MULTI-FEATURE FUSION FRAMEWORK FOR AUTOMATIC SARCASM IDENTIFICATION IN TWITTER DATA

CHRISTOPHER IFEANYI EKE

FACULTY OF COMPUTER SCIENCE AND
INFORMATION TECHNOLOGY
UNIVERSITY OF MALAYA
KUALA LUMPUR

2021

MULTI-FEATURE FUSION FRAMEWORK FOR AUTOMATIC SARCASM IDENTIFICATION IN TWITTER DATA

EKE CHRISTOPHER IFEANYI

THESIS SUBMITTED IN FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

FACULTY OF COMPUTER SCIENCE AND INFORMATION TECHNOLOGY
UNIVERSITY OF MALAYA
KUALA LUMPUR

2021

UNIVERSITY OF MALAYA

ORIGINAL LITERARY WORK DECLARATION

Name of Candidate: Christopher Ifeanyi Eke

(Matric No: WVA170031

Name of Degree: Doctor of Philosophy

Title of Thesis: Multi-Feature Fusion Framework for Automatic Sarcasm Identification in Twitter Data

Field of Study: Information Systems

I do solemnly and sincerely declare that:

(1) I am the sole author/writer of this Work;
(2) This Work is original;
(3) Any use of any work in which copyright exists was done by way of fair dealing and for permitted purposes and any excerpt or extract from, or reference to or reproduction of any copyright work has been disclosed expressly and sufficiently and the title of the Work and its authorship have been acknowledged in this Work;
(4) I do not have any actual knowledge nor do I ought reasonably to know that the making of this work constitutes an infringement of any copyright work;
(5) I hereby assign all and every rights in the copyright to this Work to the University of Malaya ("UM"), who henceforth shall be owner of the copyright in this Work and that any reproduction or use in any form or by any means whatsoever is prohibited without the written consent of UM having been first had and obtained;
(6) I am fully aware that if in the course of making this Work I have infringed any copyright whether intentionally or otherwise, I may be subject to legal action or any other action as may be determined by UM.

Candidate's Signature Date: 03 October 2021

Subscribed and solemnly declared before,

Witness's Signature Date: 03 October 2021
Name:

Designation:

MULTI-FEATURE FUSION FRAMEWORK FOR AUTOMATIC SARCASM IDENTIFICATION IN TWITTER DATA

ABSTRACT

Recently, sentiment analysis in social network research has gained much recognition. The notion behind sentiment analysis is to determine the polarity of the emotion word in an expression. Analysis of people's sentiments is a process of identifying 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 organizations. For example, sarcasm is a type of sentiment where people express their negative emotions using positive words or intensified positive words in a text. In a sarcastic utterance, the expressed statement usually deflects the different meanings than their actual composition. Various feature engineering techniques such as Bag-of-words (BoWs), N-gram, and word embedding have been investigated to detect sarcasm in textual data automatically. However, the use of the features mentioned above results in the loss of contextual information due to the methods ignoring the context of words in the text. Furthermore, there are issues bothering on the sparsity of training data in sarcasm expression. This issue makes a feature vector for each sample constructed by BoW mostly null due to the microblog's word limit. Moreover, many deep learning methods in Natural Language Processing uses word embedding learning as a standard approach for feature vector representation. Nevertheless, one of the major drawbacks of word embedding is that it does not consider the sentiment polarity of the words. Consequently, words with opposite polarities are mapped into a close vector. To address the above-named problems and enhance the predictive performance in sarcasm identification, a Multi-Feature Fusion Framework for sarcasm identification is proposed using two classification stages. The first classification stage is constructed with a lexical feature only, extracted using the BoW technique and trained using five standard classifiers, including Support Vector

Machine, Decision Tree, K-Nearest Neighbor, Logistic Regression, and Random Forest to predict the sarcastic tendency based on the lexical feature. In stage two, the extracted lexical feature is fused with the length of microblog, hashtag, discourse markers, emoticons, syntactic, pragmatic, semantic (GloVe embedding), and sentiment related features to form a feature fusion and modelled using various classifiers, including Support Vector Machine, Decision Tree, K-Nearest Neighbor, Logistic Regression, and Random Forest. The developed Multi-feature framework effectiveness is tested with various experimental analysis, which was performed to obtain classifiers' performance. The evaluation shows that the constructed classification models based on the developed framework obtained results with the highest precision of 94.7% using a Random Forest classifier. Finally, the obtained results were compared with baseline approaches, and the proposed Multi-feature fusion framework attained the average detection precision between 11.2% - 27.1% compared to the baseline methods. The comparison outcomes show the significance of the proposed framework for sarcasm identification. Thus, the data sparsity issue can be resolved by selecting the discriminative features from the sparse training set before the modelling phase and bolstering the content-based feature with contextual information can enhance the predictive performance of sarcasm classification in textual data.

Keywords: Sarcasm Identification, Twitter, Machine learning, Feature fusion, Natural language processing.

KERANGKA MULTI-CIRI UNTUK PENGENALPASTIAN SINDIRAN AUTOMATIK DALAM DATA TWITTER

ABSTRAK

Baru-baru ini, analisis sentimen dalam penyelidikan rangkaian sosial telah mendapat banyak pengiktirafan. Konsep di sebalik analisis sentimen adalah untuk menentukan kekutuban kata emosi dalam suatu ekspresi. Analisis sentimen orang adalah proses mengenal pasti maklumat subjektif dalam dokumen sumber. Proses mengenal pasti pendapat orang (sentimen) mengenai produk, politik, perkhidmatan, atau individu membawa banyak faedah kepada organisasi. Sebagai contoh, sindiran adalah sejenis sentimen di mana orang meluahkan emosi negatif mereka menggunakan kata-kata positif atau kata-kata positif yang diperhebatkan dalam teks. Dalam ucapan sindiran, pernyataan yang dinyatakan biasanya mengalihkan makna yang berbeza daripada komposisi sebenarnya. Pelbagai teknik teknik ciri seperti Bag-of-word (BoWs), N-gram, dan *embedding word* telah diselidiki untuk mengesan sindiran dalam data teks secara automatik. Namun, penggunaan ciri-ciri yang disebutkan di atas mengakibatkan kehilangan maklumat kontekstual kerana kaedah mengabaikan konteks kata dalam teks. Tambahan pula, ada masalah yang mengganggu kelangkaan data latihan dalam ekspresi sindiran. Isu ini menjadikan vektor ciri untuk setiap sampel yang dibina oleh BoW kebanyakannya batal kerana had perkataan microblog. Lebih-lebih lagi, banyak kaedah pembelajaran mendalam dalam Pemprosesan Bahasa Asli menggunakan pembelajaran penyisipan kata sebagai pendekatan standard untuk perwakilan vektor ciri. Walaupun begitu, salah satu kelemahan utama penyisipan kata adalah bahawa ia tidak menganggap polaritas sentimen kata-kata. Oleh itu, kata-kata dengan kutub bertentangan dipetakan menjadi vektor dekat. Untuk mengatasi masalah yang disebutkan di atas dan meningkatkan prestasi ramalan dalam pengenalpastian sindiran, *Multi-Feature Fusion Framework* untuk pengenalan sindiran diusulkan menggunakan dua tahap klasifikasi

dicadangkan. Tahap klasifikasi pertama dibina dengan ciri leksikal sahaja, diekstraks menggunakan teknik BoW dan dilatih menggunakan lima pengklasifikasi standard, termasuk Mesin Vektor Sokongan, Pohon Keputusan, Jiran K-Terdekat, Regresi Logistik, dan Hutan Rawak untuk meramalkan kecenderungan sarkastik berdasarkan ciri leksikal. Pada tahap kedua, ciri leksikal yang diekstrak disatukan dengan panjang mikroblog, hashtag, penanda wacana, emotikon, sintaksis, pragmatik, semantik (penyematan GloVe), dan ciri-ciri yang berkaitan dengan sentimen untuk membentuk gabungan ciri dan dimodelkan menggunakan pelbagai pengklasifikasi, termasuk Sokongan Mesin Vektor, Pohon Keputusan, Jiran terdekat-K, Regresi Logistik, dan Hutan Rawak. Keberkesanan kerangka pelbagai ciri yang dikembangkan diuji dengan pelbagai analisis eksperimental, yang dilakukan untuk mendapatkan prestasi pengklasifikasi. Penilaian menunjukkan bahawa model klasifikasi yang dibina berdasarkan kerangka yang dikembangkan memperoleh hasil dengan ketepatan tertinggi 94.7% menggunakan pengkelasan Hutan Rawak. Akhirnya, hasil yang diperoleh dibandingkan dengan pendekatan garis dasar, dan *Multi-Feature Fusion Framework* yang dicadangkan mencapai ketepatan pengesanan rata-rata antara 11.2% - 27.1% berbanding dengan kaedah garis dasar. Hasil perbandingan menunjukkan kepentingan kerangka kerja yang dicadangkan untuk pengenalan sindiran. Oleh itu, masalah sparsiti data dapat diselesaikan dengan memilih ciri diskriminatif dari set latihan jarang sebelum fasa pemodelan dan meningkatkan ciri berdasarkan kandungan dengan maklumat kontekstual dapat meningkatkan prestasi ramalan klasifikasi sindiran dalam data teks.

Kata kunci: Pengenalan Sindiran, Twitter, Pembelajaran mesin, Ciri fusion, Pemprosesan bahasa semula jadi.

ACKNOWLEDGMENTS

I would like to start by expressing my profound gratitude to Almighty God for His loving kindness, protection, provision, and sound health throughout this period of PhD training. Without His mercy, this study would not have been possible. Therefore, I said, May your name be glorified forever and ever, Amen.

Next, I want to extend my special thanks to my distinguished supervisors Dr. Azah Anir Norman and Ass. Prof. Dr. Liyana Shuib, for your immense contributions that resulted in the realization of all the works that made up of this thesis. Both of you made me learn a lot by tapping through your academic knowledge and leadership experiences. I cannot forget your kindness, time, care, and unending support whenever I call for help. Furthermore, I wish to extend my thanks to Dr. Henry Nweke and Dr. Mohammed AL-Garadi for your assistant and encouragement throughout this program. Finally, I want to thank my employer, Federal University Lafia, Nasarawa State Nigeria, for the opportunity to further my education in one of the high-ranked Universities in the world for their financial support towards realizing this thesis.

Similarly, I want to express my special thanks and appreciation to my caring and loving wife, Kanayo Precious Eke, for her relentless prayers and emotional support to see that this PhD journey comes to an end. Despite my absence, all the family affairs have been moving on smoothly. I also wish to express my thanks to my two children, Chidinma Abigail Eke and Okwukwe Deborah Eke, for their unending checking and questioning when Daddy will return home. Your endurance helped in the realization and completion of this program. Furthermore, my special thanks go to the entire family of Late Chief Anthony Eke for your prayers and support. Your constant calls, prayers, and support went a long way in the completion of this program. Finally, I wish to extend my special thanks to the entire member of the Deeper Christian Life Ministry, especially Kuala Lumpur's location, for their relentless prayers and support throughout this program.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS AND ABBREVIATIONS

AB	:	Adaboost
ANN	:	Artificial Neural Networks
API	:	Application programming interface
ARFF	:	Attribute-Relation File Format
ARTK	:	Automatic retrieval of tweets using keywords
AUC	:	Area under the curve
Bi-LSTM	:	Bi-directional long short term memory
BN	:	Bayesian Network
BOW	:	Bag-of-words
BR	:	Binary Representation
BW	:	Balanced Winnow
CBOW	:	Continuous bag-of-words
CI	:	Content information
CNN	:	Convolutional Neural network
CPU	:	Central processing unit
CLS	:	Classification token
CSV	:	Comma-separated values
CUE-CNN	:	Convolutional user embedding Convolutional Neural Network
DNN	:	Deep neural network
DT	:	Decision tree
DTM	:	Document Term matrix
FC	:	Fuzzy clustering
MFF	:	Master feature fusion
F-M	:	F-Measure

FN	:	False negative
FP	:	False positive
GB	:	Giga byte
GFI	:	Grammatical function information
GloVe	:	Global Vectors
IG	:	Information Gain
k-NN	:	k-Nearest Neighbours
KS	:	Kappa statistics
LIWC	:	Linguistic inquiry and word count
LR	:	Logistic regression
LSTM	:	Long short term memory
ME	:	Maximum entropy
MI	:	Mutual information
NB	:	Naïve Bayes
NLP	:	Natural Language processing
NLTK	:	Natural language tool kits
POS	:	Part of speech tagging
PRE	:	Precision
RB	:	Rule base
REC	:	Recall
RF	:	Random Forest
RNN	:	Recurrent Neural Network
RQ	:	Research question
SLR	:	Systematic literature review
SMO	:	Sequential minimal optimization
SVM	:	Support vector machine

TAALES	:	Tools for Analysis of Lexical sophistication
TF	:	Term frequency
TFIDF	:	Term Frequency with Inverse Document Frequency
TN	:	True negative
TP	:	True positive
TPR	:	True positive rate
URL	:	Universal resource locator
VSF	:	Visual semantic feature
WEKA	:	Waikato environment for knowledge analysis
Word2Vec	:	Word to Vector

LIST OF APPENDICES

CHAPTER 1: INTRODUCTION

This Chapter discussed the general introduction of the thesis. The Chapter is structured into ten (10) Sections. Section 1.1 gives the study's introduction, Section 1.2 presents the research motivation, Section 1.3 provides the problem statement, Section 1.4 presents the research aim, Section 1.5 gives the research objectives, and Section 1.6 provides the research questions. Moreover, Section 1.7 provides the research significance. Finally, Section 1.8 presents the research contributions, Section 1.9 provides the thesis organization, while Section 1.10 summarizes the Chapter.

1.1 Background

Social media website has become a platform and forum where users express emotions and opinions in diverse subjects such as politics, events, individuals, products, dialogue systems, review ranking, and summarization (Bharti et al., 2016; Sundararaj et al., 2021). It has also become a popular platform for global interaction and idea discussion among users. People on social media share and publish messages, thereby making their personal information globally available. Identification of subjective information of people like people's opinions, emotions, and sentiments are made possible by such information. Analysis of people's sentiment (also referred to as opinion mining) identifies subjective information in source documents. The possibility of identifying subjective information is essential and helps in the generation of structured knowledge that serves as a piece of important knowledge for decision support systems and individual decision-making (Fersini et al., 2014; Zhang et al., 2021). The process of identifying people's opinions (sentiments) about products, policies, services, or individuals brings a lot of benefits to the organizations (Wang et al., 2014; Vyas & Uma, 2019).

Many firms have realized the necessity of analyzing social media data to get the customers' emotions regarding their products, which will, in turn, increase the quality of

their products. The subjective and emotional language often requires a specific context to comprehend the meaning of what the user is discussing. 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 (Walker et al., 2012). Automatic sarcasm identification is one of the major issues in Natural Language Processing (Onan, 2017).

According to the Cambridge English dictionary, Sarcasm is defined as 'the use of remarks that mean the opposite of what one says, made to hurt someone's feelings or criticize something in a humorous way' (Dictionary, 2008). Similarly, the Macmillan English dictionary defines Sarcasm as 'the use of remarks in saying or writing the reverse of one's motive to hurt someone's perception' (Dictionary & Rundell, 2007). Accordingly, various authors have defined sarcasm in terms of NLP approaches. For instance, Yavanoglu et al. (2018) defined sarcasm identification as an activity of using NLP techniques to classify a word or sentence sequence that possesses sarcasm attributes and properties. They also referred to it as the system that learns and distinguishes between normal sentences and sarcasm within the semantic level. Moreover, Bharti et al. (2016) defined sarcasm as a sentiment where people express their negative emotion using positive words or intensified positive words in a text. In sarcasm sentiment, the negative emotion of people is communicated using a positive term in the text to reveal their sarcasm.

Sarcastic utterance represents a conflict between an individual's motive for making the utterance and the actual composition. For instance, the sarcastic expression "I love to work on holidays!" shows a conflict between the clear statement "on holidays" and the articulation "love". The contradiction and the sentiment polarities shift proves that sarcasm is a unique form 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 a text by determining the obscurity of the meaning and improving the overall sentiment classification of a large volume of user's textual data obtained from social media. However, the insufficient knowledge of the situation "Context", the environment, and the specific topic will result in difficulty detecting sarcastic utterances (Karuna & Reddy, 2020). Context understanding is one of the main challenging phases of moderation content. The term "Context" in sentiment analysis refers to supplementary support that may increase or change the content polarity. However, the sentiment classification's predictive performance will rely on context vector and learning algorithms to guarantee the reliability of the overall sarcasm classification.

Sarcasm classification can be performed using various approaches such as machine learning, lexicon, and hybrid approaches. However, the most applied approach is the machine learning approach, which deals with the creation of predictive models using an intelligent method. In the machine learning approach, there are five processes, which include the dataset collection, data preprocessing, feature extraction (also referred to as attributes extraction from the data), construction of the classification model, and evaluation of the constructed classification model (Kumar & Harish, 2018). The sarcasm dataset consists of both sarcasm and non-sarcasm expressions. On the other hand, features are the unique words or phrases, also referred to as characteristics or attributes found in the sarcasm expressions that helps in distinguishing sarcasm utterances from non-sarcasm utterances.

The main objective of sarcasm identification in a sentence is sentiment classification. Thus, the machine-learning model is often employed for sarcasm identification due to its durability and competence to observe itself in conformity with the datasets and

specifications. There are various areas that sarcasm identification has played critical roles. For instance, a sarcasm identification experiment enhances the research on sentiment analysis. In this case, emotion features serve as a bedrock for sentiment polarity identification and opinion mining classification. In addition, sarcasm identification enables companies to analyze customers' feelings regarding their products, which could improve the quality of their products (Saha et al., 2017). It is also helpful in reducing the wrong categorization of consumer's opinions towards issues, products, and services (Mukherjee & Bala, 2017b). Moreover, sarcasm identification is useful in dialogue, system review ranking, and summarization in human-computer interaction application domains (Davidov et al., 2010). Automatic identification of sarcasm has not been widely studied (González-Ibánez et al., 2011; Onan, 2017).

Previous studies have attempted to identify sarcasm in a tweet by employing various feature engineering approaches (Zhang et al., 2010; da Silva et al., 2014; Prasad et al., 2017; Jain et al., 2020). For instance, Mukherjee and Bala (2017b) employed content-based features, in which the study generally relied on the sentence to differentiate sarcastic from the non-sarcastic statement. The technique produced a reasonable performance based on the data set that was used. However, the predictive model performance relied deeply on the content-based feature, which is likely to degrade when applied to other data sets due to its dependence on word use. Hence, the obtained result is not generalized to a satisfactory extent. The literature on sarcasm detection reveals that the existing methods suffer two main problems (Al-Sallab et al., 2017; Prasad et al., 2017; Xiao et al., 2018; Jia et al., 2019). One, the BoWs technique ignores the context of the words in representation in the sentence since it is only concerned with the occurrence of the word and not where and how it is placed in the sentence (Khodak et al., 2017). In other words, different sentences can have the same vector representation, which leads to loss of contextual information and, in turn, the semantic information in the expression:

two, the sparsity of the training data (Hazarika et al., 2018). Considering the limitation on the number of words in the microblog, the value of the feature vector for each sample constructed by BoWs produces a null feature, making the training data sparse. Three, various deep learning methods in NLP uses word embedding learning as a standard approach for feature vector representation. However, one of the major drawbacks of word embedding is that it ignores the sentiment polarity of the words (Araque et al., 2017; Giatsoglou et al., 2017). Consequently, words with opposite polarities are mapped into a close vector.

Therefore, it is important to explore more approaches to overcome these drawbacks. This thesis addresses the problem mentioned above by proposing a multi-feature fusion framework for sarcasm identification in Twitter data.

1.2 Research Motivation

It is challenging to work with social media texts such as blogs, microblogs, etc. The presence of sarcasm has significantly multiplied, and identifying these occurrences is naturally hard for humans. There is no definite pattern in constructing a sarcastic expression. Since the sarcastic expression is popular in English, it is essential to automatically identify it in an expression. The main goal of the sarcasm identification task is to realize some discriminative features that will help differentiate between the sarcastic and non-sarcastic utterances.

Previous studies have proposed various feature engineerings approach such as the N-gram, Bags-of-word and word embedding for sarcasm identification in social media (Zhang et al., 2010; da Silva et al., 2014; Prasad et al., 2017; Jain et al., 2020). For instance, Dave and Desai (2016) experimented with traditional BoWs techniques to extract features in their study of sarcasm detection on textual data. They employed a Support vector machine classifier to train the model and attained an accuracy of 50%.

However, the predictive performance result revealed that the traditional bag-of-words model is inadequate to extract the discriminative features for sarcasm identification. The brain behind the low performance is that it ignores the context of the word in sarcastic expression, coupled with the hashtags, jargon and emoticons that surround social media data (Prasad et al., 2017).

In another study, Mukherjee and Bala (2017a) experimented on N-gram features that rely on word use and sentence in general in identifying sarcastic and non-sarcastic words in a sentence, leading to the dependence of the algorithm performance on the content-based features, which will degrade when applied to other. Microblog data contains highly contextual information. As a result, the application of content-based features in sentiment classification becomes relatively ineffective and requires some contextual clues (Carvalho, Sarmento, Silva, & Oliveira, 2009). Besides, the content-based features (González-Ibánez et al., 2011) that consider tweets' contents only lead to the loss of contextual information and the semantics or meaning of words in the expression (Khodak et al., 2017; Xiao et al., 2018).

Another issue in the existing studies is the sparsity of training data. Due to the word limit of microblog, it makes the value of feature vector for each sample constructed by BoW feature engineering technique produces a null feature, thus making the modelling data-sparse (Hazarika et al., 2018; Jia et al., 2019). In another study, Joshi et al. (2016) investigated features based on word embedding similarity for sarcasm identification. The feature used in their study was enhanced with the most congruent and incongruent word pair, which improved the performance. However, word embedding based features are not adequate in capturing all the sarcastic sentiment in a sarcasm expression because the word embedding technique ignores the sentiment polarity of words (Araque et al., 2017; Giatsoglou et al., 2017). Consequently, words with opposite polarity are mapped into

close vectors. Furthermore, to find the solution to the problems mentioned above, most studies in linguistic concepts related to sarcasm maintain that employing contextual features that consider tweet context enhances predictive performance (Wallace et al., 2014; Zhang et al., 2021).. A study conducted by Wallace (2015) investigated this fact by indicating the failure of traditional classifiers in a situation wherein human requires additional context. Thus, the opportunity for open research abounds for bolstering content-based features with contextual features to enhance predictive performance.

Therefore, an effective framework for sarcasm identification must be developed to capture the sentiment polarity, contextual information and addresses the sparsity of training data in classifying sarcastic utterances in sarcastic or non-sarcastic to enhance the predictive performance of the sarcasm detection model.

1.3 Problem Statement

Feature engineering in modelling is the hardest and most vital aspect of classification, and it usually determines the success or failure of a model. Previous studies have proposed various feature engineerings techniques, such as the N-gram technique, BoW techniques, and word embedding, to extract diverse features for sarcasm identification in social media (Zhang et al., 2010; da Silva et al., 2014; Prasad et al., 2017). For instance, Mukherjee and Bala (2017b) extracted content-based features. The study relied solely on the emoticon, word use, and generally in the sentence to differentiate sarcastic from non-sarcastic in a sentence. The technique produced a reasonable performance based on the data set that was used. However, the predictive model performance relied deeply on the content-based feature and ignored the contextual information on the sarcastic expression. Hence, the obtained result is not generalized to a satisfactory extent. In the related study, Dave and Desai (2016) experimented with traditional BoW techniques to extract features for sarcasm detection on textual data. They employed a support vector machine classifier

to train the model and attained an accuracy of 50%. The predictive performance result revealed that the traditional bag-of-words model is inadequate to extract the discriminative features for sarcasm identification. The brain behind the low performance is that it ignores the context and word order in sarcastic expression (Prasad et al., 2017). On the order hand, other variations and extensions of word2vec feature engineering techniques such as continuous bag-of-words (CBoW) (Ghosh et al., 2015) and skip-gram (Mikolov et al., 2013) have also been studied for sarcasm identification tasks. These techniques were able to capture some word dependency and word sequence.

Even though few studies have implemented conventional text classification-based feature engineering methods for sarcasm detection, literature studies (Al-Sallab et al., 2017; Prasad et al., 2017; Xiao et al., 2018; Jia et al., 2019) reveals that most current methods face various issues that need to be resolved to improve sarcasm identification framework. This include; one, the context of the words are ignored in representation in the sentence since it is only concerned with the occurrence of the word. This leads to loss of contextual information and, in turn, the semantic information in the expression (Khodak et al., 2017; Xiao et al., 2018). Two, the sparsity of training data issue (Hazarika et al., 2018). Due to the word limit of the microblog, the value of the feature vector for each sample constructed by BoW produces null features, which makes the modelling data sparse (Hazarika et al., 2018; Jia et al., 2019). Three, many deep learning methods in NLP uses word embedding learning as a standard approach for feature vector representation. However, one of the major drawbacks of word embedding is that it ignores the sentiment polarity of the words (Araque et al., 2017; Giatsoglou et al., 2017). Consequently, words with opposite polarities are mapped into a close vector.

Therefore, it is important to explore more methods to overcome these drawbacks and enhance predictive performance in sarcasm classification. Furthermore, even though the

current technique may have produced promising results for Twitter data with 140-word character tweets, as Twitter has extended the word usage from 140 to 280, this approach is no longer effective for Twitter data. Thus, there is room for improvement for larger databases (Joshi et al., 2017). Hence, there is a need to carry out this research.

1.4 Research Aim and Objectives

This research aims to investigate the possibility of identifying sarcastic expressions from Twitter data using a Multi-feature fusion framework. The proposed framework aimed to overcome the limitations identified in the related literature (see Section 1.3) on sarcasm identification. To achieve the goal of this study, the following research objectives are formulated.

1. To investigate the existing feature engineering and fusion approaches for sarcasm identification in Twitter data.
2. To develop a Multi-feature Fusion Framework for sarcasm identification to improve the performance address the context of words, sentiment polarity and training data sparsity issues in sarcasm expression.
3. To evaluate the performance of the proposed Multi-feature fusion Framework using the real-world datasets by evaluating the performance with the baseline methods for sarcasm classification.

1.5 Research Questions

To realize the aforementioned research objective, the following research questions (RQs) has been formulated.

Research objective 1: To investigate the existing feature engineering and fusion approaches for sarcasm identification in Twitter data.

RQ1: What are the existing feature engineering and fusion approaches employed for sarcasm identification in Twitter data?

RQ2: What are the shortcomings in the current feature engineering approach for sarcasm identification in Twitter data?

Research objective 2: To develop *a Multi-feature Fusion Framework* for sarcasm identification to improve the performance and address the context of words, sentiment polarity and training data sparsity issues in sarcasm expression.

RQ3: *What are the most useful features for sarcasm identification by researchers?*

RQ4: *How can the loss of contextual information be mitigated through the development of a Multi-feature Fusion Framework?*

RQ5: *How can the sparsity of the training data (Null features) be resolved through the development of a Multi-feature Fusion Framework?*

***RQ6**: How can the sentiment polarity of words be captured through the development of the Multi-feature Fusion Framework?*

Research objective 3: To evaluate the performance of the proposed Multi-feature fusion Framework using the real-world datasets by evaluating the performance with the baseline approaches for sarcasm classification.

RQ7: *What are the existing performance measures appropriate for evaluating the proposed Multi-feature fusion framework for sarcasm identification in Twitter data, and how much can the proposed framework's performance results be enhanced compared with the performance of the baseline methods?*

1.6 Research Scopes

The scope of this study is listed below.

1. The Twitter dataset only has been considered in this research.
2. The study is limited to the fusion of the sarcasm identification at the feature-level fusion only.
3. The study utilized only Twitter datasets composed in the English language for sarcasm identification.

1.7 Research Contributions

The following contributions of this study for sarcasm identification research domain and body of knowledge are listed below:

1. **Literature analysis:** the review of the literature performed revealed the drawbacks inherent in the existing method for sarcasm identification. An extensive analysis and critical review of sarcasm identification on textual data were explored in five aspects: the datasets, preprocessing techniques, feature engineering techniques, the modelling approach, and performance metrics. In addition, the review identified the recent open research direction to tackle issues in the sarcasm identification domain.
2. **Proposed features for sarcasm identification in Twitter Data:** The study proposes and extracts various sets of features that consist of lexical, length of microblog, hashtag, discourse markers, emoticon, syntactic, pragmatic, semantic (GloVe embedding), and sentiment related features which are selected based on observations from the characteristics of the data and evidence from the literature. The observation has been transferred to suitable features, which are now experimented with to enhance the performance of the classifiers. As the main contribution to the body of knowledge, the study identified the most substantial feature and applied them as inputs to various machine learning algorithms for sarcasm detection with promising results.

3. **Proposed Algorithms:** A feature extraction, an algorithm to extract the discriminative features, and two stages classification algorithm by considering the lexical feature in the first stage and fused features in the second stage for sarcasm identification are proposed.
4. **Multi-feature Fusion Framework:** The study developed *a Multi-feature Fusion Framework* for sarcasm identification, and results are obtained by employing various classifiers (Support Vector Machine, Logistic Regression, K-Nearest Neighbour, Decision Tree, and Random Forest). The predictive performance of the modelling showed that the developed framework can further enhance the performance of sarcasm identification and addresses the training data sparsity issue in sarcasm expression.

All proposed models in this thesis have produced research outputs that have been published in high-ranked journals and conferences. Thus, the research outputs lists are shown in Appendix A (see page 172).

1.8 Research Significance

In this Section, the significance of this study is described. This research is significant and beneficial to both organizations and the research community.

In an organization, it is observed that various companies are finding it challenging to analyze the opinion of their customers to know their sentiment about the items they purchase. Thus, developing a multi-feature fusion framework for sarcasm detection could help the company analyze customers' feelings about their products and improve the quality of their products. Sarcasm detection helps the company to analyze customers' feelings about their products and improve the quality of their products. Also, it is helpful in the reduction of incorrect classification of customer sentiment towards issues, products,

and services. It enhances sentiment analysis and product recommendations for users, which will, in turn, help businesses attract new customers.

In the research community, sarcasm identification can bring a lot of benefits to various natural language processing applications such as opinion mining, marketing research, and information categorization. Furthermore, in the human-computer interaction application domain, sarcasm identification is applicable in dialogue, summarization, and review ranking. In the research community, it can resolve issues related to the sentiment polarity of words in sarcasm expression and the ability to resolve issues related to data sparsity. It reduces the reliance on content-based features in sarcasm classification. Therefore, the proposed framework is significant and beneficial to both organizations and the research community.

1.9 Thesis Structure

The organization of the rest of this thesis is given below.

Chapter 2: This Chapter provides a concise summary of the research's domain and the sarcasm detection task approaches. It also describes the process involved in the data collection and the types of data datasets employed in sarcasm classification. Moreover, this Chapter also reviews the related work on sarcasm detection, focusing on the feature engineering techniques employed to extract discriminative features for sarcasm classification. Also, this Chapter discusses the review of the classification algorithm employed for classifying tweets as sarcastic or non-sarcastic. Furthermore, the review of performance metrics employed to evaluate the performance of the classification algorithm is also discussed. Finally, in this Chapter, some shortcomings in this study domain are also examined.

Chapter 3: This Chapter discusses the methodology employed in this research for developing the proposed approaches for sarcasm identification in Twitter data. Moreover, this Chapter discusses the dataset employed in sarcasm identification experiments. It further explained various data pre-processing steps on the acquired dataset to prepare data before the feature extraction stage to eliminate the noisy data. Moreover, a concise description of the proposed multi-feature fusion framework, the construction of the classification model, and the performance measure for measuring the model performance were provided.

Chapter 4: This Chapter presents the entire proposed Multi-feature fusion framework for sarcasm identification in the Twitter dataset. In addition, it discusses the proposed set of features employed to develop the multi-feature fusion and how the features were extracted from the dataset. Furthermore, it discusses the experimental settings and procedures employed to carry out all the experimental tasks.

Chapter 5: This Chapter provides the experimental results and the discussion of the proposed feature engineering techniques by discussing the results obtained on the proposed Multi-feature framework. In addition, it describes baseline approaches that were used as benchmark studies related to this study domain. Lastly, an evaluation of the performance of our proposed framework with that of the baseline methods was described.

Chapter 6: This Chapter brings the thesis to a conclusion by re-examining the research objectives and research question. It also summarizes the main contributions of this study, the research limitations identified in the studies, and proposes further research directions in the domain.

1.10 Chapter Summary

This Chapter discusses the background and motivation for conducting this research. It also defined the problem this study seeks to address. Also, the research aims and objectives with their corresponding research questions were outlined. Besides, a concise summary of the research contributions and the significance were provided.

Next to this Chapter, a detailed review of literature on sarcasm identification on textual data is provided.

CHAPTER 2: LITERATURE REVIEW

2.1 Introduction

Opinion mining and figurative language detection in social media is a wide area of research. Various studies have been conducted in recent times on sarcasm identification using the Twitter dataset. This Chapter presents a literature review of the existing studies on sarcasm identification in Twitter data. The Chapter is structured into eight (8) Sections. Section 2.1 introduces the Chapter. In Section 2.2, a concise description of the Twitter microblogging service is presented, which explains its usage as a source for corpus generation in this study. Section 2.3 summarizes sarcasm identification, different forms of Sarcasm, and sarcasm identification techniques such as the NLP toolkit and machine learning model. The Section also stated why it is hard to identify in text data and its advantage if effectively identified. In Section 2.4, sarcasm identification approaches are described. Section 2.5 provided the text classification process for sarcasm classification. In Section 2.6, an information fusion is described. Section 2.7 provides a review of sarcasm identification using text classification techniques. Section 2.8 provides the research issues in the existing sarcasm identification studies. Lastly, Section 2.9 provided a summary of the Chapter.

2.2 Twitter Microblog Service

The notion behind Twitter is very simple. Twitter is a microblogging site, one of the biggest online social media outlets that publish over 500 million posts per day (Davoudi et al., 2020). The current Twitter bio-data, which consist of the user's full name, education, occupation, location, short biography, and the number of tweets, tells more about the users, such as their interest, what they engage in, where they live (location) and their self-conception (Chen et al., 2016). A Twitter user can broadcast a message (often referred to as a tweet) to any individual (known as a follower) who is ready to give heed

to them. The tweet could also have other contents such as a uniform resource locator (URLs), mentions, and hashtags in addition to the text strings.

A tweet that starts with the 'RT' initial is referred to as a retweet, which is a reply of a tweet to another tweet user. A hashtag (#) is a special character often employ by Twitter users to tag their tweets. A Twitter hashtag is a string preceded by the hash symbol, which can be viewed as a topic marker (i.e., for topic grouping of tweet) or the key context expression of the tweet. Thus, users who discuss similar topics use the hashtag (Tsur & Rappoport, 2012). For example #referendum, #Tycoon etc. A mention is used to refer to another Twitter user. It uses an ampersand symbol (@) to direct its message to a specific user, which provides metadata content. A tweet that began with a mention is regarded as a response to another tweeter.

Initially, the Twitter platform allows 140 characters length of a tweet, comprising hashtags, URLs, and mentions without a limitation on the lexical order in the message. However, URLs that exceed a specified length will consequently be reduced. Interestingly, Twitter has now increased the length of its message from 140 characters to 280 characters. A complete explanation of tweet functions can be located in the Twitter documentation. There has been a fantastic growth in Twitter usage since its origin in 2006 due to its simplicity as it reduces the cost of time and burden for the users (Java et al., 2007). The research conducted on Twitter in 2014 showed more than 200 million active users of Twitter (Wehner et al., 2014) and that about 11% of the users' embedded security in their account (Udani, 2012). However, the current statistics on Twitter users showed that more than 330 million active Twitter users generate about 500 million tweets per day (Davoudi et al., 2020).

This shows that a vast number of tweets are open for public access. However, the publication of Twitter API has provided massive access to this user-generated content on

a different level to both scholars and businesses. For instance, in a business environment, Twitter provides an occasion for its customers to communicate their opinions and familiarities in connection with a brand and its product. The importance of precise knowledge of clients' needs has made numerous companies shift their attention in investigating more on the technology that provides them with an opportunity to extract precious information from the data. The increase in the degree of Twitter and other sources of content obtained from users has resulted in the necessity of devices that will facilitate companies to promptly analyze and interpret subjective data of the consumer in large magnitude. For instance, the announcement of the Hadoop (an open-source implementation) tool has set the basis for such a device, providing opportunities for businesses to utilize this ample data to facilitate an improved business strategy.

In addition to the commercial benefits of Twitter as a study platform, other disciplines have also embraced the platform. For example, online learning (Grosseck & Holotescu, 2008) and prompt news broadcasting in natural disasters (Li & Rao, 2010). The varying nature of the study as mentioned above domains has revealed the intensity and the wideness of the generated data by the service. In addition, the generated content is often humorous and subjective, becoming an important source of corpus creation for sarcasm identification studies.

2.3 Sentiment Analysis and Sarcasm Identification

Sentiment analysis is a type of text classification that involves machine learning, information retrieval, Natural Language Processing (NLP), data mining, and other research domain (Xu et al., 2019). Sentiment analysis obtains sentiment or important information from data (Kumar & Kaur, 2020). Various techniques, including text analysis, text processing and natural language, are employed for that processing. The goal of sentiment analysis is to determine the document polarity by analyzing data within the

documents. Thus, the document polarity is based on the document opinion, categorized into positive, or negative, or neutral polarity (Kumar & Kaur, 2020).

Sentiment analysis inherent many challenges, and one of them is sarcasm identification. The identification of Sarcasm is regarded as a unique case in text classification, in which the core objective is to differentiate between the sarcastic texts from the non-sarcastic counterparts. Sarcasm is figurative language, which is a noticeable characteristic of human communication. When people communicate their opinions using sarcastic expressions, they usually apply their language to attain their communication objective. In most cases, it significantly changes the meaning of the expression in contrast to the literal explanation. Thus, there is no systematic way of constructing a sarcastic expression. In such an instance, the main goal of sarcasm identification is to determine discriminative features that differentiate between sarcastic text and non-sarcastic one. As a result, it is important to analyze this figurative language to get the actual meaning of its presence in any expression. However, the analysis of this figurative language does not require only the extraction of linguistic features from the textual data but also semantics, pragmatics, and other language analysis.

Unfortunately, sarcasm identification is a challenging task in NLP. NLP is a study area that focuses on the interaction of computers and human language. It mostly focuses on the intersection of computer science, artificial intelligence and computational linguistic. NLP is needed for text analysis by allowing the machine to understand how human speaks. With the help of NLP, knowledge can be organized and analyzed to perform various tasks such as automatic summation, sentiment analysis, topic segmentation, translation, and speech recognition (Kumar & Kaur, 2020).

Sarcasm, being a special type of communication where the explicit meaning differs from the implicit one, cannot be effectively identified with conventional data mining

techniques (Yee & Pei, 2014). Sarcasm exists in many kinds of structure and order, such as verbal or written sarcasm. Verbal sarcasm is a kind of sarcasm that usually occurs in speech, which can also be referred to as spoken sarcasm. Features like pitch level and variation, speech time, tempo, and acoustic features (intensity, volume, and frequency) are found in verbal sarcasm. In addition, this kind of sarcasm uses tones and gestures like eye and hand movement to show their sarcasm.

In contrast, written sarcasm occurs in a medium such as official letters, email, social media, and product reviews. The written sarcasm, in contrast with verbal sarcasm, is easy to classify. This is because the evaluation and interpretation of the written expression can be conducted by using the NLP toolkit (Yavanoglu et al., 2018). In addition, the analysis of such expression can be likely carried out from different forms of viewpoints.

On the one hand, when sarcasm is used in communication, it becomes hard to efficiently identify by employing data mining approaches due to the differences in its implicit and explicit meanings in a sentence (Yee & Pei, 2014; Shrivastava & Kumar, 2021). On the other hand, when sarcasm utterance is expressed in textual data, it is hard for people to precisely detect if a sentence is sarcastic or not due to its ambiguity (Muresan et al., 2016), and the absence of tune and gesture in the textual data (Bharti et al., 2016). Another reason for the difficulty in detecting textual sarcasm is the absence of accurately labelled naturally occurring utterances as sarcastic that can train supervised learning algorithms (González-Ibánez et al., 2011; Muresan et al., 2016; Kumar et al., 2019). Therefore, an efficient NLP method for text classification in a sentence with sarcastic attributes and properties is required to identify sarcasm (Yavanoglu et al., 2018).

