deep learning (DL) and machine learning (ML) have been explored. However, previous approaches mainly depend on the lexical and linguistic aspects. Therefore, these techniques could not perform well in the context of sentiment accuracy. In this research, an efficient approach for detecting sarcasm is proposed. A Bidirectional Encoder Representation from a Transformer (BERT) is proposed to improve the sentiment accuracy in this research. This research also aims to compare the two models of deep learning, the BERT and LSTM (Long Short-Term Memory) models. This comparative analysis aims to provide a detailed overview of the pros and cons of each approach for the detection of sarcasm. The primary aim of this study is to examine the different existing ML and DL approaches for the identification of sarcasm. Apart from this, the comparison of BERT and LSTM contributes to the ongoing debate about whether models work best for sarcasm detection in social media. In this study, sentiment analysis's accuracy is improved by making better decisions, especially when it concerns Twitter interactions.

***Correspondence author email address:** tayyaba@biit.edu.pk
DOI: 10.21015/vtse.v12i3.1908

1 Introduction

Social media platforms, particularly Twitter, have grown at an unprecedented rate, which has led to an increase in user-generated content [1, 2]. The significant growth in data offers researchers and analysts a unique opportunity to gain a comprehensive grasp of popular attitudes, beliefs, and prevailing patterns [3]. Consequently, sentiment analysis emerged to

extract knowledge from social media data. By means of sentiment analysis, analysts have the opportunity to discover how people respond emotionally to some topics or events. This provides them instantly invaluable information about public points of view [2, 4]. The crucial thing when interpreting people's opinions and emotions regarding a particular aspect of the issue or service under investigation is to have a correct judgment on the sentiment, which can be absurd for a user as well as for the system. The existing ones cannot process metaphorical language effectively. Sarcasm is also identified as a language tool, and it primarily communicates comical meaning through social media. Identifying and separating sarcasm in social media applies to distinguishing sarcastic comments from various posts that have been posted on several social networks. Sentiment analysis in social media is complex because people use hidden tones, such as sarcasm and irony, to communicate their sentiments. The appropriate characterization of sarcasm in language changes the polarity values within a textual document and, hence, decreases its predictive precision for use with sentiment analysis. Thus, sarcasm is associated with the antithesis between shared lexical performance and the function of saying it sarcastically. Therefore, only to make the issue even more intriguing and described as relevant concerning social sciences, news headlines or online markets, people tend to incorporate some form of sarcasm, which involves oral irony, when sharing their thoughts.

In this area, most data miners apply the deep-learning technique instead of previous machine-learning approaches. Showing the unevenness or complex nature of pre-classification fundamentals, such as a feature extraction element based on traditional ML techniques, yields better results with the help of deep learning approaches. Such problems can be easily solved using deep learning methods, as no human intervention is required to adjust the features. This paper has introduced a Transformer-based Bidirectional Encoder Representation from Transformer (BERT) model for sarcasm recognition, which establishes the fact that an utterance is sarcastic or not merely by addressing that sentence with some context. It is a Transformer-based architecture. The transformer encoder applies multi-head attention and generates a cohesive embedding description of the target utterance with the background in our model. If we cut to a sample taken from Twitter, this method is much more accurate than the baseline precisely since target utterance is used as an input. Therefore, it can be concluded that Twitter has become the most significant medium where people post what is in their minds and share opinions and other real-life news, such as live tweeting, to mention a few.

Since it's been a year, data from Twitter has grown several hundred times to form big data. Twitter has a total of 315 million active users per month; approximately eighty-two per cent are from the mobile platform, and millions of tweets go through Twitter every day. People, as well as companies, resort to Twitter information so that they can know people's perspectives concerning the product or even an occurrence related to a particular company. However, Twitter is used to learn various people's opinions on political occasions, such as in films. However, it might be challenging to discover the views of other people because of the 140-word limit per tweet and the informal language that is used in tweets, such as slang, emoji hashtags, etc. Sarcasm among such tweets is more challenging to spot, in my opinion. Sarcasm refers to the delegation of negative sentiments by employing positive words. According to the Oxford Dictionary, sarcasm is a form of irony that expresses disgust. Macmillan's Dictionary defines otherwise as the act of saying or writing contrary to what a person desires- a thing that displays how one will drive someone silly and angry. According to Collins' dictionary, sarcasm refers to mockery or an ironic tone intended to insult. Sarcasm and irony are considered two almost identical concepts. Sarcasm, however, is the opposite of what he says. Sarcasm is not limited to making jokes but can also be used to criticize other persons, opinions, and many more. This is why sarcasm, in general, has a vital role to play with respect to Twitter, for instance- "I enjoyed being snubbed". Here, "enjoy" is a positive sentiment in the negative sense. So, this tweet is known as sarcastic. Therefore, the analy-

sis of sarcastic tweets is complicated. Three types of sarcasm have been shown, which are the following:

1. **Propositional sarcasm:** In Propositional sarcasm, the scope is targeted towards some proposition to which the speaker has committed. Example: If you want a delicious snack, then go to KFC.
2. **Like-prefixed sarcasm:** It operates as a subtle denial of the argument that is being put forward. For instance, I always communicate with Ram.
3. **Illocutionary sarcasm:** This type of sarcasm indicates an attitude opposite of what has been stated.

Based on the text features used for classification, sarcasm detection can be classified into three segments. The categories are Lexical and pragmatic features of the hyperbolic feature-based classification. The lexical feature-based classifier has text characteristics like unigram, bigram, and n-grams. Symbolic and figurative text refers to feature-based pragmatic classification. Hyperbole feature-based classification is one which depends on text features such as intensifiers, interjection punctuation marks and quotes. Thus, the primary motivation of this study is to identify whether a given emotional word in an expression has a positive or negative connotation. Opinion mining, a type of people's sentiment analysis, recognizes subjective information in source documents. The use it facilitates counting people's opinions (sentiments) towards goods, politics, services or individuals render to the organizations' many advantages. The identification of subjective information can only be achieved through this possibility. It assists in developing organized knowledge that acts as a single, quality piece of information for decision support systems and individual decisions [2]. For instance, effective computing and sentiment analysis can enhance recommendation systems and customer relationship management by understanding the likes or dislikes of customers or removing product suggestions that got a negative response from them [5]. The potential benefits of this research are vast, including improved sentiment analysis, enhanced recommendation systems, and a deeper understanding of linguistic complexity in digital communication, paving the way for more effective and accurate sentiment analysis tools in the future.

The majority of the Web's text content is figurative language, particularly sarcasm and irony. For example, the Internet Argumentation Corpus extracted from 4forums.com contains only 12% of sarcastic statements [6]. Unfortunately, not every social media user uses sarcasm or irony to describe his/her emotions, making it challenging to analyze the sentiments of people. The presence of sarcasm in a textual document can alter sentiment polarity, reducing precision for Sentiment Analysis. Sarcasm in statements implies a contradiction between the words stated as textual utterances and what an individual means by making sardonic claims. Vocabulary laden with emojis, text abbreviations, and slang terms have been used by social networking sites like Reddit or Twitter to denote the users' sensitive intentions while employing sarcasm. This makes the process of having a reasonable discussion even in common language on these social sites more tiresome [7]. Despite these challenges, sarcasm detection has become an attractive area of research in recent decades due to its potential to provide deep context understanding for various applications, including author profiling, irony identification, and harassment. Many computer approaches have been suggested in the literature on sarcasm detection for conversations [8]. However, as many of the previous studies were based on statements alone, it is also not easy for humans to pinpoint sarcasm in isolation. In other previous studies, different DL approaches have been used to accurately detect sarcasm in tweets. However, the biggest weakness that most researchers had while using DL algorithms was polarity incongruity as some words were unknown within the sentiment lexicon [9]. Second, DL methods typically need higher computing and larger training datasets. In order to address this problem, the researchers have tried focusing on concepts thematic for each word in a sentence that defines the polarity of sentences. This paper aims to overcome the challenges presented in these models by implementing BERT over other DL methods because of its superior performance

when detecting sarcasm from social media text. The proposed BERT model holds great promise in the field of sarcasm detection, offering a potential solution to the challenges faced by previous models. Such a model aims to fill the vacancies between current research work and provides deeper insight into linguistic complexity associated with digital communication as it improves sarcasm detection accuracy over Twitter data [10]. This research is significant as it aims to overcome the challenges in sarcasm detection and sentiment analysis, offering hope for the future of this field of study.

The rest of the paper is organized as follows: Section 2 presents the related work on sarcasm identification techniques, while proposed methodology is given in section 3. The results of conducted experiments are described in section 4. Finally, section 5 concludes this paper.

2 Literature Review

Researchers have extensively investigated the detection of sarcasm in textual forms. A comprehensive range of research lines, including lexicon-based [11], rule-based [12], pattern-based [13], and the more recent deep learning [14], have been studied. This thorough exploration provides a wealth of knowledge in the field of sarcastic text detection as shown in Figure. 1. The step-by-step details of these methods will be comprehensively explained in the following section.

