Web Mining for Social Network Analysis

By

Mohamed Kamel Abdelsalam Elhaddad
B.Sc., Military Technical College, Egypt, 2009
M.Sc., Military Technical College, Egypt, 2017

A Dissertation Submitted in Partial Fulfillment of the

Requirements for the degree of

DOCTOR OF PHILOSOPHY

In the Department of Electrical and Computer Engineering

University of Victoria

We acknowledge with respect the Lekwungen peoples on whose traditional territory the university stands and the Songhees, Esquimalt, and WSÁNEĆ peoples whose historical relationships with the land continue to this day.

Web Mining for Social Network Analysis

By

Mohamed Kamel Abdelsalam Elhaddad
B.Sc., Military Technical College, Egypt, 2009
M.Sc., Military Technical College, Egypt, 2017

Supervisory Committee

Prof. Kin Fun Li, Supervisor

(Department of Electrical and Computer Engineering)

Prof. Fayez Gebali, Co-Supervisor

(Department of Electrical and Computer Engineering)

Prof. Alex Thomo, Outside Member

(Department of Computer Science)

Abstract

Undoubtedly, the rapid development of information systems and the widespread use of electronic means and social networks have played a significant role in accelerating the pace of events worldwide, such as, in the 2012 Gaza conflict (the 8-day war), in the pro-secessionist rebellion in the 2013-2014 conflict in Eastern Ukraine, in the 2016 US Presidential elections, and in conjunction with the COVID-19 outbreak pandemic since the beginning of 2020. As the number of daily shared data grows quickly on various social networking platforms in different languages, techniques to carry out automatic classification of this huge amount of data timely and correctly are needed.

Of the many social networking platforms, Twitter is of the most used ones by netizens. It allows its users to communicate, share their opinions, and express their emotions (sentiments) in the form of short blogs easily at no cost. Moreover, unlike other social networking platforms, Twitter allows research institutions to access its public and historical data, upon request and under control. Therefore, many organizations, at different levels (e.g., governmental, commercial), are seeking to benefit from the analysis and classification of the shared tweets to serve in many application domains, for examples, sentiment analysis to evaluate and determine user's polarity from the content of their shared text, and misleading information detection to ensure the legitimacy and the credibility of the shared information. To attain this objective, one can apply numerous data representation, preprocessing, natural language processing techniques, and machine/deep learning algorithms. There are several challenges and limitations with existing approaches, including issues with the management of tweets in multiple languages, the determination of what features the feature vector should include, and the assignment of representative and descriptive weights to these features for different mining tasks. Besides, there are limitations in existing performance evaluation metrics to fully assess the developed classification systems.

In this dissertation, two novel frameworks are introduced; the first is to efficiently analyze and classify bilingual (Arabic and English) textual content of social networks, while the second is for evaluating the performance of binary classification algorithms. The first framework is designed with: (1) An approach to handle Arabic and English written tweets, and can be extended to cover data written in more languages and from other social networking platforms, (2) An effective data preparation and preprocessing techniques, (3) A novel feature selection technique that allows utilizing different types of features (content-dependent, context-dependent, and domain-dependent), in addition to (4) A novel feature extraction technique to assign weights to the linguistic features based on how representative they are in in the classes they belong to. The proposed framework is employed in performing sentiment analysis and misleading information detection. The performance of this framework is compared to state-of-the-art classification approaches utilizing 11 benchmark datasets comprising both Arabic and English textual content, demonstrating considerable

improvement over all other performance evaluation metrics. Then, this framework is utilized in a real-life case study to detect misleading information surrounding the spread of COVID-19.

In the second framework, a new multidimensional classification assessment score (MCAS) is introduced. MCAS can determine how good the classification algorithm is when dealing with binary classification problems. It takes into consideration the effect of misclassification errors on the probability of correct detection of instances from both classes. Moreover, it should be valid regardless of the size of the dataset and whether the dataset has a balanced or unbalanced distribution of its instances over the classes. An empirical and practical analysis is conducted on both synthetic and real-life datasets to compare the comportment of the proposed metric against those commonly used. The analysis reveals that the new measure can distinguish the performance of different classification techniques. Furthermore, it allows performing a class-based assessment of classification algorithms, to assess the ability of the classification algorithm when dealing with data from each class separately. This is useful if one of the classifying instances from one class is more important than instances from the other class, such as in COVID-19 testing where the detection of positive patients is much more important than negative ones.