What does the term sarcasm identification mean? It is simply a task of assigning a value or classifying a word or structure of sentences that possess the attributes and properties of sarcasm by employing natural language processing techniques (Yavanoglu

et al., 2018). It is a system that recognizes and differentiates between normal expression and sarcasm at the circumference of the semantic level. Since sentiment classification is a core objective of sarcasm identification, investigators often employ a machine learning algorithm due to its vigorous nature and the capability to adjust itself in agreement with the specified parameters and a dataset. An effective sarcasm identification design will guide an individual customer on the misleading review or personal expression composed in social or e-commerce platforms by other customers.

2.4 Sarcasm Identification Approaches

Researchers have carried out studies on sarcasm identification in textual data. Various studies approach for automatic identification of sarcasm found in the literature are lexicon-based (Riloff et al., 2013), rule-based NLP (Mukherjee & Bala, 2017a; Nezhad et al., 2021), pattern-based (Bouazizi & Ohtsuki, 2016), lexicon-based approach (Bharti et al., 2015), Corpus-based (Khodak et al., 2017), statistical-based approach (Reyes et al., 2013) and machine learning approach (González-Ibánez et al., 2011; Shrivastava & Kumar, 2021). Recently, a new technology (also referred to as a deep learning approach) has gained considerable ground in sarcasm identification research (Ghosh & Veale, 2016; Mehndiratta et al., 2017). Few researchers have employed the approach. The taxonomy of approaches is depicted in Figure 2.1, and the detailed explanations of those approaches are presented in the subSections below.

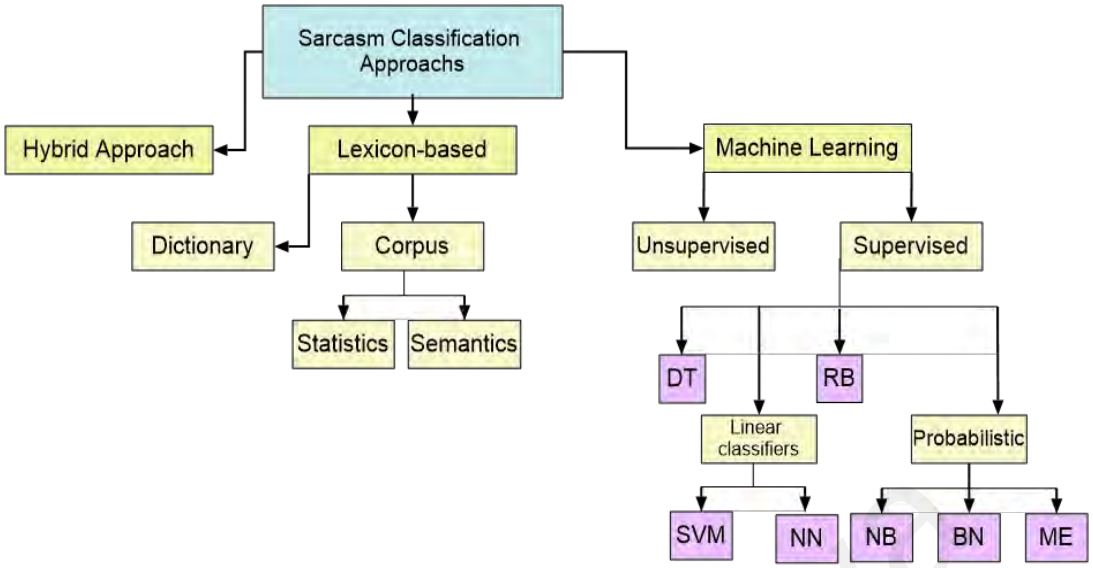

Figure 2.1: Sarcasm Identification Approaches

2.4.1 Lexicon based Approach

Sentiment analysis obtains sentiment or important information from data (Kumar & Kaur, 2020). In sentiment analysis, the lexicon-based approach is one of the unsupervised methods utilized to do many works by many researchers. The text classification in this approach is performed by comparing the sentiment lexicon against the textual feature in an expression. However, the values of sentiments are earlier obtained before the usage. Essentially, the sentiment lexicon contains the itemized word and the documents utilized to express an individual's sentiment. For example, for sarcasm analysis, a bag-of-lexicon (comprising unigram, bigram, trigram. etc.) and phrases are used to recognize sarcasm in tweets (Riloff et al., 2013; Sonawane & Kolhe, 2020).

For instance, Riloff et al. (2013) utilized a bootstrapping method to construct two lexicon bags consisting of unigram, bigram, and trigram phrases. Moreover, these phrases were employed for sarcasm identification in tweets, where the positive sentiment is used in a negative situation. Comparably, four bags-of-lexicon that consists of positive sentiment, negative sentiment, positive situation, and the negative situation has been developed (Bharti et al., 2015). However, they employed these phrases to recognize

sarcasm as negative sentiment in a positive situation and positive sentiment in a negative situation. Thus, two methods that can be used to create a sentiment lexicon are dictionary and corpus methods. Their brief description is given in the subSections below.

2.4.1.1 Dictionary-based Approach

In the dictionary-based sarcasm classification approach, three different steps are performed. Step one deals with the manual construction of opinionated words with their corresponding sentiment orientations. On the other hand, the second step focuses on the growth of the seed list by seeking for antonyms and synonyms of the seed words by using free online dictionaries such as WordNet. During the search process, when words of similar polarity with their synonyms in the list and different polarities of their existing antonym are found, then the results are added to the seed list. The iteration process continues until all the new words are exhausted in the dictionary. Finally, the third step concentrates on manual correction activity to eliminate the existence of errors. However, additional information in the WordNet dictionary such as “hyponym” and machine learning information makes it easy to produce better lists of opinion words. Consequently, this classification approach has inherent a drawback in differentiating opinion words in reverence to their domain.

2.4.1.2 Corpus-based Approach

In sarcasm classification, the Corpus-based approach is a problem-solving method that uses an object that relies on specific principles or guidelines to solve the dictionary-based. It uses syntactic, semantic, and stylistic properties of the sentence, such as the pattern of phrase and lexical structure of sentence analysis in any language for sarcasm classification. The semantic-based approach, one of the corpus-based approaches, emphasizes the meaning of word use, its structure, structural relationship, and the contextual usage in the language (Liu, 2012).

The dictionary-based problem solving using this approach is performed in two stages: In the first stage, a seed list of opinion words that contain a tag for adjective parts of speech with their corresponding polarities are constructed. In stage two, a set of linguistic controls is presented to search for more opinion words in the current corpus and their sentiment polarities. However, sentiment control relies on the notion of "Sentiment Consistency" because people often communicate similar opinions on both sides of conjunctions. Thus, the seed list of opinions can be extended. The semantic-based model is the bedrock of the corpus-based approach due to its effectiveness in nature (Katyayan & Joshi, 2019).

Accordingly, one of the studies that utilized this approach for sarcasm classification was presented (Bharti et al., 2015). The study used the Twitter dataset and the feature extraction techniques that comprise parsing, parts-of-speech tagging, and parse tree to learn the semantic arrangement. The study employed two algorithms to determine the different polarity sentiment in a tweet and the tweets that started with interjections. However, their result shows that the most sarcastic sentences begin with an interjection in a sentence. Similarly, Riloff et al. (2013) also presented a rule-based algorithm that searches for the occurrence of a negative situation and positive verb phrase in a sentence. The study utilized a well-structured iterative algorithm to extract the negative situation phrase and carried out the experimental analysis with various sets of the rule.

2.4.2 Machine learning Approach

This approach is one of the most applied approaches for sarcasm identification by researchers. This is because of its stability feature and its ability to observe itself in correspondence with a dataset and a given specification. Machine learning approach deals with the creation of a prediction model using an intelligent method. The effect of pragmatic and lexical aspects in the machine learning algorithm was studied in

(González-Ibánez et al., 2011; Sarsam et al., 2020). The machine learning approach can be further be categorized into unsupervised learning, supervised learning, semi-supervised learning, and hybrid learning. A brief explanation of these approaches is given below.

2.4.2.1 Supervised Learning

Among the machine learning algorithms, supervised learning is mostly used in sarcasm detection because of its ability to build a model by taking a labelled dataset as input data (Mohri et al., 2012) and producing a labelled output data, which helps in the construction of a decent model. This is made possible because the training datasets have already provided the result to be processed by the model. The primary purpose of this form of learning is to drive a functional correlation from the training data with well-generalized testing data. Some of the examples of this learning algorithm include Naïve Bayes (NB), Decision trees (DT), and Logistic regression (LR) (useful for either regression or classification task). Supervised learning algorithm (like NB, DT, and LR) serves as the bedrock for other learning algorithms with similar precepts (Yavanoglu et al., 2018). The machine learning algorithm (such as Support Vector Machine (SVM) and Logistic Regression (LR)) in addition to the Sequential Minimal Optimization (SMO), was also employed to differentiate sarcasm from the polarity sentiment occurring in Twitter messages (González-Ibánez et al., 2011; Castro et al., 2019b).

2.4.2.2 Semi-supervised Learning

This form of the machine learning algorithm is a mixture of supervised and unsupervised learning using a minimal quantity of annotated data and a vast number of unannotated data (Tsur et al., 2010). The presence of the unlabelled datasets and the open access to the unlabelled datasets is the feature that differentiates supervised learning from semi-supervised learning. Semi-supervised learning was created as a result of the cost-

effect of data annotation in some complex applications. Information recommendation systems and semi-supervised classification are examples of a semi-supervised learning algorithm. Davidov et al. (2010) employed this type of learning approach for automatic sarcasm identification using amazon product review datasets. In their study, a total number of 66,000 products and book reviews were collected, and both syntactic and pattern-based features were extracted. The sentiment polarity of 1 to 5 was chosen on the training phase for each training data. The authors reported a promising performance of 77% precision and 83.1 % recall on the evaluation phase.

2.4.2.3 Unsupervised learning

Unsupervised learning is employed when there are difficulties in finding the labelled sample since it does not rely on the previous training for mining the data. Thus, there is an existence of only one observation. The primary purpose of unsupervised learning is to find a correlation between the samples behind the observation. One of the notable examples of unsupervised learning is a clustering system.

The popularity of the architecture of deep learning approaches has created an opportunity for researchers in this domain to conduct a study on the automatic identification of sarcasm (Nweke et al., 2018; Eke et al., 2021). This form of learning consists of a subset of machine learning by employing neural networks to automatically learn from large datasets (Nweke et al., 2018). A neural network is a learning algorithm that processes features similar to the functioning of the nerve system in the human brain. In the neural network, each unit of the network has a connection to many other units, which can possess a summation function that combines all its input values. The neural network uses 0.0 and 1 real number value representation in terms of core and axon.

Ghosh and Veale (2016) employed a deep neural network model to identify sarcasm occurrences on Twitter datasets. In their work, they combined the algorithms that consist

of a convolutional neural network, Long Short Term Memory (LSTM) network, and recursive SVM. They got an impressive performance of the model over the baseline method for sarcasm detection system by attaining an F-score of 92% (Schifanella et al., 2016). Similarly, in their study, Joshi et al. (2016) also used features based on word embedding similarity for sarcasm identification. The feature used in their study was enhanced with the most congruent and incongruent word pair, which improved the performance.

2.4.3 Hybrid Approach

The hybrid approach comprises the fusion of other approaches, such as lexicon and machine learning-based approaches. A study that employed this approach is the learning of user-specific context presented by Amir et al. (2016); it uses a convolutional network to learn user embedding features in conjunction with the utterance-based embedding feature. The resultant features formed a hybrid Convolutional User Embedding Convolutional Neural Network (CUE-CNN) model in the domain of sarcasm detection. The result of the study produced a performance increment of 2% over single machine learning approaches for sarcasm identification.

2.5 Text Classification Process for Sarcasm Identification.

According to Nithya et al. (2012), supervised text classification is a classification that uses labelled training datasets of the text to learn and build a text classifier that can be used to classify the unlabelled test sets automatically. Human observers are often used to perform text categorization nowadays; however, these are deemed incompetent due to the huge number of files, email messages, and web addresses saved in a folder every day (Harrak et al., 2019). Moreover, manual categorization is usually slow and costly to maintain (Lytvyn et al., 2019). In addition, inconsistency is another limitation inherent in manual categorization. The above-identified limitations have shifted the text

classification from a manual to an automated base. Several techniques exist in automated text classification, such as supervised, semi-supervised, and unsupervised text classification. However, the supervised approach is most globally used as it can build a model using labelled data as input data (Mujtaba et al., 2017; Yavanoglu et al., 2018). The supervised text classification experimental process consists of 6 main steps as explained in the subsequent Sections.

2.5.1 Data Collection

The data collection phase comes first in any text classification process. The collection of datasets is based on the domain the study is considering. For example, when a study seeks to detect sarcasm on Twitter, then the Twitter data is collected. When a study seeks to analyze the disaster response and recovery through sentiment analysis, then the disaster-related data is collected in social media. In any case, once the raw data is collected, the next phase of the classification is to pre-process the data before the actual analysis can be carried out on the dataset.

2.5.2 Data Pre-Processing

Raw data collected during the data collection phase contains a lot of noisy information and requires cleaning. The purpose of cleaning is to eliminate the noise from the data before some knowledge or features can be extracted from it. Also, duplicate data are also removed during the pre-processing stage, especially the social media data (Eke et al., 2019). Data pre-processing is the data preparation phase, where the training and testing datasets are prepared. Twitter datasets are labelled as either sarcastic or non-sarcastic and are required to train the model in the training sets.

In contrast, the testing datasets are not labelled since it is mainly used for model evaluation. Therefore, the pre-processing stage mainly seeks to remove unnecessary characters or sequences, which have no value to the sentiment classification. In this phase,

the collected data will first undergo a tokenization process, also called automatic filtering. This is purposefully performed to remove retweets, duplicates, stop words, punctuations, numerals, tweets written in other languages, and tweets with the only URL. However, Parts Of Speech (POS) tagging and stemming were applied on the remaining tweets to convert the text to its original form at the end of the filtering stage.

2.5.3 Feature Engineering

Feature extraction is the third stage in the supervised learning approach with regard to the text classification task. It is a technique used to reduce the number of resources required to describe the dataset by transforming the input data into a set of features. The feature consists of linguistic, pragmatic, emotional, psychological, hyperbolic features, among others. Section 5.3 provides more explanation on these features. The most commonly used feature engineering techniques are Bag-of-words and N-gram. The Bag-of-words model is a text classification technique that uses the frequency of each word as a feature for classification. The Bag-of-word technique has been one of the widely used techniques for document representation in information retrieval for some years now and as a tool for feature generation (Salton & McGill, 1986; Yavanoglu et al., 2018). However, in the N-gram technique, n stands for the number of word features. For example, when the value of n is 1, the feature is called unigram; when n is 2, it is called bigram, and when n is 3, it is called trigram, and so on. Simplicity and scalability are some of the choices of using this technique over the bag-of-words model (Yavanoglu et al., 2018).

2.5.3.1 Feature Selection

The whole feature sets extracted from the datasets contain irrelevant features that may limit the prediction result during the classification stage. For instance, drawbacks during the text classification due to the immaterial feature content are a reduction in the accuracy,

a problem in generating a result, a decrease in the classification process, and difficulty in storage and retrieval of information. Hence, there is a need for a feature selection technique to choose the most discriminant feature subsets from the extracted feature sets for better prediction (Guyon & Elisseeff, 2003; Amini & Hu, 2021).

A thorough understanding of the aspect of the relevant datasets for the prediction that is to be carried out is needed. Feature selection techniques can be sub-divided into wrapper, filter-based, and embedding techniques (Guyon & Elisseeff, 2003). Among these three categories, the filter-based technique is widely employed (Yang & Pedersen, 1997). The filter-based technique uses statistical means to allocate a score to each feature, and the score determines the selection and rejection of the feature. Chi-Square ($\chi 2$) and Information Gain (IG) are common examples of feature selection filter-based techniques. However, the wrapper-based approach uses the query technique for the best feature selection from the different combinations and performs an evaluation using other combinations, whereas the embedding method studies the essential features of building the model. A brief description of the most commonly used feature selection technique is given below.

- ✓ **Information gain**

Information gain (IG) measures the reduction in entropy obtained by splitting the examples based on specified features. Entropy is a recognized information theory concept that defines the (im)purity of an arbitrary collection of examples (Gray, 1990).

IG is employed to compute the feature-ability or feature significance in classification experiments based on the class attribute. IG (Qabajeh & Thabtah, 2014) measure how good a definite feature separates training features based on the class labels as demonstrated in the equations defined below. Given a train data(D_t).

$$All\ train\ data\ Entropy\ (D_t) = I\ (D_t) = -\sum S_n\ log_2\ S_n \tag{2.1}$$

Where S_n represents the probability in such that D_t in a member of class n.

For attribute Atr data sets, the predicted entropy is computed as

$$Predicted\ entropy\ for\ ATT\ = I(Atr) = \sum \left(\frac{Dt_{Atr}}{Dt}\right) * I(Dt_{Atr}). \tag{2.2}$$

The IG of attribute Atr data sets is

$$IG(Atr) = I(Dt) - I(Atr). \tag{2.3}$$

✓ **Pearson correlation (PC) :**

The correlation method of feature selection is employed for dimensionality reduction in features and evaluation of discriminating ability of feature in classification experiments. This method also directly selects discriminative features from a pull of features. Correlation-based evaluates feature importance by calculating the correlation that exists between it and a class. PC coefficient determines the linear correlation between two attributes (Benesty et al., 2009). The succeeding value rests between -1 and +1, where -1 signifies absolute negative correlation, +1 represents absolute positive correlation, and 0 signifies the absence of linear correlation between the two attributes. Thus, the Pearson correlation coefficient measures the correlation between two attributes or J and K features (Hall, 1999).

$$R_{ab} = \frac{\sum(a_i - \bar{a})\ (b_i - \bar{b})}{(c-1)S_a S_b}, \tag{2.4}$$

where $\bar{a}$ and $\bar{b}$ are the mean sample for J and k, respectively, S_a and S_b are the sample SD (standard deviation) for J and K, respectively, and c is the sample size for correlation coefficient computation (Hall, 1999).

✓ **Chi-square (χ2)**

Chi-square is another most employed feature selection technique. Chi-square is a statistically based feature selection employed with other variables to test the independence of two occurrences. Specifically, the Chi-square method tests the independence of a specific feature and class occurrence and has a zero natural value. Therefore, the quantity below is estimated for an individual feature and the ranking is based on their score.

($\chi 2$) (Zheng et al., 2004) measures the independence between feature f and class c which N is the total number of documents.

$$\chi 2(\mathrm{w,c}) = \frac{N[(fc)x(NfNc)-(cNfxfNc)]^2}{(fc+cNf)x(fNc+NfNc)x(fc+fNc)+cNw+NfNc} \quad (2.5)$$

Here, fc is the total number of times f and c occur concurrently. The fNc is the number of times the f exist in the absence of c. The cNf is the number of times c exist in the absence of f. The $NfNc$ is the number of times there is no concurrence of f and c.

2.5.3.2 Feature Representation

The feature extracted is converted into a numerical value during the feature representation step in text classifications (Salton & Buckley, 1988). The feature representation technique is categorized into Term Frequency (TF), Binary representation (BR), and Term Frequency with Inverse Document Frequency (TF-IDF) (Debole & Sebastiani, 2004). In the TF representation, the value of the feature signifies the total occurrences of the feature in the document (Ramos, 2003). However, in the BR technique, the feature value 0 or 1 is used for representation where value 1 indicates the feature in the document, and value 0 signifies the absence of the feature in the document (Salton & Buckley, 1988). In TF-IDF representation, the frequency of the text in a particular document is calculated, and the result is compared with the inverse portion of the

frequency of the word in the whole document. It effectively matches a word in a query to documents that are important to the query (Ramos, 2003; Xiao & Tong, 2021).

2.5.4 Classifier Construction

At this phase, the classification model is created on the training datasets by utilizing either machine learning models or deep learning models. The created model can classify the unlabelled data as sarcastic or non-sarcastic. Several algorithms have been implemented for sarcasm identification. A few of the algorithms used in the selected studies consist of Naïve Bayes (NB), Support Vector Machine (SVM), Decision Tree (DT), Random Forest (RF), Linear Regression (LR), and Artificial Neural Network (ANN), Convolutional Neural Network (CNN), Long Short Term Memory (LSTM), Recurrent Neural Network (RNN), Bi Long Short Term Memory (Bi-LSTM) (Yang, 1999; Rizzo et al., 2020). These algorithms are described in the subSection below.

2.5.4.1 Naïve Bayes

Naïve Bayes (NB) is a classification algorithm that uses a probabilistic model to predict how data is obtained within a given class. It is a machine learning algorithm that performs a statistical analysis of numerical data (Sahami et al., 1998). It uses a labelled set of data as input data to calculate the parameter of the generative model. It is one of the simplest learning classifiers that assumes that all features do not depend on each other in a given class context (McCallum & Nigam, 1998). Moreover, NB is one of the fastest classifiers that perform well when Bag-of-words techniques are used in text representation (Rennie et al., 2003).

2.5.4.2 Decision Tree

A decision tree (DT) is a core algorithm employed in data mining for classification and prediction. It is an induced learning algorithm that is centred on the instances. It concentrates on the classification rule that displays a decision tree deduced from a group

of disorders to an irregular instance (Dai et al., 2016). The tree consists of a leaf node, path, decision node, and edges (Quinlan, 1990). DT is a classifier represented in the form of a flow-chart tree structure, in which a core node represents the attribute test, each branch denotes a test result, and each leaf node denotes a class. Thus, the whole tree tallies to a collection of a disjunctive representation rule (van der Aalst, 2001).

Furthermore, DT is employed to train instance classification, which can classify instances based on the definite attribute occurrence of the value sets. The over-fitting problem is one of the limitations inherent in a decision tree classifier. This is due to its ability to fit every data category and the noise that can extremely influence its performance. Notwithstanding, this problem can be overcome by employing multiple classifier models such as the random forest in which different trees are designed and trained by dividing the training set and the final predictions are combined over the tree.

2.5.4.3 Random Forest

Random Forest (RF) is an ensemble classification that uses sub-training sets to build a decision tree classifier. As such, DT classifies each input vector in a forest, and the most predicted classifier is selected. RF is a powerful ensemble classifier that can carry out both classification and regression tasks. It creates several decision tree models by employing sub-training sets of data. As a result, every input vector in the forest are classified, and the best classifier is chosen. However, the higher the number of these decision trees, the higher the prediction's accuracy and robustness. In a random forest, the new object based on the attributes is classified using the multiple trees that it contains.

Every tree obtains the classification, and the voting for each class is stored. Then, the selection is carried out by choosing the best voted for consideration. One of the advantages of choosing the random forest classification algorithm is that the model controls the omitted values, and the accuracy of the omitted data can be maintained. In

addition, the model can also be used to classify high dimensional large datasets. Even though the algorithm can be employed for regression tasks, it performs better in classification tasks than the regression tasks. Due to the fewer parameters tuning in this model, there is less control of the model. During the classification process, the test features are disseminated via each arbitrarily constructed tree for the classification. As the output of each tree is being predicted, the voting for the prediction is computed to obtain the greatest vote for an individual class of prediction. The process is termed majority voting. The model can be improved by employing the divide and conquer technique.

Random forest classifier produces a better performance when compared with just a single decision tree, as it overcomes over-fitting, which is inherent in most models (Liaw & Wiener, 2002; Fernández-Delgado et al., 2014). RF performs well as a strong learner when combined despite the weakness in learning individual classifiers in this group (Liaw & Wiener, 2002). Random forest solves the over-fitting problem and it produces better prediction compared to a single decision tree (Liaw & Wiener, 2002; Fernández-Delgado et al., 2014; Li et al., 2020).

2.5.4.4 Support Vector Machine

Support vector machine (SVM) is a supervised predictive model that uses statistics to construct a classification model. SVM is employed for classifying different classes in a dataset. It is a common algorithm that separates different classes of data points in datasets. It looks into the extreme data point in a dataset and builds a decision boundary (hyper-plane). This boundary has a single dimension fewer than the data point dimension. In SVM, a hyper-plane, also known as a support vector, is employed to separate the data points into two classes by decreasing the space between them by using the training sets (Cristianini & Shawe-Taylor, 2000). The data points that are nearer to other classes

pushes the boundary further to produce better prediction results. SVM algorithm maintains that only those support vectors or margins are required for the further classification task while other data points are rejected. This is because the boundary case in a class is considered by drawing a margin, and all the other points do not require earlier knowledge before the classification. However, further classification of the new data will produce a less predictive performance if the decision boundary is created without being optimized.

The dataset is split into two portions (one portion to train the model and the second portion to test the model). Accordingly, the model is built using the training set, which predicts the target value by providing attributes on the test data (Hsu et al., 2003; Pisner & Schnyer, 2020). SVM model can also be employed on a dataset with a high dimension. It refers to the data points as a vector, possessing their coordinate within the data space. Computational complexity is one of the drawbacks that are found in data points with higher dimensions for prediction. As a result, a kernel function can be employed to decrease this computational complexity. A kernel function accepts an input vector from the initial vector space and produces the dot product vectors in the feature space. Parameter tuning is needed to obtain a better prediction using a kernel. Various data mining applications, including image classification, spam mail detection, and bioinformatics, have successfully been modelled using the SVM algorithm (Fernández-Delgado et al., 2014). However, poor prediction performance usually occurs when the number of features exceeds the sample number.

2.5.4.5 K- Nearest Neighbor

K-Nearest Neighbor (KNN) is an instance-based machine learning model usually utilized for regression and classification tasks. In this form of a model, identifying the class label for each instance depends on the k-nearest neighbor of that instance. Thus, the

majority voting approach is employed in the neighbor instance to decide the class label. However, in this classification system, each neighbor's majority vote is assigned to its class instance (i.e. the k-nearest neighbor's most common class instance) (Han et al., 2011; Wang et al., 2020).

2.5.4.6 Logistic Regression

Logistic Regression (LR) is a linear predictive model that classifies event occurrence probability as a linear function of a predictor variable class (Kantardzic, 2011). In the LR algorithm, the decision boundaries are usually made by employing a linear function of the features. Logistic regression aims to augment the probability function to recognize the document class label. Parameter selection in the LR aims to attain the maximum conditional probability (Aggarwal & Zhai, 2012; Nusinovici et al., 2020). Despite LR's promising result, the class variable usually generated is out of (0-1), which is unsuitable for the probability range.

2.5.4.7 Artificial Neural Network

A neural network is a learning algorithm that possesses features similar to the functioning of the nerve system in the human brain. An artificial neural network comprises three distinct layers; input, hidden, and output layer. While the input and hidden layers consist of numerous nodes, the output layer comprises just one node. In the neural network, each network unit has a connection to any other units, which can possess a summation function that combines all its input values. The hidden layer is designed for input processing, and it connects to the output layer that garbage out the output values. The Neural network uses 0.0 and 1 real number value representation in core and axon (Yavanoglu et al., 2018).

According to Yao (1999), learning in an artificial network is categorized into unsupervised, supervised, and reinforcement learning. The unsupervised approach

centres on the relationship that exists among the input data. In that regard, there is the unavailability of "correct output" information for learning. In a supervised approach, the learning is based on comparing the actual input and the Artificial Neural Networks' target output to reduce the error function between them. In so doing, gradient descent-based optimization such as backpropagation is employed to regulate the connection weight to reduce the error iteratively.

On the other hand, reinforcement learning is a special case of a supervised approach that provides information on the correctness of the actual output. In that case, there is no knowledge of the precisely desired output. For example, in an Artificial Neural Network, a learning rule is utilized for weight modification on each input pattern, and the most commonly used rule is the Delta rule (He & Xu, 2010).

2.5.4.8 Convolutional Neural Network (CNN)

Convolutional neural network architecture consists of input, convolutional, pooling, and output layers. The input data are fed through the input layer and then passed to the convolutional layer. In the convolutional layer, feature maps are extracted, bypassing the convolutional filter on input data. However, multiple filters are utilized to input data for multiple feature extraction. The final decision is made by the fully connected layer, connected to the output and previous layers.

2.5.4.9 Recurrent Neural Network (RNN)

A recurrent neural network is a standard network that uses an edge to feed into the next time slides instead of feeding into the next layer in a similar time slide. Thus, 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.

2.5.4.10 Long Short-Term Memory (LSTM)

LSTM was created as an enhanced form of the standard recurrent neural network (Graves & Schmidhuber, 2005; Zen et al., 2016) to modify its state to verify what to retain and discard. LSTM is created by increasing the memory capability of RNNs (Salehinejad et al., 2017). The core aim of creating LSTM is to address the exploding and vanishing gradient problem found in the standard RNN. LSTM maintains the error to back-propagate using deeper layers in which learning continues over various steps during the training process. 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. Each LSTM unit consists of the input, forget, and output gate to coordinate and decide on the fraction of information to hold, discard, and move to the next step. It also decides when to read, write, and delete permission through gates that either pass or block information flow through the LSTM unit. LSTM architecture is depicted in Figure 2.2. To compute the input, forget, and output gate together with the input cell state, equations 1-6 below can be employed.

$$i_t = \sigma(W_{i_y} x_t + W_{i_z} h_{t-1} + b_i) \tag{2.6}$$

$$f_t = \sigma(W_{f_y} x_t + W_{f_z} h_{t-1} + b_f) \tag{2.7}$$

$$o_t = \sigma(W_{o_y} x_t + W_{o_z} h_{t-1} + b_o) \tag{2.8}$$

$$d_t = (W_{d_y} x_t + W_{d_z} h_{t-1} + b_d) \tag{2.9}$$

$$c_t = f_t \otimes c_{t-1} + i_t \otimes d_t \tag{2.10}$$

$$h_t = \tanh(C_t) \otimes O_t \tag{2.11}$$

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 the input cell state and three gates with the input hidden layers.

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

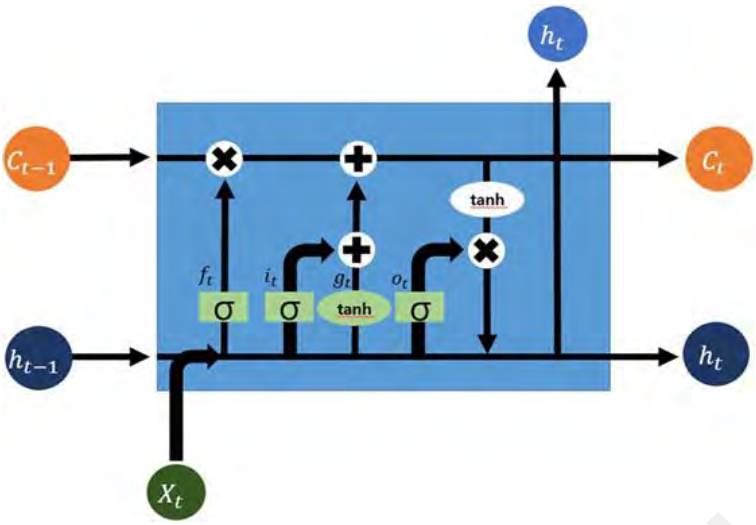

Figure 2.2: LSTM representation

2.5.4.11 Bi-Directional Long Short-Term Memory (BI-LSTM)

As indicated by (Tay et al., 2018), Bi-LSTM can capture compositional information in a sentence (for each input sentence). Bi-directional LSTM comprises the forward operation network that reads the clause information in the forward direction between words 1 and n. The backward operation network reads the clause information in the backward direction. Thus, the generated hidden states from both directions (forward and backwards) are joined to form hidden 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 (Graves et al., 2013).

$$y_t = \sigma(h^{\rightarrow}, h^{\leftrightarrow} t) \tag{2.12}$$

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 represent the final output of a Bi-LSTM layer, as shown in the equation below.

$$Y_t = [y_{t-n,} \dots \dots \dots \dots y_{t-1}\ ] \quad (2.13)$$

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

2.5.5 Performance Evaluation Measures

In the evaluation phase, the formulated classifier predicts the class of unlabelled text (sarcastic or non-sarcastic) using the training data sets. The predictive performance of the constructed classifier can be evaluated by employing the following parameters:

1. **True positive (TruPos):** The true positive result is noticed when the predicted tweet is sarcastic, and the result of the classification shows exactly sarcastic after the experimental evaluation.
2. **True negative (TruNeg):** The true negative result is obtained when the predicted tweet is not sarcastic, and the classification result also validates it as not sarcastic.
3. **False positive (FalsNeg):** True negative occurs when a true negative result is obtained when the predicted tweet is not sarcastic, but the classification result indicates that the tweet is sarcastic.
4. **False negative (FalsNeg):** Here, the true positive result is obtained when the predicted tweet is sarcastic, but the evaluation of the classification result shows that tweet is not sarcastic.
5. **Accuracy (Acc):** The accuracy provides the percentage ratio of the predicted instance. It measures 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.

$$Acc = \frac{TruPos + TruNeg}{TruPos + TruNeg + TruNeg + FalsNeg} \quad (2.14)$$

6. **Precision (Pre):** The precision provides the model accuracy in the existence of false positive instances. Thus, the model accuracy provides the overall occurrence of the false positive instance with the rejection of the positive instance. Precision is computed by finding the ratio of true positive over a positive result.

$$Pre = \frac{TruPos}{TruPos + FalsePos} \tag{2.15}$$

7. **Recall (Rec):** Recall 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 computationally represents the ratio of *true positive* against all the *true* results.

$$Rec = \frac{TruPos}{TruPos + FalseNeg} \tag{2.16}$$

8. **F-measure (F-m):** F-measure is a cumulative factor to test the overall effect of the recall and precision to find the overall impact of false negative instances and false positive instances over the whole accuracy. It represents the harmonic mean of precision and recall when there is severe equality of false positive and false negative. The standard F-M is F1, which gives precision and recall equal importance.

$$F - M = \frac{Pre\ x\ Rec}{Pre + Rec} \tag{2.17}$$

9. **Confusion Matrix:** The 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. 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 are not correctly classified. In addition, false positive instance symbolizes 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 instance. False negative symbolizes 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 2.1.

Table 2.1: Confusion matrix

	True instance	
Predicted instance	***Tru Pos***	***Fals Pos (type 1 error)***
	Fals Neg (type 2 error)	***Tru Neg***

2.6 Information Fusion Approach

The central basis of information fusion is to incorporate diverse information to enhance reliabilities, robustness, and generalization. Several research studies have been conducted to achieve the optimum features and classification algorithms to attain better classification results (Nweke et al., 2019b; He et al., 2020). Fusion approaches are generally classified into three types: data-level fusion, decision-level fusion, and feature-level fusion (Dasarathy, 1994; Mangai et al., 2010). The taxonomy of the fusion approach

is represented in Figure 2.3, and a brief explanation of each of the fusion approach is provided below.

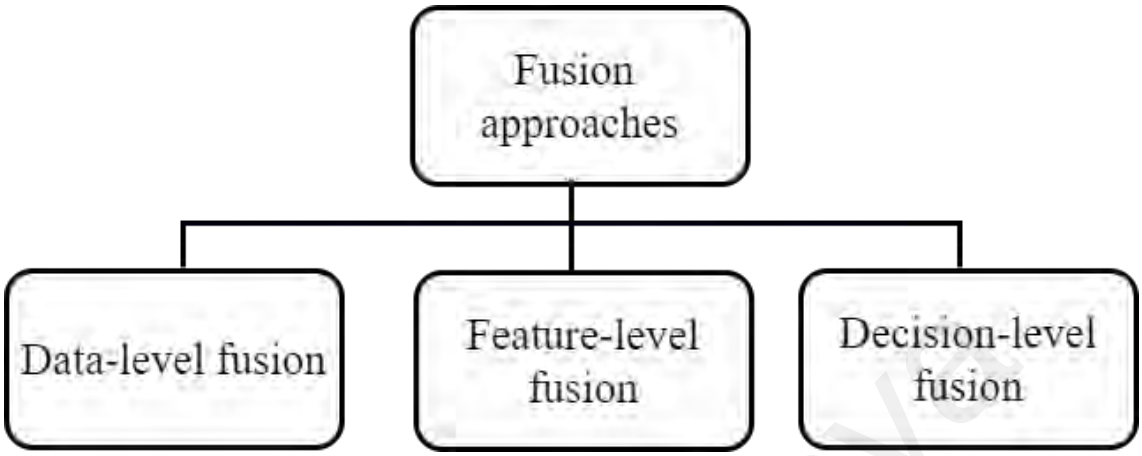

Figure 2.3: Taxonomy of fusion approaches

2.6.1 Data-level fusion.

Data level fusion is also referred to as low-level fusion, whereby various raw data sources are combined to produce new raw data that is required to be more revealing and synthetic compared to the initial (Dasarathy, 1994). For instance, Sangwan et al. (2020) proposed a multi-modal approach for sarcasm detection. The authors maintained that the conventional methods that rely on textual information for sarcasm identification are no longer enough for the tasks, and other information like visual information can offer some vital clue for sarcasm analysis. Thus, they investigated the deep learning approach that combines textual and visual data for sarcasm identification. They experimented with a Recurrent Neural Network (RNN) based on the input modality interaction for predicting sarcasm. However, the experimental analysis indicates that the inclusion of visual modalities enhances predictive performance.

2.6.2 Feature-level-fusion

Feature-level fusion dwells in the extraction, selection, and fusion of features to eliminate the irrelevant and redundant features (Dash & Liu, 1997; Yang et al., 2003;

Samanta & Das, 2009; Mangai et al., 2010). It is also called multi-features fusion. Multi-feature fusion is defined as the concatenation of two or more features/attributes to form a single master feature vector, which can be applied as an input to the machine learning model. For instance, if two features possess the same or closely related distribution, then one of them is redundant. Redundant features are those features that do not correlate well in terms of class information. In these settings, various feature vectors extracted from data are selected and integrated. The final features set are combined to form a better feature set, provided to the machine learning classifiers to get the final classification results. It is obvious from the definition that feature-level fusion is an advancement of data-level fusion. The main attraction of feature-level fusion is the ability to fuse features extracted from data with less sensitivity to noise (Kumar & Garg, 2019). Feature-level fusion is the most implemented fusion approach for sarcasm identification. Besides, other areas of feature fusion applications include medical image processing (Constantinidis et al., 2001), speech processing and video classification and retrieval (Arevalillo-Herráez et al., 2008), face recognition (Mitrakis et al., 2008; Tao & Veldhuis, 2009), gene identification in DNA sequence (Chen et al., 2007), and target object recognition (Brooks et al., 2003). Since the performance of classifiers depends on the quality of features used as input to the classifiers, this study focuses on the feature-level fusion for better performance results. Kumar and Garg (2019) proposed a feature fusion method, which utilized the fusion of pragmatic feature, sentiment feature, and Top-200 TF-IDF features to build the context using five shallow classifiers: support vector machine, K-nearest neighbor, Decision tree, Random forest, and Multilayer perception classifier. They considered the K-nearest neighbor classifier with 3 and 5 neighbors in their parameter settings whereas in SVM, they considered RBF and linear kernel. The experimental results show that the RF classifier outperformed other classifiers by attaining a precision of 69.08%. The low-performance results observed in this approach could be attributed to the deficiency of

word embedding-based features even though it included the sentiment-related features. Word embedding is imperative in sentiment analysis study, especially in the sarcasm classification task, as it captures the word semantics in the sentence. Due to the deficiency of word embedding-based features, the word co-occurrence in the text is not captured. Thus, more features that include word embedding should be explored for effective classification and performance enhancement.

2.6.2.1 Reasons for Multi-feature fusion

The reason for feature level fusion is due to the inherent drawback in the traditional feature engineering approach. For instance, the bag of words approach captures only the sequence of the word but ignores word context in sarcastic expression, which may lower the predictive performance of the classification. (Papadakis, et.al, 2016; Prasad et al., 2017 ; Xiao et al., 2018).

Secondly, the sparsity of the training data. Due to the word limit in microblog, some of the feature vector value constructed by the bag-of-word is mostly 0, which makes the training data sparse (Jia et al., 2019).

Based on this observation, a Multi-Feature Fusion Framework is proposed by fusing the lexical features with other eight proposed features to capture the context of the word and the various dimensions of sarcastic utterances present in the text.