2.1 Sarcasm Identification Approaches

2.1.1 Lexicon-based approach

T he lexicon-based approach, which uses the bag-of-lexicon (comprised of unigrams, bigrams, trigrams, etc.) and phrases to recognize irony or sarcasm in tweets, was implemented by [15]. He employed a bootstrapping approach for the selection of two bags of lexicon, each containing unigram, bigram, and trigram phrases. These expressions have been used to determine if a situation is ironic or sarcastic when a positive arousal is applied in a negative situation. Similarly, four bag-of-word sets labeling positive sentiments, negative sentiments, joyous situations, and negative situations have been implemented [14]. However, they also used sarcastic expressions to establish a context in which negative emotions were the outcome of an optimistic scenario, and vice versa.

2.1.2 The Rule-Based Approach

In sarcasm detection, a rule-based technique is based on the pattern detection method that utilizes an object on which a particular rule or policy is built. The rule-based approach utilizes semantic and syntactic features of a sentence, including the structural pattern of phrases and the sentence's lexical structure, to identify sarcastic comments in any language within grammar. Most investigators use this method as a way of achieving a better result when compared with a classifier. Along with rule-based approaches, the semantic-based approach, which gives more importance to the meaning of a word, its structure, and how words interact structurally in the language and context of use, has been introduced. The semantic-based paradigm, among others, is considered to be the base of the rule-based conception because of its powerful impact on nature. Hence, an experiment was performed by Riloff, et al. [11] using this type of determination to detect sarcasm. The research applied the dataset related to the Twitter platform. The research [14] is based on two algorithms to check polarity mood and tweets that begin with interjections.

Nevertheless, they demonstrate that cordial statements open sentences in the most sarcastic ones. According to [14], its algorithm relied on combining the rule-based search for any adverse circumstance and positive phrase of a verb in a sentence. In the study, these researchers made use of a well-organized iterative technique to identify the phrases of the questions that had negative situations and carried out the experiment using a number of rule sets.

2.1.3 Machine learning Approaches

ML is only one of the most common approaches utilized by researchers for sarcasm detection. This is thanks to the fact that the software was developed to be stable, with the ability to self-observe itself as a set of data and the criteria given. Machine learning models usually deduce intelligent predictive models. Lexical and pragmatic aspects were researched in

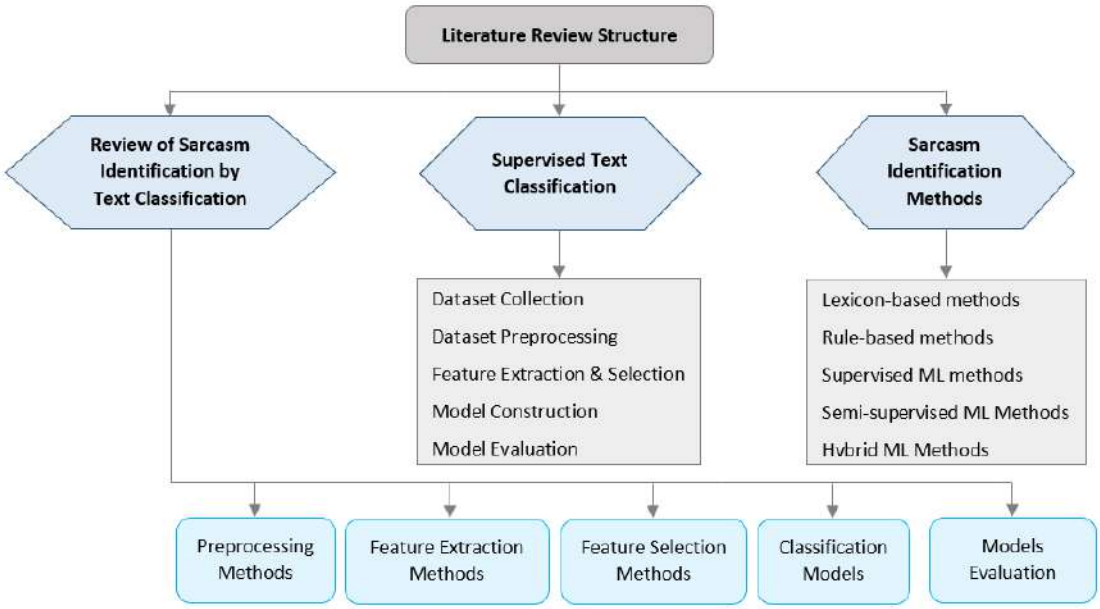

Figure 1. Taxonomy of Literature Review

[16], where machine learning methods were applied. The machine learning approach can be delimited to unsupervised, supervised, semi-supervised, structural, and hybrid learning. Approaches such as the ones that we shall discuss briefly are illustrated below.

- **Supervised Learning**
  Many ML algorithms are based on supervised learning; supervised learning is generally used in sarcastic comments' detection through labelled data since it can produce a decent model with the help of labelled input data that helps in the model's construction. Thus, because the training sets have provided the model that will process the result, this is made possible. The fundamental supervised learning algorithms (NB, DT, and LR) function as a basis from which other similar learning algorithms are borrowed [17]. Another popular approach is the SVM and LR machine learning algorithm with SMO. The key is their likelihood of determining sarcasm rather than just polarity emotion, which is typical in Twitter messages [16]. Later, the research on deep learning gave an opportunity for researchers in the domain to research automated recognition of sarcasm. It is a form of learning that incorporates a neural network to automatically learn and process the features in an input data set [18, 19, 36–39]. The neural network learning algorithm behaves in the same manner as the nervous system in the brain of a human. In the model of the neural network, every network's unit contains some connections with other ones that have a summation function that simply adds all the value units. The neural network involves 0 in the approach. 0 and 1 real numbers are represented by the core and fibre conducting the value.
  On the contrary, Ghosh and Veale [20] applied a deep neural network (DNN) to detect sarcastic tweets in user-generated datasets. In their developments, they paired algorithms including convolutional neural network (CNN), long short-term memory (LSTM), and recursive support vector machine, which contributed to the tremendous success of the system of F-score with 92%, compared to the baseline method. Experimentation on the selection of congruent and incongruent words was tested in terms of performance in [21]. This has caused an

improvement in the performance of the model.

- **Semi-Supervised Learning** This type of model is a combination of supervised and unsupervised methods. It uses only a few examples of labelled data and makes use of unlabeled data extensively. This sort of learning process is applied by Davidov, et al. [22]. The authors performed sarcasm identification automatically using datasets related to Amazon product reviews. In their research, a sizeable number (66000 of the products and reviews of books) were gathered to analyze the features that are syntactic and pattern-based. In the training phase, a sentiment polarity of 1 to 5 is considered a choice in the training data. The authors' report encouraged precision and recall of 77% and 83%, respectively. They retain 1% in the evaluation phase.
- **Hybrid-based learning**
  A hybrid technique is the formation of a new classifier out of the combination of two or more classifiers. This refers to a classifier formed as an ensemble encompassing all sorts of classifiers. It is a convolution neural network that learns user contextual features by combining them with the utterance-based embedding feature. The characteristics of the designed model were a mix of user-embedding convolution and CNN (CUE-CNN), which in the area of sarcasm identification use the newly designed approach by researchers, and the result achieved a 2% enhancement in performance over single ML techniques for detection of sarcasm.

2.2 Supervised text classification process for sarcasm identification

As Nithya, et al. [23] pointed out, supervised text categorization is the kind of classification that uses datasets having labelled training data to learn and construct the text algorithm that is used to automate classifying unlabeled testing data. The current industry practice is usually a human observer who can quickly get the correct information through saved email messages and web addresses from thousands of files every day. Besides, manual categorization is almost invariably laborious, and time lags. Not only is that, but the fact that it has subjectivity is one of the difficulties that come with manual categorization. As mentioned, the limitations converted the classification of textual data from manual data to automated data. Several types of text classification methods are present in the automated classification of text. Nonetheless, the supervised technique is the one most globally employed as it includes labels during input data construction [17]. An experimental process of text classification of supervised learning in which supervised text classification has been designed into six main steps, which will be discussed in the following sections.

2.2.1 Data collection

Data collection is the cornerstone of any text classification process. For instance, in a sarcasm study on Twitter, Twitter data serves as a crucial source of information. The study aims to understand the response and recovery of a disaster through sentiment analysis. Therefore, the classification process begins with the collection of raw data, which is then preprocessed for objective analysis in the subsequent stage.

2.2.2 Data Preprocessing

Raw data that is gathered during the phase of data collection consists of a significant amount of noisy data that needs to be removed. Cleaning is a process that eliminates unnecessary noise from data. This then leads to the discovery of new knowledge and features. The other agenda is to delete duplicate information that is highly present in social media data [24]. Data preprocessing usually means raw data preparation, as in the extraction of training and testing sets. During the training phase, data from Twitter is categorized into sarcastic and non-sarcastic sets that are used for the model training. However, since the testing datasets are solely meant to be used for model evaluation, they are not labelled. At this stage of the preprocessing, the non-sense characters are eliminated. The factors that do not affect sentiment classification are thrown away. This part will be tokenized, which is a more common term, as well as automatic filtering. This is done with the view of disregarding retweets, stopping words, dupli-

cates, punctuations, URLs, numerals, other languages' tweets, and just URL tweets. During the latter step, the tweets are put through POS tagging and stemming, thus making the content as it was originally written.