Table of Contents

List of Figures

List of Tables

List of Publications

The following manuscripts have been published:

1. M.K. Elhadad, K.F. Li, and F. Gebali, "Sentiment Analysis of Arabic and English Tweets," In: *Barolli L., Takizawa M., Xhafa F., Enokido T. (eds) Web, Artificial Intelligence and Network Applications. WAINA 2019. Advances in Intelligent Systems and Computing*, vol 927, pp. 334-348, Mar. 2019, Springer, Cham, doi: https://doi.org/10.1007/978-3-030-15035-8_32
2. M.K. Elhadad, K.F. Li, and F. Gebali, "Fake News Detection on Social Media: A Systematic Survey," In: Proceedings of the *2019 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM)*, Victoria, BC, Canada, Aug. 2019, pp. 1-8, doi: https://doi.org/10.1109/PACRIM47961.2019.8985062.
3. M. K. Elhadad, K. F. Li, and F. Gebali, "A Novel Approach for Selecting Hybrid Features from Online News Textual Metadata for Fake News Detection," In: *Barolli L., Hellinckx P., Natwichai J. (eds) Advances on P2P, Parallel, Grid, Cloud and Internet Computing. 3PGCIC 2019. Lecture Notes in Networks and Systems*, vol 96, pp. 914-925, Oct. 2020, Springer, Cham, doi: https://doi.org/10.1007/978-3-030-33509-0_86.
4. M. K. Elhadad, K. F. Li, and F. Gebali, (2020), "Detecting Misleading Information on COVID-19," In: *IEEE Access*, vol. 8, pp. 165201-165215, Sep. 2020, IEEE, doi: https://doi.org/10.1109/ACCESS.2020.3022867.
5. M. K. Elhadad, K. F. Li, and F. Gebali, "COVID-19-FAKES: A Twitter (Arabic/English) Dataset for Detecting Misleading Information on COVID-19," In: *Barolli L., Li K., Miwa H. (eds) Advances in Intelligent Networking and Collaborative Systems. INCoS 2020. Advances in Intelligent Systems and Computing*, vol 1263, pp. 256-268, Aug. 2020, Springer, Cham, doi: https://doi.org/10.1007/978-3-030-57796-4_25.
6. M. K. Elhadad, K. F. Li, and F. Gebali, "An Ensemble Deep Learning Technique to Detect COVID-19 Misleading Information," In: Barolli L., Li K., Enokido T., Takizawa M. (eds) Advances in Networked-Based Information Systems. NBiS 2020. Advances in Intelligent Systems and Computing, vol 1264, pp. 163-175, Aug. 2020, Springer, Cham, doi: https://doi.org/10.1007/978-3-030-57811-4_16.

The following journal manuscripts are under preparation:

1. M. K. Elhadad, K. F. Li, and F. Gebali, "A Novel Feature Extraction Technique for Text Mining: A Case Study on Detecting Misleading Information on Social Media".

2. M. K. Elhadad, K. F. Li, and F. Gebali, "A Novel Metric for Evaluating the Performance of Supervised Binary Machine Learning Algorithms".