2.6.3 Decision-level fusion

In decision-level fusion, also called high-level fusion, utilizes a set of classifiers and fuses several diverse machine-learning algorithms to arrive at a superior decision that provides better and unbiased results than the single classifier. The classifiers can possess similar or different features and exist in similar or different kinds (Kuncheva et al., 2001). Various classifiers exist, including artificial neural networks (Bishop, 1995), K-NN, and

various neighbors (Dasarathy, 1991). SVM with different kernels (Ho et al., 1994), RF (Liaw & Wiener, 2002) etc. However, one classifier may not be suitable for a specific application; thus, a set of classifiers is employed, and lastly, the outputs of all classifiers are fused using different techniques to get the final output. Decision-level fusion approaches have been widely utilized and tested in the various classification application domain (Ho et al., 1994; Kittler et al., 1998; Kuncheva et al., 2001; Kuncheva & Whitaker, 2003; Pan et al., 2020). Besides, different classifier fusion approaches have been defined, and the experimental analysis of some of them significantly demonstrates outstanding performance compared with individual best classifiers. However, there is lack of understanding of why some fusion approach outperforms others and the measuring criteria (Álvarez-Pato et al., 2020).

2.7 Review of Sarcasm Identification using Text Classification Technique

In this Section, a critical review of the selected primary study on various aspects was carried out. The aspects consist of datasets usage, pre-processing techniques, feature engineering techniques, the modelling approach, and performance metrics. The Section is divided into various subSections. The explanation of each subSection is given below.

2.7.1 Review of Datasets for Sarcasm Identification

The sarcasm identification dataset is an essential component of the sarcasm classification task. However, such a dataset is worthless on its own, except some features or useful knowledge are extracted from it. Related studies on sarcasm text classification showed that authors collected primary data using social media and employed two main annotation strategies like distant supervision via hashtag (Abercrombie & Hovy, 2016) and manual annotation strategy (Riloff et al., 2013). The first stage in the sarcasm identification experiment is the collection of data to be utilized for building the classification model. The analysis of the selected studies for sarcasm identification shows that datasets can be broadly categorized into homogeneous and heterogeneous data. These

data categorizations reviews are explained below, while the strengths and weaknesses of deploying the datasets for sarcasm identification are shown in Table 2.2.

2.7.1.1 Homogeneous data

In a homogeneous data, the studies utilized only one type of dataset from the Twitter platform. For instance, a study on 'Sentence level sarcasm detection in English and Filipino' carried out by (Samonte et al., 2018) utilized only Twitter datasets. The researchers collected a total number of 12,000 tweets consisting of 6000 Tagalog and 6000 English tweets. In addition, the authors employed datasets on topics such as transportation, government, politics, social media, and weather. In the study, the face pager API was utilized for the collection of data from Twitter. The parameters on the face pager were set accordingly, such as the result type (result_type); which specifies the preferred result by the users (i.e. popular, recent, or a mixture of both), the count; that specifies the maximum number of tweets to be retrieved (usually 200 maximum), and the language type; that specifies the type of language of the returned tweets.

However, similar parameter settings were used for both English and Tagalog tweets collection except in the language specification, in which *tl (for Tagalog)* was used on the Tagalog dataset. Thus, the study indicated that the nature of the datasets (balanced or Imbalanced) greatly influences the model's prediction in terms of the accuracy of sarcasm. In addition, Kumar and Harish (2018) used a content-based feature selection technique to build a classification model for sarcasm identification. The study utilized amazon product review datasets created by the study in (Filatova, 2012) and sourced from a crowdsourcing platform-*Mechanical Turk*. A total of 1,254 Amazon product reviews, consisting of 437 reviews (sarcastic) and 817 reviews (non-sarcastic), were used for the classification experiment. Interestingly, the datasets were structured using a star rating (ranging from 1 to 5) and review comments written in English.

Zhang et al. (2016) utilized Twitter datasets for sarcasm identification using a deep neural network in another study. The tweets datasets were obtained using the Twitter-streaming API with sarcasm hashtag (#sarcasm) and not hashtags (#Not) keyword. The study adopted the datasets obtained by (Rajadesingan et al., 2015b), in which a total number of 9,104 tweets annotated by the author of the tweets were used for the experiment. In this regard, similar tweets IDs provided by them were used to stream the corpus. Similarly, the contextual tweets were obtained by employing Twitter API in each tweet. However, the hashtag for sarcasm and Not (#sarcasm and #Not) were removed on the historical tweet to prevent explicit clues for sarcasm prediction. Furthermore, the author noted that both balanced and imbalanced datasets were modelled. The experimental result shows that the imbalanced dataset accuracies are greater than the balanced counterparts with the conflicting value of the F-measure. Therefore, imbalanced data create biases in sarcasm identification and performances of the model.

2.7.1.2 Heterogeneous data

The dataset used here to identify sarcasm is obtained from various social media and other platforms such as Instagram, Amazon, Tumblr, and product reviews from electronic commerce to improve the robustness and generalization of the sarcasm identification model. For instance, Schifanella et al. (2016) utilized datasets obtained from Twitter, Tumblr, and Instagram for sarcasm detection in the multimodal social platform, which comprises text and image datasets. In a previous work (Liu et al., 2014), the researchers evaluated their model by employing two corpora (English and Chinese) sarcasm features. However, the English sarcasm verification was carried out in the first corpus, which is the content of news article sets adopted from (Davidov et al., 2010), the Twitter datasets used by (Reyes et al., 2012), and Amazon datasets provided by (Burfoot & Baldwin, 2009). Then, the second corpus, which was used to verify Chinese sarcasm features, also

consisted of three different datasets obtained from Sina Weibo, Tencent Weibo, and Netease BBC to crawl various topical comments.

Invariably, the heterogeneous dataset employed in this study is highly imbalanced. Consequently, Area Under Curve (AUC) performance measure was employed for performance evaluation. It has proven successful in providing a better performance measure for imbalanced datasets than F-score by using true positive rate instead of precision. Furthermore, Davidov et al. (2010) study focused on sarcasm identification that deployed two multimodal datasets. In this study, the datasets used consist of tweets (5.9 million tweets) and Amazon product review datasets (66,000 product reviews), which were adopted from (Tsur et al., 2010). The tweets data was streamed using the #sarcasm hashtag included by the tweeter. However, there is inconsistency in using the hashtag since it is unknown to all the users; hence, most tweeters do not explicitly apply the hashtag to tag the sarcastic tweets. To this end, the tweets that included hashtag annotation can be regarded as the 'Secondary gold standard for detecting sarcastic tweets'. Still, in this study, the Amazon product review consisted of 120 products. The corpus is the content of different books and electronic products reviews. In contrast with the tweets, amazon products datasets are longer in size, as some of the review sentences contained about 2000 words. Interestingly, the sentence structure and grammar in the product review are better than the tweets datasets. Table 2.2 outlines the data types, sources, strengths, and weaknesses utilized for sarcasm identification.

Table 2.2: Dataset and volume used on the selected studies

Data Type	Data Sources	Strengths	Weaknesses	References	Number of Studies
Homogeneous data	Twitter or Product Review	Management of the data collection process is easier and cost-effective as the datasets are from a single entity. Furthermore, Twitter provides a rich source of API for data collection.	It is challenging to provide high generalization for data from one source for sarcasm identification as it involves varieties of applications	(González-Ibánez et al., 2011; Edwin & Ayu, 2013; Liebrecht et al., 2013; Riloff et al., 2013; Barbieri et al., 2014; Ptáček et al., 2014; Altrabsheh et al., 2015; Bharti et al., 2015; Bouazizi & Ohtsuki, 2015b, 2015a; Fersini et al., 2015; Ghosh et al., 2015; Khattri et al., 2015; Kunneman et al., 2015; Rajadesingan et al., 2015a; Wang et al., 2015; Amir et al., 2016; Bharti et al., 2016; Bouazizi & Ohtsuki, 2016; Ghosh & Veale, 2016; Ling & Klinger, 2016; Sulis et al., 2016; Al-Ghadhban et al., 2017; Bharti et al., 2017; Manohar & Kulkarni, 2017; Mukherjee & Bala, 2017b, 2017a; Ranjan et al., 2017; Abulaish & Kamal, 2018; Kumar & Harish, 2018; Manjusha & Raseek, 2018; Samonte et al., 2018; Sreelakshmi & Rafeeque, 2018; Kumar et al., 2019; Suhaimin et al., 2019; Ducret et al., 2020)	37
Heterogeneous data	Twitter, Amazon, Instagram, Tumblr, and Product review	The fusion of data from multiple sources helps to improve generalization, sarcasm identification model reliability, robustness, and performance result	Aggregation of data from various sources may increase computation complexity and lead to a high computation burden. Also, it is difficult to fuse a large number of datasets from multiple data sources.	(Davidov et al., 2010; Liu et al., 2014; Schifanella et al., 2016; Dharwal et al., 2017; Babanejad et al., 2020)	4

2.7.2 Review of Pre-processing Techniques for Sarcasm Identification

Pre-processing of social media data is necessary because of the irregular and informal form of data acquired. The purpose of pre-processing is to eliminate some problems

inherent in such texts, like a misuse of a letter, the use of acronyms, poor grammatical sentences, and unnecessary repetition (Cotelo et al., 2015). In the pre-processing stage, meaningless data from the acquired dataset are removed to enhance the performance of the classification model. According to the previous literature, the pre-processing techniques mostly used in sarcasm identification research include removal of stop words, space, punctuations, special symbols, conversion of uppercase letters to lower case, stemming, tokenization, POS tagging, lemmatization, removal of URLs, and hashtags. Thus, the efficiency of these preprocessing techniques is reported in various studies under consideration.

In recent studies, Al-Ghadhban et al. (2017) and Samonte et al. (2018) tested the impacts of inclusion or removal of URL, user mentions, and stops word in the textual data for sarcasm detection in Twitter. The experimental result showed that their removal enhances classification accuracy than when they are present. In addition, some researchers in their studies (Ghosh et al., 2015; Dharwal et al., 2017; Abulaish & Kamal, 2018) illustrated the application of stemming, tokenization, and conversion of upper case letters to lower case for pre-processing tasks for sarcasm identification. These studies reported that the application of such pre-processing techniques produced a better performance in classification when compared with other studies. A couple of scholars (Altrabsheh et al., 2015; Abulaish & Kamal, 2018) have also tested removing the white space character, punctuation marks, numbers, and emoticon. Their reports showed the effectiveness of applying these pre-processing techniques for improved classification tasks.

Nonetheless, Kunneman et al. (2015) tested the usage of punctuation marks as a feature for modelling in their study on ‘Signalling sarcasm from hyperbole to hashtag’. Their experiment showed a better performance in classification when punctuation marks were

present than when they were removed. Therefore, we can conclude that researchers should test the performance of the various pre-processing techniques on the sarcastic corpus to check the algorithm's accuracy in classification. The summary of the pre-processing techniques applied in the selected studies is illustrated in Table 2.3. The analysis from Table 2.3 shows that many studies used basic pre-processing techniques, which revealed the effectiveness of the pre-processing in attaining a better accuracy in the classification task.

Table 2.3: Pre-processing techniques used in the selected studies

Pre-processing techniques	References
Removal of Twitter user mentions, URL, hashtag, duplicates, quotes, elongation, punctuation marks, retweet symbols, less than 3 or 4 words, neutral tweets, manual labelling, and stop words	(Davidov et al., 2010; González-Ibánez et al., 2011; Fersini et al., 2015; Rajadesingan et al., 2015a; Schifanella et al., 2016; Zhang et al., 2016; Bharti et al., 2017; Mukherjee & Bala, 2017b, 2017a; Sreelakshmi & Rafeeque, 2018)
Conversion of numeric characters into alphabets, lower case, removal of local repetition, punctuation marks, blank spaces, special characters, stop words, and digits	(Edwin & Ayu, 2013; Altrabsheh et al., 2015; Kumar & Harish, 2018)
Tokenization, stripped with a punctuation mark, retain capital letters, part of speech tagging, stemming, stop word removal, conversion to capital letters, upper to lower case conversion, stemming, removal of URL and user mentions	(Liebrecht et al., 2013; Riloff et al., 2013; Barbieri et al., 2014; Ptáček et al., 2014; Ghosh et al., 2015; Khattri et al., 2015; Wang et al., 2015; Dharwal et al., 2017; Ranjan et al., 2017)
Removal of a retweet, hashtag, irrelevant tweet, emoji, links, lemmatization, tokenization, acronyms, and URL removal, part of speech tagging (POS)	(Bouazizi & Ohtsuki, 2016; Ling & Klinger, 2016; Al-Ghadhban et al., 2017; Samonte et al., 2018; Ducret et al., 2020)
Removal of hashtag, unwanted space using a regular expression, replacements of emoticon and acronyms using dictionaries, tokenization, stop word removal.	(Manjusha & Raseek, 2018)
Tokenization, removal of URLs, @mention, retweets, hashtags, ampersands, and extra white space, upper to lower case, double quotes, 1emoticons, numbers, and dots	(Abulaish & Kamal, 2018)
C1eaning, instance selection, normalization, transformation, POS tagging, tokenization	(Manohar & Kulkarni, 2017)
Tokenization (punctuation, emoticons, and capitalization information were kept), removal of less than three letter word	(Kunneman et al., 2015)
Removal of social media markers such as profile references, retweets and hashtags, parsing, and splitting of multiple sentences using the Stanford splitter.	(Ghosh & Veale, 2016)
Tokenization, spell checking and stop word removal.	(Suhaimin et al., 2019)

Pre-processing techniques	References
URLs, @mention, hashtag, and numbers in tweets are replaced with a placeholder, emoji	(Kumar et al., 2019; Nayel et al., 2021)
The pre-processing technique that was used was not mentioned	(Liu et al., 2014; Bharti et al., 2015; Bouazizi & Ohtsuki, 2015b, 2015a; Amir et al., 2016; Bharti et al., 2016; Babanejad et al., 2020)

2.7.3 Review of Feature Engineering techniques for sarcasm identification

Feature engineering is one of the major steps in any classification problem. Feature creation for modelling is the hardest and most vital aspect of classification, and it usually determines the success or failure of a model. Feature engineering is very important, especially when a few independence correlates well with the class. Every classification problem needs a different feature set, and as a result, the feature extraction process can be as important as choosing the best classifiers (Domingos, 2012). The quality of the feature extracted from the data depends on how well the data processing stage is performed. Three major stages are involved in feature engineering: feature extraction, feature representation, and subset feature selection (Mujtaba et al., 2018). The output of the feature engineering stage is in the form of the feature vectors (in numerical form), which serves as an input to the learning algorithm (SVM, RF, DT, etc.) for classification model construction and validation. A detailed explanation of these stages was given in Section 3, and the review is presented in the subsequent subSection.

2.7.3.1 Review of Feature Extraction Techniques

In sarcasm identification, feature extraction is extracting relevant and discriminant information from the sarcastic dataset, which will help train the model for sarcasm identification. The review of the selected studies showed that the semantic properties of the sentence features were used in most studies; researchers also utilized an automatic feature extraction technique to extract content-based features. This was carried out by using the algorithm and various statistical methods. The content-based feature extraction technique consists of BoWs (da Silva et al., 2014), word embedding (word to vector) (Lee

et al., 2018), and N-gram (Sintsova & Pu, 2016) technique. Word embedding (word vector) uses a contextual word vector that includes GloVe embedding feature (Pennington et al., 2014) trained 42B corpus as employed in (Ghosh et al., 2015; Eke et al., 2020; Potamias et al., 2020). As revealed in Table 2.5, most studies utilized the N-gram feature extraction technique on the selected studies. For instance, some authors (González-Ibánez et al., 2011; Rajadesingan et al., 2015a; Kumar & Harish, 2018) utilized the N-gram feature extraction technique for sarcasm detection and reported that the N-gram technique is useful in extracting lexical features. One of the motivations of the N-gram model usage by the researcher is due to its simplicity and scalability (the matching scale of all the enormous sample datasets) properties. In another study (Suhaimin et al., 2017), on sarcasm detection in the bilingual text, various NLP techniques were used to extract the combination of various features such as lexical, pragmatic, syntactic, prosodic, and idiosyncratic. These features were trained using a non-linear SVM algorithm. However, the result shows that NLP selected features outperformed the baseline features such as bag-of-words, which demonstrated better performance of the proposed method. The summary of the features extraction techniques used in the selected studies is shown in Table 2.5.

2.7.3.2 Review of features used for sarcasm Identification

In sarcasm detection, the quality of feature used plays a vital role in determining the sarcastic utterances present in the text. The features used can be classified into linguistic and content-based features. In the linguistic category, textual features that consist of hyperbole, lexical, and pragmatic features are used. The lexical features use textual properties, including bigram, trigram, unigram, etc., for identifying sarcasm in a text (Riloff et al., 2013). In a lexical-based feature, the corpus related to vocabularies of words is employed to identify sarcasm existence in the textual documents (Sulis et al., 2016). Similarly, hyperbole is also employed as one of the important features for detecting

sarcasm text documents. Hyperbole text consists of interjection words (such as wao, aha), punctuation marks (? and !), quotes (‘ ’, “ ”) and intensifiers (noun (NN), adverbs (ADV), adjectives (ADJ)) to identify sarcasm in the text (Barbieri et al., 2014; Abulaish & Kamal, 2018). For instance, Barbieri et al. (2014), in their study on sarcastic sentiment identification in tweets data, employed hyperbolic words as the set of features for classification. However, this feature was extracted based on the proposed standalone algorithm that followed a certain procedure set yet; sarcasm cannot be properly expressed in a certain predefined set of procedures.

On the order hand, the pragmatic feature consists of symbolic texts that consist of emoticons or emojis utilized in the expressions (Sulis et al., 2016). Various studies (González-Ibánez et al., 2011; Ghosh et al., 2015; Sulis et al., 2016; Samonte et al., 2018) utilized different linguistic features for sarcasm detection in the texts. For instance, lexicon-based features and pragmatic features (emoticons and user mentions) were extracted in a study by González-Ibánez et al. (2011) for sarcasm identification. The experimental analysis showed that the combination of such features improved the accuracy of the prediction. However, selecting suitable features for sarcasm detection in expression has not been properly investigated. Besides the linguistic features, various studies (Mukherjee & Bala, 2017b, 2017a; Onan, 2017) also investigated the content-based features by considering the presence or absence of a term in tweets. For instance, Mukherjee and Bala (2017b) employed content-based features. The study relied solely on the content of word use generally in the sentence to differentiate sarcastic from non-sarcastic in a sentence. The study produced a reasonable performance based on the data set that was used. However, the predictive model performance relied deeply on the content-based feature, which is likely to degrade when applied to other data sets due to its dependence on word use. Hence, the obtained result is not generalized to a satisfactory

extent. Table Table 2.4 depicts the comparative studies on the used features for sarcasm identification.

Table 2.4: The summary of features used for sarcasm identification

Feature(s) used	Reference
Punctuation and Pattern-based feature	(Davidov et al., 2010)
Lexical and pragmatic features	(González-Ibánez et al., 2011)
Sentiment polarity and interjection word	(Edwin & Ayu, 2013)
Sentiment and pattern feature	(Liebrecht et al., 2013)
Pragmatic and pattern feature	(Riloff et al., 2013)
Parts of the speech and sentiment feature	(Ptáček et al., 2014)
Parts of speech, pragmatics and pattern features	(Barbieri et al., 2014)
Punctuation symbols, linguistic features and syntactic features	(Liu et al., 2014)
Sentiment-based, lexical and punctuation features	(Bouazizi & Ohtsuki, 2015b)
Pragmatics and Parts of Speech	(Fersini et al., 2015)
Sentiment feature	(Khattri et al., 2015)
Pragmatics	(Ghosh et al., 2015)
Parts of the speech feature	(Bharti et al., 2015)
Sentiment, punctuation, syntactic and pattern	(Bouazizi & Ohtsuki, 2015a)
Pragmatics and polarity label	(Altrabsheh et al., 2015)
Sentiment-based feature	(Wang et al., 2015)
Punctuation and pragmatic features	(Kunneman et al., 2015)
Lexical and subjectivity features	(Schifanella et al., 2016)
Sentiment-based features	(Bouazizi & Ohtsuki, 2016)
Parts of speech and sentiment feature	(Ghosh & Veale, 2016)
Contextual features	(Zhang et al., 2016)
Behavioural features (Likes and dislikes)	(Bharti et al., 2016)
Sentiment and emotion-based features	(Sulis et al., 2016)
Content word, function word, and parts of speech feature	(Mukherjee & Bala, 2017b)
Sentiment-based feature	(Manohar & Kulkarni, 2017)
Punctuation marks, Dots, positive words and bracket	(Al-Ghadhban et al., 2017)
Sentiment polarity feature	(Ranjan et al., 2017)
Function words, content words and parts of speech features	(Mukherjee & Bala, 2017a)
Sentiment and topic features	(Dharwal et al., 2017)
Interjections and intensifiers	(Bharti et al., 2017)
Hyperbolic, question mark and intensifiers	(Abulaish & Kamal, 2018)
Lexical, pragmatic, hyperbole, quotations and punctuation marks	(Samonte et al., 2018)
Sentiment and emoticon features	(Sreelakshmi & Rafeeque, 2018)
punctuation and sentiment features	(Manjusha & Raseek, 2018)
Punctuation mark, capital letter and ‘or’ conjunction	(Kumar et al., 2019)
Pragmatic features, Malay prosodic, syntactic feature, POS features	(Suhaimin et al., 2019)
Emoji features	(Lemmens et al., 2020)
Linguistic (complexity, stylistic, psychological)	(Ducret et al., 2020)

Feature(s) used	Reference
Affective and contextual features	(Babanejad et al., 2020)

2.7.3.3 Review of Feature Representation Techniques

In addition to the feature extraction techniques, the study revealed that the feature representation techniques mostly used to convert the extracted feature into numerals are term frequency (TF), which is used to determine the frequency and occurrence of sarcasm in the extracted features. For instance, the contextual features extracted from the target author's historical tweets in a study by Suhaimin et al. (2019) were represented with TF and IDF. In that regard, the feature values of TF-IDF were used to sort the history tweets to choose the constant number of contextual tweets word (feature), having the greatest values of TF-IDF. In another study on sarcasm detection and sentiment analysis classification (Suhaimin et al., 2019), three NLP categories of features (pragmatic, syntactic, and prosodic), proposed by (Suhaimin et al., 2018), was adopted due to the demonstration of its improvement in sarcasm detection. Thus, the extracted features were represented using the term frequency-inverse document frequency (TF-IDF) and binary representation (BR).

2.7.3.4 Review of Feature Selection Techniques.

In feature selection, certain criteria are followed to discover suitable feature sets (Guyon & Elisseeff, 2003), and it is broadly employed in sarcasm detection. Notwithstanding, only a few studies in the selected studies on sarcasm identification utilized the feature selection technique to investigate the outcome of the different subgroups on the classification accuracy. The feature selection techniques that were used in the selected studies are chi-square ($\chi 2$), information gain (IG), and mutual information (MI), which are briefly explained below.

Chi-square (χ2): Chi-square is a statistical test used for measuring the absence of the independence that exists between a particular class (c) and term of features (f) (Kumar & Harish, 2018).

Information Gain (IG): Information gain is a feature selection technique that is used to determine the information gain by knowing the value of the attribute within a feature vector (Yang & Pedersen, 1997).

Mutual Information (MI): It is a statistical measure that is commonly used to model two random variables (word association and related application) that are mutually dependent (Yang & Pedersen, 1997).

For instance, Kumar and Harish (2018) employed chi-square (χ2), mutual information (MI), and Information Gain (IG) as conventional feature selection techniques to select the discriminative features for sarcasm classification. The researcher tested their presence, and the experimental finding shows that using these feature Section techniques reduced the high dimensional feature space and increased the classifier's classification accuracy. For example, SVM and RF classifiers yielded a maximum accuracy when MI and IG selection schemes were applied in classification. In a related study (Muresan et al., 2016), the N-gram lexical features were extracted using linguistic inquiry and word count (LIWC) and WordNet-Affect dictionary (Strapparava & Valitutti, 2004; Pennebaker et al., 2015). Furthermore, pragmatic features such as emoticon and punctuation were extracted. However, the discriminative features were selected by employing the chi-square (χ2) selection scheme before modelling. The review showed that five (5) out of the 40 selected studies used chi-square to select discriminative features, three (3) studies used information gain, one study used chi-square, information gain & mutual information (MI), 31 studies, however, did not report the use of any feature selection scheme to select the important feature from the extracted one. The summary of the feature representation

techniques is shown in Table 2.6, while the feature selection scheme utilized in the analyzed studies is shown in Table 2.7.

Table 2.5: Feature Extraction Techniques used in the selected studies

Feature Extraction Techniques	Shortcomings	Reference
N-gram	Loss of contextual information	(González-Ibánez et al., 2011)
N-gram	Loss of contextual information	(Edwin & Ayu, 2013)
N-gram	Loss of contextual information	(Riloff et al., 2013)
POS tagging	Ignores the sentiment polarity	(Ptáček et al., 2014)
N-gram	Loss of contextual information	(Barbieri et al., 2014)
Bag of Word	Ignores the context and data sparsity issue	(Fersini et al., 2015)
N-gram	Loss of contextual information	(Khattri et al., 2015)
N-gram	Loss of contextual information	(Ghosh et al., 2015)
N-gram	Loss of contextual information	(Rajadesingan et al., 2015b)
POS tagging	Ignores the sentiment polarity	(Bharti et al., 2015)
N-gram	Loss of contextual information	(Altrabsheh et al., 2015)
N-gram	Loss of contextual information	(Kunneman et al., 2015)
Bag of word and N-gram	Loss of contextual information and data sparsity issue	(Ling & Klinger, 2016)
N-gram and word embedding	Loss of contextual information and ignores the sentiment polarity	(Schifanella et al., 2016)
N-gram	Loss of contextual information	(Bouazizi & Ohtsuki, 2016)
Bag of words and POS tagging	Ignores the context and data sparsity issue	(Ghosh & Veale, 2016)
N-gram and Bag of word	Loss of contextual information and data sparsity issue	(Amir et al., 2016)
N-gram	Loss of contextual information	(Mukherjee & Bala, 2017b)
Parts of speech N-gram	Loss of contextual information	(Mukherjee & Bala, 2017a)
N-gram	Loss of contextual information	(Dharwal et al., 2017)
N-gram	Ignores the sentiment polarity	(Sreelakshmi & Rafeeque, 2018)
N-gram	Ignores the sentiment polarity	(Manjusha & Raseek, 2018)
N-gram	Ignores the sentiment polarity	(Kumar & Harish, 2018)

Table 2.6: Feature representation techniques used in the selected studies

Feature representation technique	Reference
BR	(Riloff et al., 2013; Liu et al., 2014; Ghosh et al., 2015; Khattri et al., 2015; Amir et al., 2016; Schifanella et al., 2016; Sulis et al., 2016; Mukherjee & Bala, 2017a; Sreelakshmi & Rafeeque, 2018)
TF	(Davidov et al., 2010; González-Ibánez et al., 2011; Liebrecht et al., 2013; Kumar & Harish, 2018; Manjusha & Raseek, 2018)
TF-IDF	(Ptáček et al., 2014; Dharwal et al., 2017; Samonte et al., 2018; Suhaimin et al., 2019)
BR and TF	(Barbieri et al., 2014)
TF and TF-IDF	(Zhang et al., 2016; Ranjan et al., 2017; Nayel et al., 2021)
The Feature representation technique that was used was not mentioned	(Edwin & Ayu, 2013; Altrabsheh et al., 2015; Bharti et al., 2015; Bouazizi & Ohtsuki, 2015b; Fersini et al., 2015; Kunneman et al., 2015; Rajadesingan et al., 2015a; Wang et al., 2015; Bharti et al., 2016;

	Bouazizi & Ohtsuki, 2016; Ghosh & Veale, 2016; Ling & Klinger, 2016; Al-Ghadhban et al., 2017; Bharti et al., 2017; Manohar & Kulkarni, 2017; Mukherjee & Bala, 2017b; Abulaish & Kamal, 2018; Kumar et al., 2019) (Babanejad et al., 2020)

Table 2.7: Feature selection techniques used in the selected studies

Feature selection technique	Reference
Chi-square	(González-Ibánez et al., 2011; Liebrecht et al., 2013; Dharwal et al., 2017; Manjusha & Raseek, 2018; Sreelakshmi & Rafeeque, 2018)
Information gain	(Barbieri et al., 2014; Liu et al., 2014; Sulis et al., 2016)
Chi-square, Information gain and Mutual information	(Kumar & Harish, 2018)
The Feature selection technique that was used was not mentioned	(Davidov et al., 2010; Edwin & Ayu, 2013; Riloff et al., 2013; Ptáček et al., 2014; Altrabsheh et al., 2015; Bharti et al., 2015; Bouazizi & Ohtsuki, 2015b, 2015a; Fersini et al., 2015; Ghosh et al., 2015; Khattri et al., 2015; Kunneman et al., 2015; Rajadesingan et al., 2015a; Wang et al., 2015; Amir et al., 2016; Bharti et al., 2016; Bouazizi & Ohtsuki, 2016; Ghosh & Veale, 2016; Ling & Klinger, 2016; Schifanella et al., 2016; Zhang et al., 2016; Al-Ghadhban et al., 2017; Bharti et al., 2017; Manohar & Kulkarni, 2017; Mukherjee & Bala, 2017b, 2017a; Ranjan et al., 2017; Abulaish & Kamal, 2018; Samonte et al., 2018; Kumar et al., 2019; Suhaimin et al., 2019; Babanejad et al., 2020; Nayel et al., 2021)

2.7.4 Review of Classification Techniques for Sarcasm Identification.

Various classification algorithms, according to our findings, have been used for sarcasm identification in social media. The review summary of the classification algorithms used in the selected studies is depicted in Table 2.8.

2.7.4.1 Conventional Machine Learning Model.

Table 2.8 shows that each study has utilized one or more classifiers. In addition, some studies utilized multiple classifiers to compare the performance of each classifier with the proposed method. It is evident from Table 2.8 that some studies employed only one learning algorithm for classification. Moreover, different researchers on sarcasm identification used different datasets. Thus, the comparison of different classifier's performance in classification in such an instance becomes difficult. For instance, a few recent studies (Liebrecht et al., 2013; Kunneman et al., 2015) employed only balanced

winnow classifiers for sarcasm identification. A balanced winnow allocates scores to each class label in these studies, and good performance was obtained when area under curve (AUC) metrics were used, which showed its confidence in such a label.

In another study, random forest (RF), support vector machine (SVM), K-nearest neighbor (K-NN), and maximum entropy (ME) were used to classify sarcasm on tweets datasets using pattern related features. The performance classifier result showed that RF outperformed SVM, K-NN, and ME by attaining an accuracy of 81.3% F-measure. Ling and Klinger (2016), in their study on the 'Comparative analysis classification of differences between irony and sarcasm', compared the performance of the DT, ME, and SVM classifiers. The empirical analysis showed that the ME model performed better than the decision tree and SVM classifiers. Sulis et al. (2016) investigated the classifier performance of NB, DT, RF, LR, and SVM in modelling the differences among the three figurative messages (#sarcasm, #Not, and #Irony) on Twitter. Among these classifiers, the highest result of f-measure was obtained by applying RF classifier in distinguishing #Irony vs #Not. However, when similar datasets used in (Barbieri et al., 2014) were employed for the #Irony vs #Sarcasm classification experiment, the performance result showed an improvement of F-measure from 0.62 to 0.70.

Moreover, Abulaish and Kamal (2018) compared the performance of the NB, DT, and Bagging (ensemble) classifier to classify hyperbolic and self-deprecating features for sarcasm identification in the tweets datasets (balanced and unbalanced). They reported the performance result of the experiment in the form of precision, f-measure, and recall in applying all the three classifiers, that the DT attained the highest values in f-measure and recall. In contrast, the bagging classifier achieved the best precision value in both datasets.

2.7.4.2 Deep Learning Model

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. (2017) presented a method of automatic sarcasm identification in textual data using a Deep Convolutional Neural Network (DCNN). Their study used sentiment polarity as a feature set and extracted feature vectors 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 (2016) proposed a Deep Neural Model (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% (Schifanella et al., 2016). Similarly, Onan (2019) 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 model mentioned above'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.

Table 2.8: Classification algorithm used in the selected studies

Studies	SVM	NB	RF	ME	DT	LR	KNN	ANN/DNN	FC	RB	AB	BW
(Davidov et al., 2010)	×	×	×	×	×	×	✔	×	×	×	×	×
(González-Ibánez et al., 2011)	✔	×	×	×	×	✔	×	×	×	×	×	×
(Edwin & Ayu, 2013)	✔	✔	×	✔	×	×	×	×	×	×	×	×
(Liebrecht et al., 2013)	×	×	×	×	×	×	×	×	×	×	×	✔
(Riloff et al., 2013)	✔	×	×	×	×	×	×	×	×	×	×	×
(Ptáček et al., 2014)	×	×	×	×	✔	×	×	×	×	×	×	×
(Barbieri et al., 2014)	✔	×	×	✔	×	×	×	×	×	×	×	×
(Liu et al., 2014)	✔	✔	×	✔	×	×	×	×	×	×	×	×
(Bouazizi & Ohtsuki, 2015b)	×	×	✔	×	×	×	×	×	×	×	×	×
(Fersini et al., 2015)	✔	✔	×	×	✔	×	×	×	×	×	×	×
(Khattri et al., 2015)	×	×	×	×	×	×	×	×	×	✔	×	×
(Ghosh et al., 2015)	✔	×	×	×	×	×	×	×	×	×	×	×
(Rajadesingan et al., 2015b)	✔	×	×	×	✔	✔	×	×	×	×	×	×
(Bharti et al., 2015)	×	×	×	×	×	×	×	×	×	×	×	×
(Bouazizi & Ohtsuki, 2015a)	✔	✔	×	✔	×	×	×	×	×	×	×	×
(Altrabsheh et al., 2015)	×	✔	✔	✔	×	×	×	×	×	✔	×	×
(Wang et al., 2015)	✔	×	×	×	×	×	×	×	×	×	×	×
(Kunneman et al., 2015)	×	×	×	×	×	×	×	×	×	×	×	✔
(Ling & Klinger, 2016)	✔	×	×	✔	✔	×	×	×	×	×	×	×
(Schifanella et al., 2016)	✔	×	×	×	×	×	×	×	×	×	×	×
(Bouazizi & Ohtsuki, 2016)	✔	×	✔	✔	×	×	✔	×	×	×	×	×
(Ghosh & Veale, 2016)	✔	×	×	×	×	×	×	✔	×	×	×	×
(Amir et al., 2016)	×	×	×	×	×	×	×	✔	×	×	×	×
(Zhang et al., 2016)	×	×	×	×	×	×	×	✔	×	×	×	×
(Bharti et al., 2016)	×	×	×	×	×	×	×	×	×	×	×	×
(Sulis et al., 2016)	✔	✔	✔	×	✔	✔	×	×	×	×	×	×
(Mukherjee & Bala, 2017b)	×	✔	×	×	×	×	×	×	✔	×	×	×
(Manohar & Kulkarni, 2017)	×	×	×	×	×	×	×	×	×	×	×	×
(Al-Ghadhban et al., 2017)	×	✔	×	×	×	×	×	×	×	×	×	×
(Ranjan et al., 2017)	✔	✔	×	×	×	×	×	×	×	×	×	×
(Mukherjee & Bala, 2017a)	×	✔	×	✔	×	×	×	×	×	×	×	×
(Dharwal et al., 2017)	✔	×	×	×	×	✔	×	×	×	×	×	×
(Bharti et al., 2017)	✔	✔	✔	×	✔	×	×	×	×	×	✔	×
(Abulaish & Kamal, 2018)		✔	✔	×	✔	×	×	×	×	×	×	×
(Samonte et al., 2018)	✔	✔	×	✔		×	×	×	×	×	×	×

(Sreelakshmi & Rafeeque, 2018)	✔	×	×	×	✔	×	×	×	×	×	×	×
(Manjusha & Raseek, 2018)	✔	✔	×	×	×	×	✔	✔	×	×	×	×
(Kumar & Harish, 2018)	×	×	✔	×	×	×	×	×	×	×	×	×
(Kumar et al., 2019)	×	×	×	×	×	×	×	✔	×	×	×	×
(Suhaimin et al., 2019)	✔	×	×	×	×	×	×	×	×	×	×	×
(Babanejad et al., 2020)	×	×	×	×	×	×	×	✔	×	×	×	×
(Nayel et al., 2021)	✔	✔	×	×	×	✔	×	×	×	×	×	×
Total:	23	5	7	9	8	5	3	6	1	2	1	2

In a recent study (Castro et al., 2019a) a multimodal features consisting of textual, speech, and video features were employed to recognise Sarcasm. The textual features in the data sets were represented using Bidirectional Encoder Representation from Transformer (Devlin et al., 2018), a specification for sentence representation. On the other hand, speech feature extraction was extracted using Libnsa, a well-known library for speech extraction (Carr & Zukowski, 2019), by considering only the low-level feature for audio data to exploit the audio modality information. Also, pool five layers of an ImageNet (Deng et al., 2009) 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, (Onan & Toçoğlu, 2021) 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. In addition, 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%, indicating the proposed framework's effectiveness. It is obvious from Table 2.8 that SVM and NB are the most used classifiers for sarcasm identification in social platform. Many deep learning methods in NLP use word embedding learning as a standard approach for feature vector representation. However, one of the major

drawbacks of word embedding is that it ignores the sentiment polarity of the words (Araque et al., 2017; Giatsoglou et al., 2017; Agrawal et al., 2020). Consequently, words with opposite polarities are mapped into a close vector. Hence, this study seeks to address the issue.

2.7.5 Review of Performance Measure

The performance evaluation of sarcasm classification can be measured using various performance metrics such as accuracy (ACC), recall (REC), F-measure (F-M), precision (PR), the Area Under Curve (AUC), and Kappa Statistics (KS). In addition, the values of False Positive (FP), False Negative (FN), True Positive (TP), and True Negative (TN), which are the contents of the confusion matrix, can be used for the computation of these metrics. The detailed description and the computation of these measures are given in Section 3.7. Moreover, selecting the performance metrics depends on the goal for which sarcasm is being identified. Although this review indicated precision, accuracy, recall, and F-measure as the most employed performance metrics, these metrics may be inadequate to correctly evaluate the classifier's performance. This is because of the class imbalance in various datasets found in most selected studies. In such a situation, AUC would be the best option due to its suitability in evaluating the classification performance related to an individual class (Provost & Fawcett, 1997; Provost et al., 1998). For instance, Samonte et al. (2018) collected two sets of tweets dataset (English and Filipino) on a range of domains such as social media, politics, weather, government, and transportation to build a model for sarcasm identification in a multilingual platform.

In the study, the author employed only accuracy metrics to measure the performance of the classification. The English datasets comprised 1101 sarcastic and 13998 non-sarcastic, whereas Filipino datasets consisted of 894 sarcastic and 14229 non-sarcastic.

Here, the two sets of data are naturally imbalanced, and in such a case, there may be biases in using the only accuracy as performance metrics. Thus, the right measure to accurately determine the performance of the algorithm for sarcasm identification is AUC. In another study, Liu et al. (2014) employed two corpora to classify English and Chinese sarcasm features. The first corpus consists of Twitter, Amazon product review, and News article datasets. Among this corpus, the Twitter dataset comprised 3200 sarcastic and 36,800 non-sarcastic, Amazon product (471 sarcastic and 5020 non-sarcastic), News article (223 sarcastic and 4000 non-sarcastic). However, the second corpus consist of three Chinese topic comments crawled from Tencent Weibo (359 sarcastic and 5128 non-sarcastic), Sina Weibo (238 sarcastic and 3621 non-sarcastic), and Netease BBC (546 sarcastic and 9810 non-sarcastic). It is obvious that all the class distributions of the corpus used in the classification experiment are highly imbalanced. Thus, the authors employed Area under the curve (AUC) to measure the performance of the classification models accurately. This is because; AUC has a strong resistance to the skewness in datasets compared to the F-score when employing TPR instead of precision. The summary of the performance measure used in the selected studies is shown in Table 2.9.