2.2.3 Feature Extraction

Feature sets from the entire dataset consist of unimportant variables that might affect the classification by limiting the prediction results. The failure of the text classifier because of the non-materials in nature is a decrease in accuracy, a problem in producing the result, and a recent classification process, which is a difficulty in storing and retrieving the information. It follows, therefore, that there had to be a selective technique to eliminate feature subsets that are not sufficiently discriminant and to retain the subsets that are for improved prediction. To make the best use of the dataset, it is crucial to evaluate the characteristics of the dataset that could impact the prediction. The feature selection method can be classified as a type of wrapper, random, and embedding. Particle removal by the filters is the most commonly used method in trace metal analysis [25]. The bag-based method takes into account each feature as a bag, and a ranking using the specific statistical score is assigned to each feature. The feature selection and removal are based on the feature score. One of the most frequently used filter-based approaches is the chi-square (2) and information gain (IG) feature selection. However, unlike the wrapper-based strategy that chooses the best attribute and evaluates the rest of the ways using the query tool, the embedding method looks into the most significant features throughout the construction of the model.

2.2.4 Representation of Features

In the case of the classification of textual data, a feature representation uniquely converts the extracted feature to numerical data during the representation of a feature. A technique of feature representation can be either term frequency (TF), binary representation (BR), or term frequency with inverse document frequency (TFIDF). The TF method puts the feature value as a sum of this feature in the entire document. Nevertheless, in the BOW method, the Boolean functions of value 0 or 1 are used. On the other hand, value 1 shows that the feature is present in the document, and value 0 means that the feature does not exist. The comparison is made on the basis of the inverse document, and the frequency of the text in a specific paper is determined; after that, it is compared with the frequency of the words in the whole of the document. It is done through the capability of detecting a specific word in the query and connecting it to current documents that are pertinent to the question.

2.2.5 A Construction of a Classifier

A classification model is built with the dataset containing a training method using an ML algorithm. The generated classification model can label unlabeled data as sarcasm or non-sarcasm. Different classifiers are deployed as methods for the identification of sarcasm. Some classifiers such as SVM, DT, RF, ANN, and NB have been incorporated in recent studies. These sample classifiers are introduced in the subsection below.

- **Naïve Bayes** NB is a technique of classification from the machine that uses a probabilistic model to predict the class from which a given data is coming. This algorithm is machine learning in which multiple statistical analyses of numerical data are conducted. It takes a labelled dataset as input data to create a metric of the generative model s optimization. Regarding the classifiers, it is one of the easiest to learn as it denies relationships between the various features when talking about a specific class. Furthermore, NB has proven to be one of the quickest classifiers and shows the best result when the BOW technique is utilized in the representation of text.
- **Decision Tree** This ML classifier is considered to be the primary knowledge discovery of data mining to classify the data as well as for data prediction. DT is an algorithm of learning that is embedded in the instances and is built based on the rule derived upside down from the set of orders under it to an individual case. The tree has been identified as the leaf node, path, decision node, and edge. DT classifier is modelled

in the form of a 2-way tree, in which the middle node represents the test of the attribute, each outgrowth means a result of the testing, and each leaf node represents a class. Therefore, the entire tree answers the question of whether there exists inconsistency or not in decision tree learning. Instance classification, which is achieved through the employment of DT, classifies sample instances by counting the definite attribute appearance of the value sets. One of the more probable downsides of the decision tree classifier is an over-fitting problem. This capacity exists in its tendency to group every type of data with noise, which can seriously impair accuracy. However, using a solution such as the RF, where multiple decision trees are developed by splitting the sets of training and the outcomes of the predictions are determined through each tree, is a practical approach to solving this problem.

- **Random Forest** The RF is an ensemble classifier that coincides with tree nodes by different training sets. DT would accordingly infer the classes for all different input values (vectors) in the forest, and the most frequent ML algorithm would be chosen. RF process solves the overfitting problem, and comparative prediction is better than a single decision tree.
- **Support Vector Machine** This type of classifier can be said to be a supervised learning technique that is based on the statistics learning theory. Classification of the data is to be performed by separating it into a training dataset and a testing dataset. At the same time, it applies the training data to construct a model for predicting the output of the test data for the unseen data that is employed. In the SVM, an agent called a hyper-plane is used for classifying the two-class data sets at the time of reduction in the space among them by the sets of training. Different applications like sarcasm detection, image classification, and bioinformatics have been successfully performed using an SVM classifier.
- **Maxent-Type Algorithm** Having the option to pick the large entropy value classifier that best fits with the data of training from all models is what the classifier, based on the maximum entropy, does. This model does not impose the independent assumption of the feature, which is why it has relaxed conditions and a less strict structure compared to other classifiers. In the same vein, the maximum entropy classifier is fitted with an optimization problem that should be optimized when the parameters of the model are calculated. The consequent requires more training epochs as compared to other ML classifiers like the NB classifier [12].
- **Artificial Neural Network** It is an algorithm of learning that acts like the brain of a human, and it has a mechanism for processing and transferring information. The main building block of the artificial neural network is formed by three constituent layers, which are input, output, and hidden. In this classifier, the current unit is connected with all the other units of the network; the summation function is made to combine all the values of the input. In this case, the hidden layer of a network is designed to process the input, indicating that it is connecting to the output layer, which gives particular values at the output. This type of classifier would apply a 0. 0 and 1 as the actual number assigned in terms of core and axon [17]. The study [26] reported that artificial learning network learning is either unsupervised, supervised, or reinforcement style. The unsupervised data-centred solution is based on the relationship among the given data input. This is because verification of such essential data is not possible due to the learner's lack of "right response" knowledge. By comparing the input of the network with the intended output of the artificial neural network, supervised learning reduces the error function that is present in these networks. As a result, general gradient descent-based optimization algorithms based on the backpropagation technique are used so that weights alternately change to minimize the error. While some supervised learning methods

are reinforcement-based in nature and give information on the actual output accuracy, the rest of the methods belong to the unsupervised learning family. In such a scenario, there is no information about the desired output result to be obtained without precedents. In an artificial neural network, the delta rule is the learning rule; it is used to modify the weights on each pattern of input. The most common learning rule is the delta rule.

2.3 Sarcasm Identification Review using a Technique of Text Classification

2.3.1 Preprocessing Techniques Review for Identification of Sarcasm

Data preprocessing is needed because of the informal and irregular data it acquires. The step of preprocessing is to remove the obstacles that have to deal with these types of texts, such as mistypes, use of acronyms, complex sentence structure, and unnecessary repetitions. In the preprocessing step, information that is deemed to be useless is deleted to boost the classifier's performance. Several techniques of preprocessing have been used in the research of sarcasm identification, including stop word removal, blank space, punctuation and special symbols removal, conversion of uppercase to lowercase, stemming, tokenizing, pos tagging, URLs and hashtags removal, and lemmatization. Hence, the efficiency of such techniques of preprocessing is shown in different research studies to get insights of this nature. Samonte, et al. [27] experimented with the inclusion or sophistication of URLs and user mentions in tweets for the identification of sarcasm on Twitter. During the investigation, it was seen that omitting them brings up a higher accuracy rate of classification than what they exist. The study Abulaish and Kamal [28] used templates for preprocessing tasks such as stemming, tokenization, and conversion of capital letters to small letters. These studies were commended for the efficiency of their machine-learning classifiers, which indicated a higher rate of correct classification compared to other studies. The authors applied the same method; they omitted white spaces, characters, punctuation marks, numerals, and emoticons. Their papers stated that these preprocessing actions had been applied, which in turn showed improvement in the tasks of classification.

Nonetheless, Ghosh, et al. [9] explored the ability of punctuation marks to take on the role of modelling while dealing with a study on 'Unveiling sarcasm from hyperbole to hashtags'. Their experiment revealed that a better categorization occurred when marks of punctuation were added, while a worse classification occurred when the marks were removed. This finding suggests that punctuation marks can play a significant role in sarcasm identification, and their inclusion or exclusion can impact the accuracy of the classification. Thus, the researchers should try out several preprocessing approaches and determine the performance of each technique on the sarcastic corpus and the accuracy of the algorithm in classification. As presented in Table 5 below, the techniques of preprocessing used in the selected papers are summarized as follows. The output of Table 5 shows that a large number of studies either utilized simple data preprocessing techniques or did not disclose the techniques they employed, which shows that data preprocessing plays a crucial role in achieving good classification accuracy.