Acronyms

ACC	Accuracy
ANN	Artificial Neural Network
API	Application Programming Interface
AUC	Area Under the Curve
BNB	Bernoulli Naïve Bayes
BoVW	Bag of Visual Words
BoW	Bag of Words
CNN	Convolutional Neural Network
CSV	Comma Separated Values
DSS	Decision Support Systems
DT	Decision Trees
EDA	Exploratory Data Analysis
EHRs	Electronic Health Records
ERF	Extremely Random Forests
ERR	Error Rate
eWOM	Electronic Word of Mouth
F1	F1-Score
GIS	Geographical Information Systems
HTML	Hypertext Markup Language
ICSDF	Inverse Class Space Density Frequency
IDA	Initial Data Analysis
IDF	Inverse Document Frequency
IG	Information Gain
JSON	JavaScript Object Notation
kNN	K-Nearest Neighbor
LR	Logistic Regression
LSTM	Long Short-Term Memory
LSVM	Linear Support Vector Machines
MENA	Middle East and North Africa Region
MID	Misleading Information Detection
ML	Machine Learning
MNB	Multinomial Naïve Bayes
MSE	Mean Square Error
MT	Machine Translation
MTF	Modified Term Frequency
MTF-IDF	Modified Term Frequency-Inverse Document Frequency
MTF-MIDF	Modified Term Frequency-Modified Inverse Document Frequency
NB	Naïve Bayes
NLP	Natural Language Processing
NLTK	Natural Language Toolkit
NN	Neural Network
P	Precision
PoS	Part-of-Speech
PN	Proper Nouns
PTM	Probabilistic Topic Model
R	Recall

RCF	Relevance Category Frequency
RFFS	Relevance Frequency Feature Selection
RNTN	Recursive Neural Tensor Networks
SA	Sentiment Analysis
SN	Sensitivity
SNA	Social network analysis
SP	Specificity
SQL	Structured Query Language
SQRT-TF-IGMimp	Square Root of Term Frequency-Inverse Gravity Moment
SLM	Statistical Language Model
STWS	Supervised Term Weighting Scheme
SVM	Support Vector Machines
TDG	Twitter Data Grants
TF	Term Frequency
TF-IGMimp	Term Frequency-Inverse Gravity Moment
TF-MIDF	Term Frequency-Modified Inverse Document Frequency
TF-RFST	Term Frequency-Ranking of Fuzzy Logic with the Semantic Relationship of Terms
TF-RTF	Term Frequency-Ranking of Term Frequency
TPR	True Positive Rate
URL	Uniform Resource Locator
VoIP	Voice Over IP
VSM	Vector Space Model
WLTF	Weighted Logarithmic Term Frequency
word2vec	Word-to-Vector
WWW	World Wide Web
XGBoost	eXtreme Gradient Boosting
XML	Extensible Markup Language

Acknowledgments

First of all, I would like to express my deep gratitude, thanks, and prayers to the *Mighty Allah,* the most gracious and the most merciful, who blessed us with the ability to think, taught us everything, and gave me the power to finish this work.

I would also like to thank my *beloved wife Nayera*, my lovely daughter, *Karma*, and my dear sons, *Adam* and *Zeyad,* for their sacrifices, patience, love, moral support, and continuous encouragement during all the stages of developing this work and in my entire life.

There are no words that could express my gratitude, appreciation, love, and respect to my parents, *Kamel* and *Hadir*, for their sacrifices, encouragement, guidance, and for every second they spend and invest in my upbringing and education. Many thanks to my brothers, *Mahmoud* and *Wael*, for their continuous encouragement, support, love, and for being real brothers.

Finally, yet importantly, it is my pleasure to express my sincere thanks to my supervisors for their leadership. First and foremost, I am deeply indebted to my mentor, *Prof./ Kin Fun Li*, for his time, effort, advice, and for fueling my motivation and enthusiasm during this research. I also would like to express my gratitude to *Prof./ Fayez Gebali*, for his time, effort, useful notes, and his continuous encouragement and support. They offered not only guidance and leadership, but they were able to hold discussions on a scientific basis, besides their enlightened vision and rational approach, they were able to elicit and guide my research, sometimes gently, and sharply at other times.

Also, I honorably would like to express my deep appreciation to *Prof./ Alex Thomo*, for his continuous support and valuable advice from which I have learned a lot.

Last but not least, I wish to express my deepest gratitude towards the government of Egypt for funding my Ph.D. research, all my leaders and colleagues in the Computer Engineering and Artificial Intelligence Department at the Military Technical Collage for their support, and my friends for being by my side when needed.

Mohamed Kamel Elhaddad
Victoria, B.C., Canada, July 2021

Dedication

To my beloved wife, you are crucial for my success story.