Table 2.9: The frequency of performance metrics in the selected studies

Studies	ACC	PR	REC	F-M	AUC	KS
(Davidov et al., 2010)	✔	✔	✔	✔	×	×
(González-Ibánez et al., 2011)	✔	×	×	×	×	×
(Edwin & Ayu, 2013)	✔	×	×	×	×	×
(Liebrecht et al., 2013)	✔	×	✔	×	✔	×
(Riloff et al., 2013)	×	✔	✔	✔	×	×
(Ptáček et al., 2014)	×	✔	✔	✔	×	×
(Barbieri et al., 2014)	×	×	×	✔	×	×
(Liu et al., 2014)	×	×	×	×	✔	×
(Bouazizi & Ohtsuki, 2015b)	✔	✔	✔	×	×	×
(Fersini et al., 2015)	✔	✔	✔	✔	×	×
(Khattri et al., 2015)	×	✔	✔	✔	×	×
(Ghosh et al., 2015)	×	✔	✔	✔	×	×
(Rajadesingan et al., 2015b)	✔	×	×	×	✔	×

(Bharti et al., 2015)	×	✔	✔	✔	×	×
(Bouazizi & Ohtsuki, 2015a)	✔	✔	✔	✔	×	×
(Altrabsheh et al., 2015)	✔	✔	✔	✔	✔	×
(Wang et al., 2015)	✔	×	×	×	×	×
(Kunneman et al., 2015)	×	✔	×	×	✔	×
(Ling & Klinger, 2016)	✔	×	×	×	×	×
(Schifanella et al., 2016)	✔	×	×	×	×	×
(Bouazizi & Ohtsuki, 2016)	✔	✔	✔	✔	×	×
(Ghosh & Veale, 2016)	×	✔	✔	✔	×	×
(Amir et al., 2016)	✔	×	×	×	×	×
(Zhang et al., 2016)	×	✔	✔	✔	×	×
(Bharti et al., 2016)	×	✔	✔	✔	×	×
(Sulis et al., 2016)	×	×	×	✔	×	×
(Mukherjee & Bala, 2017b)	✔	✔	✔	✔	×	×
(Manohar & Kulkarni, 2017)	✔	×	×	×	×	×
(Al-Ghadhban et al., 2017)	✔	✔	✔	✔	×	×
(Ranjan et al., 2017)	✔	✔	✔	✔	×	×
(Mukherjee & Bala, 2017a)	✔	✔	✔	✔	×	×
(Dharwal et al., 2017)	×	×	×	✔	×	×
(Bharti et al., 2017)	×	✔	✔	✔	×	×
(Abulaish & Kamal, 2018)	×	✔	✔	✔	×	×
(Samonte et al., 2018)	✔	✔	✔	✔	×	✔
(Sreelakshmi & Rafeeque, 2018)	✔	✔	✔	✔	×	×
(Manjusha & Raseek, 2018)	×	✔	✔	✔	×	×
(Kumar & Harish, 2018)	✔	✔	✔	✔	×	×
(Kumar et al., 2019)	✔	✔	✔	✔	×	×
(Suhaimin et al., 2019)	×	×	×	✔	×	×
(Babanejad et al., 2020)	×	×	×	✔	×	×
(Nayel et al., 2021)	✔	✔	✔	✔	×	×
** **ACC**= accuracy, **PR**=Precision, **REC**=recall, **F-M**=F-measure, **AUC**=Area under the curve, **KS**=kappa statistics						

2.8 Research issues of sarcasm identification approach in the existing literature

In this subsection, the issues found in the review of the existing studies on sarcasm identification approaches are discussed. The major issues found in the literature are related to the datasets, feature engineering, and performance metrics. These require considerable further investigation to create an efficient classification model in the domain

of sarcasm identification. The discussion of these issues is provided in the subsequent Section, and it stands as a starting point for the presented study in this thesis.

2.8.1 Issues Related to the Datasets

One of the major issues in the sarcasm identification domain is the lack of a standard dataset. There is a scarcity of publicly available datasets for sarcasm identification, which has made most researchers create privately owned datasets. However, when there is the availability of the public dataset (Twitter), the authors, in some cases, provide only the tweets IDs, and users, in most cases, find it difficult to access the dataset. For instance, Jia et al. (2019) studied sarcasm detection using a deep learning approach intended to use (Ptáček et al., 2014) dataset. Still, they failed to recollect the dataset with the tweet ID since the author provided only the tweets ID.

Consequently, this situation has resulted in the biases of the data since most of the training and testing sets are created by the researchers, thus leading to the scarcity of the standard data that can be used for comparison purposes with the proposed framework to evaluate the unbiased in terms of the performances. In most cases, there is also an imbalance in the class distribution of the datasets, which makes the number of sarcastic text data and non-sarcastic correspond not to the same size. In addition, the misspelling of words has also become a common mistake in microblogs while composing a textual message. Humans, without any effort, can quickly correct such errors manually, but it is challenging for machine learning to detect and correct such misspelt words. However, such words can correspond to a specific dictionary that has been removed during the pre-processing stage. Thus, it can drastically influence the sentence polarity. Not only that, machine learning could ignore such wrongly spelt words and replace them with closely related ones. Notwithstanding, such errors are pervasive in sarcasm detection.

Furthermore, people have also been familiar with using emotional symbols like emoji and emoticons in social media to display their state of mind, especially in a microblog that restricts the number of characters per chat. Ambiguity is likely to occur among the users with regards to the specific meaning of emoji. Thus, it can change the overall sentiment of the sentence as the emoji features are not incorporated into most of the current system. Addressing the issues related to the dataset calls for the publications of the standard datasets by researchers and making them available and accessible by the researchers, which will solve the problem of biases in the data. Secondly, the application of AUC performance metrics, suitable for evaluating the classifier's performance in the imbalanced datasets, is required. Thirdly, a technique that can detect and correct the misspelt word and investigate and incorporate emoji and emoticon features in sarcasm detection studies is also needed.

2.8.2 Issues Related to the Feature Engineering

Feature engineering is the core aspect of any text classification as it improves the performance of the predictive models in any classification task (Domingos, 2012). Most of the reviewed studies attempted to classify tweets into sarcastic or non-sarcastic by proposing several discriminative features (see Section 2.6.3, Table 4). However, the review showed that most of these studies had proposed various feature engineerings techniques such as the N-gram technique, BoW techniques, and word embedding for sarcasm identification in social media (Zhang et al., 2010; da Silva et al., 2014; Prasad et al., 2017). The experimental results show that these techniques did not attain optimum performance due to some inherent limitations. One, those traditional feature engineering techniques only are not adequate to extract discriminative features for sarcasm classification. This is because such techniques have always focused on expression contents only, leaving the contextual information in isolation, enhancing the predictive performance. Besides, the content-based features obtained using either the bag-of-words

or N-gram based feature engineering technique relies on word use and sentence in general in identifying sarcastic and non-sarcastic utterance in a sentence, leading to the dependence of the algorithm performance on the content-based features, which will degrade the model performance (Mukherjee & Bala, 2017a). Thirdly, the BoW features are extremely imbalanced when combined with other features due to their high dimension in nature (Jia et al., 2019), which will result in the BoW feature dominance in the classification performance. Another issue found in the related techniques is the sparsity of training data. Due to the word limit of microblog, the feature vector's value for each sample constructed by BoW produces a null feature, making the modelling data sparse. This study proposed a Multi-feature fusion framework for sarcasm classification using Twitter data to address the aforementioned problem. A study proposed by Jia et al. (2019) on Chinese irony detection maintained that the feature fusion approach outperformed the traditional BoW features.

2.8.3 Issues Related to the Performance Metrics

In sarcasm identification research, the review of the existing studies revealed that precision, recall, accuracy, and f-measure are mostly used performance metrics to evaluate the performance of the proposed techniques (Davidov et al., 2010; Riloff et al., 2013; Bouazizi & Ohtsuki, 2016; Samonte et al., 2018). See Section 2.6.5. Most of these studies reported an enhanced performance of the computational model when those metrics were employed without giving the dataset's class distribution details. For instance, the Riloff dataset (Riloff et al., 2013), the first publicly available dataset for sarcasm detection, consists of 308 sarcastic and 1648 non-sarcastic. Thus, the dataset is not balanced in class distribution. In their study on sarcasm analysis, the authors employed precision, recall, and f-measure to evaluate the performance of the proposed method. However, these metrics may not be adequate to measure the performance of the model accurately.

In such an instance, AUC would be the best choice of the metrics due to its suitability in evaluating the classification's performance related to an individual class (Provost & Fawcett, 1997; Provost et al., 1998). Besides, AUC has a strong resistance to skewness in datasets using the true positive ratio (TPR) compared with F-Measure. Thus, it must be used along with other performance metrics when there is class imbalance distribution to measure the proposed technique effectively. In addition, the use of sampling techniques such as SMOTE or over-sampling is another option to balance the distribution in the dataset when there is an imbalance in the dataset (Japkowicz, 2000; Tang & Liu, 2005).

Table 2.10: Summary of the Issues in the existing studies

Research issues categories	**Issues**	**References**
Datasets	Scarcity of comprehensive dataset.	Jia et al. (2019)
	Tweets IDs only is provided by the dataset owner when made available, and users in most cases find it difficult to access the dataset.	(Ptáček et al., 2014; Subramanian et al., 2019)
	Imbalance in dataset class distribution.	(Banerjee et al., 2020)
	Ambiguity among the users with regards to the specific meaning of emoji	(Malave & Dhage, 2020)
Feature Engineering Techniques	BoW model ignores the semantic and context of words, training data sparsity issue	(Khodak et al., 2017; Jia et al., 2019)
	N-gram also suffers data sparsity and high dimensionality problem	(Hazarika et al., 2018)
	Word embedding ignores the sentiment polarity of the word	(Araque et al., 2017; Giatsoglou et al., 2017).
Performance metrics	Omission of AUC performance metrics when there is an imbalance in the distribution feature class	(Riloff et al., 2013)

Based on the identified literature gaps from the review of the existing studies, this research focuses on feature engineering techniques by proposing a Multi-feature fusion framework that contains sentiment related features, word embedding features, and other types of contextual features for sarcasm identification in Twitter. This is to efficiently

learn the contextual features to improve the sarcasm detection performance and address the context of words, sentiment polarity and training data sparsity issues in sarcasm expression.

2.9 Chapter Summary

The Chapter presents a comprehensive review of classification techniques for sarcasm identification on the social media platform. The review covered the aspects of datasets usage, pre-processing techniques, feature engineering techniques (consisting of feature extraction, representation, and selection), the classification approach, and the performance metrics. The study showed few standard and publicly available datasets for sarcasm identification in social microblogs such as Twitter so that researchers are required to crawl their datasets. Content-based features were mainly used features, whereas N-gram and POS tagger were the most used feature extraction techniques due to their simplicity in usage. BR and TF were the most used feature representation schemes in the selected studies. BR technique is very effective in sentiment feature representation, as the sarcasm is checked on the textual data. For example, sentiment one indicates sarcasm in the sentence, whereas sentiment 0 indicates the absence of sarcasm. TF was also used to check the frequency of occurrence of the feature in the training sets; this can increase the likelihood of the feature in the test set.

To eliminate the non-discriminative features, various studies applied feature selection schemes such as Chi-squared and Information gain. Most studies applied supervised machine learning algorithms such as SVM, NB, RF, ME, and DT in the classification phase. The review showed that the SVM algorithm is mainly used, followed by NB, RF, and ME. This is so because it obtained better results compared to other classifiers. Only a few studies used rule-based, and NLP approaches. A deep learning approach has gained ground in sarcasm identification in recent studies because learning and feature

engineering is done automatically without human intervention. Performance metrics such as precision, recall, accuracy, and F-measure were used as a performance measure to measure the classification algorithm's performance. It was found that accuracy was mostly used in the selected studies. However, relying only on the accuracy of performance measures will not produce a better result when imbalanced datasets are used. Hence, AUC is a more suitable metrics for performance measures where there are datasets imbalances.

A comprehensive investigation of datasets' characteristics, types, strengths, and weaknesses for sarcasm identification in the social media textual data was carried out. In addition, outline taxonomy, various features representation, and extraction for efficient algorithm development are presented. The survey also analyzed various data preparation (pre-processing) techniques and recent classification algorithms for sarcasm identification. Finally, to set the pace for developing the new ground, the study identifies current research issues and provides suggestions to address some of the issues in the sarcasm identification domain. However, the review revealed that most studies on sarcasm identification have always focused on the content-based features, leaving the contextual information in isolation and failing to capture the semantics or meaning of words in the expression, which could enhance the predictive performance. Thus, it is important to explore a Multi-feature framework incorporating contextual information with the content-based feature to improve the sarcasm classification performance using Twitter data.

CHAPTER 3: RESEARCH METHODOLOGY

3.1 Introduction

This Chapter provides a general research methodology for the proposed multi-feature fusion framework for sarcasm identification in Twitter data. Lately, several methods have been employed for sarcasm detection in textual data. However, these methods inherent some issues that hindered them from achieving optimum performance (see Section 2.8). Thus, this research proposes a multi-feature fusion framework to minimize the issue and enhance the sarcasm classification's predictive performance using a machine learning approach. The feature fusion framework explores different discriminative features that can improve the predictive performance in sarcasm classification. The features were combined to form cumulative features (feature fusion) to address the context of words and data sparsity issues in classifying sarcastic text. As a result, the developed framework produced enhanced performance compared with the existing methods with minimal resources and less computational time.

The research methodology of this study is based on the experimental quantitative study. A quantitative study is a type of study that establishes and solves the problem using numerical data. The quantitative study is built on quantity measurement and emphasises collecting, analyzing, and experimenting on data to conclude (Hoy & Adams, 2015).

To achieve this research's main goal through the objectives specified, the study shall adopt the following research methodologies using the proposed framework to realize all the objectives for meeting the aim and ultimately answer the research questions. There are seven different phases of methodology design for this research, as shown in Figure 3.1. A concise description of these phases is presented in the subSection below. However, the specific research methodology and the working of the proposed framework is extensively described in Chapter 4 of this thesis.

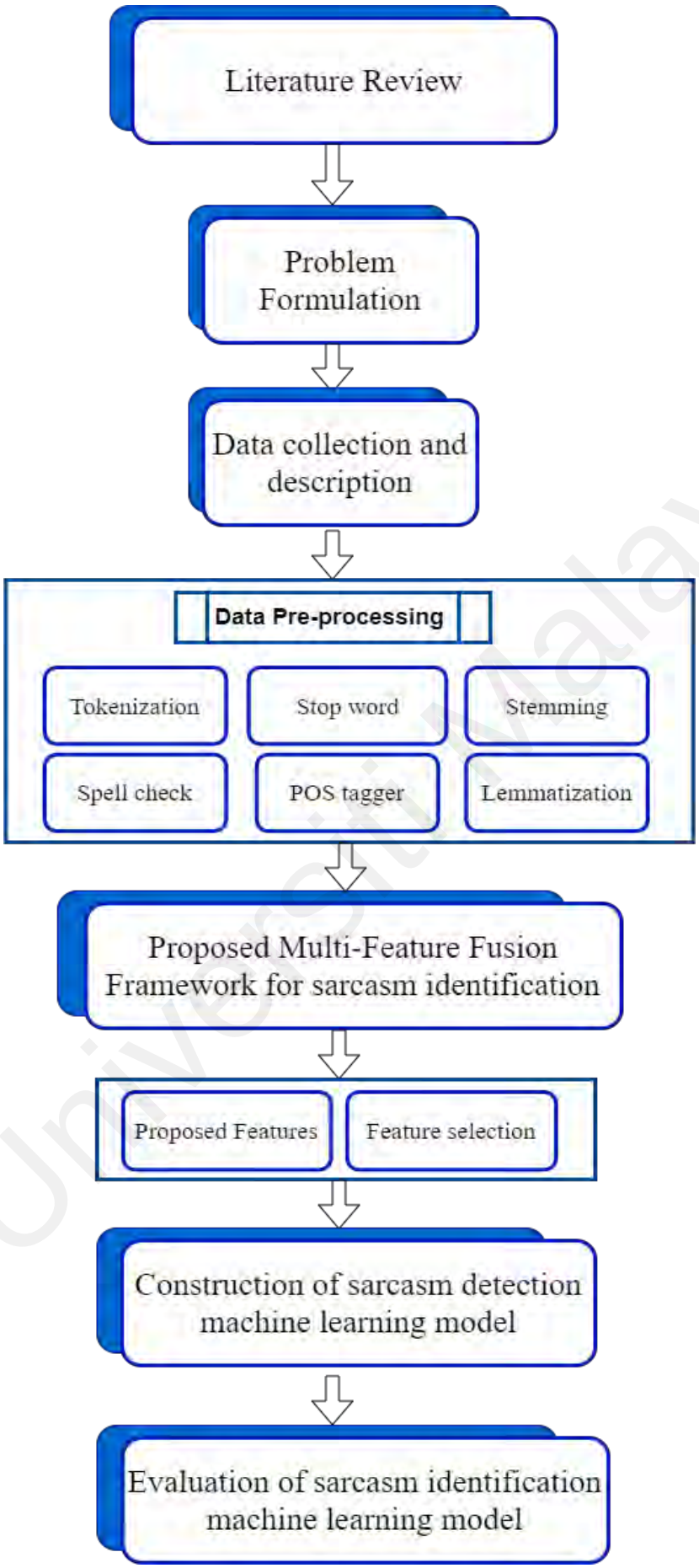

Figure 3.1: Detailed Research Methodology

3.2 Review of Related Literature

The first step to undertake in this study is to survey existing literature on the sarcasm identification domain. In this study, a review of academic literature in the domain of sarcasm detection was carried out under dataset usage, pre-processing techniques, feature engineering techniques, the modelling approach, and performance metrics. Six academic databases (including Science Direct, IEEE Xplore, and ACM) were systematically selected and extensively reviewed under the six aspects mentioned above. Based on the literature survey, the existing approaches for the sarcasm identification task, strengths, and drawbacks were identified. However, the literature survey recognised the research gap and methodological framework that includes data collection, pre-processing, feature engineering, classification model construction, and evaluation of the constructed model for sarcasm identification.

Moreover, feature engineering and information fusion approaches were extensively investigated. Based on the investigation, it has been noticed that effective sarcasm identification requires the development of a framework with the fusion of multiple discriminative features and the development of a context-based feature technique for sarcasm classification. The literature survey process also identified research gaps and various limitations described in the problem identification and formulation Section (see Section 2.8).

3.3 Problem Formulation

Various studies have employed different feature engineering approaches such as bag-of-Word (BoW) and N-gram and word embedding for sarcasm identification (Zhang et al., 2010; da Silva et al., 2014; Prasad et al., 2017). For instance, Dave and Desai (2016) experimented with traditional BoW techniques for feature extraction for sarcasm detection study on textual data. SVM classifier to train the model and attained an accuracy

of 50%. The predictive performance result revealed that the traditional BoW technique is inadequate to extract discriminative features for sarcasm identification. Even though few studies have implemented conventional text classification-based feature engineering methods for sarcasm detection, literature studies on sarcasm identification (Al-Sallab et al., 2017; Prasad et al., 2017; Xiao et al., 2018; Jia et al., 2019; Chia et al., 2021) reveals that most current techniques face various issues that need to be resolved to improve sarcasm classification framework. These include one, loss of contextual information on sarcasm expression. The BoW technique ignores the context of the words in representation since it is only concerned with their occurrence. This leads to loss of contextual information and, in turn, the meaning of words in the expression. Consequently, different expressions can possess a similar vector representation: two, the sparsity of training data. Due to the word limit of microblog, the feature vector's value for each sample constructed by BoW is mostly 0, making the training data-sparse (Hazarika et al., 2018; Kapil & Ekbal, 2021). This issue can create a severe problem during the model training because some words could be seen in the testing set but not found in the training set, making most of the training features sparse. Three, many deep learning methods in NLP use word embedding learning as a standard approach for feature vector representation. However, one of the major drawbacks of word embedding is that it ignores the sentiment polarity of the words (Araque et al., 2017; Giatsoglou et al., 2017; Agrawal et al., 2020). Consequently, words with opposite polarities are mapped into a close vector.

A Multi-feature Fusion Framework for sarcasm Identification in Twitter data that uses two classification stages, which enhances the predictive performance of classifying sarcastic text, is required to address those problems. Therefore, it is important to explore more techniques to overcome this drawback. Hence, there is a need to carry out this research.

3.4 Dataset Collection and Description

The sarcasm identification process begins with the acquisition of a suitable dataset. Dataset is very crucial in any data mining study. A live streaming dataset was collected from Twitter using Automatic Retrieval of Tweets using the Keywords (ARTK) for sarcasm classification purposes. The Dataset acquisition was carried out by using the Twitter streaming API for both sarcastic and non-sarcastic collections. Data collection for this research took place between June 2019 and September 2019. Twitter is a leading microblog site that enables users to exchange their ideas, news, and emotion with their co-users. One of the major advantages of Twitter data is that one can collect as many tweets as possible because people post messages daily. The Twitter application program interface (API) provides a connection between Twitter servers and users to make archived tweets easily accessible. API facilitated the extraction of public tweets. Each of the tweets extracted using the API provides extensive information about the users. (Kwak et al., 2010), This includes the user identification, URL, user name, user account information, and tweet text (the major textual data required for the analysis as it contains the emotional, behavioural, and other information and thoughts) (Eichstaedt et al., 2015). This information has been utilized to construct a feature set for the effective classification of Twitter data (Eichstaedt et al., 2015; Preoţiuc-Pietro et al., 2015). It has also been used to develop a proposed multi-feature fusion framework by identifying the machine learning model's significant features for model training in differentiating between the sarcastic and non-sarcastic expressions.

To build the datasets of sarcastic and non-sarcastic, self-annotated tweets by tweets owners were streamed from Twitter and utilized. Tweets expression having the hashtag '#sarcasm' or '#sarcastic' is considered sarcastic, a similar concept used in (Schifanella et al., 2016; Mukherjee & Bala, 2017b). However, tweets without such hashtags are considered non-sarcastic by following the same concept utilized in (Sreelakshmi &

Rafeeque, 2018) or tweets with keywords #notsarcasm or #notsarcastic (Mukherjee & Bala, 2017b). In this research, balanced tweet datasets of 29,931volume of tweets that contained 15,000 sarcastic and 14,931 non-sarcastic tweets are used for the analysis. The summary of dataset1 is depicted in Table 3.1.

Table 3.1: Summary of Dataset

Data source	Twitter
Data collection approach	Automatic retrieval of tweets using keywords (ARTK)
Language of tweets	English
Data classes	Sarcastic and non-sarcastic
Search period	Between June 2019 and September 2019
Sarcastic data volume	15,000
Non-sarcastic data volume	14,913
Total volume of data	29,931
Annotation	Self-annotated by tweet owner
Sarcastic annotation	'#sarcasm' or '#sarcastic'
Non-sarcastic annotation	#notsarcasm or #notsarcastic or without any hashtag

3.5 Data Pre-processing

One of the drawbacks of obtaining data set from Twitter is the noise that comes along with the data. Twitter data (tweets) may be in the form of simple text, user's mentions (@user), and reference to URLs or a content tag, also known as hashtags (#). In this stage, the sarcastic and non-sarcastic data were pre-processed to prepare before the feature extraction and classification task. This is carried out in various steps to remove noise from the sarcastic datasets, including retweets, duplicates, numerals, tweets written in other languages, and tweets with the only URL. These noisy data do not contribute to the enhancement of classification accuracy and are, therefore, eliminated. In addition, the text data were converted to the lower case, and other basic pre-processing techniques such as tokenization, stop word removal, spell check, stemming, lemmatizing. POS tagging were also employed, which were implemented using Python library and Natural Language

Processing (NLP) toolkit. They are briefly described below, and the flowchart is depicted in Figure 3.2.

- ✓ ***Tokenization****:* This is a process of a splitting sequence of words or sentences into smaller chunks called tokens, such as words, phrases, and symbols that are useful on their own. The tokenization process also eliminates the empty white space characters found in textual documents. A token refers to a sequence of characters found in a particular document joined together to create an appropriate semantic unit useful later during the analysis. Thus, the tokenization output becomes an input for further future analysis. Tokenization tasks can be performed using the NLP toolkit.
- ✓ ***Stop word removal***: These are common words that consist of articles and prepositions (such as a, an, the, etc.) that do not influence the context of the expression and do not have any contribution to the text analysis. NLTK corpus stop word was employed to remove the stop word from the data set. It should be noted that empirical analysis was performed to examine the model performance in the existence or nonexistence of stop words in the text. The reason is that few studies on text classification indicated that the absence of stop words reduces the performance of the classification (Sarker & Gonzalez, 2015; Lauren et al., 2018). Contradictorily, several pieces of research demonstrated that the existence of stop words in the text reduces classification performance. In our study, the experimental analysis of the stop word indicated that stop words lower the performance results due to the noisy factor (Jo, 2013; Adeva et al., 2014; Sarker & Gonzalez, 2015). Therefore, stop words were eliminated, in turn, to improve the classification results. The pre-processing phase is an input to the next classification phase, known as the feature engineering phase.
- ✓ ***Spell correction***: This is a process of checking for the spelling of the text to correct the wrong spelt text. A PyEnchant (Bird et al., 2009) spell checker python library was employed to correct all the misspelt words.

- ✓ ***Stemming:*** Stemming is restoring the derived words into their root form or obtaining the root word called the stem by removing the prefixes and suffixes from the word. The stemming process reduces the keyword space's number and enhances the classification performance when a single keyword is obtained from different keywords. For example, the word 'stealing' can be stemmed to 'steal'. However, the Port stemmer library was employed for the word stemming task. Various studies stated that the stemming procedure contributes to classification performance (Buchan et al., 2017; Wang et al., 2017). Thus, the stemming process was performed in this study to enhance the classification performance.
- ✓ ***Lemmatizing***: The removal of prefixes and the suffixes in a derived word sometimes render the word meaningless. Lemmatization is another normalization technique that truncates the inflectional of a word using morphological and vocabulary analysis of a particular word to transform it into a dictionary form. Lemmatizer, therefore, inputs the missing characters to the stemmed word to bring meaningfulness out of it. This procedure normalizes the word into basic forms. Unlike stemming, lemmatization does not yield the word stem but substitute the suffix of the input word with a different word to generate its normalized form. For instance, the word 'concluded' can be stemmed to the word 'conclud', which can then be lemmatized to the 'conclude.'
- ✓ ***Parts-Of-Speech (POS) tagging***: POS tagger reads the textual documents and allocates parts of speech to each token based on its definition. The tagger allocates various parts of speech such as verb, noun, adverb, adjectives, conjunctions, interjections, etc. Most computational sciences application needs fine-grained POS tagging. For instance, noun tagging can exist in different forms, such as singular nouns, possessive nouns, and plural nouns. POS tagger uses various notations. For example, NN notation represents a singular common noun, NNS represents plural

common nouns and NP for a singular proper noun. However, the POS tagger for tagging uses stochastic and rule-based algorithms.

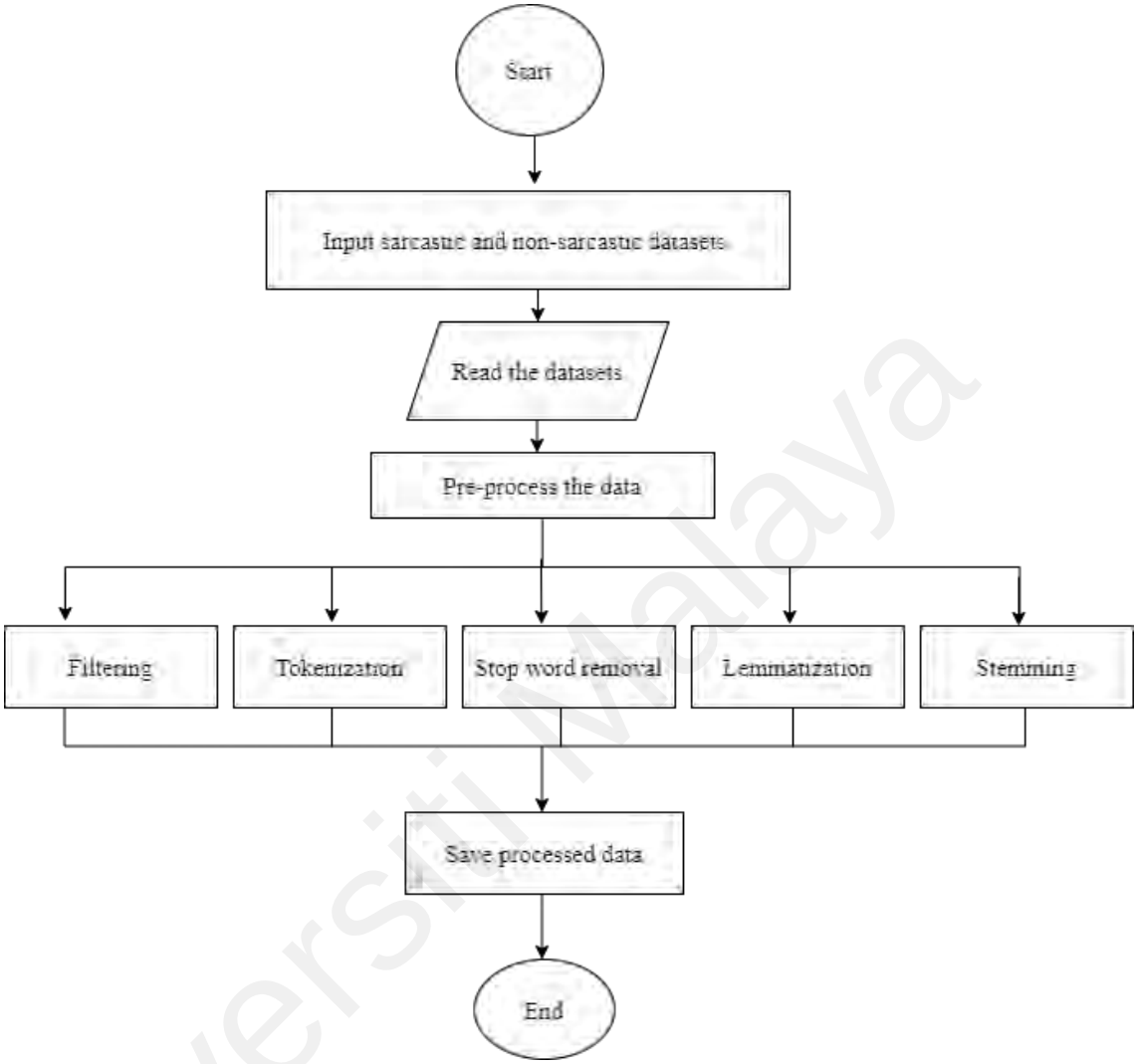

Figure 3.2: Data pre-processing flowchart

3.6 Proposed Multi-feature fusion framework for Sarcasm Identification

This Section described the methodology employed for the development of a multi-feature fusion framework for sarcasm identification. The proposed framework aims to improve the sarcasm identification performance to attain effective sarcasm classification and address the drawbacks described in the problem statement (Section 1.3). The framework uses a varied feature set to develop feature fusion and model a context using a machine learning algorithm.

One of the key processes in sarcasm identification is feature engineering. The quality of features employed for the classification task determines the degree of performance. One of the major issues in the existing techniques for sarcasm identification tasks is the reliance on the content-based features only (Mukherjee & Bala, 2017a), leaving the contextual information in isolation. For instance, (Dave & Desai, 2016) experimented with traditional bag-of-words techniques to extract features during their study on sarcasm detection on textual data. They employed a support vector machine classifier to train the model and attained an accuracy of 50%. However, the predictive performance result revealed that the traditional bag-of-words model is inadequate to extract the discriminative features for sarcasm identification. The brain behind the low performance is that it ignores the context of the word in sarcastic expression, coupled with the hashtags, jargon, and emoticons that surround social media data (Prasad et al., 2017). In addition, the N-gram-based technique relies on word use and sentence, in general, to identify sarcastic and non-sarcastic words in a sentence, leading to the dependence of the algorithm performance on the content-based features, which will degrade when applied to other datasets (Mukherjee & Bala, 2017a). Even though few studies have implemented conventional text classification based feature engineering methods for sarcasm detection, literature studies on sarcasm identification (Al-Sallab et al., 2017; Prasad et al., 2017; Xiao et al., 2018; Jia et al., 2019) reveals that most current methods face various limitations that need to be resolved to improve sarcasm classification framework. These include the loss of contextual information (context of the word being ignored); two, the training data sparsity issue. Three, 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.

Therefore, a Multi-feature fusion framework is proposed to overcome the limitations of existing approaches by addressing the problem of the context of words and data

sparsity issue in expression for sarcasm classification, which will help improve the sarcasm detection performance realize effective sarcasm classification. The layout of the framework is depicted in Chapter 4, Figure 4.1. The methodology employed for the feature fusion framework consists of five processes, such as data collection, data pre-processing, proposed features, feature fusion process, construction of sarcasm classification model, and evaluation of the constructed sarcasm classification models. A brief description of each process is presented in the subsequent Section, while the detailed process with the algorithm and experimentation is presented in Chapter 4 of this thesis. The fusion framework was developed using the Twitter dataset. However, the data collection and pre-processing stage have been described in Sections 3.4 and 3.5.

3.6.1 Proposed Set of Features

In this study, some discriminative features for effective sarcasm identification are proposed and extracted from the processed data. Proposing a discriminative set of features is the main step in constructing an effective classification model in the various application domains (Libbrecht & Noble, 2015). One of the major contributions of this research to the literature is the extracted sarcastic features used to formulate feature fusion utilized to construct a sarcasm identification model with high predictive results. Nine different kinds of features that consist of lexical, pragmatics, sentiment, emoticon, hashtag, discourse markers, syntactic, length of microblog and semantic (word embedding) features are extracted from the processed dataset. The extracted features which are extensively described in Chapter 4 (Section 4.3), were employed together with a machine learning model to construct a sarcasm detection model.

3.6.2 Feature Selection Algorithm

This study investigates the effect of the feature selection algorithm. Two feature selection algorithms were chosen: Pearson correlation and information gain to determine

the features with discriminating ability (Yang & Pedersen, 1997). As described in Chapter 2 of this thesis, this feature selection algorithm is most widely used to select discriminative features. This report provides a detailed explanation of this feature selection algorithm in Chapter 2 (Section 2.5.3.1). However, the feature selection aims to select the subsets of features from the proposed features to form the feature fusion used in the classification phase. The process of feature selection scheme helps to eliminate the redundant features and also reduce the computational resources. Redundant features are those features that do not contribute to differentiating classes from each other. They can thus be removed without incurring much loss of information.

3.7 Construction of Machine Learning Model

In this stage, various classification algorithms employed to construct the sarcasm detection model on the proposed feature fusion framework were selected. The output of the feature fusion process produces fused features. The fused feature is then employed to input the machine learning algorithm to construct a classifier to train on the feature fusion. However, the decision on choosing the best classifier for a particular dataset is quite challenging. In the existing studies, two or more machine learning algorithms are tested to find the best algorithm since it is difficult to find a single classifier that can attain the best performance in all application domains (Wolpert & Macready, 1995). This is because of the variations in the philosophy of the learning process. Thus, five different classifiers that include the Decision tree, Support Vector Machine, Logistic Regression, K-Nearest Neighbor, and Random Forest, has been employed to determine the model performance of the feature fusion framework for sarcasm identification. As a guide in selecting a machine-learning algorithm to be utilized in this study, three points have been employed to scale down the selection. First, specific literature on the classification algorithm for sarcasm detection is essential in selecting specific classifiers.

The distinction of the machine learning model may be restricted to a particular domain (Macià et al., 2013). Therefore, the literature survey carried out in Section 2 serves as a guide in selecting the classifier. Second, a text mining study review was also used to guide model selection (Sebastiani, 2002; Korde & Mahender, 2012). Third, the comparative results on comprehensive datasets are also guided in selecting classification algorithms (Fernández-Delgado et al., 2014). Thus, the machine learning algorithm such as SVM, KNN, RF, DT, and LR classifiers (Hall, 1999) has been tested in the proposed Multi-feature Fusion Framework. An extensive description of the algorithm is reported in the subsequent Section. In this study, a feature analysis and selection scheme was investigated to identify and select the discriminative features and eliminate redundant features that do not contribute to the classification results. An extensive description of the algorithm is reported in Chapter 2.

3.8 Development of Feature Fusion framework

The feature fusion is developed based on the proposed sets of features. Among the features constructed, the lexical feature is extracted based on the BoW technique that uses TF-IDF, resulting in the dimension of lexical features. Firstly, classifiers are trained based on the lexical feature extracted by the bag-of-word model to obtain a prediction. Next, other groups of features that consist of a length of microblog, hashtag, discourse markers, emoticons, syntactic, pragmatic, semantic (GloVe embedding), and sentiment related features are extracted accordingly and fused with a lexical feature one after the other until all the features were added to test the effect of each of the feature in the fusion framework. Lastly, in this step, the fused feature is utilized as an input to the classifiers and trained on the fused feature to obtain the fused feature's performance. The fused features can capture both the semantic, sentiment polarity and contextual information from the sarcastic expression due to some contextual features such as hashtag feature, discourse marker, GloVe embedding, emoticon, and sentiment related features. In this

experimented process, the contextual information issues are resolved. Lastly, the feature selection technique was performed on the lexical feature to obtain the top 200 discriminative features. The process eliminated the null features, also referred to as redundant features.

Next, the feature selection technique was also performed on each of other feature sets such as hashtag, emoticons, syntactic, pragmatic semantic (GloVe embedding), and sentiment related features to check the discriminating power of each subset and eliminate the redundant feature. Next, the features selected from the lexical, hashtag, emoticons, syntactic, pragmatic semantic (GloVe embedding), and sentiment related features were fused to discourse marker and length of microblog to form a new fused feature. Thus, the new fused feature was employed to train the model based on feature selection. This experimental procedure resolved the training data sparsity problem, and all the null features were eliminated before the modelling phase. However, the feature selection was performed by using two feature selection algorithms. Firstly, by using the Pearson correlation algorithm, and secondly, the information gain feature selection algorithm. Thus, the feature fusion classification with feature selection algorithm obtained improvement results over the feature fusion classification without feature selection and lexical-based feature classification, which shows the significance of our proposed feature fusion framework for sarcasm identification.

3.9 Evaluation of Machine Learning Model

Various experiments were performed to measure the multi-feature fusion framework's efficiency for sarcasm identification on the dataset. This study utilized ‘Precision’ as the major performance evaluation. However, other performance metrics include recall, f-measure, and accuracy, have been employed as a supplemental to evaluate the framework's performance. As described in Chapter 2 (2.8.3; Issues Related to Evaluation

Metrics), the selection of evaluation metrics should be thoroughly considered not to obtain misleading evaluation results. This issue is commonly found in machine learning tasks with an imbalance in the dataset's class distribution. In such a situation, AUC metrics (Dobbins et al., 2017) are the best option because of their robustness compared with recall, accuracy, precision, and f-measure in class imbalance situations. This study also employed a 10-fold cross-validation experimental approach during the evaluation phase. In that approach, the initial dataset is arbitrarily separated into two exclusive portions, whereas one portion is used for training the algorithm and the other for testing. A detailed discussion of the performance metric is presented in Chapter 2, Section 2.5.7 of this thesis. Lastly, the proposed feature fusion framework was used to compare four state-of-the-art baseline approaches on the sarcasm identification task. Thus, the evaluation aims to know how suitable and adequate the proposed framework identifies sarcasm and examines which approach is more appropriate in classifying text as sarcastic.

3.10 Chapter Summary

This chapter discussed the general research methodology deployed to implement and evaluate the proposed multi-feature fusion framework for sarcasm identification. The Chapter began by describing the survey of literation that led to the formulation of the problem. Next, a discussion on the datasets employed for the study is provided, followed by the pre-processing techniques employed on data preparation and normalization. Moreover, the methodology for the proposed Multi-Feature Fusion Framework for sarcasm identification was described, including the proposed features and feature fusion process. Furthermore, the sarcasm classification model's construction for the proposed framework was described along with its components. Finally, the feature selection algorithms and evaluation measures deployed to measure the effectiveness of the proposed framework was also presented. However, the details of the proposed framework and its contributions to sarcasm detection studies are provided in Chapter 4 of this thesis.

CHAPTER 4: MULTI-FEATURES FUSION FRAMEWORK FOR SARCASM IDENTIFICATION USING CONTENT AND CONTEXTUAL FEATURES

4.1 Introduction

This Chapter provides a detailed development of the proposed framework for sarcasm identification in Twitter data. It presents a Multi-feature Fusion Framework for sarcasm identification to enhance the predictive performance and overcome the limitations mentioned above in the most related techniques by addressing the context of words and data sparsity and sentiment polarity issues in sarcasm expression.