2.3.2 Feature Extraction Techniques Review for Sarcasm Identification

In the task of sarcasm detection, in the feature extraction, significant and contrastive data from the sarcastic dataset are extracted, which will be used in the training process of the machine for predicting sarcastic or non-sarcastic sentences. The analysis of the papers of the specific studies showed that in most cases, researchers applied the semantic properties of the sentence features, which include the underlying meaning, context, and tone of the sentence. The content-based and linguistic features were extracted by automatic feature extraction. This was done by applying the algorithm and multiple statistical models. The employed the N-gram feature extraction method for their chosen studies [29]. For example, authors [30] applied an n-gram technique for sarcasm detec-

tion; their finding was that this technique makes the extraction of the lexical features possible. The language model n-gram used by the researcher is one of the motivations because it has simple and scalability properties (about the number of substantial sample datasets). In this case, Suhaimin, et al. [29] conducted a make-believe study on sarcasm detection in the bilingual text using a variety of NLP techniques to get the blend of particular features such as lexical-pragmatic-syntactic-prosodic-idiomatic. These elements had been trained on an SVM algorithm with a non-linear feature. Alternatively, the result demonstrates that the features constructed by NLP improved over traditional global bag-of-words and achieved better results than those of the advanced method. Also, lexicon sentiments were extracted in [16]. Lexicon sentiments were accompanied by pragmatic features (emoticons and user mentions) for sarcasm identification. The testing established that the presence of these factors made it more precise in the case of prediction.

2.3.3 Feature Selection Techniques Review for Sarcasm Identification

Feature selection is a process that uses a set of criteria to identify proper sets of features. It is widely exploited in the detection of sarcasm. While very few of the articles that discuss sarcasm detection in this group of selected studies employed the feature selection method, it is a crucial aspect of the process. Some of the studies use the techniques of feature selection, Chi-square (2), information gain (IG), and mutual information (MI), among others, which are elaborated as.

- **Chi-square (2)** is a statistical test that helps to obtain the lack of independence between class (c) and terms of feature (f). This technique is particularly useful in the context of sarcasm identification, as it can reveal the relationship between different features and the class they belong to IG. IG is a technique of feature selection that determines the information gained by knowing the values of a feature vector belonging to a given attribute.
- **MI:** It can mathematically express the dependence between two random variables (word association and related applicability) through a lot of work.

In this regard, Kumar and Harish [31] performed three of the orthodox feature selection techniques: mutual information (MI), information gain (IG), and chi-square (2), which were used to determine the discriminant features for sarcasm classification. The researcher investigated if these techniques were present, and from the findings of the experiments, the utilization of feature selection techniques minimized clutter from the ample dimensional feature space and further improved the performance of the classifier. As an illustration, SVM and RF algorithms made the highest accuracy when MI and IG selection mechanisms were implemented in the Italian dataset [32]. For the validation of linguistic traits for emotion, the N-gram lexical features were retrieved based on the inquiry of linguistic and word count paramount process such as emoji, as well as punctuation, was extracted. Nonetheless, the variables were chosen from those features, making use of the discrimination feature selection scheme through the two before modelling. The analysis revealed five techniques that use Chi-square to choose the disparate features and three methods that use information gain. Another method is Chi-square, information gain, and mutual information, and the rest of the 31 studies did not disclose any protocol for selecting essential features from the features that have been extracted.

2.3.4 A Classification Techniques Review for Sarcasm Identification

The assessment of the results leads us to conclude that different algorithms are applied to detect sarcasm in posts on social media. The experiments made use of the multiple classifiers separately to see the performance of the proposed approach in terms of each classifier. In some instances, just one learning algorithm was utilized for classification. On the other hand, the researchers who examined sarcasm identification in their work used different datasets.

Therefore, it becomes uncertain whether the best classifier can be determined by taking into account all the different metrics in such a situation. However, it used the upside-down classifier to extract sarcasm. In the course of these studies, average recall lets the algorithm score the labels, giving good results when the area under the curve (AUC) metric is utilized, which shows confidence in such labels. In another one, from the tweets dataset, random forest (RF), support vector machine (SVM), K-nearest neighbour (K-NN), and maximum entropy (ME) were used for the sarcasm classification task with feature-related entities. The test classification demonstrated that the RF model gave better results compared to SVM, K-NN, and ME, which achieved 81% accuracy and 3% F-measure. Ling and Klinger performed a comparative analysis of the 'Classification of differences between irony and sarcasm', utilizing the algorithms of DT, ME, and SVM algorithms. The experimental result displayed that the ME model scored more than the decision tree and SVM classification algorithms. Suhaimin, et al. [29] hypothesized the test performance of NB, DT, RF, LR, and SVM in modelling the linguistic classification among the three figurative messages 'Sarcasm', 'No', and 'Irony' on Twitter.

Out of these ML classifiers, the most significant outcome of the f-measure was acquired using the RF classifier to discern Irony and Not. However, the outstanding performance of sarcasm detection that was carried out by [11]) using similar datasets was much higher, with an F-measure of 0. 70. 62 to 0. 70. Aside from that, [31] compared the efficiency of NB (Naive Bayes), DT (Decision Tree), and B (ensemble) classifier to distinguish hyperbole and self-depreciation features for sarcasm recognition in the (balanced and unbalanced) tweets data. The researchers present the result of the performance in the case of each classifier's precision, f-measure, and recall. The bagging classifier obtains better precision values in two separate datasets, whereas the best f-measure and recall results are achieved by the DT classifier in both data sets. The most employed classifiers for sarcasm identification on social grounds are SVM and NB. Among the 40 chosen analyses, only some of the analyses used an SVM classifier.

3 Research Methodology

In this study, workflow architecture is shown in Figure. 2. Workflow architecture is identifying the sarcasm. A textual dataset collected from the Twitter Platform is used to carry out the experiments [33]. A Twitter platform's data is being preprocessed initially to remove any noisy data. An approach to textual feature engineering is categorized into word embedding techniques. Preprocess data is further categorized into two halves, such as 20% of the dataset contributing as a test set and 80% of the dataset contributing as a train set. Two different DL techniques, BERT and LSTM, are used for dataset processing. After processing these two techniques, a comparison is performed. The training set is the whole batch from which models of applications are trained. Real-time testing of an outperforming BERT model was done on the basis of a selected test set performed in real time. Different evaluation parameters are applied to validate the model's higher performance. In this study, F1 score, accuracy precision, and recall are used as model evaluation parameters. The three elements of the proposed methodology's structure can be seen in Figure. 2: tweet preprocessing, feature engineering, and sarcasm recognition.

3.1 Data Collection

In this study, two types of data sets are being referred to. One is drawn from the Kaggle site, while another is drawn through a Twitter API. The given study relies on an open-source textual data download from the Kaggle. This dataset is available for free on the Internet in a well-known database repository, Kaggle. This data can be produced by the collection of ironic and satirical speeches on Twitter's social platform, which has a total of 54618 tweets. The data set file splits into two columns. As shown in Figure. 3, the class column represents the target label, which means it is classified as regular, sarcastic, figurative, and ironic classes. The column of tweets shows a textual message that provides what a Twitter user has written. The dataset is comprised of 54618 samples containing two key features, i.e. "Tweet" and 'Class'. Here, "Tweet" features essentially represent an object data type due to textual Twitter data, providing actual words for data analysis.

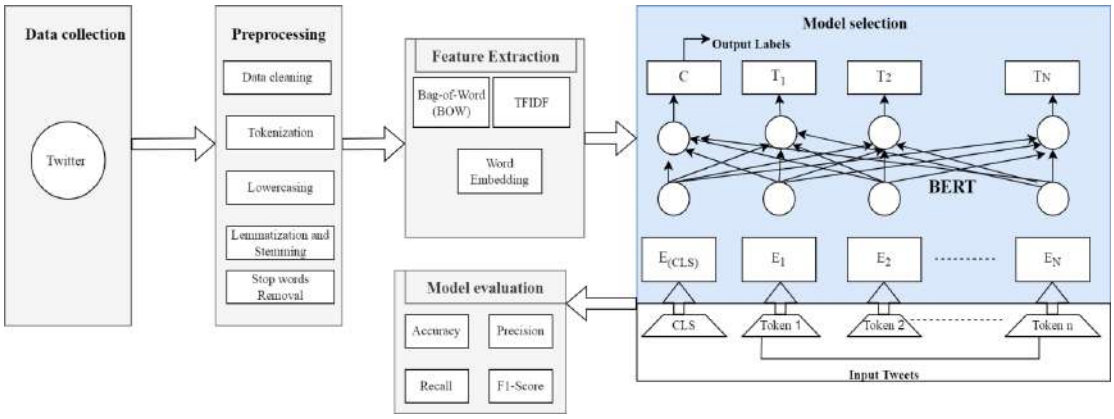

Figure 2. Proposed Methodology Architecture

In the "Tweet" feature, linguistic expressions used for the identification of sarcasm are contained and serve as the critical input of the data study. A "class" feature is also an object data type. An object data type is allocated to every Twitter class. Therefore, it is mandatory to divide each attribute of a target class into either Regular, Sarcasm, Figurative, or Irony. The pattern of these features is helpful for the subsequent descriptive analysis and sarcasm detection model training as this provides an insight into the textual properties of tweets' two classes, which were collected using them. In this case, the occurrence of the Regular class attribute is 18557; sarcasm occurred 15404 in a dataset, irony occurred 12784 times, and figurative occurred 7873 times within a dataset.