To my lovely kids, Adam, Karma, and Zeyad, you are the sheer joy and happiness of my life.

To my parents, without you, I would not have had any success.

To the souls of the martyrs, who sacrificed their lives for the sake of defending motherland, Egypt, throughout the ages.

Chapter 1

Introduction

1.1 Overview and Motivation

Through the years the World Wide Web (WWW) structure and architecture have been in continuous growth and expansion. With the vast development of information technology and the widespread use of the Internet, social network platforms, also known as social media, arise [1]. These social network platforms (*Facebook, Twitter, Snapchat, YouTube*, etc.) are web-based and mobile-based Internet applications that allow their users to create, access, exchange, and share different content (views, feelings, experiences, advice, news, etc.) easily and at no cost, making the entire world connected [2].

Although the contents of social network platforms are accessible through Application Programming Interface (API)s, for its commercial value, many social network platforms such as Facebook, LinkedIn and Skype are making it increasingly difficult for researchers to obtain full access to their 'raw' data [2]. In contrast, in 2014 Twitter introduced a project called Twitter Data Grants (TDG), which allows research institutions to access Twitter's public and historical data. This access permits researchers to perform different tasks on its massive set of data [3], e.g., analyzing sentiments, detecting botnets, detecting misleading information, etc. Twitter is considered the most popular and commonly used social network platform. On Twitter, users can easily communicate with each other or share emotions, stories, concerns, and it provides better means to get quick responses and feedback on different global issues in the form of short blogs of at most 280 words in length [4].

Practically speaking, the latest statistics of Twitter, in Q1-2021, shows that Twitter has almost 23 percent of Internet users on it, with 353 million monthly active users who published around 6,000 tweets every second with a total of more than 500 million daily posted tweets [5]. This large number of users with such a huge amount of posted content on Twitter is increasing at an astronomical pace. This fact results in the exponential growth of the available contents and makes it a challenge to be managed, especially in a real-time environment, with existing techniques. These contents could be formed from any type of data; it could be textual data, multimedia, or a mixture of them [6], as shown in Figure 1.1.

In recent years, most countries were suffering from economic problems, political issues, wars, terrorism, and violent conflicts. The shared data, Electronic Word of Mouth (eWOM) statements, on these social network platforms could be harmful to some while they could be helpful to others. Without having any control over the consumed data, and with no standard or measure to analyze and determine the truthfulness, validity, and credibility of it, a proliferation of much misleading

information is anticipated. Such misleading information may be systematic and follow up on agendas to serve specific goals, objectives, and interests for a faction or the interests of certain countries or institutions. As a result, it may influence the public opinion of users of these social network platforms by making some posts, that are usually, about interesting topics, but quite often, with fabricated headlines with some misleading content.

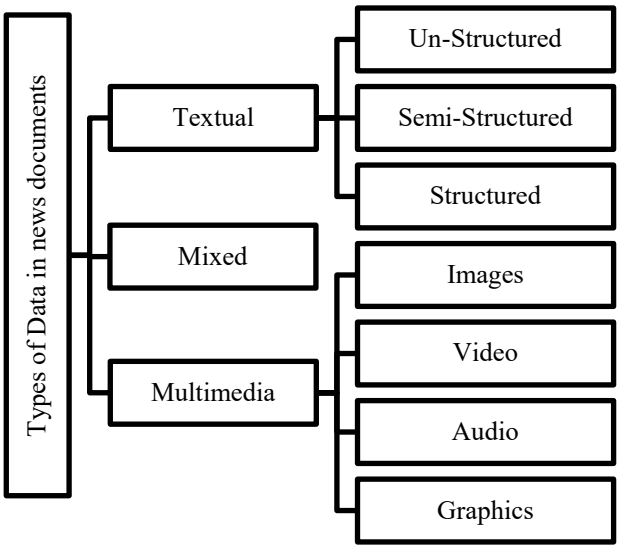


Figure 1.1 Types of Data in Social Network Platforms.