However, the substantial contributions are the proposed and extraction of various sets of features from Twitter that consist of lexical, length of microblog, hashtag, discourse markers, emoticon, syntactic, pragmatic, semantic (GloVe embedding), and sentiment related features, which are selected based on observations from the characteristics of the data and evidence from the literature. In addition, a multi-feature fusion was developed, and finally, the feature fusion was employed to construct the classification model for sarcasm identification.

The remainder of this Chapter is structured into six (6) Sections. Section 4.2 presents the proposed multi-feature fusion framework. In Section 4.3, feature extraction is presented. Section 4.4 gives the detailed development of the multi-feature fusion process. Feature analysis and selection is described in Section 4.5. Section 4.6 give a detailed experimental design, while Section 4.7 summarizes the Chapter.

4.2 Proposed Multi-Feature Fusion Framework for Sarcasm Identification

This Section describes the multi-feature fusion framework for sarcasm identification. In this framework, the Automatic Retrieval of Tweets using a Keyword (ARTK) approach was employed to acquire the dataset utilized in this study. The dataset undergoes the pre-processing stage, as explained in Chapter 3, Section 3.5. As presented in Section 4.3 of

this Section, various sets of proposed features are extracted from the processed data. The output of the extracted features is employed to develop a Multi-Feature Fusion Framework using two classification stages that use the lexical feature only in the first stage and the fusion of lexical feature and eight other features in the second stage. However, the decision to choose the best classifier for a particular dataset is quite challenging. In existing studies, two or more machine learning algorithms are tested to find the best algorithm since it is difficult to find a single classifier that can attain the best performance in all application domains (Wolpert & Macready, 1995). This is because of the variations in the philosophy of the learning process. Thus, five different classifiers: DT, SVM, LR, K-NN, and RF, have been employed to assess the feature fusion framework's model performance. As a guide in selecting classifiers utilized in this study, three points have been used to scale down the selection: one, the specific literature on the classification algorithm for sarcasm detection helped in classifiers selection.

The machine learning model distinction may be restricted to a particular domain (Macià et al., 2013). Therefore, the literature survey carried out in Section 2 serves as a guide in selecting the classifier. Two, a text mining study review was also used as a guide for model selection (Sebastiani, 2002; Korde & Mahender, 2012). Third, the comparative results on comprehensive datasets also guided selecting the classification algorithm (Fernández-Delgado et al., 2014). Thus, classifiers, including SVM, KNN, RF, DT, and LR classifiers found in WEKA (Hall et al., 2009), were tested in this feature fusion framework. The extensive description of the algorithm is reported in the subsequent Section. In this study, feature analysis and selection schemes were also investigated to identify and select the discriminative features and eliminate the redundant ones that do not contribute to the classification results.

Various experiments were performed to measure the efficiency of the proposed framework for sarcasm identification with the dataset. This study utilized 'Precision' as the major performance evaluation. However, other performance metrics such as recall, f-measure, and accuracy have been employed as a supplement for the framework evaluation.

The selection of evaluation metrics should be thoroughly considered in order not to obtain a misleading evaluation result. This issue is commonly found in machine learning tasks where the dataset is an imbalance in the class distribution. Consequently, the AUC metric is the right choice in such a case because of its robustness compared with recall, f-measure, accuracy, and precision in class imbalance situations. However, the dataset utilized in this study is balanced. This study also employed a 10-fold cross-validation experimental approach during the evaluation phase. The initial dataset is arbitrarily separated into two exclusive portions in that approach, whereas one portion is used for model training and the other for model testing. The discussion of the performance metric is presented in Section V subSection A of this report. Lastly, the proposed feature fusion framework was used to compare four state-of-the-art baseline approaches on sarcasm identification. Thus, the evaluation aims to know how suitable and adequate the proposed framework identifies sarcasm and examines which approach is more appropriate in classifying text as sarcastic or non-sarcastic. The proposed framework is developed to overcome the limitations mentioned above of most related techniques by addressing the context of words, the training data sparsity and sentiment polarity issues for sarcasm classification. The proposed framework is depicted in Figure 4.1. The detailed discussion of data collection and preprocessing components are provided in Chapter 3, Section 3.4 and 3.5.

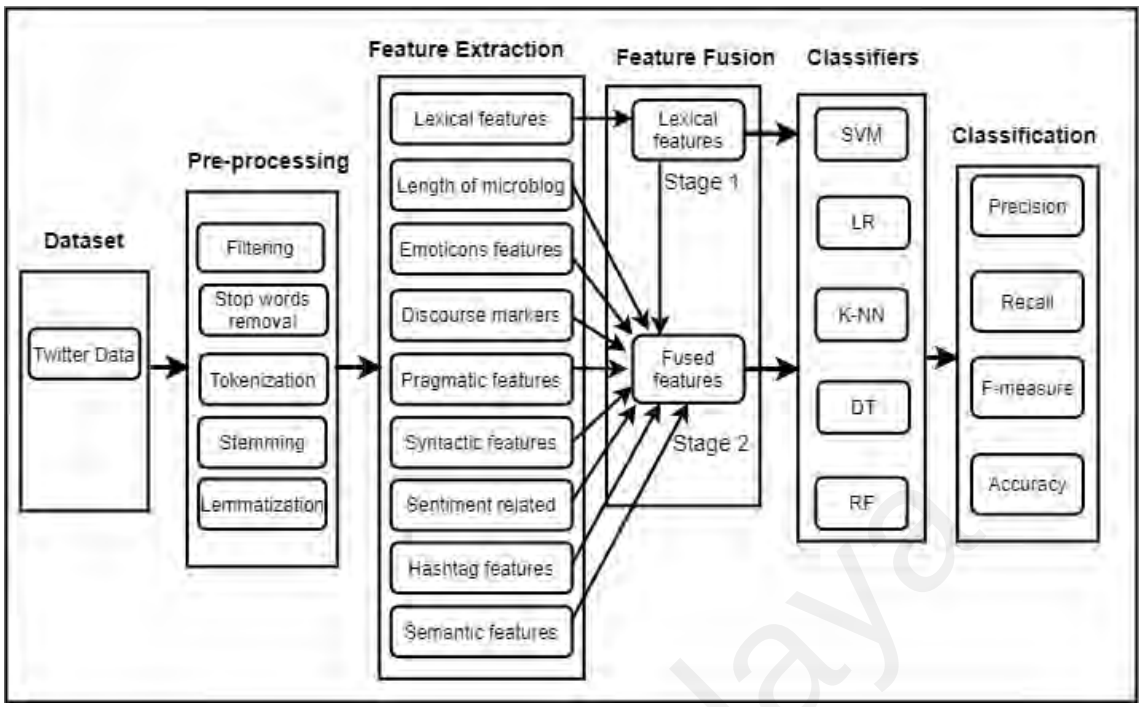

Figure 4.1: Multi-Feature Fusion Framework for Sarcasm Identification.

4.2.1 Data collection

The detailed discussion of data collection components, including the data collection approach, the Language of tweets, data classes, the search period, sarcastic data volume, non-sarcastic data volume, total volume of data, the annotation (sarcastic & non-sarcastic), are provided in Chapter 3, Section 3.4.

4.2.2 Data pre-processing

In this stage, the sarcastic and non-sarcastic data were pre-processed to prepare the data for feature extraction and classification tasks. This is carried out in various steps to remove noise from the sarcasm datasets, including retweets, duplicates, numerals, tweets written in other languages, and tweets with the only URL. These noisy data do not contribute to the enhancement of classification accuracy and are, therefore, eliminated. The text data were converted to the lower case and basic pre-processing techniques such as tokenization, stop word removal, spell check, stemming, and lemmatizing. POS tagging is also employed were implemented using the Python library and Natural

Language Processing (NLP) toolkit. A detailed discussion of the pre-processing components are provided in Chapter 3, Section 3.5.

4.2.3 Feature Extraction

Feature engineering is one of the key processes in any text classification task. The features with discriminative power in differentiating sarcastic from the non-sarcastic text are extracted from the processed data in the feature engineering stage. Apart from feature extraction, other feature engineering schemes such as feature representation and subset feature selection are investigated in this stage. Previous studies have relied on the content-based feature, for example, BoW features, in isolation for sarcasm detection without considering contextual features. Performance results obtained with content-based features revealed that these features alone are not sufficient to capture all the sarcastic tendencies in the text accurately. To enhance the performance of the model, some comprehensive novel features have been proposed to augment the content-based features. These features are presented in this Section for the development of feature fusion for sarcasm identification. They include lexical, length of microblog, hashtag, discourse markers, emoticons, syntactic, pragmatic, semantic (GloVe embedding), and sentiment-related features selected based on observations from the characteristics of the data and evidence from the literature. The observation has been transferred to suitable features, which are now experimented with to enhance the classifiers' performance. The observation was made after the analysis of each feature through the performance results that they attained.

The feature set that attained 50% precision and above is deemed as a potential feature for the feature fusion development framework. The analysis results of each feature are given in Appendix C of this thesis. These features were extracted from tweet content. These features, as described below, were employed in conjunction with the classification algorithm to construct a model for sarcasm identification. The previous section stated that we utilized SVM, RF, LR, KNN, and DT classifiers. Investigating all proposed features

were performed (see result section) by using the feature selection technique to identify the discriminative features for fusion with promising results. The discussion on the feature extraction components is provided below, and the summary is depicted in Table 4.1.

4.2.3.1 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, sarcasm's expression uses contradicting sentiment that can be observed in expressing a negative situation using the positive sentiment, as found in the study on sarcasm analysis conducted by Riloff et al. (2013). For example, 'I love being always cheated.' This study investigated a contradiction between the word's sentiment and other components in the tweets to recognize such sarcastic statements. To this end, sentiment related features are extracted from each tweet and counted. In this study, seven subsets of sentiment related features are defined, which include positive sentiment words, negative sentiment words, highly positive sentiment words, highly negative sentiment words, co-existence between positive sentiment & negative sentiment words, co-existence of positive and negative sentiment words with hashtags, and co-existence of positive and negative sentiment words with emoticons. To extract sentiment related features from the tweet's content, a dictionary that consists of positive words and negative words is created using the SentiStrength (Thelwall et al., 2012) database. SentiStrength is a sentiment lexicon that utilizes linguistic rules and information to detect English text sentiment. The lexicon usually provides the polarity sentiment (positive and negative) of questions, negation, emotion, booster, idioms, slang, and emoticons. The sentiment score uses integers ranging from -5 to +5, in which the larger absolute value represents the stronger sentiment. The first two features are extracted using the two lists by computing the number of sentiment words that tend to be positive or negative. The next two features (highly positive and negative positive words) are extracted by checking

if any of the positive or negative sentiment words are associated with highly emotional parts of speech (adjective (JJ), verb (RB), adverb (VB)) tags tweets. If it occurs, an integer value of 1 is recorded; otherwise, 0 is recorded. Lastly, the last three features are extracted by checking the co-existence of positive sentiment & negative sentiment words, positive sentiment & negative sentiment words with the hashtag, and positive sentiment & negative sentiment words with an emoticon in the same tweet by recording integer one if there is co-occurrence otherwise 0. Therefore, the sentiment-based feature contains seven subsets of feature.

4.2.3.2 Pragmatic (Punctuation related) features.

This study utilizes punctuation marks as pragmatic features. Punctuation has an important effect on text analysis, especially in sentiment analysis. Punctuation symbols are mostly used as an explicit mark that brings out the sarcastic expression in the text. In punctuation related features, six different sets of features were considered and were extracted from tweets content. To extract punctuation marks from the tweets, a regular expression is employed to check the punctuation marks present in the sarcastic expressions. After that, the number of times each of them is used is computed. Firstly, the number of question marks were calculated and extracted as a feature (?). The second feature was obtained by counting the number of exclamation marks in the text (!). The third feature calculated the number of ellipses (.) in the text. The fourth feature considered the presence of capitalization in the text. It computed the number of occurrences, i.e. it searches for the word that is “All-capitals” and extracted it as a feature in the text. The fifth feature calculated the quoted words, which are the words that are in a quote, and added them as a feature. Lastly, the sixth feature calculated the repeated vowels in the text and added them as a feature. Thus, these six features formed a feature set for related pragmatic features.

4.2.3.3 Length of microblog feature

Zhang et al. (2014) noted that opinion mining results could be influenced by the number of words that a text expression contains. Also, the author reported that most of the non-sentimental statements commonly occur in a longer text. In such an instance, there is difficulty in analyzing such text to find accurate sentiment. Length of microblog defines the depth of sentence in conversation, and it is important to determine if speech is sarcastic or not. Despite that, this feature was mentioned in sentiment analysis by Zhang et al. (2014), the impact of deploying the length of the word for sarcasm identification is yet to be investigated by any study. After careful analysis, it was discovered how some of the sarcastic text differed in lengths based on utterance and proposed sarcasm detection features. Even though it has been preliminarily studied in sentiment analysis, this is the first study to comprehensively investigate and implement inclusion of the features for distinguishing sarcastic and non-sarcastic tweets. Thence, the length of the text is considered as a feature for sarcasm identification in this study. To extract the length of the microblogging feature, each tweet's length is calculated and measured as an integer in the text by employing a "Counter" python library. The feature's outcome is implemented using the lens function to compute each tweet's length and results represented in numeric data.

4.2.3.4 Syntactic features

Syntactic feature performs a significant function in providing information regarding the tweets' text syntactic structure. In this study, three features that include POS feature, interjection word, & laughing expression are defined as syntactic features extracted from the processed tweet's content. This study employed the NLTK tokenizer library to perform tokenization tasks on the processed tweets to extract the syntactic feature. Firstly, we extracted the POS feature using the parts of speech dictionary as the basis, and the

count of its presence in the sarcastic text is taken. We only focused on the parts of speech details with some emotional contents such as nouns, adverbs, and adjectives.

Furthermore, the mapping of each of the POS tags and each corresponding POS group was established, and only the tokenized words that correspond with the chosen three parts of speech groups as aforementioned were preserved in the text. The study employed the same framework used in (Berry & Castellanos, 2004) and extracted ADV+ADJ+N (adverb, adjective, and noun). Secondly, to extract the second feature, we identified laughter words that are used to express pleasures or joy. Thus, laughing features were added, which is the sum of internet laughs, represented with lol, hahaha, hehe, rofl, and imao, which we refer to as a new punctuation way. The feature is extracted by creating a dictionary list that contains the most common laughing words and using it 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.

4.2.3.5 Emoticon Feature

Emoticons are a pictorial representation of facial expressions using punctuation and letters. A study on sarcasm analysis conducted in Jain et al. (2017) noted that emoticons play a significant role in uttering sarcastic statements because it expresses the user's mood. For instance, a smiley emoticon with negative situation words produces a sarcastic utterance and vice-versa. In this class of feature, emoticons that consist of positive emoticon like :-(, :(, :-|, ;-(, ;-<,|- {, negative emoticon like :-), :), :o, :-}, , ;-}, :->, ;-), and sarcastic emoticons such as (, [:, ;], -?[), p, P] are considered in this study. Emoticons are usually employed in ironic or sarcastic expressions. People use these emoticons to make a joke or funny when using sarcasm as a wit. This research employed regular

expressions to identify emoticons that consist of Sad, Happy, Laughing, Surprise and Winking by computing their frequency in each tweet to extract emoticon features. Then the frequencies obtained are regarded and added as a feature set.

4.2.3.6 Lexical features

In this study, the Bag-of-Words model uses the term frequency-inverse document frequency (TF-IDF) to represent a lexical feature. Bag of Words-based features is the most useful feature in sentiment analysis. The lexical level feature uses TF-IDF to obtain the most descriptive terms in tweets data. To extract the linguistic feature using the BoW model, a pre-processed step is performed on the tweets dataset to eliminate the microblog typos and internet slang. Next, an NLTK library is employed to tokenize the whole tweet dataset by splitting the tweets into individual words, also known as a token. Furthermore, a dictionary list is constructed based on the extracted words. Lastly, the TF-IDF feature is produced by employing the built-in function in WEKA, which is then utilized as an input to the machine learning algorithm. Thus, the Bag-of-Words feature extraction process was performed in the Weka machine learning algorithm environment using the "StringToWordVector" function found in WEKA.

4.2.3.7 Hashtag features

Sometimes, emotional content is expressed by using hashtags. The hashtag is employed to disambiguate the actual intention of the Twitter user to pass a message. For instance, in a tweet, "Thanks a lot for always helping me, # I hate you." In this utterance, the hashtag "#i hate you" shows that the user is not really expressing thanks to the intended but tremendously hating him for not helping him when the need arises. We call the above expression a negative hashtag tweet. Hashtag features could be positive or negative hashtags. In this study, three sets of hashtag features are defined: a positive hashtag, a negative hashtag, and the co-existence of the positive and negative hashtag. The hashtag

features are extracted by creating a dictionary that consists of a list of negative hashtag words such as "#hate, #pity, #waste, #discrimination, etc." and a list of a positive hashtags such as "#happy, #perfect, #great, #goodness, etc." However, using this dictionary, 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-existence of positive hashtags and negative hashtags in the same tweet. However, if there is co-existence in the same tweet, an integer one (1) is measured; otherwise, zero (0) is measured. Thus, the three sets of features are extracted and added as a feature set.

4.2.3.8 Discourse markers

In social media platforms, people use various 'discourse markers' in making utterances. It has definite functions and aids in expressing an idea. Discourse markers such as temporal compression and counter-factuality have been utilized in irony detection studies (Reyes et al., 2013). It is used to mark the upcoming words' relationship to previous discourse (utterance used in a social context). This feature is very important in sarcasm identification because it helps comprehend utterances by previewing what's coming up. Counter-factuality concentrates on implicit marks: discourse words that suggest contradiction or conflicts in a text. For example: yet, nevertheless, nonetheless, about, etc.

On the other hand, temporal compression concentrates on identifying words associated with opposition in time, i.e., words that show a sudden change in description. Temporal compression can be represented using temporal verbs like suddenly, abruptly, etc. A dictionary containing a list of counter-factuality and temporal compression words is created to extract discourse marker features. Using the semantic dictionary list, the number of counter-factuality and temporal compression words present in the tweets is computed and used as a discourse marker feature.

4.2.3.9 Semantic (word embedding) feature.

Word embedding features employed to extract the semantic features are Global Vectors (GloVe). GloVe embedding is a powerful word embedding learning scheme that learns vector representation of words employing dimensionality reduction on the co-occurrence matrix (a count-based model). This is done by constructing a large matrix of co-occurrence information, with the content of information on how frequently each "word" stored in rows appears in the column. It is an unsupervised technique used to obtain a meaningful vector that corresponds to individual words in a corpus (George et al., 2019). In this model, different words repel against each other where similar words cluster together. In GloVe, the counts' matrix is pre-processed by normalizing the counts and log smoothing them. With GloVe embedding, one can use the co-occurrence matrix to obtain a semantic relationship between words (Pennington et al., 2014). One of the benefits of GloVe over other word-embedding schemes like word2vec is that GloVe does not capture only the local context information of the words (local statistics), but also captures word co-occurrence, also known as global statistics in a corpus to obtain word vectors. The GloVe allows parallel implementation, which makes it easy to train on a large corpus. It also combines the best features of two model families: the local content window methods and the global matrix factorization, to create a new one (Pennington et al., 2014).

Table 4.1: Summary of the Extracted features for classification

NO	Groups	Features
1	Lexical features	Features based on bag-of-words which uses TF-IDF as a lexical level feature.
2	Length of microblog	Length of microblog feature.

3	Hashtag feature	Positive hashtags, Negative hashtags, co-existence of the positive and negative hashtag.
4	Discourse marker features	Discourse markers such as temporal compression and counter factuality.
5	Emoticon features	Positive, negative, and sarcastic emoticon.
6	Syntactic features	Laughing expression, POS (Noun, verb, adverb and adjectives), and Interjection.
7	Pragmatic features	Exclamation mark, Question mark, Ellipsis, Quoted word, All capitals, Repeated vowels.
8	Word embedding	GloVe embedding features.
9	Sentiment related features	Positive sentiment words, Negative sentiment words, Highly emotional positive content, highly emotional negative content, contrast related features between the sentiment components.

4.2.4 Proposed Feature Extraction and Fusion Process Algorithm

This section describes the proposed feature extraction and feature fusion algorithm. It discusses the steps for extracting features and creating the master feature for the proposed feature fusion. However, data is pre-processed before the actual feature extraction takes place. The overall step is divided into three different segments, namely: data pre-processing, feature extraction, and feature fusion. In the data pre-processing, the raw tweet data *rt* is first loaded into memory. Next, six pre-processing operations are performed before extracting discriminative features using the pre-processing ($\wp$) function. It includes correction of misspelt words using $\wp_s$ on a raw tweet, stop word removal from the raw tweet by applying $\wp_w$ on *rt,* lower case conversion of all words by

applying $\wp_l$ on *rt,* number value removal by applying $\wp_n$ on *rt*, tokenization of sarcastic expression into a unique token by applying $\wp_t$, lastly, the processed tweets *rt* is stored in storage location called *P.* In the feature extraction stage, on the other hand, the processed data is loaded to the memory for sarcasm classification. For every processed tweet, a set of features in a numerical form is extracted using the feature extraction function represented with $\Im$. These features include Sentiment features (FS), Pragmatic features (FP), Lexical features (FL), Hashtag feature (FH), Discourse markers feature (FDM), Sematic (Glove Embedding) features (FG), Emoticon feature (FE), Length of microblog feature (FLM), and Syntactic feature (FST) features. Each feature is extracted in a numerical form, referred to as an individual feature (IF). Furthermore, the feature fusion operation is performed using the fusion function represented with φ. The feature fusion involves fusing FS, FP, FL, FH, FDM, FG, FE, FLM, and FST using φ to form the Multi-feature fusion (MFF). Finally, the fused feature is converted to the ARFF file format and provided input to classifiers for the classification step.

Algorithm 1: Feature extraction and feature fusion process.

Definition of terms.

rt	:	raw tweet data
n	:	number of a row in the tweet
P	:	pre-process tweet data
$\wp_t$	:	Tokenization Function
$\wp_c$	:	Special Character removal function
$\wp_l$:	:	Lower Case conversion function
$\wp_n$	:	Number Value Removal function
$\wp_w$	:	Stop word removal function
$\wp_s$	:	Spell Checking function
$\Im_S$	:	Sentiment related feature extraction function
$\Im_P$	:	Pragmatic (punctuation) feature extraction function

$\Im_L$: Lexical feature extraction function

$\Im_H$: Hashtag feature extraction function

$\Im_{DM}$: Discourse Markers feature extraction function

$\Im_G$: Semantic (Glove embedding) feature extraction function

$\Im_E$: Emoticon feature extraction function

$\Im_{LM}$: Length of microblog feature extraction function

$\Im_{ST}$: Syntactic feature extraction function

IF : Individual features

φ : Feature fusion function

MFF : Multi-Feature fusion

Input: Raw Twitter data (rt)

Output: Sets of features and feature fusion as input to machine learning classifiers.

Procedure: FeatExtract (rt)

1: i ← 1

2: While i <= n

3: rt ← LOAD rt (i) from tweet data

4: rt_s ← $\wp_s$(rt) // perform spell check on the raw tweets

5: rt_w ← $\wp_w$(rt_s) // stop word removal from the raw tweets

6: rt_l ← $\wp_l$(rt_w) // convert raw tweet to lower case

7: rt_c ← $\wp_c$(rt_l) // remove special character from tweet

8: rt_n ← $\wp_n$(rt_c) // remove numerical values

9: rt_t ← $\wp_t$(rt_n) // tokenize the tweets

10: P(i) ← rt_t // pre-process tweets

11: i ← i + 1

12: END

13: i ← 1

14: While i <=n

15: P ← LOAD P(i) from pre-processed tweet data

16: FS←$\Im_S$(P(i)) // extract sentiment features from the pre-process tweet

17: FP ← $\Im_p$((Pi)) // extract pragmatic features from the pre-process tweet

```
18:             FL ← ℑL (P(i))   // extract lexical feature from the pre-process tweet
19:             FH ← ℑH((Pi))   // extract hashtag feature from the pre-process tweet
20:             FDM ← ℑDM((Pi))  // extract discourse markers feature from the pre-process tweet
21:             FG ← ℑG(P(i))     // extract semantic features from the pre-process tweet
22:             FE ← ℑE((Pi))   // extract emoticon feature from the pre-process tweet
23:             FLM ← ℑLM((Pi))   // extract length of microblog feature from the pre-process tweet
24:             FST ← ℑST (P(i))     // extract syntactic feature from the pre-process tweet
25:          IF← [FS, FP, FL, FH, FDM, FG, FE, FLM, FST] // sets of features extracted
                WRITE IF // append the extracted features to file
26:       i ← i + 1
27:          END
28:      i ← 1
29:      While i <=n
30:      MFF ← φ (FS, FP, FL, FH, FDM, FG, FE, FLM, FST) // fusion of all feature sets
31:           WRITE MFF//Append the  Multi-feature fusion
32:      END
```

4.2.5 Construction of Multi-Feature Fusion Framework Machine Learning Classification Models.

This Section describes the multi-feature fusion framework development process. The multi-feature fusion framework development process for sarcasm identification that uses two classification stages is described as follows. The first stage classification is constructed using a lexical feature extracted by Bag-of-Words (BoW) only that uses TF-IDF, trained using five standard classifiers, including Support Vector Machine, Decision Tree, K-Nearest Neighbor, Logistic Regression, and Random Forest to predict the sarcastic tendency based on the lexical feature. However, this prediction does not capture the text's semantics, context, and word co-occurrence or relatedness. As a result, the second stage classification is performed. In stage two, the extracted lexical sarcastic

tendency feature is fused with eight other proposed feature that consists of a length of microblog, hashtag, discourse markers, emoticons, syntactic, pragmatic, semantic (GloVe embedding), and sentiment related features were added one after the other until all the features are added to test the effect of each of the feature in the multi-feature fusion framework. The fused feature (feature fusion) is then employed to input machine learning classifiers to model a context on the fused features to obtain the feature fusion performance by employing various classifiers (SVM, DT, K-NN, LR, and RF). The fused features capture both the semantic and contextual information from the sarcastic expression. Next, the feature selection algorithm was performed on the lexical feature to obtain the top 200 discriminative features. The feature selection algorithm was also performed on each of the other features with two or more subsets, such as hashtag, emoticons, syntactic, pragmatic semantic (GloVe embedding), and sentiment related features, to check the discriminating power of each subset. However, any of the features with a low threshold is eliminated. Next, the features selected from the lexical, hashtag, emoticons, syntactic, pragmatic semantic (GloVe embedding), and sentiment-related features were fused with discourse marker feature and length of microblog feature to form a new feature fusion (feature fusion with feature selection). Lastly, the new feature fusion was employed as an input to the machine learning classifier to train the model based on feature selection. However, the feature selection was performed by using two feature selection algorithms. Firstly, using the Pearson correlation algorithm, and secondly, using the information gain feature selection algorithm. The effectiveness of the developed multi-feature fusion framework is tested with various experimental analysis, which was performed to obtain classifiers' performance. Thus, the feature fusion classification (with feature selection technique) obtained improved results over the feature fusion (without feature selection technique) and lexical-based feature classification, which shows the

significance of the proposed Multi-feature fusion framework sarcasm identification. The flowchart of the proposed methodology is shown in Figure 4.2.

Algorithm 2: Two stages classification of the proposed framework

Definition of terms

U : U: Lexical feature content.

W : W: Eight other groups of features

V : V: Classification label.

C : C: Lexical-based sarcasm tendency feature

1: **Input:** Training set T = {(U_1, W_1, V_1), (U_2, W_2, V_2), ...,(U_n, W_n, V_n)}; 2 sets classifier k_1, and k_2; a testing object M = (u, w*)*;

Output: The label of M;

2: Train:

3: create lexical feature training set T_1= {(U_1, V_1), (U_2, V_2),...,(U_n, V_n)};

4: train k_1 on T_1;

5: for i=1to n do

6: apply k_1 on U_i to get C;

7: End for

8: create fusion feature training set T_2 = T2 = {(C1, W1, V1), (C2, W2, V2),..., (Cn, Wn, Vn)};

9: train K_2 on T_2;

10: Test:

11: apply k_1 over M = (U) to obtain its label C_U;

12: apply k_2 over M^1 = (C_U, w) to obtain its label V;

13: Return V;

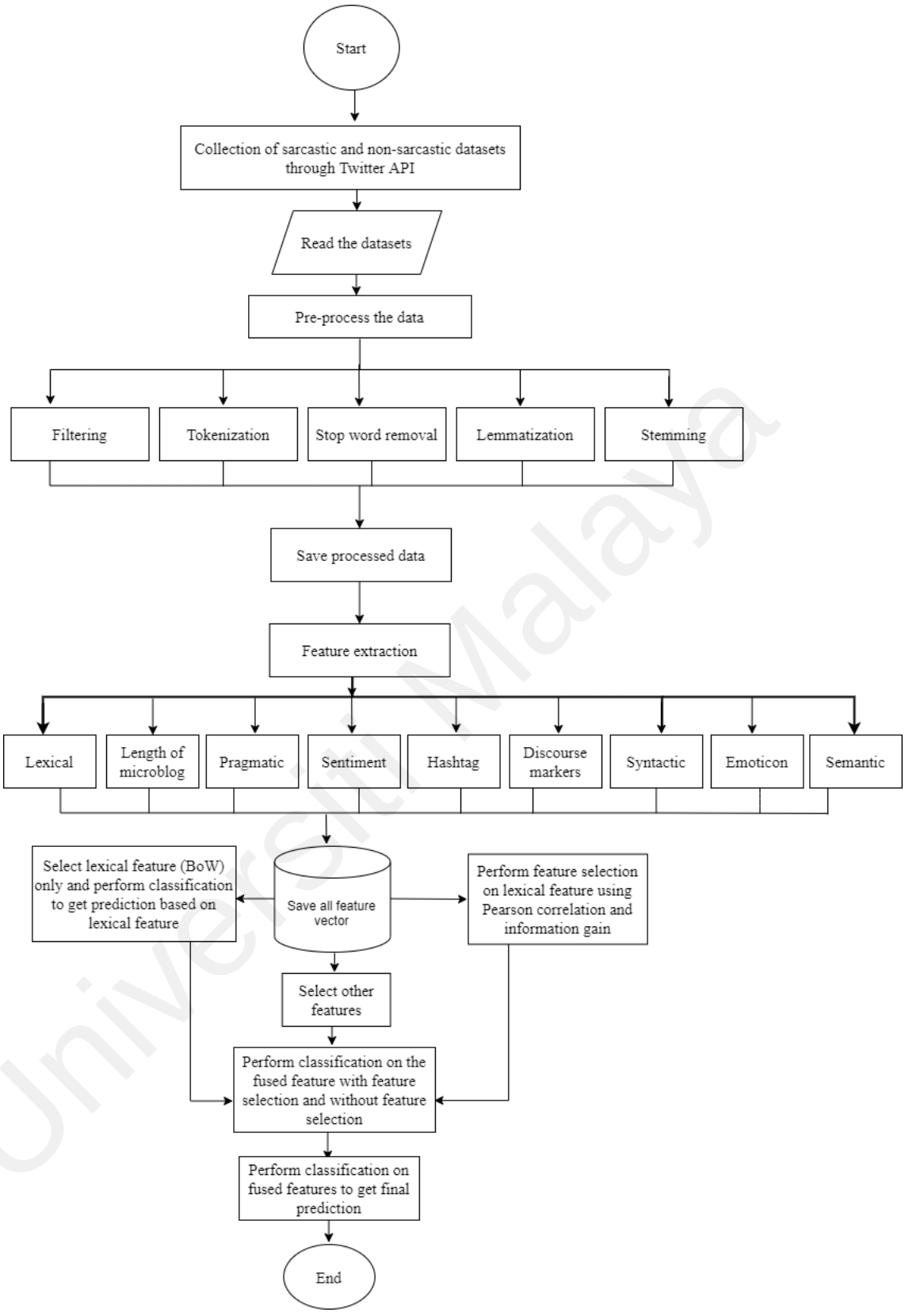

Figure 4.2: The Flowchart of the proposed Framework.

4.2.6 Feature analysis and selection

All the extracted subset features from each group of features may not be relevant for the effective development of the sarcasm identification framework. The utilization of irrelevant features could lead to high computation time, reduction in predictive performance, and model overfitting (Chowdhury et al., 2017; Nweke et al., 2019a). Thus, feature selection algorithms were investigated. The feature selection technique is essential in any text classification task. It can reduce the computation time and eliminate the irrelevant features that do not contribute to the classification performance. Analysis of features is conducted to identify the most performing features. As a result, some features are insignificant and do not add any value to the classifier's performance. In this case, features with discriminating ability were selected using various feature selection techniques.

However, choosing the best feature selection algorithm dimensionality reduction still poses a challenge since the working of feature selection algorithms relies on the nature of the training data. Hence, two feature selection algorithms have been investigated and compared to evaluate the impact of feature selection on the performance of classification model on the developed feature fusion framework, namely Pearson correlation (Guyon & Elisseeff, 2003) and information gain (Guyon & Elisseeff, 2003) to find the discriminative power in each feature (Yang & Pedersen, 1997). However, the above two feature selection algorithms were chosen as they outperformed other features selection algorithms tested after the analysis on about five different features selection algorithms. In the Pearson correlation technique, the selection of features is made by computing the correlation between the feature vectors and each class on the training data. However, the ranking of features was made by correlation and features that attained 0.00248 correlation threshold and above were selected for the modelling stage. Similarly, the information gain feature selection technique is a filter-based technique that uses a statistical approach to

allocate a score to each feature. However, the selection and rejection of feature is determined by the threshold score. In this study, a feature that attained a threshold score of 0.000924 and above were selected for the modelling stage.

4.2.7 Performance Evaluation of the Constructed Multi-Feature Fusion framework Classification models.

Various experiments were performed to measure the efficiency of the Multi-feature fusion framework for sarcasm identification on the dataset. This study utilized 'Precision' as the major performance evaluation. However, other performance metrics such as recall, f-measure, and accuracy were employed as supplemental to evaluate the framework's performance. As described in Chapter 2 (2.8.3; Issues Related to Evaluation Metrics), the selection of evaluation metrics should be thoroughly considered in order not to obtain misleading evaluation results. This issue is commonly found in machine learning tasks where there is an imbalance in the class distribution of the dataset. In this study, a balanced dataset was utilized. The detailed description is provided in Chapter 2, Section 2.5.7 of this thesis.

4.3 Experimental design

This Section presents various experimental designs to construct a classification model for sarcasm identification on a feature fusion framework. Extensive sets of experiments were performed to evaluate the predictive performance of the classifiers. The classification experiment was carried out to analyze the sarcasm expression (sarcastic and non-sarcastic) in a given tweet. The data pre-processing, normalization, and feature extraction tasks were performed in Jupyter notebook, an integrated development environment (IDE) for python programming language on both sarcastic and non-sarcastic data. However, the feature extraction for each group of features was stored as a .csv file. Subsets of features explained in Section 4.3 have been employed in the sarcasm analysis

experiment as input to various machine learning algorithms. This study has experimented with five different machine learning models that consist of Logistic Regression, K-Nearest Neighbor, Support Vector Machine, Decision Tree, and Random Forest to estimate the existence of sarcastic sentiment in the given tweets. The purpose of employing different models is to get the best performance result. The detailed justification for choosing the classifiers above is provided in Chapter 3, Section 3.6.3. The impact of feature selection on the classification model results was also investigated. Thus, Pearson correlation and information gain feature selection techniques were tested on the feature fusion and compared to evaluate their impact on the model performance. All experiments were performed using 10-fold cross-validation (Liu & Özsu, 2009). All the sarcasm classification and features selection process experiments were implemented in java by applying the machine learning toolkit WEKA 3.9 (Waikato Environment for Knowledge Analysis), open-source software consisting of various machine learning algorithms.

Moreover, Table 4.3 presents the parameter settings of classifiers used during the experiment. The same set of parameters were employed for all the experimental settings to measure the multi-feature fusion framework for sarcasm identification. Four standard evaluation metrics such as precision, f-measure, recall, and accuracy were tested and weighted over both classes (sarcastic and non-sarcastic) during the experiment. The weights were obtained based on class ratios. However, this study utilized 'Precision' as the major performance evaluation. However, other performance metrics that include recall, f-measure, and accuracy have been employed as supplemental framework evaluation. Lastly, the significance of the proposed multi-feature fusion framework was evaluated using four baseline techniques. All experiments were performed on a system running on window 10 with 64-bit operating systems. The system uses an Intel Core™ i7-4770 CPU @ 3.400GHz with 16GB of random access memory (RAM). The summary

of the experimental environment is shown in Table 4.2, whereas the parameter tuning utilized in all the experimental settings are depicted in Table 4.3.

The aforementioned experimental settings were conducted sequentially in four different settings as described under the subSections below to measure the proposed sarcasm identification framework's performance.

Table 4.2: List of Experimental Environment.

S/N	Experiments	Environment
1	Data-preprocessing and normalization	Python programming environment
2	Feature Extraction and feature fusion	Python programming environment
3	Feature selection	Weka tool kit environment
4	Sarcasm classification	Weka tool kit environment

Table 4.3: Parameter Optimization and tuning values of Classifiers.

Classifier	**Parameters**	**Values**
Support Vector Machine	Batch size	100
	Kernel	Polykernel-E1.0-C250007
	Complexity	1.0
	Epsilon	1.0E-12
	Tolerance parameter	0.001
Logistic Regression	Batch size	100
	Ridge	1.0E-8
	Maxlts	-1
K- Nearest Neighbors	Batch size	100
	K	10
Decision Tree (J48)	Batch size	100
	Confidence factor	0.25
	Number of folds	3.0
	MinNumObj	2.0
	Seed	1.0
Random Forest	Batch size	100
	numExecutionSlots	1.0
	numIterations	100
	Seed	1.0

4.3.1 Experimental Setting 1 (Classification based on the Lexical feature).

The first experimental setting is based on the lexical feature (BoW) and machine learning classifiers. In this setting, the lexical feature is extracted from the processed data using the Bag-of-Words model. The obtained features are then employed and fed as an input to the machine learning algorithms to construct the classification model. The feature is trained using machine learning classifiers to predict the sarcastic tendency based on the lexical feature. In this setting, a total of five analysis (lexical feature x five classifiers) were performed to measure the performance of the constructed classification model. Figure 4.3 illustrates the flow of the settings and the results of the precision, recall, f-measure, and accuracy of all the experiments is depicted in Table 5.1. The purpose of this experiment is to test the effectiveness of the lexical features for sarcasm detection.

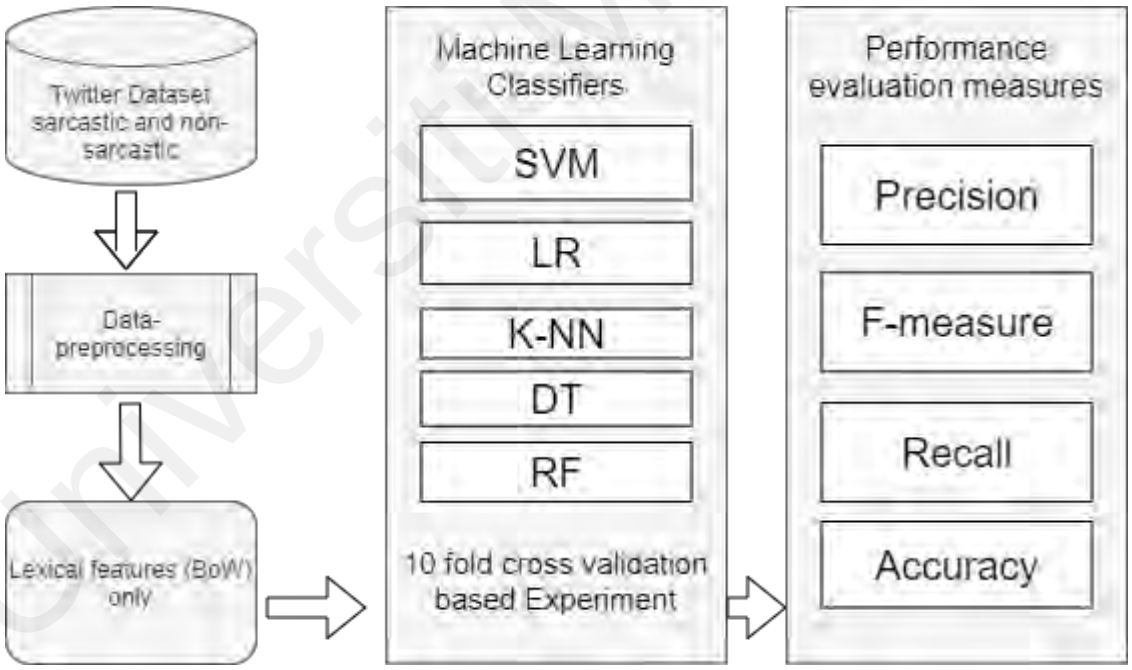

Figure 4.3: Design of Experimental Settings 1.