	tweets	class
1662	Thank goodness #NCAA has handle on most pressi...	figurative
7734	This is where Leonard hold up the #SARCASM sig...	sarcasm
1759	A man who killed many under his car is lovd. A...	figurative
3279	But did yall catch that "police" presenter for...	irony
2709	"Easy Like Heaven - The Cure..." This video is...	irony

Figure 3. Sample dataset with class label

3.1.1 Twitter Application Programming Interface

It is another way of sourcing data from social media sites such as Twitter. Using the Twitter API is a systematic and automated way to access a vast amount of open data, such as tweets, along with users' profiles and other associated information. Using this API has a systematic approach to collecting data following set parameters such as hashtags, keywords, and user names. Due to the requirement for capturing potentially up-to-the-minute and dynamically evolving data, an average Twitter set is chosen as a source of information. Search queries that define the limits of research, such as essential hashtags for sarcasm or specific user accounts known to share sarcastic content, are required in every process of data collection. The dataset to be analyzed further is comprised of the collected tweets and their metadata. Considering that Twitter APIs prove effective in evaluating the overall situation of user-generated content for this platform, research is aimed at describing top pictures and properties of sarcasm in general. The use of APIs essentially means that it is practical, works on a large scale, and does not violate the moral norm while accessing reliable publicly available information on Twitter for scientific analysis. It is also convenient for developers to work with the Twitter API and be able to retrieve tweets. The developer can target a search and retrieve the tweets that match precisely or set an ongoing stream to accumulate constantly each time an inform case happens. In this case, for data collection purposes, it is a 'stream' that is to be termed the necessity of tweeting. Using the Twitter API, tweets with the hashtag "sarcasm" are collected to identify the sarcastic tweets.

Similarly, different tweets are being collected on

different topics in order to neglect sarcastic tweets as well as to delete those tweets that are "sarcastic". In this scenario, a total of 76799 tweets have been collected, and these are categorized into two categories, i.e. 37583 sarcastic tweets and of non-sarcasm proportion, there are as many as 39216. Here, '1' is used for sarcastic tweets, and '0' describes Non-sarcastic tweets. In this case, the tweet column is comprised of the textual tweets of the users, whereas the target column is comprised of discrete numbers, i.e. 1 and 0, where it gives 1 in case of sarcasm and zero in not.

3.2 Data Preprocessing

Data Preprocessing consists of the following steps:

3.2.1 Preprocessing of the Text

Excessive noise is being eliminated with the help of advanced techniques of text preprocessing, which are introduced to the concerned sarcasm dataset. The limitations of the learning model arise due to noise in social media text that cannot identify sarcasm. First, we converted the column with users' tweets into a string type and then dropped the null value data from our dataset. After preprocessing, the final dataset consists of 54618 tweets only. There are our systematic approaches to the practices of cleaning texts.

3.2.2 Data cleaning

During this process, unnecessary elements that do not add emotions to the text are often incorporated in data downloaded from web pages and have to be eliminated. These disconnected elements and metadata included space, hashtags, quotes, retweets, emojis, and URLs, among others. Numerals are cancelled, and all alphabets are converged to lowercase letters. Remove all those Tweets that do not have any scenario.

3.2.3 Remove a Noise

The measurement of the noise removal factor is captured through text analysis. The fact that there will be apprehensions in achieving the advantages of data for classification purposes is essential to think about. The processes used for data are used to achieve optimal results. Noise removal is a process of deleting all the non-text elements from a document, such as punctuation, stop words, URLs, and others. The subsequent step uses the data that has been cleaned. To hasten the feature mining technique, delete all numerals and non-ASCII alphabets brought by the Twitter application and tweets that have a single word together with many string literals.

3.2.4 URLs removal

It has no worth in the tweets as it only refers to a specific page of the web application and shows no other detail. In the preprocessing of sarcasm detection methods, they are deleted.

3.2.5 Remove the Stop Words

The second stage of curtailing the properties during the preprocessing is dropping out the stop words and unnecessary details. The list of stop words contains predicate shifters, such as in, of, and to, as well as some articles, such as a, the, the, an, and some other words that could be used frequently. Of course, they bear no relation to the sarcastic accent of a sentence. As a result, they are taken away prior to processing.

3.2.6 Truncated Elongated Words

Many characters sprinkle such sentences as "goooooood", "looooooovve" and others. These words mark sarcastic intentions in tweets. In this tweet, the calculation of the number of alphabet parts in duplication and vowel doubling was not stated to be considered a preprocessing step. Some additional characteristics can be derived by determining the primary character for every word, such as good goal, love, or control.

3.2.7 Remove an Unnecessary Punctuation

In this case, all unnecessary punctuation from its body's text is removed. In the training of the model, they only burdened the learning model's cost. Prior to extracting the characteristics from tweets, punctuations.

3.2.8 Lowercase conversion

All the data from tweets will be changed into lowercase letters since it will help with preprocessing. A lower-

case conversion technique makes the entire remaining textual data insensitive, thereby allowing for 275 repetitions in successfully separating from and learning grammar sequence patterns.

3.2.9 Lemmatization Technique

A Lemmatization technique is implemented for the corpus of tweets. The data normalization is achieved through lemmatization. It converts the various inflected forms of a word into its root form, which means that it did not alter in meaning. By using this approach, the complexity of a learning model is reduced, and such an endeavour yields high performance in text categorization. For instance, in Twitter text, the word caring might adopt a more casual tone by being careful as it is an advanced form of knowledge with respect to training any model.

3.2.10 Tokenization

A method of tokenization is used to normalize tweets. AN approach of tokenization is used in creating a stream of words using words from all the various tweets, whereas lemmatization is applied to fix any choice like liked into that likes and made are fixed as make. In the majority of cases, knowledge about tweets can be determined from a word's base form. The final step in this process would, therefore, be tokenization – the conversion of preprocessed text representing tweets into clean tokens. When all extra noise has been eliminated from the spoken word, they are then tokenized and become a sentence. Next, the sentences are transformed into a token of words. Then, relevant to the sarcastic tweet, the text is introduced along with later words.

3.2.11 Parts-of-Speech Tagging

This method is generally utilized to relate a word to the structural class, which primarily assists in understanding how each one processes its role within the phrase. Nouns, verbs, adverbs, and adjectives represent the essentials of POS Tagging. In essence, the core of part-of-speech taggers is based on accepting a collection of words as an input and producing a list as output that contains tuples where each word has been assigned with needed tags. A sample dataset after preprocessing is shown in Figure. 4.

3.3 Feature Extraction

It is also a core part of the process of model building. The most significant techniques, such as NLP field, Bag of Word (BOW), and TFID Frequency, are used to represent texts using text data and construct a BERT model. These three techniques of feature extraction are being evaluated to compare the performance among these techniques [34]. Feature extraction, however, is characterized by several techniques that are used in this section.

3.3.1 Bag of Word

This technique is applied to text-type datasets [6]. Using this approach, the text-oriented data would be transformed into some numeric type of data that is understandable. This transformed data relies on how long a word appears in a particular case sentence. Each numbered figure in the dataset is a stand-in for word frequency count. In the feature map for the visual representation of words, an occurrence of a word is simply one or zero, where it is found in sentences. The BOW has a variety of limitations. For one, it uses lots of processing resources, not providing any information about the grammar image on sentences and word placement in text.

3.3.2 Term Frequency-Inverse Document Frequency (TFIDF)

This approach is applied to map tweets' textual data into an ML structure vector [33]. This method is a fast numerical statistic method primarily for determining whether there should be some incentive to include a record of any word into the set. Viewed in relation to BOW, the TFIDF method used for feature extraction also maintains the importance of each word. TF indicates the occurrence of a word in any given text. The IDF determines the necessity of every word in a book. The major problem with the TFIDF is that it fails to understand meaning in the context of a word within texts.

	tweets	class	clean_text
1662	Thank goodness #NCAA has handle on most pressi...	figurative	thank goodness ncaa handle pressing issues day...
7734	This is where Leonard hold up the #SARCASM sig...	sarcasm	leonard hold sarcasm sign bbt
1759	A man who killed many under his car is lovd. A...	figurative	man killed many car lovd man acquitted courts ...
3279	But did yall catch that "police" presenter for...	irony	yall catch police presenter best hip hop video...
2709	"Easy Like Heaven - The Cure..." This video is...	irony	easy like heaven cure video longer available d...

Figure 4. Sample dataset after preprocessing

3.3.3 Word Embedding

The idea of this technique is grounded after the failure of BOW and TFIDF methods of feature extraction as they could not define the sentence's context within any text they resorted to. Word Embedding refers to a set of feature-learning methods that are based on the process whereby words or phrases from the lexicon get mapped onto vectors whose components have actual values. This is because word embedding was utilized in the feature extraction stage of our study. Since the word embedding feature extraction technique is modified to prepare a vectorized textual dataset of our tweet for deep learning methods [4]. The following sentence is fully processed by the tweet and transformed into an integer sequence. Each integer in the series infers from each index of a dictionary token. Every binary-type token has a coefficient in the vector of sequence.

3.4 Splitting of the Dataset

In this case, a splitting ratio is kept at 80:20 by categorizing the tweet-based textual data into two halves. Machine learning and deep learning techniques are trained by it, and we use them on 80% of our research data. Testing is performed on 20% of the research data to check the implemented classifier's performance outcome. The data is being separated to prevent the applied model's overfitting issue. The data partitioning substantiates our real-time performance analysis findings. Now, to locate sarcasm in any writing materials, different models have been researched.

3.5 Selection of Model

To show the BERT's particular strengths in natural language understanding related to this study, we demonstrate the capability of a BERT (Bidirectional Encoder Representations from Transformers) model that has been used for sarcasm detection. However, a clear consensus has been observed across the literature that traditional models fall short of depicting contextually diverse subtleties, especially when language is particularly unclear and ambiguous on social media platforms like Twitter [34]. As a result of the thorough analysis presented in the literature, BERT was chosen for sarcastic tone identification. To begin with, the earlier version of deep learning known as Long Short Term Memory (LSTM) is outlined that it cannot grasp some arduous contextual dependencies occurring in sarcasm sentences and mainly those challenging dynamically ambiguous linguistic substances utilized on social media sites such as Twitter. This study points out that the advantages of the BERT model are shown to describe contextual language and bidirectional attention properties inside sarcastic wording [35].

Further, the study outlines a range of benefits upon the adoption of the BERT model, such as its ability to perform the different techniques of NLP and attain knowledge of the pre-trained dataset. While sarcasm detection on Twitter is based upon implicit context-sensitive areas, a BERT model has been applied to integrate such contextual information within the pre-trained embedding, which would enhance its quality. By empirically analyzing the literature review data, we can conclude that BERT representation is one of the most popular models because it outperforms other traditional approaches. In this particular case, BERT is relevant because it can identify the sparse data problems common to Twitter datasets, which contributes significantly to validating its applicability.

Basically, the literature review creates substantial grounds for why the BERT approach was selected since it represents one with notable features that prove to be able to respond in many different ways to sarcastic content. Despite the fact that BERT has shown state-of-the-art performance on multiple tasks, sarcasm detection is not one of its main areas of application. Thus, the first contribution of this study aims to develop a BERT-based analysis method for sarcasm identification from Twitter language. In order to analyze the relationship between different words in a document, BERT uses an approach that is based on the transformer. BERT is built to produce a language expression model. As a result, an encoder is required to utilize the token as its input. For the implementation, we used an official BERT tokenization script that is updated continually with all new improvements. Then, the segments, masks, and tokens are derived from the encoding process. All of these will correspond to one input layer.

3.5.1 BERT Model's Architecture

The BERT model manages the optimal representation of all feature sequences through meaningful pre-trained data. Figure. 5 below provides the BERT's network structure. The morphology of BERT network layouts is thought to be based on the transformer design. Let n be the tokens of the input series and so of its embedding vector's dimension. The input layer of the BERT model is a matrix, and it also has an output in another form, which is nothing but a kind of text. As a result, N BERT layers can easily be in sequence. Instead, the Transformer block contains 12 layers that are employed in the base model of BERT. This model employs mutual conditioning in contexts in order to pre-train a complex bidirectional representation based on unlabeled data.

For this reason, present approaches for many NLP tasks can be developed in just one additional output layer by fine-tuning the pre-trained BERT model. On a vast number of unlabeled texts, the chosen model is being pre-trained, which encompasses Wikipedia and BookCorpus– a popular model of NLP uses sentiment analysis on Twitter. A Bert-tokenizer of the BERT model with TensorFlow 2.0, a text classification can be carried out based on this language model. Epoch drop rate is followed by 0, 2, and 10, which causes the batch size to be 30 [35].

3.5.2 Long short-term memory (LSTM)

LSTM employs the approach of deep learning and is a neural network designed for classification problems. The LSTM model is usually applied to the data that has a long sequence, for instance, in the case of long comments. This model was able to learn or extract a series of long datasets. LSTMs were proposed to address the limitations of the RNN approach. The architecture of the linked LSTM network is comprised of neural networks as well as many blocks in memory. Combining the input, output, and forget gate, the sequential attribute learning over long periods is done using the LSTM method. Based on how the model's gates are controlled in cells of a neural network, data pattern flow could be regulated by LSTM [35]. LSTM employs the approach of deep learning and is a neural network designed for classification problems. The LSTM model is usually applied to the data that has a long sequence, for instance, in the case of long comments. This model was able to learn or extract a series of long datasets. LSTMs were proposed to address the limitations of the RNN approach. The architecture of the linked LSTM network is comprised of neural networks as well as many blocks in memory. Combining the input, output, and forget gate, the sequential attribute learning over long periods is done using the LSTM method. Based on how the model's gates are controlled in cells of a neural network, data pattern flow could be regulated by LSTM.

3.6 Evaluation Matrix

The accuracy, precision, and recall of the model used to detect sarcasm are evaluated using four main criteria. These metrics are crucial in measuring the performance of the model quantitatively for detecting sarcastic tweets within Twitter, a popular social networking platform. The mathematical state for these evaluation matrices is presented below: True Positive as TP, True Negative as TN, False Positive as FP, and False Negative as FN.

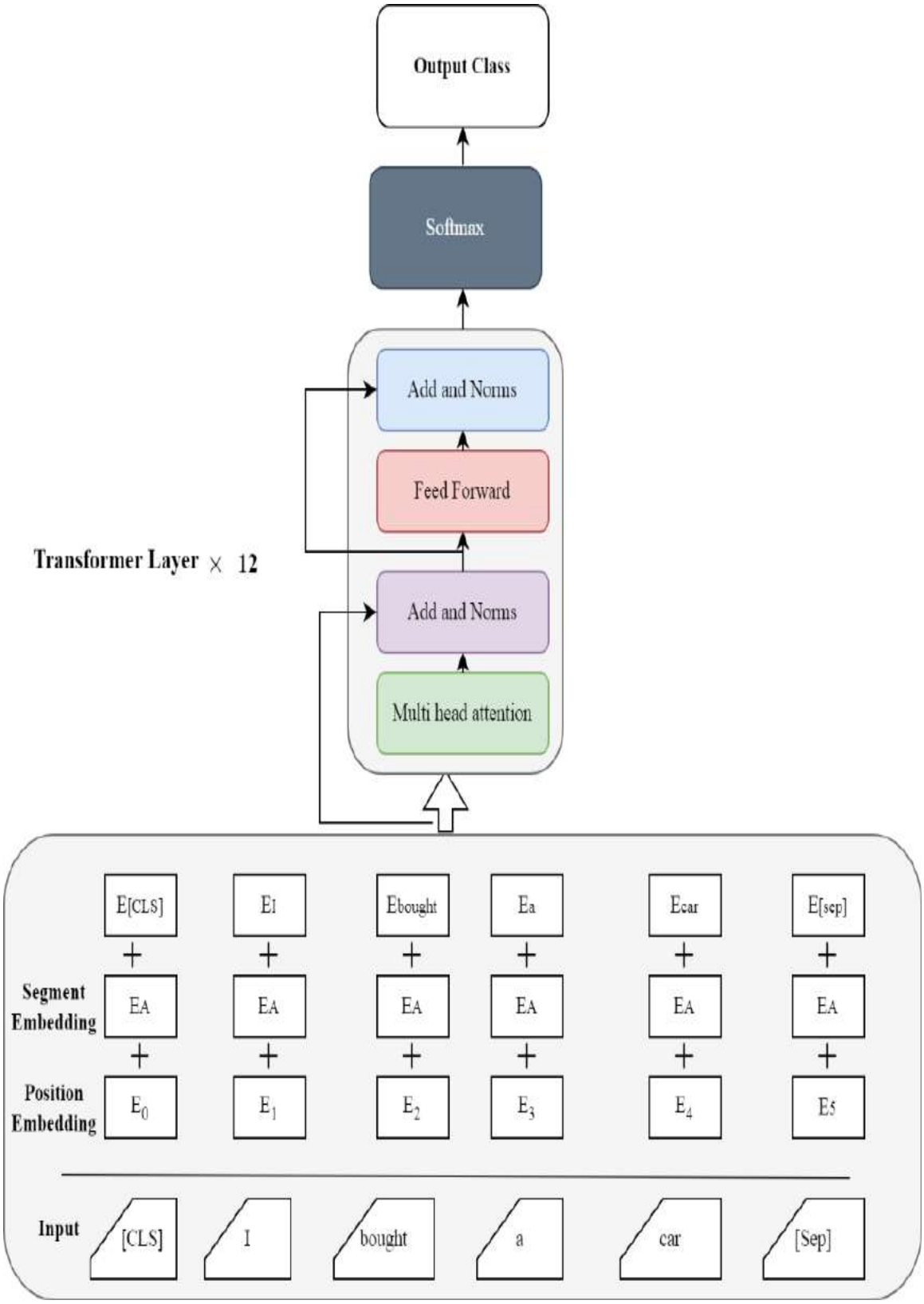

Figure 5. BERT Model Architecture

3.6.1 Accuracy

The measure of accuracy, a key evaluation metric, refers to the entire correctness defined by a BERT prediction model. The latter is denoted as dividing total instances by a ratio of TP and TN correctly detected sarcastic tweets. The accuracy matrix assesses the correct predictions throughout the entire instance. A high score of accuracy is ideal for the model as it reflects an excellent ability to identify sarcastic tweets from non-sarcastic ones.