4.3.2 Experimental Setting 2 (Classification based on the Fused Feature).

The second experimental setting is based on feature fusion and machine learning classifiers. In this setting, the experiment setting on each of the extracted group of the feature that consists of the lexical feature, the length of microblog, hashtag, discourse

markers, emoticons, syntactic, pragmatic, semantic (GloVe embedding), and sentiment related features, were performed and the obtained features are then employed and fed as an input to the machine learning algorithms for the construction of the classification model. Each group of features is trained using machine learning classifiers to predict the sarcastic tendency based on each feature set to test the performance of each feature. Next, all the features were fused together by adding them one after the other until all the features were added to test the effect of each feature in the fusion framework. The fused feature (feature fusion) is then employed as an input to machine learning classifiers to model a context on the fused features to obtain the performance of the feature fusion by employing various classifiers (Support Vector Machine, Decision Tree, K-Nearest Neighbor, Logistic Regression, and Random Forest). The fused feature, also known as a multi-feature fusion (MFF) in the form of a feature matrix, is then employed and fed as an input to the machine learning algorithms to construct the classification model. Finally, the feature is trained using a basic classifier to get a prediction on feature fusion. In this setting, a total of five analysis (proposed feature fusion x five classifiers) were performed to measure the performance of the constructed classification model. Figure 4.4 illustrates the flow of the settings and the results of the precision, recall, f-measure, and accuracy of all the experiment is depicted in Table 5.2. This experiment aims to address the loss of contextual information issue in sarcastic expression by modelling the fused feature that consists of contextual feature and content-based features. Thus, experimental settings two will address research question 4 (RQ4).

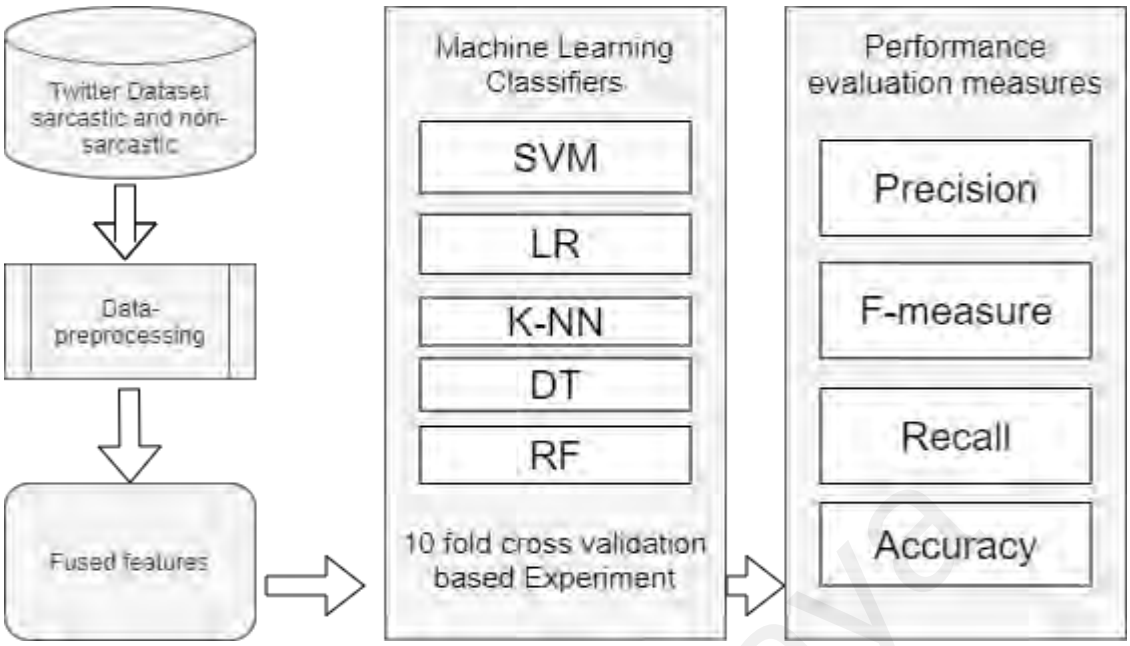

Figure 4.4: Design of Experimental Settings 2.

4.3.3 Experimental Setting 3 (Classification based on the Fused Features and Feature Selection).

The third experimental setting is based on feature fusion, feature selection, and basic classifiers. Various experimental analysis in sarcasm identification has indicated that redundant features could decrease performance result and high computational time (Forslid & Wikén, 2015; Dharwal et al., 2017). In this experimental setting, the feature selection technique described in Section 4.5 was applied to feature fusion to select features with discriminating power and reduce the high dimensional feature vector space. The feature selection algorithm was initially performed on the lexical feature to obtain the top 200 discriminative features. Besides, the feature selection algorithm was also performed on each of the other types of features with two or more subsets, such as hashtags, emoticons, syntactic, pragmatic semantic (GloVe embedding), and sentiment-related features to check the discriminating power of each subset. However, any of the features with a low threshold is eliminated. Next, the features selected from the lexical, hashtag, emoticons, syntactic, pragmatic semantic (GloVe embedding), and sentiment-

related features were fused with discourse marker feature and length of microblog feature to form a new feature fusion (feature fusion with feature selection). The feature selection was performed in two faces. Firstly, using the Pearson correlation algorithm, and secondly by using the information gain feature selection algorithm. However, each feature selection technique's output in the form of a feature matrix (FM) is then employed and fed as an input to the machine learning algorithms for the construction of an effective classification model. The feature is trained using a basic classifier to get a prediction on feature fusion. In this setting, ten analysis (feature fusion x two feature selection algorithm x five classifiers) were performed to measure the performance of the constructed classification model. Figure 4.5 illustrates the flow of the settings and the results of the precision, recall, f-measure, and accuracy of all the experiments depicted in Table 5.4 and Table 5.5. This experiment aims to address the training data sparsity issue in sarcastic expression by modelling the fused feature that consists of contextual feature & content-based features, and performing feature selections techniques that select the features with discriminative powers eliminating the null features. Thus, experimental settings three will address research question five (RQ5).

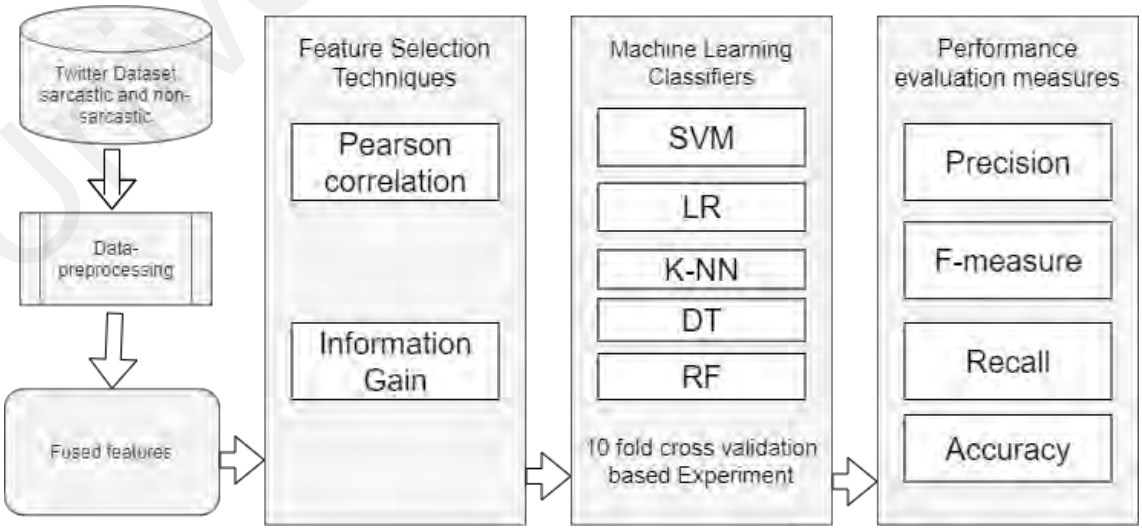

Figure 4.5: Design of Experimental Settings 3.

4.3.4 Experimental Setting 4 (Classification based on the Evaluation of the Proposed Framework with the baselines).

The fourth experimental setting is based on the comparison of the multi-feature fusion framework with baseline methods. Due to the lack of comprehensive public datasets for evaluating the significance of the proposed framework, four baseline approaches were established and experimented on the dataset utilized in this study. The first baseline approach is based on the BoW technique as employed in (Ghosh & Veale, 2016; Khodak et al., 2017; Hazarika et al., 2018) studies. The second baseline is based on word embedding (word vector), which is another important baseline that uses a contextual word vector that includes GloVe embedding feature (Pennington et al., 2014) trained 42B corpus as employed in (Ghosh et al., 2015; Potamias et al., 2020). The third baseline is a feature fusion method proposed by Kumar and Garg (2019), which utilized the fusion of pragmatic feature, sentiment feature, and Top-200 TF-IDF features to build the context using shallow classifiers. The fourth baseline is a proposed approach studied by Sundararajan et al. (2020) that proposed stacking ensemble feature-based sarcasm detection in Twitter. In this experimental setting, methods used in the baseline mentioned above were implemented on a processed sarcasm dataset and represented accordingly. Therefore, five master features represented in a numeric format were arranged. However, the five master features (MFs) were then employed and fed as input to the machine learning algorithms to construct the classification model. The MFs are trained using machine learning classifiers to get performance results for each baseline. This experiment aims to evaluate the performance of the five classifiers on the four baseline methods. The performance results attained from the four baselines are compared with the proposed framework. In this setting, a total of twenty-eight analysis (feature fusion + four baselines x five classifiers+ 3 additional settings for baseline 3 and 4) were performed to measure the performance of the constructed classification model. Figure 4.6 illustrates the flow of

the settings, and the results of the precision, recall, f-measure, and accuracy obtained from all the experiments are depicted in Table 5.7. This experiment aims to find how much performance results are enhanced by developing the proposed framework for sarcastic expression by modelling the fused feature that consists of contextual features & content-based features and performing feature selections techniques that select the features with discriminative power. Thus, experimental setting 4 will address the first part of research question 7 (RQ7).

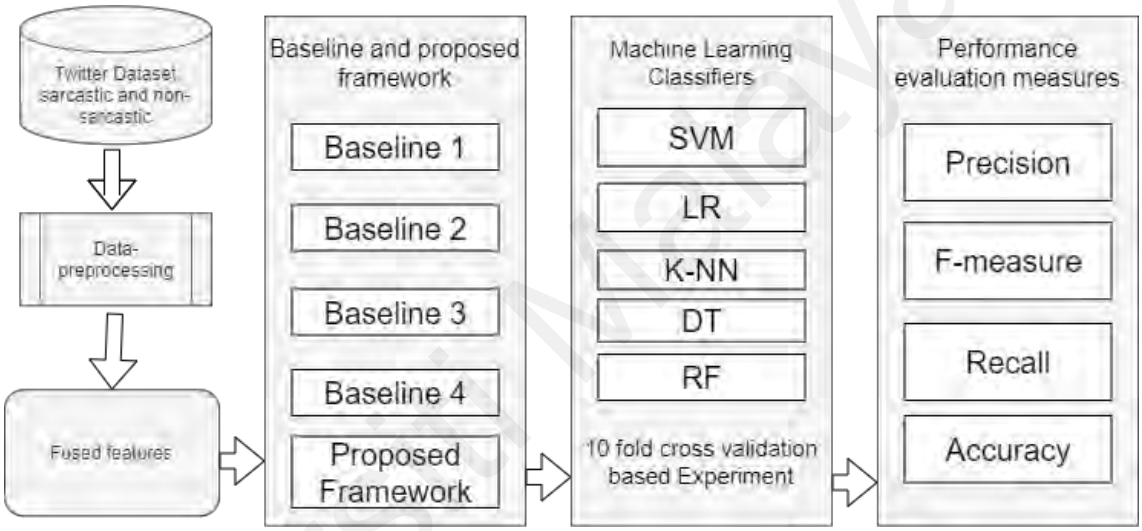

Figure 4.6: Design of Experimental Settings 4.

4.4: The Summary of Experimental settings.

Experimental settings	Classification based on	Analysis arrangements	Total analysis	Classifiers	Performance metrics
Experimental settings 1	Lexical features only	Lexical x five classifiers	5	SVM, LR, KNN, DT & RF	Precision, recall, f-measure, accuracy
Experimental settings 2	Fused features	Proposed feature fusion x five classifiers	5	SVM, LR, KNN, DT & RF	Precision, recall, f-measure, accuracy
Experimental settings 3	Fused features with feature selection techniques	Feature fusion x two feature selection algorithm x five classifiers	10	SVM, LR, KNN, DT & RF	Precision, recall, f-measure, accuracy
Experimental settings 4	Proposed framework and baselines	Feature fusion + four baselines x five classifiers+ 3 additional settings for baseline 3 and 4	28	SVM, LR, KNN, DT & RF	Precision, recall, f-measure, accuracy

4.4 Chapter Summary.

This Chapter presents the implementation of the effective multi-feature fusion framework for sarcasm identification. Nine sets of comprehensive features were proposed and extracted from tweets to construct a machine learning model for classifying tweets as either sarcastic or non-sarcastic using two stages classification approach. The first stage classification is constructed using a lexical feature extracted using the BoW model only, that uses TF-IDF and trained using five standard classifiers, which include Support Vector Machine, Decision Tree, K-Nearest Neighbour, Logistic Regression, and Random Forest to predict the sarcastic tendency based on the lexical feature. In stage two, the extracted lexical sarcastic tendency feature is fused with eight other proposed feature that consists of a length of microblog, hashtag, discourse markers, emoticons, syntactic, pragmatic, semantic (GloVe embedding), and sentiment related features to form a new master feature (MF). The fused feature (MF) is then employed as an input to machine learning classifiers to model a context on the fused features to obtain the feature fusion performance by employing various classifiers (SVM, DT, K-NN, LR, and RF) to build a machine learning algorithms. To identify the most discriminative features, two feature selection algorithms were investigated: Information gain and Pearson correlation. Various feature analysis experiments were conducted to select the features with the substantial discriminative ability to enhance the results of the predictive performance. We conducted extensive experiments to measure the performance of the five selected classifiers. The aforementioned experimental settings were conducted sequentially in four different settings to measure the proposed sarcasm identification framework's performance, as described in Section 4.6. In the next Section, detailed results and a discussion of the developed framework will be presented.

CHAPTER 5: RESULTS AND DISCUSSIONS

5.1 Introduction

This Chapter presents and discusses the predictive results of the experiments performed in Chapter 4 based on the Twitter dataset. All five classifiers were run on the proposed feature fusion using 10-fold cross-validation (Liu & Özsu, 2009). Precision was used as a major evaluation matric. Besides, this thesis also reported the classification performance of the f-measure, recall, accuracy on each classifier as a supplementary measure for effective evaluation of the proposed framework. The experimental results were very suitable and assisted in predicting the best classifiers for sarcastic classification. The presented results are based on four experimental settings described in Section 4.6. Firstly the experimental results obtained on the first classification stage, based on lexical features only, are presented. Secondly, the second classification stage results, also known as feature fusion, consists of the fusion of lexical-based sarcastic feature and eight other proposed features were obtained. Thirdly, the proposed feature fusion results based on feature selection by experimenting with two feature selection techniques (Pearson correlation and information gain) are presented. Lastly, the results of the evaluation of the proposed multi-feature fusion with four baseline approaches are presented. The subSection below provides all the results.

5.2 Results of Experimental Setting 1.

In this Section, the experimental setting 1 result is presented. The result is based on the extracted lexical feature, which was then employed and fed as an input to five different machine learning algorithms: SVM, LR, KNN, DT, and RF. The performance results in terms of precision, recall, f-measure, and accuracy of 5 analysis (lexical feature x five classification algorithm) are presented in Table 5.1. The visualization of the results is depicted in Figure 5.1. However, it can be observed from Table 5.1 that the performance results of precision, recall, f-measure, and accuracy fall in the range of 78% and 83.5%.

The values show that all the classifiers understand the sarcastic expression based on the lexical feature. The results show that the random forest classifier attained the highest performance in precision, with 0.835 overall classifiers. The results also show that it outperformed other classifiers in terms of f-measure, recall, and accuracy. Even LR also showed good performance in the classification, indicating that it understood the sarcastic expressions.

Moreover, Table 5.1 shows that low-performance results were recorded in KNN and DT classifiers, whereby the two classifiers obtained precision results of 78.4% and 79.2%. It shows that both classifiers had a low understanding of sarcastic expressions based on the lexical features. The SVM and LR classifiers show a negligible difference in precision performance. A conclusion can be made based on the result that the RF performance is attributed to the ensemble properties.

Table 5.1: Performance Results obtained by considering Lexical Feature Only

Classifier	Precision (%)	Recall (%)	F-measure (%)	Accuracy (%)
SVM	0.812	0.803	0.801	0.802
LR	0.810	0.806	0.805	0.806
KNN	0.784	0.776	0.774	0.776
DT	0.792	0.790	0.789	0.790
RF	0.835	0.832	0.832	0.832

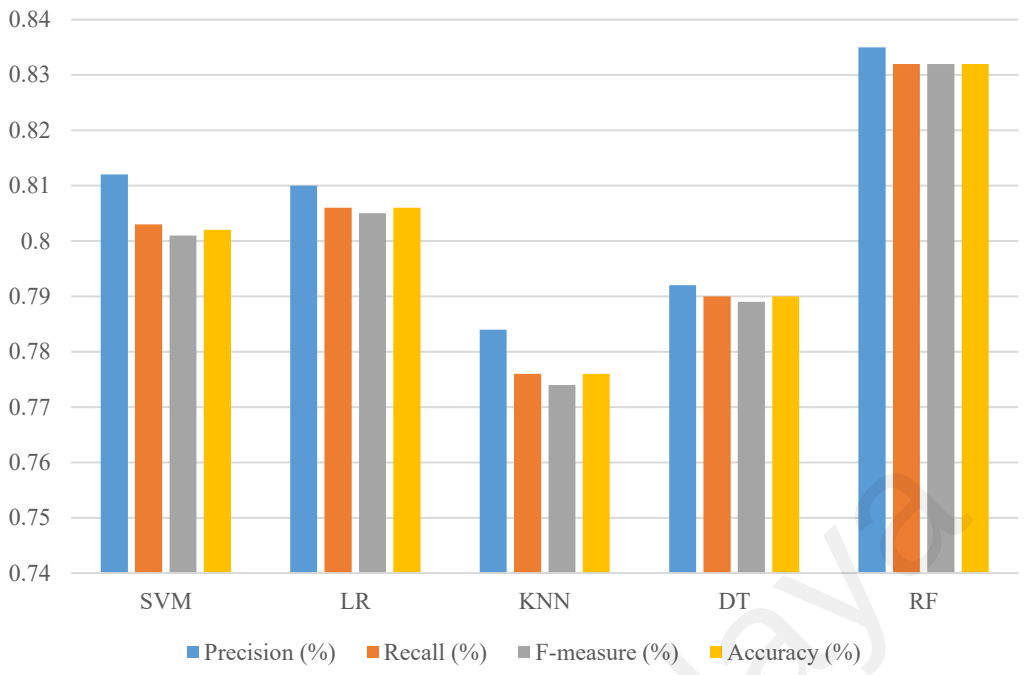

Figure 5.1: Performance results of different classification algorithms on the lexical feature only.

5.3 Results of Experimental Setting 2.

In this Section, the experimental setting 2 result is presented. The result is based on the fusion of lexical sarcastic tendency feature and other proposed features consisting of a length of microblog, hashtag, discourse markers, emoticons, syntactic, pragmatic, semantic (GloVe embedding), and sentiment related features into a fused feature (feature fusion). The fused feature is then employed and fed as an input to five different machine learning algorithms that consist of SVM, LR, KNN, DT, and RF. The predictive performance results on the fused features in terms of precision, recall, f-measure, and accuracy of five analysis (feature fusion x five classification algorithm) are presented in Table 5.2. The visualization of the results is depicted in Figure 5.2. As shown in Table 5.2, the proposed Framework (Multi-Feature Fusion) effectively evaluated the model. The table shows the values obtained from the simulated result by comparing different classifiers on sarcasm analysis. It can be observed from the table that the DT and RF had a good classification performance. It shows that both classifiers understand the sarcastic

utterances, which shows that both classifiers can classify well the sarcastic utterances. We can also imagine from the table that the DT classifier outperformed KNN. It can also be observed that the last result is obtained with the KNN classifier with a precision of 91%. It shows that the classifiers had a lesser understanding of sarcastic expressions but still can produce better results. It is obvious from the experiment results that out of the five models tested with, LR and SVM are competing in terms of precision by attaining 93.4% precision each. However, the LR outperforms the SVM in terms of f-measure, recall, and accuracy. However, when the results are compared with the results obtained with the lexical feature (BoW), it can be observed that there is an improvement in the performance in all the models (see Table 5.3 and Figure 5.3 - Figure 5.6). For instance, the RF classifier attained an additional 9.5% and 9.7% results for precision and f-measure, respectively, which shows the significance of the proposed multi-feature fusion framework in the sarcasm analysis task.

Thus, the results of the experiments show that the proposed multi-feature fusion framework in the sarcasm analysis task that consists of lexica, length of microblog, hashtag, discourse markers, emoticons, syntactic, pragmatic, semantic (GloVe embedding), and sentiment related features enhanced the predictive performance of the sarcasm classification.

Table 5.2: Performance results obtained by considering fused features

Classifier	Precision (%)	Recall (%)	F-Measure (%)	Accuracy (%)
SVM	0.934	0.928	0.928	0.927
LR	0.934	0.932	0.932	0.931
KNN	0.910	0.910	0.909	0.910
DT	0.932	0.931	0.932	0.931
RF	0.930	0.929	0.929	0.929

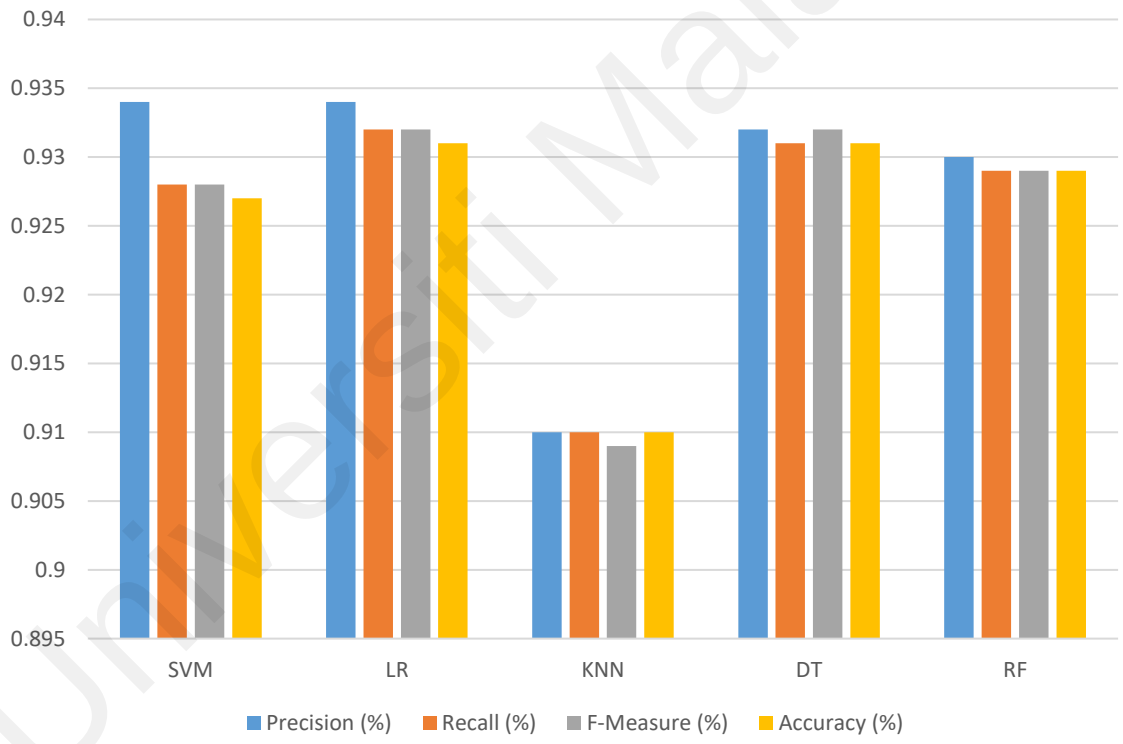

Figure 5.2: Performance results of different classification algorithms on the fused features.

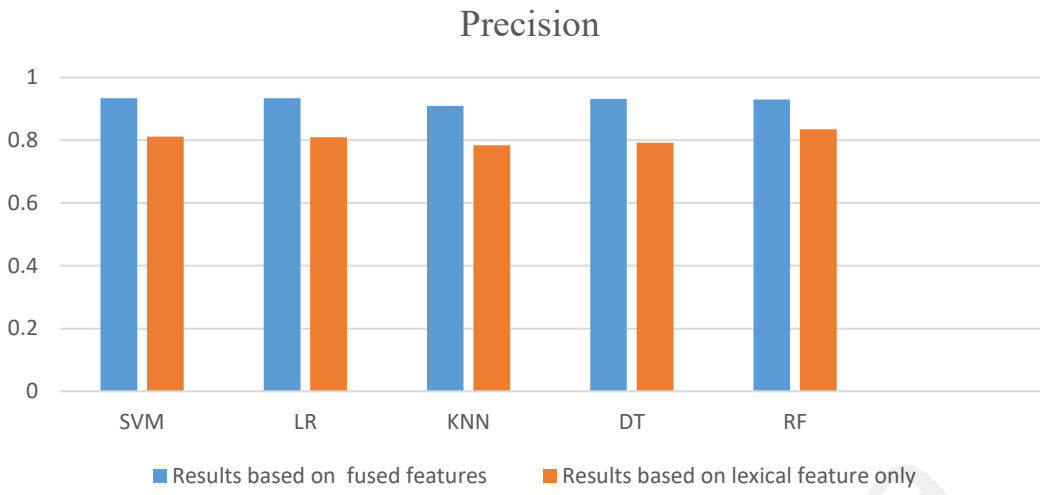

Figure 5.3: Comparison results of Precision on different feature sets.

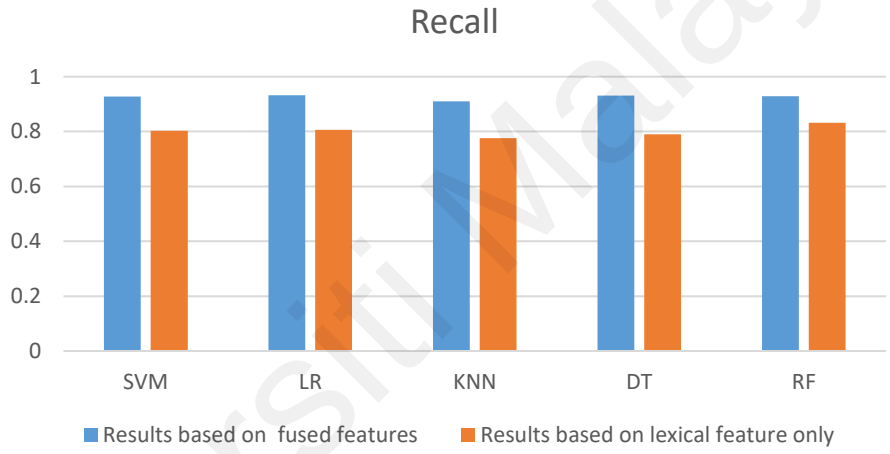

Figure 5.4: Comparison results of Recall on different feature sets.

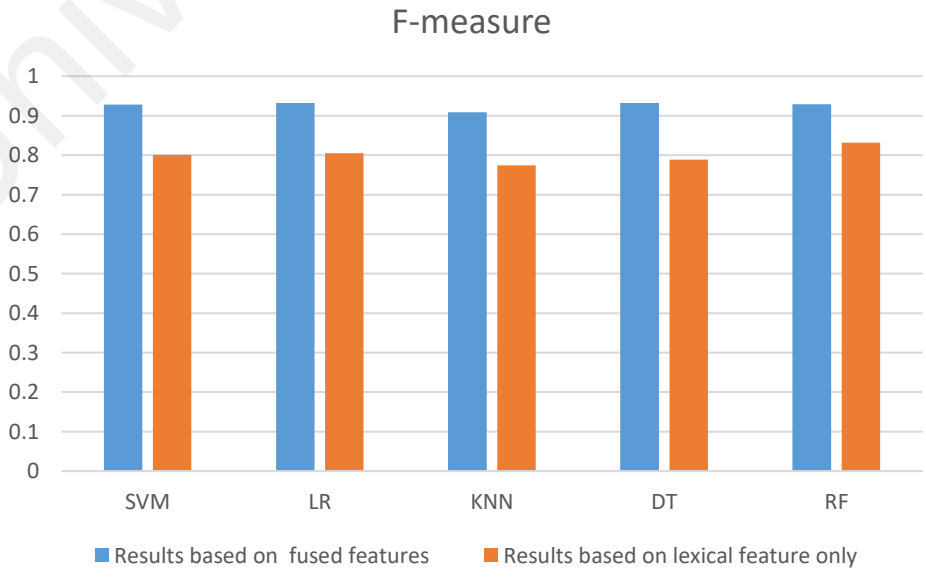

Figure 5.5: Comparison results of F-measure on different feature sets

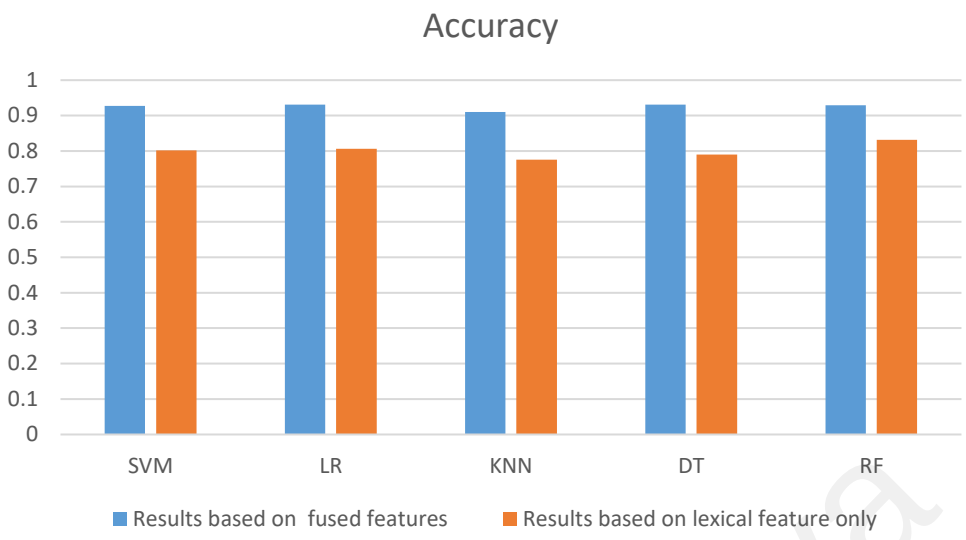

Figure 5.6: Comparison results of Precision on the different feature set.

Table 5.3: The differences in Precision, Recall, F-measure, and Accuracy for five classifiers on different feature sets.

Classifier	Precision	Recall	F-measure	Accuracy
SVM	0.122	0.125	0.127	0.125
LR	0.124	0.126	0.127	0.125
KNN	0.126	0.134	0.135	0.134
DT	0.14	0.141	0.143	0.141
RF	0.095	0.097	0.097	0.097

5.4 Results of Experimental Setting 3.

In this Section, the experimental setting 3 result is presented. The result is based on the application of the feature selection algorithm on feature fusion. The feature selection was performed in two faces. Firstly, using the Pearson correlation algorithm, and secondly by using the information gain feature selection algorithm. However, each feature selection technique's output in the form of a feature matrix (FM) is then employed and fed as an input to the machine learning algorithms for the construction of an effective classification model. We experimented with all the five classifiers that consist of SVM,

LR, KNN, DT, and RF on the proposed multi-feature fusion framework to recognize the most discriminative features that may enhance the performance of the classifiers and lower the classification time. The predictive performance results on each of the feature selection algorithms in terms of precision, recall, f-measure, and accuracy of 10 analysis (feature fusion x two feature selection x five classification algorithm) is presented in Table 5.4 and Table 5.5, provides a comparison of the results of the five classifiers with each feature selection algorithm. In contrast, Figure 5.7 and Figure 5.8 represent their visualization. The comparison of Table 5.2 with Table 5.4 indicates that the use of the Pearson correlation feature selection algorithm on the fused feature (feature fusion) has slightly enhanced the precision performance for RF (0.947), KNN (0.917), LR (0.940), SVM (0.937) and DT (0.935). The results also show that it outperformed other classifiers in terms of recall, f-measure, and accuracy.

Similarly, Table 5.5 depicts the experimental results attained by employing an information gain feature selection algorithm on the feature fusion. The comparative results with Table 5.2 also show a slight improvement in precision with RF (0.944), KNN (0.917), DT(0.936), SVM (0.937), and LR (0.940) remained the same. Accordingly, the algorithm also outperformed the other four classifiers regarding recall, f-measure, and accuracy. However, a slight variation in performance results is noticed on both of the feature selection algorithms.

In overall performance, it can be observed that RF outperformed all the four other classifiers by attaining a precision of 94.7%, which shows an enhancement of 1.7% precision with the Pearson correlation feature selection algorithm. We assumed that the random forest's performance result is attributed to the ensemble scheme, whereby approximately 300 decision trees are combined and together with 10 features to attain a consensus of sarcasm classification. RF classifier is one of the powerful learning models

that train on various datasets, including large datasets, and it can handle large input features while parameters remain the same. The model approximates missing data due to its ability to maintain accuracy when there is missing data as it balances errors in the dataset even when there is an imbalance in class distribution. We can also assume that the decline in RF model performance in experimental 2 settings could be attributed to redundant features. In conclusion, the utilization of two feature selection algorithms marginally enhanced the precision performance results in comparison with the normal settings results (see Table 5.2).

Table 5.4: performance results attained on fused features using Pearson correlation.

Classifier	Precision (%)	Recall (%)	F-measure (%)	Accuracy (%)
SVM	0.937	0.933	0.932	0.933
LR	0.940	0.938	0.938	0.938
KNN	0.917	0.917	0.917	0.916
DT	0.935	0.934	0.934	0.934
RF	0.947	0.946	0.946	0.945

Table 5.5: Performance results attained on fused features using information gain.

Classifier	Precision (%)	Recall (%)	F-Measure (%)	Accuracy (%)
SVM	0.937	0.933	0.932	0.932
LR	0.940	0.938	0.938	0.937
KNN	0.917	0.917	0.917	0.916
DT	0.936	0.935	0.934	0.935
RF	0.944	0.943	0.943	0.943

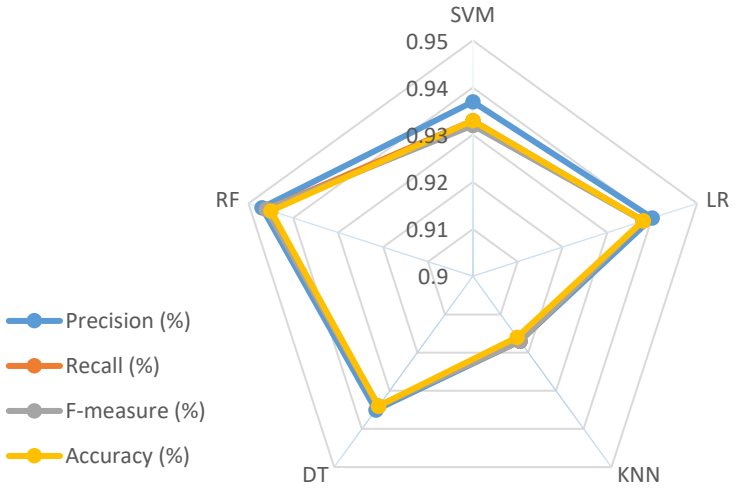

Figure 5.7: Fused feature with Pearson correlation

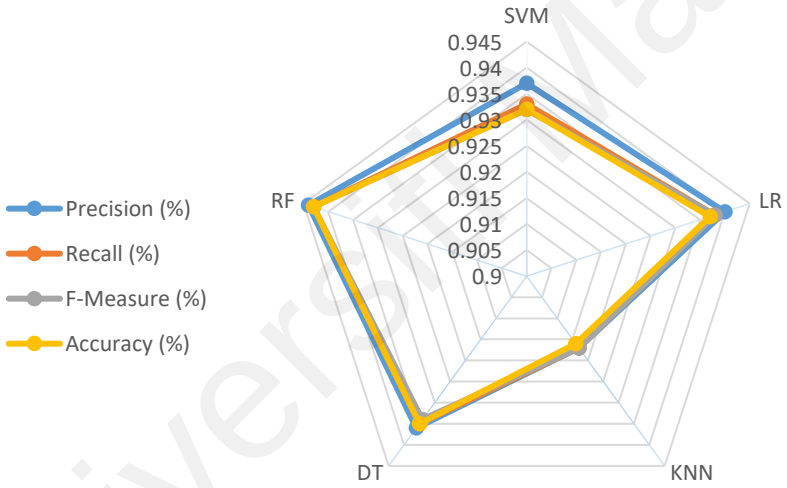

Figure 5.8: Fused feature with information gain.

5.5 Results of Experimental Setting 4.

In this Section, the experimental setting 4 result is presented. The result is based on the evaluation of the proposed framework with a baselines approach. To measure the proposed framework's significance, we performed an extensive set of experiments on our dataset to evaluate classifiers' performance using four baseline approaches for sarcasm identification in Twitter data. The four baseline methods were established to compare

with the proposed framework. The first baseline approach is based on the bag-of-words (BoW) technique experimented in a study conducted by (Khodak et al., 2017). The second baseline is based on word embeddings (word vector), which is another important baseline that uses a contextual word vector that includes GloVe embedding feature (Pennington et al., 2014) trained on 42B corpus as experimented on a study conducted in (Ghosh et al., 2015). The third baseline considered the feature fusion method proposed by Kumar and Garg (2019), which utilized the fusion of pragmatic feature, sentiment feature, and Top-200 TF-IDF features to build the context using five shallow classifiers, which include support vector machine, K-Nearest Neighbor, Decision Tree, Random Forest, and Multilayer Perception classifier. They considered the K-Nearest Neighbor classifier with 3 and 5 neighbors in their parameter settings, whereas, in SVM, they considered RBF and linear kernel. The results obtained from each of the settings are shown in Table 5.6. Out of the seven parameter settings, the best performing settings was obtained on a random forest classifier. Thus, the result of the RF classifier was utilized for the evaluation with the proposed framework.

In Baseline 4, this research considered the proposed approach by Sundararajan et al. (2020) that proposed stacking ensemble feature-based sarcasm detection in Twitter. The study utilized lexical features, emoticon features, internet slang, and hyperbolic features. In their settings, they utilized random forest and AdaBoost on the proposed stacking-based ensemble method. The results obtained from the settings are shown in Table 5.6. However, the best performing settings were obtained on the stack-based ensemble method.

The performance results attained from the baselines were compared with the proposed feature fusion framework. In this setting, a total of thirty analysis as presented in Table 5.7. Five experimental settings were performed to measure classifiers performance by

employing these four baselines. All settings were maintained as utilized. However, only the proposed features were substituted with the baseline features. Thus, similar experiments were performed to determine the best settings for each baseline.

The first Baseline attained a promising result with a Random Forest classifier (precision = 0.835). In Baseline 2, a Random Forest classifier attained the best result (precision 0.721). Baseline 3 achieved the highest result with a Random Forest classifier (precision = 0.787), whereas Baseline 4 obtained the best result with a stack ensemble classifier (precision = 0.666). However, each baseline results' performance evaluation based on each set of experiments is represented in Table 5.6. We compared the best result from our proposed framework with the best results from each baseline. The comparison results are shown in Table 5.7. The last row of the table shows the performance of our proposed framework. With the Random Forest classifier, the best precision of 94.7% was obtained on the proposed framework using the Pearson Correlation feature selection algorithm, which indicates the significance of the proposed Multi-feature Fusion Framework for classifying tweets as sarcastic and non-sarcastic. Thus, our proposed framework outperformed Baseline 1 by 11.2%, Baseline 2 by 22.6%, Baseline 3 by 16%, and Baseline 4 by 28.1% precision during the evaluation experiments. Besides, our framework also shows a relatively higher f-measure when compared with the baselines. In Figure 5.10, the visualization of the comparison is represented. In summary, the comparison results indicate that the developed framework offers a possible solution for sarcasm identification in Twitter data.

Table 5.6: Evaluation Experiments of the baselines.

Baselines	Classifiers	Precision	Recall	F-measure	Accuracy
BL1	SVM	0.812	0.803	0.801	0.802
	LR	0.810	0.806	0.805	0.806
	KNN	0.784	0.776	0.774	0.776
	DT	0.792	0.790	0.789	0.790
	RF	**0.835**	**0.832**	**0.832**	**0.832**
BL2	SVM	0.662	0.659	0.659	0.659
	LR	0.665	0.659	0.659	0.664
	KNN	0.689	0.688	0.688	0.688
	DT	0.685	0.683	0.683	0.683
	RF	**0.721**	**0.720**	**0.720**	**0.720**
BL3	KNN with Neighbor =3	0.737	0.737	0.736	0.736
	KNN with Neighbor =5	0.736	0.735	0.734	0.734
	RF	**0.787**	**0.764**	**0.759**	**0.763**
	MLP	0.740	0.576	0.487	0.575
	DT	0.758	0.757	0.757	0.757
	SVC with Linear Kernel	0.733	0.707	0.698	0.706
	SVC with RBF Kernel	0.731	0.672	0.648	0.671
BL4	RF	0.664	**0.610**	**0.574**	**0.610**
	AdaBoost	0.646	0.523	0.398	0.523
	Stacking ensemble (RF+AdaBoost)	**0.666**	0.606	0.566	0.605

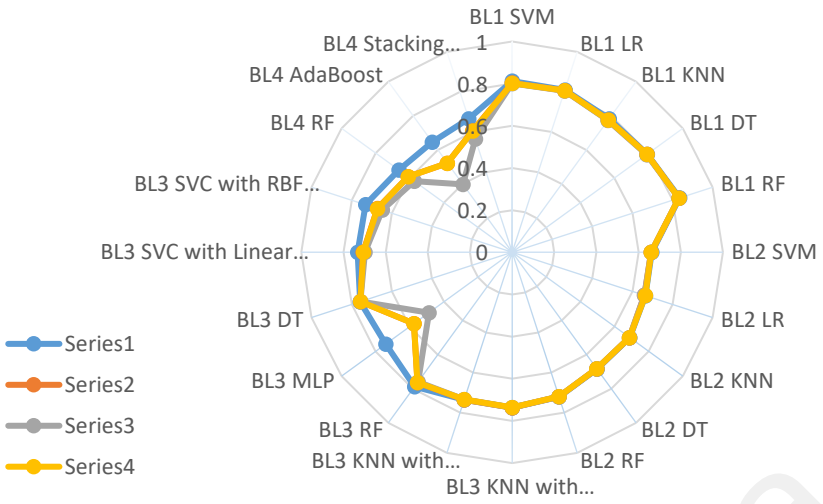

Figure 5.9: Evaluation of four baselines approaches.

Table 5.7: Precision results comparison of the proposed framework with baselines.

Baselines / Proposed framework	Precision (%)	Recall (%)	F-measure (%)	Accuracy (%)
BL1 (Hazarika et al., 2018)	0.835	0.832	0.832	0.832
BL2 (Potamias et al., 2020)	0.710	0.709	0.709	0.710
BL3 (Kumar & Garg, 2019)	0.787	0.728	0.727	0.728
BL4 (Sundararajan et al., 2020)	0.666	0.610	0.574	0.610
Our Proposed Framework	0.947	0.946	0.946	0.945

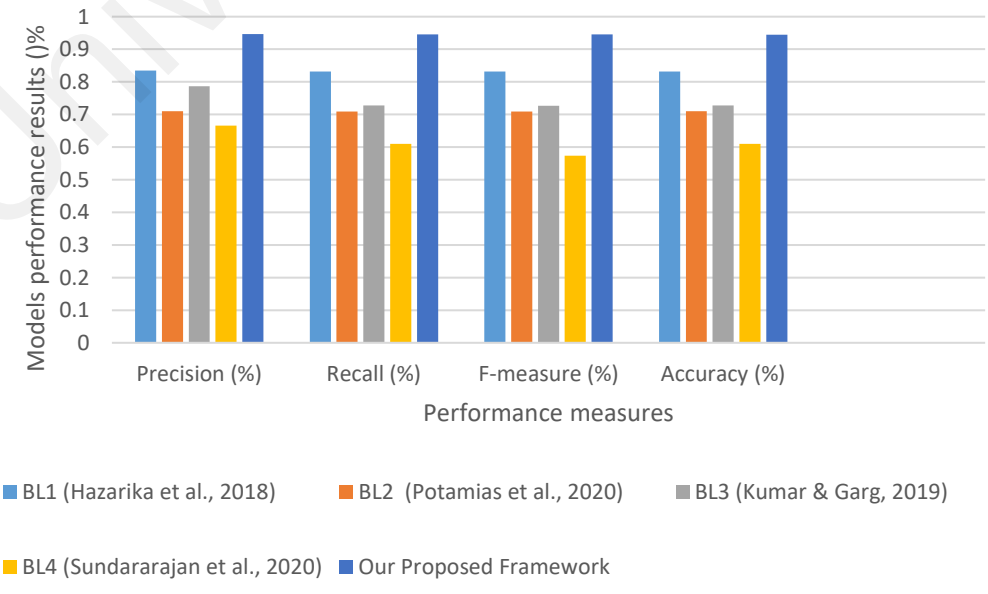

Figure 5.10: Comparison of our proposed framework with baselines.

5.8: The Summary of the Experimental settings results

Results of Experimental Settings 1					
Performance results obtained by considering the lexical features only	Classifier	Precision (%)	Recall (%)	F-measure (%)	Accuracy (%)
	SVM	0.812	0.803	0.801	0.802
	LR	0.810	0.806	0.805	0.806
	KNN	0.784	0.776	0.774	0.776
	DT	0.792	0.790	0.789	0.790
Results of Experimental Settings 2					
Performance results obtained by considering fused features	Classifier	Precision (%)	Recall (%)	F-Measure (%)	Accuracy (%)
	SVM	0.934	0.928	0.928	0.927
	LR	0.934	0.932	0.932	0.931
	KNN	0.910	0.910	0.909	0.910
	DT	0.932	0.931	0.932	0.931
Results of Experimental Settings 3					
Performance results attained on fused features using Pearson correlation	Classifier	Precision (%)	Recall (%)	F-measure (%)	Accuracy (%)
	SVM	0.937	0.933	0.932	0.933
	LR	0.940	0.938	0.938	0.938
	KNN	0.917	0.917	0.917	0.916
	DT	0.935	0.934	0.934	0.934
	RF	0.947	0.946	0.946	0.945
Performance results attained on fused features using Information gain	SVM	0.937	0.933	0.932	0.932
	LR	0.940	0.938	0.938	0.937
	KNN	0.917	0.917	0.917	0.916
	DT	0.936	0.935	0.934	0.935
	RF	0.944	0.943	0.943	0.943
Results of Experimental Settings 4 (Performance results of the baselines)					
Baselines	Classifiers	Precision	Recall	F-measure	Accuracy
BL1	SVM	0.812	0.803	0.801	0.802

	LR	0.810	0.806	0.805	0.806
	KNN	0.784	0.776	0.774	0.776
	DT	0.792	0.790	0.789	0.790
	RF	**0.835**	**0.832**	**0.832**	**0.832**
BL2	SVM	0.662	0.659	0.659	0.659
	LR	0.665	0.659	0.659	0.664
	KNN	0.689	0.688	0.688	0.688
	DT	0.685	0.683	0.683	0.683
	RF	**0.721**	**0.720**	**0.720**	**0.720**
BL3	KNN with Neighbor =3	0.737	0.737	0.736	0.736
	KNN with Neighbor =5	0.736	0.735	0.734	0.734
	RF	**0.787**	**0.764**	**0.759**	**0.763**
	MLP	0.740	0.576	0.487	0.575
	DT	0.758	0.757	0.757	0.757
	SVC with Linear Kernel	0.733	0.707	0.698	0.706
	SVC with RBF Kernel	0.731	0.672	0.648	0.671
BL4	RF	0.664	**0.610**	**0.574**	**0.610**
	AdaBoost	0.646	0.523	0.398	0.523
	Stacking ensemble (RF+AdaBoost)	**0.666**	0.606	0.566	0.605

5.6 Discussions

The observation on the performance results of this study indicates that the proposed multi-feature fusion can classify tweets as sarcastic or non-sarcastic, with predictive results ranging from 78.4% to 94.7%. Moreover, a significant difference was noticed in most of the analysis. The findings from the analysis indicate that lexical feature (BoW) only is not sufficient in classifying tweets as sarcastic or non-sarcastic. Thus, a multi-

feature fusion framework was developed to bolster lexical features with contextual features to enhance the classification's performance. In the subsequence Section, the feature framework's results analysis is discussed under four aspects: the machine learning algorithm, the effect of the contextual features, the feature selection algorithm, and the comparison of the baseline with the proposed framework.

5.6.1 Results analysis of machine learning algorithm

As described in Chapter 2 and 3 of this thesis on the decision on choosing the best classifier, it was noted that the best classifier that can perform well on a particular dataset is quite challenging tasks that lie on the basic theory of algorithm and how it correlates with the attributes of data. In the existing studies, two or more machine learning algorithms are tested to find the best algorithm since it is difficult to find a single classifier that can attain the best performance in all application domains (Wolpert & Macready, 1995). This is because of the variations in the philosophy of the learning process. Machine learning models are made up of various constituents due to the model's composite nature (Vanschoren et al., 2012). However, the machine learning model's superiority may be restricted to a particular domain (Macià et al., 2013). Therefore, the survey of literature carried out in Chapter 2 guided in selecting the classifiers. As mentioned in Chapter 3, the two points employed in narrowing down the classifier's selection as a guide in finalizing the selection. Thus, five different classifiers, including DT, SVM, LR, K-NN, and RF, were selected to assess the feature fusion framework's model performance. As a result, these classifiers were employed with the proposed feature fusion. Among the five selected classifiers, SVM and LR attained the highest performance in classifying tweets as sarcastic or non-sarcastic without the feature selection techniques.

SVM classification task requires the use of threshold function to separate classes using margins. However, it uses the training set to build a model that predicts the target value

of data, giving only the test data attributes (Hsu et al., 2003). In a support vector machine, a hyper-plane, also known as a support vector, is used to separate the two-class data points by reducing the space between them with the help of training sets (Cristianini & Shawe-Taylor, 2000). SVM is not susceptible to overfitting problems that are common with various machine learning algorithms. Logistic regression classifies event occurrence probability as a linear function of a predictor variable class (Kantardzic, 2011). In the LR algorithm, the decision boundaries are usually made by employing a linear function of the features. Logistic regression aims to augment the probability function to recognize the document class label. The results show that both SVM and LR attained similar results in precision, but LR outperformed SVM in terms of recall, f-measure, and accuracy. It can also be observed that RF has a lower performance than SVM without applying feature selection techniques. However, it can be observed that when feature selection techniques were applied to the fused features, the performance results of RF improved and outperformed all the five selected classifiers. Thus, RF's low-performance results without feature selection can be attributed to some redundant features occupying the vector spaces. Among all the five classifiers, KNN had the least performance because it is regarded as a lazy learner. KNN uses the training sets directly for classification instead of learning from it first. Thus, the obtained results with KNN is not generalized, and it is not strong to noisy data (Liu et al., 2004). The decision tree classifier also recorded low performance in classifying sarcastic tweets. This could be attributed to the continuous data representation in the master feature vector, which hampers the optimal thresholds required to create a decision tree (Dreiseitl et al., 2001). Thus decision tree may be unsteady in classifying sarcastic tweets.

5.6.2 Results analysis on the effect of contextual information in addressing the loss of contextual information issue.

The utilization of contextual features in sarcasm identification has recently gained ground in social media platforms. Microblog data contains highly contextual information. As a result, the application of content-based features in sentiment classification becomes relatively ineffective and requires contextual clues (Carvalho, Sarmento, Silva, & De Oliveira, 2009). The term "context" in sentiment analysis is defined as a supplementary source of evidence that can either increase or shift the polarity of the content in expression (Kumar & Garg, 2019). Most studies on sarcasm-related linguistic concepts maintain that employing contextual features that consider tweet context enhances predictive performance (Wallace, 2015). Experimental result 1 presented the results by considering the lexical features only. Lexical features are content-based features, having the drawback of loss of contextual information. Table 5.1 depicts the predictive result obtained by considering lexical features only. The highest predictive result obtained in this experiment is 0.835 precision on the RF classifier. However, when contextual features were added to the content-based feature (Lexical), a significant improvement in predictive performance results was obtained (0.934 precision), as depicted in Section 5.3, Table 5.2. It can be seen from Table 5.2 that RF attained a precision of 0.930, SVM attained a precision of 0.935. LR attained a precision of 0.935, and KNN attained a precision of 0.910, and DT attained a precision of 0.932, which shows the significance of contextual features in sarcasm classification. The differences in the predictive performance obtained by comparing the result in Table 5.1 and Table 5.2 is depicted in Table 5.3.

Thus, it can be inferred that bolstering content-based features with contextual features enhances the predictive performance of sarcasm classification.

5.6.3 Results analysis of Feature Selection Techniques in addressing the Training data Sparsity issues

The predictive performance of models always relies on the quality of the features utilized. The irrelevant feature may produce results that are not comprehensive enough. Thus, it is essential to investigate the performance's effect on the feature selection techniques applied to the proposed feature fusion. This eliminates the redundant features and features with no discriminating ability before the classification task (Hall & Smith, 1998). One of the feature selection goals is to decide on the features to eliminate or retain and utilize for classification. In classifiers construction, feature selection involves selecting discriminating features out of proposed features using statistical analysis approaches. For instance, the proposed lexical features extracted using the BoW model inherent the problem of data sparsity. In such a case, feature vectors extracted by the bag-of-words technique occupy high dimensional vector space. Therefore, not all the feature vectors are relevant, which can cause overfitting in models. Thus, this study utilizes feature selection to eliminate redundant features and reduce training and execution times (Libbrecht & Noble, 2015). Moreover, selecting the discriminating and relevant feature can lower the overfitting problem commonly found in the machine learning model in the training dataset (Sebastiani, 2002). However, the classifier's improvement determines whether the selected features or the fusion of all proposed features have the most discriminating power.

In this study, two feature selection techniques were investigated to test if the performance results could be enhanced in precision, recall, f-measure, and accuracy. A classification algorithm constructed with feature fusion and feature selection is suggested to evaluate the predictive performance in precision, f-measure, recall, and accuracy. Applying feature selection techniques shows that the two feature selection techniques (Pearson correlation and information gain) tested attained almost the same results except

in RF classifiers. However, using Pearson correlation feature selection outperformed the information gain by attaining 94.7% precision over 94.4%, as shown in Table 5.4 and Table 5.5, respectively. Furthermore, when the results obtained in Table 5.4 and Table 5.5 are compared with those obtained from Table 5.2, predictive performance is observed. Hence, there is a significant enhancement in classifiers performance in applying feature selection techniques. Thus, the reduction in Table 5.2 compared with Table 5.4 and Table 5.5 can be attributed to the null features (data sparsity) in the training sets. Consequently, applying the feature selection techniques can eliminate null features and enhance the sarcasm classification's predictive performance.

5.6.4 Result analysis of the proposed framework and baseline approach

The proposed framework was compared with four existing state-of-the-art baseline approaches to investigate the effectiveness of the proposed multi-feature fusion framework for sarcasm identification. The four baselines were created on the dataset utilized in this study. The first baseline is based on the lexical (bag-of-words) feature. The second baseline is based on the word embedding (GLoVe) feature. The third baseline is based on the proposed approach by (Kumar & Garg, 2019), and the fourth baselines are based on the proposed approach (Sundararajan et al., 2020). The purpose of selecting these baselines for evaluation is because they comprised the most utilized features employed in the literature. Besides, those baseline studies are recent studies related to the domain of sarcasm identification. The evaluation comparison shows that the proposed feature fusion framework outperformed the four state-of-the-arts baselines because it overcomes the limitation found in the related studies. As described in Chapter 2, the comprehensiveness of the proposed feature in which machine learning can be learned efficiently is essential in developing an efficient feature fusion framework.

The first baseline uses bag-of-words feature representation. However, simple representation features using bag-of-words, in which each word in the dataset is regarded as a feature, may lead to inadequate features for constructing the machine learning model. This phenomenon is noticed in the predictive performance results in which the proposed framework performed better than bag-of-words. This is due to the drawback imposed in the BoW feature engineering technique as it is concerned with the word's existence, not the word's position in the sentence. This brings the loss of contextual information and word semantics in the representation (Nigam et al., 2000; Sebastiani, 2002). Besides, there is also an issue of data sparsity in vector representation since each expression has a word limit. This issue can create a severe problem during the model training because some words could be seen in the testing set only but certainly not found in the training set, making most of the training features sparse. However, when some of the words found in the testing sets are missing in the training sets, there will be a divergence between the testing and training sets, resulting in poor performance results in the classifiers. This problem is common when constructing machine learning classifiers in sentiment analysis tasks such as sarcasm. Moreover, not all words can be regarded as significant features, and as a result, highly convergence words can be selected and utilized as discriminating features for effective classifiers construction. Thus, results obtained using the BoW feature engineering approach are not comprehensive and generalized for sarcasm identification.

The second baseline uses word embedding (word vector) feature representation to classify a tweet as sarcastic or non-sarcastic. Low-performance results were also recorded on the second baseline method. The brain behind the low performance on using the word embedding feature is due to the limitation inherent in such representation. One of the major limitations of word embedding is that it ignores the sentiment polarity of words (Araque et al., 2017; Giatsoglou et al., 2017). Though word embedding-based word vector

captures the word's context, words with 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.

The third baseline combines various features that included utilized pragmatic feature, sentiment feature, and Top-200 TF-IDF features to build the sarcasm classification using shallow classifiers. Low-performance results are also observed in the baseline because the approach did not utilize word embedding-based features even though it included the sentiment-related features. Word embedding is imperative in sentiment analysis study, especially in the sarcasm classification task, as it captures the word semantics in the sentence. However, due to the deficiency of word embedding-based features, the word co-occurrence in the text is not captured. Thus, more features that include word embedding should be explored for effective classification and performance enhancement.

The fourth baseline is based on the combination of various features such as lexical, emoticon, internet slang, and hyperbolic feature. This baseline attained the lowest performance. Though this baseline contains some discriminative features for sarcasm classification yet, these features are not adequate and comprehensive enough because the pragmatic feature is missing. Pragmatic features are markers that describe the “meaning in the context.” They understand the way utterances are made. The pragmatic markers such as emoticons, punctuation marks, capitalization, vocalization signals are often employed in sarcastic utterances. Thus, pragmatic features should be considered as important features in sarcasm classification. Also, some important features that are paramount for sarcasm classification, such as sentiment-related features and context

embedding features, are missing in the study. As a result, low-performance results were obtained on the baseline 4 study approach.

Therefore, the comparison results indicate that the multi-feature fusion framework utilizing the proposed features is more effective for sarcasm classification when compared with the four baseline approaches.

5.7 Chapter summary

This Chapter provides the results and discussions of the experimental settings presented in Chapter 4 of this thesis to evaluate the proposed feature fusion framework's significance. It further discusses results analysis of machine learning algorithm, feature selection techniques, and the proposed framework comparison with baselines to show the significance of the proposed framework. We experimented with various feature analysis to select the features with substantial discriminative ability to enhance the results of the predictive performance. Precision was utilized as the major performance measure due to its robustness in measuring classifiers. The highest classification result was attained by employing classifiers that use feature selection techniques to select the features with discriminative power. Random forest classifier with Pearson correlation feature selection technique attained the highest precision (0.947) and f-measure (0.946), followed by logistic regression with precision (0.940) and f-measure (0.938). The comparison of the highest result outperformed the four baselines approach in terms of performance results, which shows the importance of the proposed framework for sarcasm identification. Thus, the results show that the fusion of length of microblog, semantic, sentiment, pragmatic, emoticon, discourse marker, syntactic, and hashtag features, with the lexical feature, resolves data sparsity by augmenting lexical feature with eight other features, thereby reducing data sparsity. Also, the results show that the context of words can be captured by bolstering lexical features with some contextual features such as semantic features,

discourse markers, NLP feature (POS), sentiment, pragmatics, and hashtag features. Furthermore, RF and LR classification algorithms are more suitable for classifying sarcastic tweets. Moreover, the promising results show that researchers and practitioners can utilize the proposed framework to enhance sentiment classification and opinion mining due to its ability to recognize sarcastic utterances in Twitter data.

CHAPTER 6: CONCLUSION

6.1 Introduction

This Chapter presents the conclusion of this thesis and describes possible further research directions. This thesis investigated and explored the existing methods for sarcasm classification. A live streaming dataset obtained from Twitter using Twitter API was utilized for classification experiments in this study. Various sets of features were proposed and extracted from the dataset that consists of lexical, length of microblog, hashtag, discourse markers, emoticon, syntactic, pragmatic, semantic (GloVe embedding), and sentiment related features, which are selected based on observations from the characteristics of the data and evidence from the literature. This study proposed a multi-feature fusion framework for sarcasm identification to enhance the predictive performance and address the limitations inherent in the related studies. To measure the significance of the proposed framework, various extensive sets of experiments were performed on the dataset to evaluate classifiers' performance using four baseline methods for sarcasm identification in Twitter data. The four baseline methods were established to compare with the proposed framework. The experimental results indicate that the proposed framework outperformed the four baseline methods in precision, recall, f-measure, and accuracy.

Each research question presented in Chapter 1 has been answered and discussed as presented in Chapter 2 to 5 of this thesis. This thesis concluded by re-visiting the research objectives defined in Chapter 1 by describing how they were accomplished. In addition, it presents the major contributions of this study together with the limitations identified in this study and the future directions. This study aims to achieve three objectives and seven research questions as described in subsection 6.2 below. The summary of the findings is shown in Table 6.1

6.2 Reappraisal of the research objectives and research questions

In this Section, the research questions and objectives presented in Chapter 1 of this study are re-visited to describe how they are achieved. It also provides concise outcomes of individual research questions of each objective. Thus, various research objectives are mapped against the research questions to discuss the research finding.

Research objective 1: To investigate the existing feature engineering and fusion approaches for sarcasm identification in Twitter data.

To attain this objective, a review of academic literature in the domain of sarcasm detection was carried out under dataset usage, pre-processing techniques, feature engineering techniques, the modelling approach, and performance metrics. 43 primary studies from 6 academic databases (including Science Direct, IEE Xplore, and ACM) were systematically selected and extensively reviewed under the six aspects mentioned above to accomplish the first objective. In addition, the outcomes of the individual research question under objective 1 is presented below.

RQ1: *What are the existing feature engineering and fusion approaches employed for sarcasm identification in Twitter data?*

The literature review identified several existing feature engineering and fusion approaches for sarcasm identification, namely BoWs, N-gram, word embedding, data-level fusion, feature level fusion, and multi-classifiers fusion. This answer is provided through an extensive review of the literature (see Chapter 2). A detailed, comprehensive discussion on feature engineering approaches, including their limitations, is provided in Chapter 2 (Section 2.5.3, 2.6, 2.7.3, and 2.8.3).

RQ2: *What are the shortcomings in the current feature engineering approach for sarcasm identification in Twitter data?*

The literature review identified three major limitations on the existing feature engineering approaches for sarcasm identification, including ignoring contextual and semantic information in the sarcasm expression, training data sparsity in vector representation, and ignoring the sentiment polarity in the word embedding feature engineering approach in sarcastic utterances. As a result of these shortcomings, these approaches attained low classification performance for classifying tweets into sarcastic and non-sarcastic. A detailed discussion on the shortcomings of the existing feature engineering approach is presented in Chapter 2 (Section 2.7.3 and 2.8.2). Thus, this thesis proposes a Multi-Feature Fusion Framework for sarcasm identification to overcome the existing feature engineering technique problem by addressing the context of words, sentiment polarity issue of word embeddings, and training data sparsity in expression classifying tweets as sarcastic or non-sarcastic. A detailed explanation of the proposed framework is presented in Chapter 4 and 5 of this thesis.

Research objective 2: To develop a Multi-feature Fusion Framework for sarcasm identification to improve the performance and address the context of words, training data sparsity and sentiment polarity issues in sarcasm expression.

To attain this objective, a Multi-feature Fusion Framework is proposed using two classification stages. The first stage classification is constructed using a lexical feature extracted by the BoW approach only. It is trained using five standard classifiers, including SVM, DT, KNN, LR, and RF, to predict the sarcastic tendency based on the lexical feature. In stage two, the lexical feature is fused with other extracted features, which include the length of microblog, the emoticon, the synthetic, the semantic, the sentiment, the discourse markers, the hashtag, and the pragmatic features to form a feature fusion, which is employed to model a context to obtain a final prediction using various classifiers. The effectiveness of the developed multi-feature fusion framework is tested with various

experimental analysis, which was performed to obtain the performance of classifiers. The detailed development of the feature fusion technique for sarcasm identification is presented in Chapter 4 of this thesis. Lastly, the formulated research questions to realize the stated objective is described below.

RQ3: *What are the most useful features for sarcasm identification by researchers?*

The findings from the literature revealed that content-based features such as unigram, bigram, trigram, rating features, word features, acronym feature punctuation features, and emoticon features were the most useful features by the researchers for sarcasm identification. However, it is not encouraged to rely only on the content-based features for classification in sarcasm identification. This is because of the limited accuracy of the classification performance due to the limitations inherent in those features. One issue with the content-based feature is the loss of contextual information and grammar even though the word frequency is retained. Secondly, the content-based training data contains null features, thereby making the training data sparse. To avoid these limitations, a combination of contextual and content-based features is necessary to enhance classification accuracy. This thesis explains a detailed review of the features used for sarcasm classification in Chapter 2 (Section 2.7.3.2).

RQ4*: How can the loss of contextual information be mitigated through the development of the Multi-feature Fusion Framework?*

To answer this research question, this thesis developed a Multi-feature fusion framework for sarcasm identification. The loss of contextual information is mitigated by employing the Multi-Feature Fusion Framework that contains contextual information features such as GloVe embedding features, Discourse marker feature, hashtag features, semantic and syntactic features. The experimental results that answered this research question are

presented in Section 5.3, Table 5.2 of this thesis. Thus, to avoid loss of contextual information in sarcasm classification, context-based features should be considered in addition to content-based features, which will enhance the model's predictive performance.

RQ5*: How can the sparsity of the training data (Null features) be resolve through the development of the Multi-feature Fusion Framework?*

The training data sparsity can be resolved by performing a feature selection technique on the feature fusion to select features with discriminating power and eliminating the null features to reduce the high dimensional feature vector space. The feature selection algorithm was initially performed on the lexical feature to obtain the top 100 discriminative features. The feature selection algorithm was also performed on each of the other features with two or more subsets, such as hashtags, emoticons, syntactic, pragmatic semantic (GloVe embedding) sentiment-related features, to check the discriminating power of each subset. However, any of the features with a low threshold is eliminated. Next, the features selected from the lexical, hashtag, emoticons, syntactic, pragmatic semantic (GloVe embedding), and sentiment related features were fused with discourse marker feature and length of microblog feature to form a new fused feature (feature fusion with feature selection). The feature selection was performed using two feature selection algorithms. Firstly, by using the Pearson correlation algorithm, and secondly, the information gain feature selection algorithm. The experimental results that answered this research question are represented in Chapter 4, Section 5.4 (Table 5.4 and Table 5.5).

RQ6*: How can the sentiment polarity of words be captured through the development of the Multi-feature Fusion Framework?*

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. To answer this research question, this thesis developed a Multi-feature Fusion Framework that contains sentiment related features (such as positive words, highly emotional positive words, negative words, and highly emotional negative words), polarity sentiment such as (positive sentiment and negative sentiment), and hashtag features (positive and negative hashtag) features. The experimental results that answered this research question are presented in Section 5.3, Table 5.2 of this thesis. Thus, it is important to bolster word embedding features with sentiment-related features for feature vector representation to avoid this limitation.

Research objective 3: To evaluate the performance of the proposed Multi-feature fusion Framework using the real-world datasets by evaluating the performance with the baseline methods for sarcasm classification.

The third objective of this research is to assess the significance and effectiveness of the proposed multi-feature fusion framework for sarcasm identification. To do so, the proposed framework is compared with the state-of-the-art baseline methods. However, the experimental results indicated that the proposed framework outperformed the existing state-of-the-arts baseline methods for sarcasm classification. Chapter 5 (Section 5.5) provides a detailed evaluation of the proposed framework with baselines.

RQ7: *What are the existing performance measures appropriate for evaluating the proposed Multi-feature fusion framework for sarcasm identification in Twitter data, and how much can the proposed framework's performance results be enhanced compared with the performance of the baseline methods?*

Standard performance measures that have been identified for measuring the proposed feature fusion framework for sarcasm identification are precision, recall, f-measure, and accuracy. The reason for using the metrics mentioned above is provided in Chapter 2 (Section 2.7.5) and Chapter 3 (Section 3.9) of this thesis. However, the overall prediction precision of 94.7% was obtained on the proposed Multi-feature fusion framework for sarcasm identification. However, the performance of the proposed feature fusion was compared with four state-of-the-arts baseline methods. The comparative results indicated that the proposed framework attained better performance results ranging from 11.2% to 27.1% precision compared with the baseline methods. Chapter 5 (Section 5.5) provides a detailed discussion of the comparison. The experimental results that answered this research question are represented in Table 5.6 and Table 5.7.

Table 6.1: Summary of the findings

Research Objectives	Methodology	Status
RO1: To investigate the existing feature engineering and fusion approaches for sarcasm identification in Twitter data.	Literature analysis and comparison study	Achieved
RO2: To develop a Multi-feature Fusion Framework for sarcasm identification to improve the performance and address the context of words, training data sparsity and sentiment polarity issue in sarcasm expression.	✓ Development of Multi-feature Fusion Framework that contains ➢ Contextual information features such GloVe embedding features, Discourse marker feature, hashtag features, semantic and syntactic features (context of the words issue) ➢ Sentiment related features (such as positive words, highly emotional positive words, negative words, and highly emotional negative words), polarity sentiment such as (positive sentiment and negative sentiment), and hashtag features (positive and negative hashtag) (features sentiment polarity issue) ✓ Development of a feature selection technique on the feature fusion to select features with discriminating power and eliminating the null features to reduce the high dimensional feature vector space (sparsity issue)	Achieved

RO3: To evaluate the performance of the proposed Multi-feature fusion Framework using the real-world datasets by evaluating the performance with the baseline methods for sarcasm classification.	✓ The overall prediction precision of 94.7% was obtained on the proposed Multi-feature fusion framework for sarcasm identification. ✓ The comparative results indicated that the proposed framework attained better performance results ranging from 11.2% to 27.1% precision compared with the baseline methods.	Achieved

6.3 Limitation and Further Research Direction.

This Section presents the study limitation for the sarcasm identification framework and identifies further research directions to enhance the sarcasm classification study.

6.3.1 The exploitation of multi-modal data and new features

The current study employed a tweet dataset for the experiments. Though, tweets data are effective in detecting sarcasm in textual data in which this study focused on. To ensure the comprehensives in this study, other social media corpus such as product review, internet argumentation corpus are required for sarcasm identification study. In addition, for more generic research and large-scale application, other datasets consisting of homogeneous – heterogeneous data (Chapter 2 SubSection 2.7.1) can be considered to construct a classification model for sarcasm identification. In such multi-modal data, audio and visual features can be integrated for the classifier's construction. This study also made use of both content-based and contextual features in the classification phase for sarcasm identification. However, further research can take advantage of the behavioural features (Bharti et al., 2016; Zhang et al., 2016) to identify sarcasm. A study conducted by (Schifanella et al., 2016) for sarcasm identification made use of the visual semantics feature (VSF), in which the sarcasm can only be understood through the semantics in the image and was able to attain a higher accuracy when combined with N-gram using the SVM classifier. Therefore, future research is important to explore various novel features such as behavioural, audio, and visual features for sarcasm identification.

6.3.2 Multilingual-based approach

The majority of the existing works on sarcasm identification utilized only English language datasets. However, most people usually express their emotions better in their native languages than in English. Thus, mining such opinions becomes problematic because many people do not have an interest in such research; that is why most existing works on sarcasm classification paid more attention to textual data expressed only in English. As such, further research that will focus on feature extraction on varieties of languages such as Chinese, Mexican, Turkish, Spanish, etc., and modification of classifiers is urgently required to be applicable in sarcasm classification in more than one language.

6.3.3 Application of Deep learning methods

Most researchers in the data mining domain are now shifting from traditional machine learning to Deep learning methods due to the cumbersomeness inherent in the pre-classification phase, especially the feature extraction phase in the traditional machine learning approaches sarcasm identification. The deep learning model applies a computational approach that consists of various processing layers for learning data representation with varying abstraction degrees. The deep learning approach is required to overcome such issues, as the features are automatically represented and not engineered by human intervention. The classification accuracy of the sarcasm detection can be enhanced by applying different deep learning techniques for effective feature extractions such as word to vector (word2vec) conversion, N-gram, and bag-of-words. Some deep learning classification algorithms, such as Recurrent Neural Networks (RNN) and Convolutional Neural Network (CNN), have reported good performance when applied in sarcasm identification. Deep learning has also enhanced the performance accuracy in many texts and web mining classification (Dumais & Chen, 2000). As such, deep learning

methods can benefit multi-feature fusion framework classification that consists of multiple features that were manually extracted.

6.3.4 Clustering-based approach

The clustering-based approach deploys an unsupervised learning approach (Yang, 1993) that is mostly applicable in pattern recognition, but this is still an infant in the domain of sarcasm identification. Most researchers in the selected studies implemented a supervised learning approach to build a classification model and obtained a good result despite the limitations inherent in such approaches. One of the key issues in supervised learning is the labelling of the datasets to construct the training sets. Such tasks require linguistic experts, and they are time-consuming. Thus, a tremendous amount of time is required in the preparation, and disagreement could arise in a situation where more than one expert is engaged for annotation. So, focusing more on the unsupervised approach (clustering) for modelling sarcasm identification helps eliminate such labelling exertion. However, finding pattern within two or more classes via unsupervised grouping still remain problematic. Thus, an adequate study is needed to develop a complete automated unsupervised algorithm for sarcasm classification and attain better prediction than the proposed framework in this thesis.

6.3.5 Transfer learning based on BERT Model

A transfer learning technique based on BERT (Bidirectional Encoder Representation from Transformers) is another open research direction for sarcasm identification as it has recorded promising results in many NLP tasks. BERT is the first deep bidirectional and unsupervised language model, which uses only plain text data to pre-train the model. Unlike the existing models constrained on unidirectional by employing a mask language model that randomly masks some tokens from the input, BERT removes such barriers and allows training on deep bidirectional transformers. In addition, it pre-train text pair

representation by employing the next sentence prediction (NSP) task. The configuration of BERT consists of two innovative prediction tasks such as Next Sentence Prediction and Masked LM. Studies have revealed that the pre-trained BERT model produces a better performance when compared with ELMO and OpenAI GPT in the sequence of the downstream task in NLP (Devlin et al., 2018). Thus, transfer learning that captures more discriminative features that can enhance the sarcasm classification performance is highly required.

6.4 Conclusion

This Chapter brings the thesis to a conclusion by reappraising the research questions and objectives of this study. Moreover, the Chapter presents the discussions on how questions and research objectives were achieved. However, the findings show the significance of the proposed multi-feature fusion framework and context-based feature technique for sarcasm identification. The proposed approaches outperformed the existing baseline methods compared. Finally, this Chapter provides the limitations of this thesis and open research directions to address the limitations as mentioned above in the sarcasm classification study domain.

REFERENCES

Abercrombie, G., & Hovy, D. (2016). *Putting sarcasm detection into context: The effects of class imbalance and manual labelling on supervised machine classification of twitter conversations.* Paper presented at the Proceedings of the ACL 2016 Student Research Workshop.

Abulaish, M., & Kamal, A. (2018). *Self-Deprecating Sarcasm Detection: An Amalgamation of Rule-Based and Machine Learning Approach*. Paper presented at the 2018 IEEE/WIC/ACM International Conference on Web Intelligence (WI).

Adeva, J. G., Atxa, J. P., Carrillo, M. U., & Zengotitabengoa, E. A. (2014). Automatic text classification to support systematic reviews in medicine. *Expert Systems with Applications, 41*(4), 1498-1508.

Aggarwal, C. C., & Zhai, C. (2012). A survey of text classification algorithms. In *Mining text data* (pp. 163-222): Springer.

Agrawal, A., An, A., & Papagelis, M. (2020). *Leveraging transitions of emotions for sarcasm detection.* Paper presented at the Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval.

Al-Ghadhban, D., Alnkhilan, E., Tatwany, L., & Alrazgan, M. (2017). *Arabic sarcasm detection in Twitter.* Paper presented at the 2017 International Conference on Engineering & MIS (ICEMIS).

Al-Sallab, A., Baly, R., Hajj, H., Shaban, K. B., El-Hajj, W., & Badaro, G. (2017). Aroma: A recursive deep learning model for opinion mining in arabic as a low resource language. *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP), 16*(4), 1-20.

Altrabsheh, N., Cocea, M., & Fallahkhair, S. (2015). Detecting sarcasm from students' feedback in Twitter. In *Design for teaching and learning in a networked world* (pp. 551-555): Springer.

Álvarez-Pato, V. M., Sánchez, C. N., Domínguez-Soberanes, J., Méndoza-Pérez, D. E., & Velázquez, R. J. F. (2020). A Multisensor Data Fusion Approach for Predicting Consumer Acceptance of Food Products. *9*(6), 774.

Amini, F., & Hu, G. J. E. S. w. A. (2021). A two-layer feature selection method using genetic algorithm and elastic net. *166*, 114072.

Amir, S., Wallace, B. C., Lyu, H., & Silva, P. C. M. J. (2016). Modelling context with user embeddings for sarcasm detection in social media. *arXiv preprint arXiv:1607.00976*.

Araque, O., Corcuera-Platas, I., Sánchez-Rada, J. F., & Iglesias, C. A. (2017). Enhancing deep learning sentiment analysis with ensemble techniques in social applications. *Expert Systems with Applications, 77*, 236-246.

Arevalillo-Herráez, M., Domingo, J., & Ferri, F. J. J. P. R. L. (2008). Combining similarity measures in content-based image retrieval. *29*(16), 2174-2181.

Babanejad, N., Davoudi, H., An, A., & Papagelis, M. (2020). *Affective and Contextual Embedding for Sarcasm Detection.* Paper presented at the Proceedings of the 28th International Conference on Computational Linguistics.

Banerjee, A., Bhattacharjee, M., Ghosh, K., Chatterjee, S. J. M. T., & Applications. (2020). Synthetic minority oversampling in addressing imbalanced sarcasm detection in social media. *79*(47), 35995-36031.

Barbieri, F., Saggion, H., & Ronzano, F. (2014). *Modelling sarcasm in twitter, a novel approach.* Paper presented at the Proceedings of the 5th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis.

Benesty, J., Chen, J., Huang, Y., & Cohen, I. (2009). Pearson correlation coefficient. In *Noise reduction in speech processing* (pp. 1-4): Springer.

Berry, M. W., & Castellanos, M. (2004). Survey of text mining. *Computing Reviews, 45*(9), 548.

Bharti, S., Vachha, B., Pradhan, R., Babu, K., & Jena, S. (2016). Sarcastic sentiment detection in tweets streamed in real time: a big data approach. *Digital Communications and Networks, 2*(3), 108-121.

Bharti, S. K., Babu, K. S., & Jena, S. K. (2015). *Parsing-based Sarcasm Sentiment Recognition in Twitter Data*. Paper presented at the Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2015 - ASONAM '15.

Bharti, S. K., Naidu, R., & Babu, K. S. (2017). *Hyperbolic feature-based sarcasm detection in tweets: a machine learning approach.* Paper presented at the 2017 14th IEEE India Council International Conference (INDICON).