$$Accuray = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{TP + TN}{TP + TN + FP + FN}\right)_i \quad (1)$$

3.6.2 Recall

The recall, also known as the sensitivity or the actual positivity rate, is a factor used to quantify the model's power to detect tweets that are sarcastic within multiple rows, which would be considered sarcastic in reality. It is defined as the amount of TP over its denominator, which consists of FN and TP.

$$Recall = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{TP}{TP + FN}\right)_i \quad (2)$$

The recall matrix, a significant aspect of our evaluation, is instrumental in estimating the capability of the model to capture all sarcastic instances. A high recall value signifies that the model can detect many actual sarcastic tweets, demonstrating the model's thoroughness in capturing all sarcastic instances.

3.6.3 Precision

This matrix, a key part of our evaluation, observes the model's correctness, particularly for identifying sarcasm tweets among cases that are marked as sarcastic. Precision is defined as the ratio of TP to TP+FP. A significant precision score means that the majority of tweets, which are classified as being sarcasm seasoned, were indeed sarcastic, demonstrating the model's accuracy in identifying sarcastic tweets.

$$Precision = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{TP}{TP + FP}\right)_i \quad (3)$$

When Precision says that the shared tweet is pointed to a sarcastic tone, it helps in understanding how accurate this model can be. A significant precision score means that the majority of tweets, which are classified as being sarcasm seasoned, were indeed sarcastic.

3.6.4 F1-Score

It is an example of the harmonic mean applied to the level of precision and recall in case they are almost the same as the case of true positives and false positives. F-score calculates an accurate measurement, which considers both cases of false positive and negative. This comes in especially handy when there is class imbalance.

$$F1 - score = \frac{1}{N}\sum_{i=1}^{N} 2 \times \left(\frac{recall \times precision}{recall + precision}\right)_i \quad (4)$$

4 Experimental Results

With the help of Twitter and Kaggle databases, a BERT model is assessed. Utilizing the % split technique, four BERT models were examined. In this case, 20% of the dataset is categorized for the testing of features and the rest of the 80% of the dataset is used for the training of the dataset [69]. According to scientific studies, datasets are most effective when used as follows: 70–80% for training the model and 20–30% for testing. By using this data-splitting strategy, the models are able to produce accurate findings without exaggerating their accuracy. Both the Tweeter and Kaggle datasets performed well during BERT model training using an average of 273 features as shown in Table. 1.

The introduced model attained flawless training accuracy for both datasets. Table 1 shows that the BERT model created for this study improves both the precision and cost of the testing and training samples. This proves that the paradigm is applicable to both databases. The results of LSTM model compared to BERT model are presented in Table. 2.

4.1 The Models Prediction Performance

This study makes use of two databases that include sarcastic expressions. We present the results of our evaluations of the installed models' predictive abilities. For the purpose of evaluating the models using the

Table 1. BERT Model's Results

	Dataset	F1-score	Precision	Recall	Accuracy
BERT Model	Kaggle	91.4%	91.4%	91.9%	92.0%
	Twitter	89.7%	89.5%	90%	90.0%

Table 2. LSTM Model's Results

	Dataset	F1-score	Precision	Recall	Accuracy
LSTM Model	Kaggle	84.9%	85.0%	85.0%	85.0%
	Twitter	88.0%	87.4%	87.5%	88.0%

F1-score matrices, accuracy, recall, and precision. Table. 3 display the classification reports for the Twitter and Kaggle datasets. It shows how much stronger the constructed model is compared to the existing baseline techniques.

A remarkable overall accuracy of 96% was achieved by the BERT model that was applied to the Kaggle database. Due to informational confusion, the baseline Twitter dataset used by the BERT model has low accuracy in detecting sarcasm. The model that was put into action, however, was able to more accurately anticipate all emotions by making use of discriminative and robust properties. See Table 3 for the sarcastic classes' accuracy and confusion matrices in the BERT model of the Kaggle database. In the appropriate rows of the confusion matrix, you can see the values that have been diagonally projected, as well as the inter-emotional confusion. The BERT model educated on the data set Kaggle yielded the following confusion matrices: sarcasm (90% accuracy), figurative (99% accuracy), regular (96% accuracy), and irony (90% accuracy). The Twitter dataset's confusion matrix reveals that irony achieved an accuracy level of 93%, sarcasm level of 99%, figurative level of 97%, and regular level of. Similarly, the LSTM model's confusion matrix shows that among the biggest accuracies for figurative, irony, sarcasm, and regular language were 96%, 93%, 99%, and 99%, respectively, while utilizing the Kaggle dataset. The confusion matrix of the Twitter dataset shows that sarcasm, figurative, regular, and irony each attained maximum accuracy levels of 92%, 93%, 99%, and 90%, respectively. For both datasets, a tiny, enhanced BERT model (about 2.75 to 2.80 MB) was sufficient for class forecasting. The implemented model's computing time ranges from 900 to 1300 seconds due to its unique features and straightforward transformer architecture.

4.2 Comparison with State-of-the-Art

By comparing the BERT model's performance with the current baseline methodologies utilizing Kaggle and Twitter databases, we were able to demonstrate the implemented model's resilience. An extensive synopsis of the comparison study. On rare occasions, however, the prediction rates of the proposed models for a particular class of sarcastic are only slightly lower than the baseline models. As an example, the model developed by BERT predicted a 90% accuracy rate for the Kaggle dataset, but the Kaggle technique in only managed a 94% accuracy rate for the sarcastic remarks.

In contrast to the baseline, the proposed model performed better, achieving a total preciseness of 91% when the baseline only managed 77%. Using Twitter data to accurately anticipate sarcasm, a lightweight BERT system with a low computing time is helpful for real-time applications. So, it is safe to claim that an

Table 3. Performance of BERT model on Kaggle and Twitter datasets

	F1-score		Precision		Recall		Accuracy	
	Kaggle	Twitter	Kaggle	Twitter	Kaggle	Twitter	Kaggle	Twitter
Figurative	86%	86%	85%	85%	75%	75%	90%	90%
Irony	89%	89%	84%	84%	74%	74%	99%	99%
Sarcasm	79%	79%	87%	87%	78%	78%	96%	96%
Regular	88%	88%	81%	81%	72%	72%	90%	90%

applied BERT method is more accurate, general, and reliable than the baseline approaches.

4.3 Comparative analysis with LSTM model

After using the data cleaning approaches, 273 noise characteristics were recovered from snarky comments, as shown in the results section. An analysis shows that a BERT model architecture achieved 97% accuracy on the Kaggle set and 92% accuracy on Twitter data, surpassing deep learning methods like LSTM. However, the LSTM model received 90% on the Kaggle dataset and 70% on the Twitter dataset. To sum up, the proposed BERT model outperformed the conventional deep learning methods by achieving greater weighted accuracy with two databases.

5 Conclusion

It is considered that sarcasm is hard for computers and people to understand. To prove the BERT model's efficacy, the study uses a large dataset of pre-trained values. Without further contextual data, such as parenting comments or prior user comments, we were able to automatically and effectively discern sarcastic sentences. Using LSTM architecture, this study sought to deduce how sarcasm relates to the emotional undercurrents in sarcastic text. By instantly training the task at hand on a linked activity, the BERT simulations can be made more efficient. It can find out if the statement's sentiment is more essential than the impact feelings have on the model's performance, especially on small-scale datasets for sarcasm identification.

Consequently, we achieved new state-of-the-art results in sarcasm identification by smashing three datasets. Models trained using BERT outperformed those trained using older methods for sarcasm detection by 11.53%. Models that use a writer's background as personality attributes as supplementary data perform worse than BERT models that rely solely on message content, according to previous research. This was an outstanding outcome, according to our assessment. Also, LSTM transferred learning (with sentiment as the intermediate job) may be able to produce even better results when the dataset size is limited for the target problem, which is sarcasm detection. We believe our models can lay a solid groundwork for future work that uses context-based data, such as user embedding, to enhance the models and achieve even more excellent state-of-the-art performance. Simultaneous integration of several intermediate tasks can improve the model's performance; however, domain-specific information must be preserved.