Bird, S., Klein, E., & Loper, E. (2009). *Natural language processing with Python: analyzing text with the natural language toolkit*: " O'Reilly Media, Inc.".

Bouazizi, M., & Ohtsuki, T. (2015a). *Opinion mining in Twitter: How to make use of sarcasm to enhance sentiment analysis.* Paper presented at the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM).

Bouazizi, M., & Ohtsuki, T. (2015b). *Sarcasm Detection in Twitter:" All Your Products Are Incredibly Amazing!!!"-Are They Really?* Paper presented at the 2015 IEEE Global Communications Conference (GLOBECOM).

Bouazizi, M., & Ohtsuki, T. O. (2016). A pattern-based approach for sarcasm detection on twitter. *IEEE Access, 4*, 5477-5488.

Brooks, R. R., Ramanathan, P., & Sayeed, A. M. J. P. o. t. I. (2003). Distributed target classification and tracking in sensor networks. *91*(8), 1163-1171.

Buchan, K., Filannino, M., & Uzuner, Ö. (2017). Automatic prediction of coronary artery disease from clinical narratives. *Journal of biomedical informatics, 72*, 23-32.

Burfoot, C., & Baldwin, T. (2009). *Automatic satire detection: Are you having a laugh?* Paper presented at the Proceedings of the ACL-IJCNLP 2009 conference short papers.

Carr, C., & Zukowski, Z. (2019). *Curating Generative Raw Audio Music with DOME.* Paper presented at the Joint Proceedings of the ACM IUI 2019 Workshops.

Carvalho, P., Sarmento, L., Silva, M. J., & De Oliveira, E. (2009). *Clues for detecting irony in user-generated contents: oh...!! it's so easy.* Paper presented at the Proceedings of the 1st international CIKM workshop on Topic-sentiment analysis for mass opinion.

Carvalho, P., Sarmento, L., Silva, M. J., & Oliveira, E. d. (2009). *Clues for detecting irony in user-generated contents: oh...!! it's "so easy" ;-)*. Paper presented at the Proceedings of the 1st international CIKM workshop on Topic-sentiment analysis for mass opinion, Hong Kong, China. https://doi.org/10.1145/1651461.1651471

Castro, S., Hazarika, D., Pérez-Rosas, V., Zimmermann, R., Mihalcea, R., & Poria, S. (2019a). Towards Multimodal Sarcasm Detection (An _Obviously_ Perfect Paper). *arXiv preprint arXiv:1906.01815*.

Castro, S., Hazarika, D., Pérez-Rosas, V., Zimmermann, R., Mihalcea, R., & Poria, S. J. a. p. a. (2019b). Towards multimodal sarcasm detection (an _Obviously_ perfect paper).

Chen, J., Liu, Y., & Zou, M. (2016). Home location profiling for users in social media. *Information & Management, 53*(1), 135-143.

Chen, X., Zhao, Y., Zhang, Y.-Q., & Harrison, R. (2007). *Combining SVM classifiers using genetic fuzzy systems based on AUC for gene expression data analysis.* Paper presented at the International Symposium on Bioinformatics Research and Applications.

Chia, Z. L., Ptaszynski, M., Masui, F., Leliwa, G., Wroczynski, M. J. I. P., & Management. (2021). Machine Learning and feature engineering-based study into sarcasm and irony classification with application to cyberbullying detection. *58*(4), 102600.

Chowdhury, A. K., Tjondronegoro, D., Chandran, V., & Trost, S. G. (2017). Ensemble methods for classification of physical activities from wrist accelerometry. *Medicine & Science in Sports & Exercise, 49*(9), 1965-1973.

Constantinidis, A., Fairhurst, M. C., & Rahman, A. F. R. J. P. R. (2001). A new multi-expert decision combination algorithm and its application to the detection of circumscribed masses in digital mammograms. *34*(8), 1527-1537.

Cotelo, J. M., Cruz, F. L., Troyano, J. A., & Ortega, F. J. (2015). A modular approach for lexical normalization applied to Spanish tweets. *Expert Systems with Applications, 42*(10), 4743-4754.

Cristianini, N., & Shawe-Taylor, J. (2000). *An introduction to support vector machines and other kernel-based learning methods*: Cambridge university press.

da Silva, N. F. F., Hruschka, E. R., & Hruschka, E. R. (2014). Tweet sentiment analysis with classifier ensembles. *Decision Support Systems, 66*, 170-179. doi:10.1016/j.dss.2014.07.003

Dai, Q.-Y., Zhang, C.-p., & Wu, H. (2016). Research of decision tree classification algorithm in data mining. *International Journal of Database Theory and Application, 9*(5), 1-8.

Dasarathy, B. V. (1994). *Decision fusion* (Vol. 1994): IEEE Computer Society Press Los Alamitos, CA.

Dasarathy, B. V. J. I. C. S. T. (1991). Nearest neighbor (NN) norms: NN pattern classification techniques.

Dash, M., & Liu, H. J. I. d. a. (1997). Feature selection for classification. *1*(3), 131-156.

Dave, A. D., & Desai, N. P. (2016). *A comprehensive study of classification techniques for sarcasm detection on textual data.* Paper presented at the 2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT).

Davidov, D., Tsur, O., & Rappoport, A. (2010). *Semi-supervised recognition of sarcastic sentences in twitter and amazon.* Paper presented at the Proceedings of the fourteenth conference on computational natural language learning.

Davoudi, A., Klein, A. Z., Sarker, A., & Gonzalez-Hernandez, G. J. A. S. o. T. S. P. (2020). Towards automatic bot detection in Twitter for health-related tasks. *2020*, 136.

Debole, F., & Sebastiani, F. (2004). Supervised term weighting for automated text categorization. In *Text mining and its applications* (pp. 81-97): Springer.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Fei-Fei, L. (2009). *Imagenet: A large-scale hierarchical image database.* Paper presented at the 2009 IEEE conference on computer vision and pattern recognition.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Dharwal, P., Choudhury, T., Mittal, R., & Kumar, P. (2017). *Automatic sarcasm detection using feature selection.* Paper presented at the 2017 3rd International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT).

Dictionary, C. (2008). Cambridge advanced learner's dictionary. In: PONS-Worterbucher, Klett Ernst Verlag GmbH.

Dictionary, M. E., & Rundell, M. (2007). Macmillan English Dictionary. In: Macmillan Education.

Dobbins, C., Rawassizadeh, R., & Momeni, E. (2017). Detecting physical activity within lifelogs towards preventing obesity and aiding ambient assisted living. *Neurocomputing, 230*, 110-132.

Domingos, P. (2012). A few useful things to know about machine learning. *Communications of the ACM, 55*(10), 78-87.

Dreiseitl, S., Ohno-Machado, L., Kittler, H., Vinterbo, S., Billhardt, H., & Binder, M. (2001). A comparison of machine learning methods for the diagnosis of pigmented skin lesions. *Journal of biomedical informatics, 34*(1), 28-36.

Ducret, M., Kruse, L., Martinez, C., Feldman, A., & Peng, J. (2020). *You Don't Say... Linguistic Features in Sarcasm Detection.* Paper presented at the CEUR Workshop Proceedings.

Dumais, S., & Chen, H. (2000). Hierarchical classification of web content. InProceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (pp. 256-263). In: ACM Press.

Edwin, L., & Ayu, P. (2013). indonesian social media sentiment analysis with sarcasm detection.pdf.

Eichstaedt, J. C., Schwartz, H. A., Kern, M. L., Park, G., Labarthe, D. R., Merchant, R. M., . . . Sap, M. (2015). Psychological language on Twitter predicts county-level heart disease mortality. *Psychological science, 26*(2), 159-169.

Eke, C. I., Norman, A., Shuib, L., Fatokun, F. B., & Omame, I. (2020). *The Significance of Global Vectors Representation in Sarcasm Analysis.* Paper presented at the 2020 International Conference in Mathematics, Computer Engineering and Computer Science (ICMCECS).

Eke, C. I., Norman, A. A., Shuib, L., & Nweke, H. F. (2019). A Survey of User Profiling: State-of-the-Art, Challenges, and Solutions. *IEEE Access, 7*, 144907-144924. doi:10.1109/ACCESS.2019.2944243

Eke, C. I., Norman, A. A., & Shuib, L. J. I. A. (2021). Context-Based Feature Technique for Sarcasm Identification in Benchmark Datasets Using Deep Learning and BERT Model. *9*, 48501-48518.

Fernández-Delgado, M., Cernadas, E., Barro, S., & Amorim, D. (2014). Do we need hundreds of classifiers to solve real world classification problems? *The Journal of Machine Learning Research, 15*(1), 3133-3181.

Fersini, E., Messina, E., & Pozzi, F. A. (2014). Sentiment analysis: Bayesian ensemble learning. *Decision Support Systems, 68*, 26-38.

Fersini, E., Pozzi, F. A., & Messina, E. (2015). *Detecting irony and sarcasm in microblogs: The role of expressive signals and ensemble classifiers.* Paper presented at the 2015 IEEE International Conference on Data Science and Advanced Analytics (DSAA).

Filatova, E. (2012). *Irony and Sarcasm: Corpus Generation and Analysis Using Crowdsourcing.* Paper presented at the Lrec.

Forslid, E., & Wikén, N. (2015). Automatic irony-and sarcasm detection in Social media. In.

George, A., Ganesh, H. B., Kumar, M. A., & Soman, K. (2019). Significance of Global Vectors Representation in Protein Sequences Analysis. In *Computer Aided Intervention and Diagnostics in Clinical and Medical Images* (pp. 261-269): Springer.

Ghosh, A., & Veale, T. (2016). *Fracking sarcasm using neural network.* Paper presented at the Proceedings of the 7th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis.

Ghosh, D., Guo, W., & Muresan, S. (2015). *Sarcastic or not: Word embeddings to predict the literal or sarcastic meaning of words.* Paper presented at the proceedings of the 2015 conference on empirical methods in natural language processing.

Giatsoglou, M., Vozalis, M. G., Diamantaras, K., Vakali, A., Sarigiannidis, G., & Chatzisavvas, K. C. (2017). Sentiment analysis leveraging emotions and word embeddings. *Expert Systems with Applications, 69*, 214-224.

González-Ibánez, R., Muresan, S., & Wacholder, N. (2011). *Identifying sarcasm in Twitter: a closer look.* Paper presented at the Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Short Papers-Volume 2.

Graves, A., Jaitly, N., & Mohamed, A.-r. (2013). *Hybrid speech recognition with deep bidirectional LSTM.* Paper presented at the 2013 IEEE workshop on automatic speech recognition and understanding.

Graves, A., & Schmidhuber, J. (2005). Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural networks, 18*(5-6), 602-610.

Gray, R. M. (1990). Entropy and information. In *Entropy and Information Theory* (pp. 21-55): Springer.

Grosseck, G., & Holotescu, C. (2008). *Can we use Twitter for educational activities.* Paper presented at the 4th international scientific conference, eLearning and software for education, Bucharest, Romania.

Guyon, I., & Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of machine learning research, 3*(Mar), 1157-1182.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I. H. (2009). The WEKA data mining software: an update. *ACM SIGKDD explorations newsletter, 11*(1), 10-18.

Hall, M. A. (1999). *Correlation-based feature selection for machine learning.* The University of Waikato,

Hall, M. A. (1999). Correlation-based feature selection for machine learning.

Hall, M. A., & Smith, L. A. (1998). Practical feature subset selection for machine learning.

Han, J., Pei, J., & Kamber, M. (2011). *Data mining: concepts and techniques*: Elsevier.

Harrak, F., Bouchet, F., & Luengo, V. (2019). *Categorizing students' questions using an ensemble hybrid approach.* Paper presented at the Educational Data Mining.

Hazarika, D., Poria, S., Gorantla, S., Cambria, E., Zimmermann, R., & Mihalcea, R. (2018). Cascade: Contextual sarcasm detection in online discussion forums. *arXiv preprint arXiv:1805.06413*.

He, Q., Li, X., Kim, D. N., Jia, X., Gu, X., Zhen, X., & Zhou, L. J. I. F. (2020). Feasibility study of a multi-criteria decision-making based hierarchical model for multi-modality feature and multi-classifier fusion: Applications in medical prognosis prediction. *55*, 207-219.

He, X., & Xu, S. (2010). *Process neural networks: Theory and applications*: Springer Science & Business Media.

Ho, T. K., Hull, J. J., Srihari, S. N. J. I. t. o. p. a., & intelligence, m. (1994). Decision combination in multiple classifier systems. *16*(1), 66-75.

Hsu, C.-W., Chang, C.-C., & Lin, C.-J. (2003). A practical guide to support vector classification.

Jain, D., Kumar, A., & Garg, G. (2020). Sarcasm detection in mash-up language using soft-attention based bi-directional LSTM and feature-rich CNN. *Applied Soft Computing*, 106198.

Jain, T., Agrawal, N., Goyal, G., & Aggrawal, N. (2017). *Sarcasm detection of tweets: A comparative study.* Paper presented at the 2017 Tenth International Conference on Contemporary Computing (IC3).

Japkowicz, N. (2000). *The class imbalance problem: Significance and strategies.* Paper presented at the Proc. of the Int'l Conf. on Artificial Intelligence.

Java, A., Song, X., Finin, T., & Tseng, B. (2007). *Why we twitter: understanding microblogging usage and communities.* Paper presented at the Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 workshop on Web mining and social network analysis.

Jia, X., Deng, Z., Min, F., & Liu, D. (2019). Three-way decisions based feature fusion for Chinese irony detection. *International Journal of Approximate Reasoning, 113*, 324-335.

Jo, T. (2013). *Application of table based similarity to classification of bio-medical documents.* Paper presented at the 2013 IEEE International Conference on Granular Computing (GrC).

Joshi, A., Agrawal, S., Bhattacharyya, P., & Carman, M. J. (2017). *Expect the unexpected: Harnessing sentence completion for sarcasm detection.* Paper presented at the International Conference of the Pacific Association for Computational Linguistics.

Joshi, A., Tripathi, V., Patel, K., Bhattacharyya, P., & Carman, M. (2016). Are Word Embedding-based Features Useful for Sarcasm Detection? *arXiv preprint arXiv:1610.00883*.

Kantardzic, M. (2011). *Data mining: concepts, models, methods, and algorithms*: John Wiley & Sons.

Kapil, P., & Ekbal, A. J. a. p. a. (2021). Leveraging Multi-domain, Heterogeneous Data using Deep Multitask Learning for Hate Speech Detection.

Karuna, Y., & Reddy, G. R. (2020). Broadband subspace decomposition of convoluted speech data using polynomial EVD algorithms. *Multimedia Tools and Applications, 79*(7), 5281-5299.

Katyayan, P., & Joshi, N. (2019). Sarcasm Detection Approaches for English Language. In *Smart Techniques for a Smarter Planet* (pp. 167-183): Springer.

Khattri, A., Joshi, A., Bhattacharyya, P., & Carman, M. (2015). *Your sentiment precedes you: Using an author's historical tweets to predict sarcasm.* Paper presented at the Proceedings of the 6th workshop on computational approaches to subjectivity, sentiment and social media analysis.

Khodak, M., Saunshi, N., & Vodrahalli, K. (2017). A large self-annotated corpus for sarcasm. *arXiv preprint arXiv:1704.05579*.

Kittler, J., Hatef, M., Duin, R. P., Matas, J. J. I. t. o. p. a., & intelligence, m. (1998). On combining classifiers. *20*(3), 226-239.

Korde, V., & Mahender, C. N. (2012). Text classification and classifiers: A survey. *International Journal of Artificial Intelligence & Applications, 3*(2), 85.

Kumar, A., & Garg, G. (2019). Empirical study of shallow and deep learning models for sarcasm detection using context in benchmark datasets. *Journal of Ambient Intelligence and Humanized Computing*, 1-16.

Kumar, A., Sangwan, S. R., Arora, A., Nayyar, A., & Abdel-Basset, M. (2019). Sarcasm Detection Using Soft Attention-Based Bidirectional Long Short-Term Memory Model With Convolution Network. *IEEE Access, 7*, 23319-23328.

Kumar, H. K., & Harish, B. (2018). Sarcasm classification: A novel approach by using Content Based Feature Selection Method. *Procedia computer science, 143*, 378-386.

Kumar, R., & Kaur, J. (2020). Random Forest-Based Sarcastic Tweet Classification Using Multiple Feature Collection. In *Multimedia Big Data Computing for IoT Applications* (pp. 131-160): Springer.

Kumar, S., Atreja, S., Singh, A., & Jain, M. (2019). *Adversarial adaptation of scene graph models for understanding civic issues.* Paper presented at the The World Wide Web Conference.

Kuncheva, L. I., Bezdek, J. C., & Duin, R. P. J. P. r. (2001). Decision templates for multiple classifier fusion: an experimental comparison. *34*(2), 299-314.

Kuncheva, L. I., & Whitaker, C. J. J. M. l. (2003). Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. *51*(2), 181-207.

Kunneman, F., Liebrecht, C., Van Mulken, M., & Van den Bosch, A. (2015). Signaling sarcasm: From hyperbole to hashtag. *Information Processing & Management, 51*(4), 500-509.

Kwak, H., Lee, C., Park, H., & Moon, S. (2010). *What is Twitter, a social network or a news media?* Paper presented at the Proceedings of the 19th international conference on World wide web.

Lauren, P., Qu, G., Zhang, F., & Lendasse, A. (2018). Discriminant document embeddings with an extreme learning machine for classifying clinical narratives. *Neurocomputing, 277*, 129-138.

Lee, H.-S., Lee, H.-R., Park, J.-U., & Han, Y.-S. (2018). An abusive text detection system based on enhanced abusive and non-abusive word lists. *Decision Support Systems, 113*, 22-31. doi:10.1016/j.dss.2018.06.009

Lemmens, J., Burtenshaw, B., Lotfi, E., Markov, I., & Daelemans, W. (2020). *Sarcasm detection using an ensemble approach.* Paper presented at the Proceedings of the Second Workshop on Figurative Language Processing.

Li, J., & Rao, H. R. (2010). Twitter as a rapid response news service: An exploration in the context of the 2008 China earthquake. *The Electronic Journal of Information Systems in Developing Countries, 42*(1), 1-22.

Li, J., Tian, Y., Zhu, Y., Zhou, T., Li, J., Ding, K., & Li, J. J. A. i. i. m. (2020). A multicenter random forest model for effective prognosis prediction in collaborative clinical research network. *103*, 101814.

Liaw, A., & Wiener, M. (2002). Classification and regression by randomForest. *R news, 2*(3), 18-22.

Libbrecht, M. W., & Noble, W. S. (2015). Machine learning applications in genetics and genomics. *Nature Reviews Genetics, 16*(6), 321-332.

Liebrecht, C., Kunneman, F., & van Den Bosch, A. (2013). The perfect solution for detecting sarcasm in tweets# not.

Ling, J., & Klinger, R. (2016). *An empirical, quantitative analysis of the differences between sarcasm and irony.* Paper presented at the European Semantic Web Conference.

Liu, B. (2012). Sentiment analysis and opinion mining. *Synthesis lectures on human language technologies, 5*(1), 1-167.

Liu, L., & Özsu, M. T. (2009). *Encyclopedia of database systems* (Vol. 6): Springer New York, NY, USA:.

Liu, P., Chen, W., Ou, G., Wang, T., Yang, D., & Lei, K. (2014). Sarcasm Detection in Social Media Based on Imbalanced Classification. In *Web-Age Information Management* (pp. 459-471).

Liu, T., Moore, A., Gray, A., & Yang, K. (2004). An investigation of practical approximate nearest neighbor algorithms. pages 825--832. In: MIT Press.

Lytvyn, V., Vysotska, V., Rusyn, B., Pohreliuk, L., Berezin, P., & Naum, O. (2019). *Textual Content Categorizing Technology Development Based on Ontology.* Paper presented at the MoMLeT.

Macià, N., Bernadó-Mansilla, E., Orriols-Puig, A., & Ho, T. K. (2013). Learner excellence biased by data set selection: A case for data characterisation and artificial data sets. *Pattern Recognition, 46*(3), 1054-1066.

Malave, N., & Dhage, S. N. (2020). Sarcasm detection on twitter: user behavior approach. In *Intelligent Systems, Technologies and Applications* (pp. 65-76): Springer.

Mangai, U. G., Samanta, S., Das, S., & Chowdhury, P. R. J. I. T. r. (2010). A survey of decision fusion and feature fusion strategies for pattern classification. *27*(4), 293-307.

Manjusha, P., & Raseek, C. (2018). *Convolutional Neural Network Based Simile Classification System.* Paper presented at the 2018 International Conference on Emerging Trends and Innovations In Engineering And Technological Research (ICETIETR).

Manohar, M. Y., & Kulkarni, P. (2017). *Improvement sarcasm analysis using NLP and corpus based approach.* Paper presented at the 2017 International Conference on Intelligent Computing and Control Systems (ICICCS).

McCallum, A., & Nigam, K. (1998). *A comparison of event models for naive bayes text classification.* Paper presented at the AAAI-98 workshop on learning for text categorization.

Mehndiratta, P., Sachdeva, S., & Soni, D. (2017). Detection of Sarcasm in Text Data using Deep Convolutional Neural Networks. *Scalable Computing: Practice and Experience, 18*(3), 219-228.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). *Distributed representations of words and phrases and their compositionality.* Paper presented at the Advances in neural information processing systems.

Mitrakis, N. E., Theocharis, J. B., Petridis, V. J. F. S., & Systems. (2008). A multilayered neuro-fuzzy classifier with self-organizing properties. *159*(23), 3132-3159.

Mohri, M., Rostamizadeh, A., & Talwalkar, A. (2012). *Foundations of machine learning*: MIT press.

Mujtaba, G., Shuib, L., Idris, N., Hoo, W. L., Raj, R. G., Khowaja, K., . . . Nweke, H. F. (2018). Clinical text classification research trends: Systematic literature review and open issues. *Expert Systems with Applications*.

Mujtaba, G., Shuib, L., Raj, R. G., Majeed, N., & Al-Garadi, M. A. (2017). Email classification research trends: Review and open issues. *IEEE Access, 5*, 9044-9064.

Mukherjee, S., & Bala, P. K. (2017a). Detecting sarcasm in customer tweets: an NLP based approach. *Industrial Management & Data Systems, 117*(6), 1109-1126.

Mukherjee, S., & Bala, P. K. (2017b). Sarcasm detection in microblogs using Naïve Bayes and fuzzy clustering. *Technology in Society, 48*, 19-27. doi:10.1016/j.techsoc.2016.10.003

Muresan, S., Gonzalez-Ibanez, R., Ghosh, D., & Wacholder, N. (2016). Identification of nonliteral language in social media: A case study on sarcasm. *Journal of the Association for Information Science and Technology, 67*(11), 2725-2737.

Nayel, H., Amer, E., Allam, A., & Abdallah, H. (2021). *Machine learning-based model for sentiment and sarcasm detection.* Paper presented at the Proceedings of the Sixth Arabic Natural Language Processing Workshop.

Nezhad, Z. B., Deihimi, M. A. J. J. o. I., & Technology, C. (2021). Sarcasm detection in Persian. *20*(1), 1-20.

Nigam, K., McCallum, A. K., Thrun, S., & Mitchell, T. (2000). Text classification from labeled and unlabeled documents using EM. *Machine learning, 39*(2-3), 103-134.

Nithya, K., Kalaivaani, P. D., & Thangarajan, R. (2012). *An enhanced data mining model for text classification.* Paper presented at the Computing, Communication and Applications (ICCCA), 2012 International Conference on.

Nusinovici, S., Tham, Y. C., Yan, M. Y. C., Ting, D. S. W., Li, J., Sabanayagam, C., . . . Cheng, C.-Y. J. J. o. c. e. (2020). Logistic regression was as good as machine learning for predicting major chronic diseases. *122*, 56-69.

Nweke, H. F., Teh, Y. W., Al-Garadi, M. A., & Alo, U. R. (2018). Deep learning algorithms for human activity recognition using mobile and wearable sensor networks: State of the art and research challenges. *Expert Systems with Applications, 105*, 233-261.

Nweke, H. F., Teh, Y. W., Mujtaba, G., & Al-Garadi, M. A. (2019a). Data fusion and multiple classifier systems for human activity detection and health monitoring: Review and open research directions. *Information Fusion, 46*, 147-170.

Nweke, H. F., Teh, Y. W., Mujtaba, G., & Al-Garadi, M. A. J. I. F. (2019b). Data fusion and multiple classifier systems for human activity detection and health monitoring: Review and open research directions. *46*, 147-170.

Onan, A. (2017). *Sarcasm identification on twitter: a machine learning approach.* Paper presented at the Computer Science On-line Conference.

Onan, A. (2019). *Topic-enriched word embeddings for sarcasm identification.* Paper presented at the Computer Science On-line Conference.

Onan, A., & Toçoğlu, M. A. J. I. A. (2021). A Term Weighted Neural Language Model and Stacked Bidirectional LSTM Based Framework for Sarcasm Identification. *9*, 7701-7722.

Pan, Y., Zhang, L., Wu, X., & Skibniewski, M. J. J. I. F. (2020). Multi-classifier information fusion in risk analysis. *60*, 121-136.

Pennebaker, J. W., Boyd, R. L., Jordan, K., & Blackburn, K. (2015). *The development and psychometric properties of LIWC2015*. Retrieved from

Pennington, J., Socher, R., & Manning, C. (2014). *Glove: Global vectors for word representation.* Paper presented at the Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP).

Pisner, D. A., & Schnyer, D. M. (2020). Support vector machine. In *Machine Learning* (pp. 101-121): Elsevier.

Potamias, R. A., Siolas, G., & Stafylopatis, A.-G. (2020). A transformer-based approach to irony and sarcasm detection. *Neural Computing and Applications*, 1-12.

Prasad, A. G., Sanjana, S., Bhat, S. M., & Harish, B. (2017). *Sentiment analysis for sarcasm detection on streaming short text data.* Paper presented at the 2017 2nd International Conference on Knowledge Engineering and Applications (ICKEA).

Preoţiuc-Pietro, D., Eichstaedt, J., Park, G., Sap, M., Smith, L., Tobolsky, V., . . . Ungar, L. (2015). *The role of personality, age, and gender in tweeting about mental illness.* Paper presented at the Proceedings of the 2nd workshop on computational linguistics and clinical psychology: From linguistic signal to clinical reality.

Provost, F. J., & Fawcett, T. (1997). *Analysis and visualization of classifier performance: Comparison under imprecise class and cost distributions.* Paper presented at the KDD.

Provost, F. J., Fawcett, T., & Kohavi, R. (1998). *The case against accuracy estimation for comparing induction algorithms.* Paper presented at the ICML.

Ptáček, T., Habernal, I., & Hong, J. (2014). *Sarcasm detection on czech and english twitter.* Paper presented at the Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers.

Qabajeh, I., & Thabtah, F. (2014). *An Experimental Study for Assessing Email Classification Attributes Using Feature Selection Methods.* Paper presented at the Advanced Computer Science Applications and Technologies (ACSAT), 2014 3rd International Conference on.

Quinlan, J. R. (1990). Decision trees and decision-making. *IEEE Transactions on Systems, Man, and Cybernetics, 20*(2), 339-346.

Rajadesingan, A., Zafarani, R., & Liu, H. (2015a). *Sarcasm Detection on Twitter*. Paper presented at the Proceedings of the Eighth ACM International Conference on Web Search and Data Mining - WSDM '15.

Rajadesingan, A., Zafarani, R., & Liu, H. (2015b). *Sarcasm detection on twitter: A behavioral modeling approach.* Paper presented at the Proceedings of the Eighth ACM International Conference on Web Search and Data Mining.

Ramos, J. (2003). *Using tf-idf to determine word relevance in document queries.* Paper presented at the Proceedings of the first instructional conference on machine learning.

Ranjan, P., Yadav, J., & Saha, S. (2017). Proposed Approach for Sarcasm Detection in Twitter. *Indian Journal of Science and Technology, 10*(25), 1-8. doi:10.17485/ijst/2017/v10i25/114443

Rennie, J. D., Shih, L., Teevan, J., & Karger, D. R. (2003). *Tackling the poor assumptions of naive bayes text classifiers.* Paper presented at the Proceedings of the 20th international conference on machine learning (ICML-03).

Reyes, A., Rosso, P., & Buscaldi, D. (2012). From humor recognition to irony detection: The figurative language of social media. *Data & Knowledge Engineering, 74*, 1-12.

Reyes, A., Rosso, P., & Veale, T. (2013). A multidimensional approach for detecting irony in twitter. *Language resources and evaluation, 47*(1), 239-268.

Riloff, E., Qadir, A., Surve, P., De Silva, L., Gilbert, N., & Huang, R. (2013). *Sarcasm as contrast between a positive sentiment and negative situation.* Paper presented at the Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing.

Rizzo, F., Caracoglia, L. J. C., & Structures. (2020). Artificial Neural Network model to predict the flutter velocity of suspension bridges. *233*, 106236.

Saha, S., Yadav, J., & Ranjan, P. (2017). Proposed approach for sarcasm detection in twitter. *Indian Journal of Science and Technology, 10*(25).

Sahami, M., Dumais, S., Heckerman, D., & Horvitz, E. (1998). *A Bayesian approach to filtering junk e-mail.* Paper presented at the Learning for Text Categorization: Papers from the 1998 workshop.

Salehinejad, H., Sankar, S., Barfett, J., Colak, E., & Valaee, S. (2017). Recent advances in recurrent neural networks. *arXiv preprint arXiv:1801.01078*.

Salton, G., & Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *Information Processing & Management, 24*(5), 513-523.

Salton, G., & McGill, M. J. (1986). Introduction to modern information retrieval.

Samanta, S., & Das, S. (2009). *A fast supervised method of feature ranking and selection for pattern classification.* Paper presented at the International Conference on Pattern Recognition and Machine Intelligence.

Samonte, M. J. C., Dollete, C. J. T., Capanas, P. M. M., Flores, M. L. C., & Soriano, C. B. (2018). *Sentence-Level Sarcasm Detection in English and Filipino Tweets*. Paper presented at the Proceedings of the 4th International Conference on Industrial and Business Engineering - ICIBE' 18. http://delivery.acm.org/10.1145/3290000/3288172/p181-Samonte.pdf?ip=103.18.0.19&id=3288172&acc=ACTIVE%20SERVICE&key=69AF3716A20387ED%2EE7759EC8BE158239%2E4D4702B0C3E38B35%2E4D4702B0C3E38B35&__acm__=1562041412_216ad611ed7438dea30eb1738af6b7df

Sangwan, S., Akhtar, M. S., Behera, P., & Ekbal, A. (2020). *I didn't mean what I wrote! Exploring Multimodality for Sarcasm Detection.* Paper presented at the 2020 International Joint Conference on Neural Networks (IJCNN).

Sarker, A., & Gonzalez, G. (2015). Portable automatic text classification for adverse drug reaction detection via multi-corpus training. *Journal of biomedical informatics, 53*, 196-207.

Sarsam, S. M., Al-Samarraie, H., Alzahrani, A. I., & Wright, B. J. I. J. o. M. R. (2020). Sarcasm detection using machine learning algorithms in Twitter: A systematic review. *62*(5), 578-598.

Schifanella, R., de Juan, P., Tetreault, J., & Cao, L. (2016). *Detecting sarcasm in multimodal social platforms.* Paper presented at the Proceedings of the 2016 ACM on Multimedia Conference.

Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM Computing Surveys (CSUR), 34*(1), 1-47.

Shrivastava, M., & Kumar, S. J. T. i. S. (2021). A pragmatic and intelligent model for sarcasm detection in social media text. *64*, 101489.

Sintsova, V., & Pu, P. (2016). Dystemo. *ACM Transactions on Intelligent Systems and Technology, 8*(1), 1-22. doi:10.1145/2912147

Sonawane, S. S., & Kolhe, S. R. J. P. C. S. (2020). TCSD: Term Co-occurrence Based Sarcasm Detection from Twitter Trends. *167*, 830-839.

Sreelakshmi, K., & Rafeeque, P. (2018). *An Effective Approach for Detection of Sarcasm in Tweets.* Paper presented at the 2018 International CET Conference on Control, Communication, and Computing (IC4).

Strapparava, C., & Valitutti, A. (2004). *Wordnet affect: an affective extension of wordnet.* Paper presented at the Lrec.

Subramanian, J., Sridharan, V., Shu, K., & Liu, H. (2019). *Exploiting emojis for sarcasm detection.* Paper presented at the International Conference on Social Computing,

Behavioral-Cultural Modeling and Prediction and Behavior Representation in Modeling and Simulation.

Suhaimin, M. S. M., Hijazi, M. H. A., Alfred, R., & Coenen, F. (2017). *Natural language processing based features for sarcasm detection: An investigation using bilingual social media texts.* Paper presented at the 2017 8th International Conference on Information Technology (ICIT).

Suhaimin, M. S. M., Hijazi, M. H. A., Alfred, R., & Coenen, F. (2018). Mechanism for Sarcasm Detection and Classification in Malay Social Media. *Advanced Science Letters, 24*(2), 1388-1392.

Suhaimin, M. S. M., Hijazi, M. H. A., Alfred, R., & Coenen, F. (2019). Modified framework for sarcasm detection and classification in sentiment analysis. *Indonesian Journal of Electrical Engineering and Computer Science, 13*(3), 1175-1183.

Sulis, E., Farías, D. I. H., Rosso, P., Patti, V., & Ruffo, G. (2016). Figurative messages and affect in Twitter: Differences between# irony,# sarcasm and# not. *Knowledge-Based Systems, 108*, 132-143.

Sundararaj, V., Rejeesh, M. J. J. o. R., & Services, C. (2021). A detailed behavioral analysis on consumer and customer changing behavior with respect to social networking sites. *58*, 102190.

Sundararajan, K., Saravana, J. V., & Palanisamy, A. (2020). Textual Feature Ensemble-Based Sarcasm Detection in Twitter Data. In *Intelligence in Big Data Technologies—Beyond the Hype* (pp. 443-450): Springer.

Tang, L., & Liu, H. (2005). *Bias analysis in text classification for highly skewed data.* Paper presented at the Fifth IEEE International Conference on Data Mining (ICDM'05).

Tao, Q., & Veldhuis, R. J. P. R. (2009). Threshold-optimized decision-level fusion and its application to biometrics. *42*(5), 823-836.

Tay, Y., Tuan, L. A., Hui, S. C., & Su, J. (2018). Reasoning with sarcasm by reading in-between. *arXiv preprint arXiv:1805.02856*.

Thelwall, M., Buckley, K., & Paltoglou, G. (2012). Sentiment strength detection for the social web. *Journal of the American Society for information Science and Technology, 63*(1), 163-173.

Tsur, O., Davidov, D., & Rappoport, A. (2010). *ICWSM—a great catchy name: Semi-supervised recognition of sarcastic sentences in online product reviews.* Paper presented at the Fourth International AAAI Conference on Weblogs and Social Media.

Tsur, O., & Rappoport, A. (2012). *What's in a hashtag?: content based prediction of the spread of ideas in microblogging communities.* Paper presented at the Proceedings of the fifth ACM international conference on Web search and data mining.

Udani, G. J. B. T. (2012). An exhaustive study of twitter users across the world.[Online] Available http://www. beevolve. com/twitter-statistics.

van der Aalst, W. M. (2001). Exterminating the dynamic change bug: A concrete approach to support workflow change. *Information Systems Frontiers, 3*(3), 297-317.

Vanschoren, J., Blockeel, H., Pfahringer, B., & Holmes, G. (2012). Experiment databases. *Machine Learning, 87*(2), 127-158.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., . . . Polosukhin, I. J. a. p. a. (2017). Attention is all you need.

Vyas, V., & Uma, V. (2019). Approaches to sentiment analysis on product reviews. In *Sentiment Analysis and Knowledge Discovery in Contemporary Business* (pp. 15-30): IGI Global.

Walker, M. A., Tree, J. E. F., Anand, P., Abbott, R., & King, J. (2012). *A Corpus for Research on Deliberation and Debate.* Paper presented at the LREC.

Wallace, B. C. (2015). Computational irony: A survey and new perspectives. *Artificial intelligence review, 43*(4), 467-483.

Wallace, B. C., Kertz, L., & Charniak, E. (2014). *Humans require context to infer ironic intent (so computers probably do, too).* Paper presented at the Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers).

Wang, G., Sun, J., Ma, J., Xu, K., & Gu, J. (2014). Sentiment classification: The contribution of ensemble learning. *Decision Support Systems, 57*, 77-93.

Wang, Y., Coiera, E., Runciman, W., & Magrabi, F. (2017). Using multiclass classification to automate the identification of patient safety incident reports by type and severity. *BMC medical informatics and decision making, 17*(1), 84.

Wang, Z.-W., Wang, S.-K., Wan, B.-T., & Song, W. W. J. I. J. o. D. S. N. (2020). A novel multi-label classification algorithm based on K-nearest neighbor and random walk. *16*(3), 1550147720911892.

Wang, Z., Wu, Z., Wang, R., & Ren, Y. (2015). *Twitter sarcasm detection exploiting a context-based model.* Paper presented at the International Conference on Web Information Systems Engineering.

Wehner, M. R., Chren, M.-M., Shive, M. L., Resneck, J. S., Pagoto, S., Seidenberg, A. B., & Linos, E. (2014). Twitter: an opportunity for public health campaigns. *The Lancet, 384*(9938), 131-132.

Wolpert, D. H., & Macready, W. G. (1995). *No free lunch theorems for search*. Retrieved from

Xiao, S., & Tong, W. (2021). *Prediction of User Consumption Behavior Data Based on the Combined Model of TF-IDF and Logistic Regression.* Paper presented at the Journal of Physics: Conference Series.

Xiao, Z., Li, X., Wang, L., Yang, Q., Du, J., & Sangaiah, A. K. (2018). Using convolution control block for Chinese sentiment analysis. *Journal of Parallel and Distributed Computing, 116*, 18-26.

Xu, G., Meng, Y., Qiu, X., Yu, Z., & Wu, X. J. I. A. (2019). Sentiment analysis of comment texts based on BiLSTM. *7*, 51522-51532.

Yang, J., Yang, J.-y., Zhang, D., & Lu, J.-f. J. P. r. (2003). Feature fusion: parallel strategy vs. serial strategy. *36*(6), 1369-1381.

Yang, M.-S. (1993). A survey of fuzzy clustering. *Mathematical and Computer modelling, 18*(11), 1-16.

Yang, Y. (1999). An evaluation of statistical approaches to text categorization. *Information retrieval, 1*(1-2), 69-90.

Yang, Y., & Pedersen, J. O. (1997). *A comparative study on feature selection in text categorization.* Paper presented at the Icml.

Yao, X. (1999). Evolving artificial neural networks. *Proceedings of the IEEE, 87*(9), 1423-1447.

Yavanoglu, U., Ibisoglu, T. Y., & Wıcana, S. G. (2018). Technical Review: Sarcasm Detection Algorithms. *International Journal of Semantic Computing, 12*(03), 457-478.

Yee Liau, B., & Pei Tan, P. (2014). Gaining customer knowledge in low cost airlines through text mining. *Industrial Management & Data Systems, 114*(9), 1344-1359.

Zen, H., Agiomyrgiannakis, Y., Egberts, N., Henderson, F., & Szczepaniak, P. (2016). Fast, compact, and high quality LSTM-RNN based statistical parametric speech synthesizers for mobile devices. *arXiv preprint arXiv:1606.06061*.

Zhang, L., Qian, G., Fan, W., Hua, K., & Zhang, L. (2014). Sentiment analysis based on light reviews. *Ruan Jian Xue Bao/Journal of Software, 25*(12), 2790-2807.

Zhang, M., Zhang, Y., & Fu, G. (2016). *Tweet sarcasm detection using deep neural network.* Paper presented at the Proceedings of COLING 2016, The 26th International Conference on Computational Linguistics: Technical Papers.

Zhang, P., Zhu, X., Tan, J., & Guo, L. (2010). *Classifier and cluster ensembles for mining concept drifting data streams.* Paper presented at the 2010 IEEE International Conference on Data Mining.

Zhang, Y., Tiwari, P., Song, D., Mao, X., Wang, P., Li, X., & Pandey, H. M. J. N. N. (2021). Learning interaction dynamics with an interactive LSTM for conversational sentiment analysis. *133*, 40-56.

Zheng, Z., Wu, X., & Srihari, R. (2004). Feature selection for text categorization on imbalanced data. *ACM Sigkdd Explorations Newsletter, 6*(1), 80-89.