Author Contributions

Tayyaba Javed: Conceptualization, Methodology, Software, Writing- Original draft preparation **Muhammad Asif Nauman**: Data curation, Visualization, Investigation. **Rushna Zahid**: Software, Validation, Writing- Reviewing and Editing

Compliance with Ethical Standards

It is declare that all authors don't have any conflict of interest. It is also declare that this article does not contain any studies with human participants or animals performed by any of the authors. Furthermore, informed consent was obtained from all individual participants included in the study.

Funding Information

This research received no external funding.

References

[1] A. Baruah, K. Das, F. Barbhuiya and K. Dey, "Context-aware sarcasm detection using BERT," in Proceedings of the Second Workshop on Figurative Language Processing, pp. 83-87, 2020.

[2] H. Gregory et al., "A transformer approach to contextual sarcasm detection in twitter," in Proceedings of the second workshop on figurative language processing, pp. 270-275, 2020.

[3] X. Dong, C. Li and J. D. Choi, "Transformer-based context-aware sarcasm detection in conversation threads from social media," arXiv preprint arXiv:2005.11424, 2020.

[4] C. I. Eke, A. A. Norman and L. Shuib, "Context-based feature technique for sarcasm identification in benchmark datasets using deep learning and BERT model," IEEE Access, vol. 9, pp. 48501-48518, 2021.

[5] M. S. Jahan and M. Oussalah, "A systematic review of hate speech automatic detection using natural language processing," Neurocomputing, vol. 546, pp. 126232, 2023.

[6] A. Thakkar and K. Chaudhari, "Predicting stock trend using an integrated term frequency–inverse document frequency-based feature weight matrix with neural networks," Applied Soft Computing, vol. 96, pp. 106684, 2020.

[7] S. Khotijah, J. Tirtawangsa and A. A. Suryani, "Using lstm for context based approach of sarcasm detection in twitter," in Proceedings of the 11th international conference on advances in information technology, pp. 1-7, 2020.

[8] V. Sukhavasi and V. Dondeti, "Sarcasm detection using optimized bi-directional long short-term memory," Knowledge and Information Systems, pp. 1-29, 2024.

[9] A. Avvaru, S. Vobilisetty and R. Mamidi, "Detecting sarcasm in conversation context using transformer-based models," in Proceedings of the second workshop on figurative language processing, pp. 98-103, 2020.

[10] H. Srivastava, V. Varshney, S. Kumari and S. Srivastava, "A novel hierarchical BERT architecture for sarcasm detection," in Proceedings of the Second Workshop on Figurative Language Processing, pp. 93-97, 2020.

[11] E. Riloff et al., "Sarcasm as contrast between a positive sentiment and negative situation," in Proceedings of the 2013 conference on empirical methods in natural language processing, pp. 704-714, 2013.

[12] N. Chatterjee, T. Aggarwal and R. Maheshwari, "Sarcasm detection using deep learning-based techniques," Deep Learning-Based Approaches for Sentiment Analysis, pp. 237-258, 2020.

[13] M. Nirmala, A. H. Gandomi, M. R. Babu, L. D. Babu and R. Patan, "An Emoticon-Based Novel Sarcasm Pattern Detection Strategy to Identify Sarcasm in Microblogging Social Networks," IEEE Transactions on Computational Social Systems, 2023.

[14] A. Ray, S. Mishra, A. Nunna and P. Bhattacharyya, "A multimodal corpus for emotion recognition in sarcasm," arXiv preprint arXiv:2206.02119, 2022.

[15] J. Venskus, P. Treigys and J. Markevičiūtė, "Unsupervised marine vessel trajectory prediction using LSTM network and wild bootstrapping techniques," Nonlinear analysis: modelling and control., vol. 26, no. 4, pp. 718-737, 2021.

[16] R. González-Ibánez, S. Muresan and N. Wacholder, "Identifying sarcasm in twitter: a closer look," in Proceedings of the 49th annual meeting of the association for computational linguistics: human language technologies, pp. 581-586, 2011.

[17] U. Yavanoglu, T. Y. Ibisoglu and S. G. Wıcana, "Sarcasm detection algorithms," International Journal of Semantic Computing, vol. 12, no. 03, pp. 457-478, 2018.

[18] R. Jahangir et al., "Text-independent speaker identification through feature fusion and deep neural network," IEEE Access, vol. 8, pp. 32187-32202, 2020.

[19] R. Jahangir et al., "Speaker identification through artificial intelligence techniques: A comprehensive review and research challenges," Expert Systems with Applications, vol. 171, pp. 114591, 2021.

[20] A. Ghosh and T. Veale, "Fracking sarcasm using neural network," in Proceedings of the 7th workshop on computational approaches to subjectivity, sentiment and social media analysis, pp. 161-169, 2016.

[21] A. Joshi, P. Bhattacharyya and M. J. Carman, "Automatic sarcasm detection: A survey," ACM Computing Surveys (CSUR), vol. 50, no. 5, pp. 1-22, 2017.

[22] D. Davidov, O. Tsur and A. Rappoport, "Semi-supervised recognition of sarcasm in Twitter and Amazon," in Proceedings of the fourteenth conference on computational natural language learning, pp. 107-116, 2010.

[23] K. Nithya, P. D. Kalaivaani and R. Thangarajan, "An enhanced data mining model for text classification," in 2012 international conference on computing, communication and applications, pp. 1-4, 2012.

[24] C. I. Eke, A. A. Norman, L. Shuib and H. F. Nweke, "A survey of user profiling: State-of-the-art, challenges, and solutions," IEEE Access, vol. 7, pp. 144907-144924, 2019.

[25] M. Ghosh, R. Guha, R. Sarkar and A. Abraham, "A wrapper-filter feature selection technique based on ant colony optimization," Neural Computing and Applications, vol. 32, pp. 7839-7857, 2020.

[26] Y. Hong, B. Hou, H. Jiang and J. Zhang, "Machine learning and artificial neural network accelerated computational discoveries in materials science," Wiley Interdisciplinary Reviews: Computational Molecular Science, vol. 10, no. 3, pp. e1450, 2020.

[27] M. J. C. Samonte, C. J. T. Dollete, P. M. M. Capanas, M. L. C. Flores and C. B. Soriano, "Sentence-level sarcasm detection in English and Filipino tweets," in Proceedings of the 4th international conference on industrial and business engineering, pp. 181-186, 2018.

[28] M. Abulaish and A. Kamal, "Self-deprecating sarcasm detection: an amalgamation of rule-based and machine learning approach," in 2018 IEEE/WIC/ACM International Conference on Web Intelligence (WI), pp. 574-579, 2018.

[29] M. S. M. Suhaimin, M. H. A. Hijazi, R. Alfred and F. Coenen, "Mechanism for sarcasm detection and classification in malay social media," Advanced Science Letters, vol. 24, no. 2, pp. 1388-1392, 2018.

[30] D. Khurana, A. Koli, K. Khatter and S. Singh, "Natural language processing: state of the art, current trends and challenges," Multimedia tools and applications, vol. 82, no. 3, pp. 3713-3744, 2023.

[31] H. K. Kumar and B. Harish, "Sarcasm classification: a novel approach by using content based feature selection method," Procedia computer science, vol. 143, pp. 378-386, 2018.

[32] A. Giachanou and F. Crestani, "Like it or not: A survey of twitter sentiment analysis methods," ACM Computing Surveys (CSUR), vol. 49, no. 2, pp. 1-41, 2016.

[33] Y. Yunitasari, A. Musdholifah and A. K. Sari, "Sarcasm detection for sentiment analysis in Indonesian tweets," IJCCS (Indonesian Journal of Computing and Cybernetics Systems), vol. 13, no. 1, pp. 53-62, 2019.

[34] S. Minaee et al., "Deep learning–based text classification: a comprehensive review," ACM computing surveys (CSUR), vol. 54, no. 3, pp. 1-40, 2021.

[35] C. Shorten, T. M. Khoshgoftaar and B. Furht, "Text data augmentation for deep learning," Journal of big Data, vol. 8, no. 1, pp. 101, 2021.

[36] R. Jahangir, Y. W. Teh, F. Hanif and G. Mujtaba, "Deep learning approaches for speech emotion recognition: State of the art and research challenges," Multimedia Tools and Applications, vol. 80, no. 16, pp. 23745-23812, 2021.

[37] O. Saidani et al., "An efficient human activity recognition using hybrid features and transformer model," IEEE Access, vol. 11, pp. 101373-101386, 2023.

[38] A. A. Khan et al., "An efficient text-independent speaker identification using feature fusion and transformer model," Comput. Mater. Contin, vol. 75, no. 2, pp. 4085-4100, 2023.

[39] H. Y. Ghafoor et al., "Sensors-based human activity recognition using hybrid features and deep capsule network," IEEE Sensors Journal, 2024.