The main goal of these fabricated headlines is to attract viewers, making these headlines become a trending item [7]. Additionally, the spread of such misleading information could affect the operation of the financial markets, the response during critical situations, and terrorist attacks, etc. [8], causing financial or political damage to public figures, agencies, or even countries. Furthermore, they could be employed in information warfare between countries, companies, and institutions. For example, social networks have played active roles in accelerating the pace of events in countries all over the world, such as, in the 2012 Gaza conflict (the 8-day war) [9], in the pro-secessionist rebellion in the 2013-2014 conflict in Eastern Ukraine [10], in the 2016 US Presidential elections [11], and in conjunction with the outbreak of the COVID-19 pandemic since the beginning of 2020 [12].

In the same context, the Middle East and North Africa region (MENA) has been an extremely fertile ground for misleading information, but it has undeniably hit new heights since the COVID-19 pandemic. While the world struggles to deal with the COVID-19 global health pandemic, it seems that the dissemination of distorted and inaccurate facts in the MENA region is outpacing the disease's real spread. This has escalated and intensified the information wars that were already raging in the region, in an unparalleled onslaught of information. Since almost all the countries of the MENA region are Arabic speaking countries that use the English language as their second language, and the shared data could probably be a combination of both Arabic and English languages, and since each

country speaks different Arabic/English dialects [13] [14], a need for studying the daily jotted data on social network platforms in both languages in different dialects arises.

Hence, a growing interest in the research community arises to benefit from the available social network platforms and perform analysis in many aspects, using quantitative approaches, qualitative approaches, or a combination of both. Researchers are devoting a great effort to develop automated techniques to be deployed in managing, analyzing, and classifying the textual content of social network platforms. Due to the intertwining of most of the available data on social networks, the existing techniques are mostly relying on the manual analysis and interpretation of data. This manual analysis, also known as thematic analysis, involves at least two expert analysts who are intimately familiar with the data domain and serve as an interface between data and end-users [15]. These manual approaches are slow, labor-intensive, highly subjective, biased, and impractical due to the huge volumes of data on social network platforms [16].

The process of the automatic analysis and categorization of the consumed data on social network platforms, or part of it, represents an exciting and productive area of study. It could be done from many perspectives and for many reasons and applications depending on the collected data and the required task(s). Machine learning (ML) techniques are the most common approaches to perform the automated analysis and categorization of social network content [17]. This categorization is done based on comparing a given data with some pre-known corpora that contain different categories of data on the domain of interest and the training datasets [18].

One of the main challenges in building machine learning models is the lack of clarity, purity, and comprehensiveness of the available training data. Additionally, these data are rare, noisy, and sometimes multilingual. Therefore, they must be transformed, through a series of operations, so that they are prepared, processed, and transmuted into an appropriate format for different machine learning techniques. These operations are not straightforward but require special design and implementation depending on the type of data that we are dealing with [19].

For representing the content of textual news documents, different data representation methods could be used, such as Word-to-Vector (word2vec), Vector Space Model (VSM), Probabilistic Topic Model (PTM), and Statistical Language Model (SLM). Among these methods, the VSM is the most used one [20]. With VSM, textual documents are transformed into vectors in an n-dimensional space, where n is the number of unique terms in the dataset, i.e., converting the textual news documents into a compact form of their content with a high number of terms (features) that describe the documents. Most often, the size of these obtained features exceeds the number of the transformed documents that are used in training the detection model. For example, we could have a dataset that contains 1,000 textual documents, and the extracted non-redundant features that build up the feature vector could be 10,000 or more depending on the data we have.

Hence, there is an urgent need to devise an efficient, automatic, and adaptive framework to collect online data, perform analysis, and categorize bilingual textual content of social network platforms.

1.2 Problem Statement

The task of automatic classification of bilingual textual content in social network platforms, as one of web mining tasks, has become an interesting area for research. It is the key method for managing the huge amount of daily jotted textual data in multiple languages on social network platforms. For this task to be accomplished, textual contents are processed and transformed from the full-text version to a document vector by mapping each document into a compact form of its content, which makes handling them much easier and reduces their complexity.

A general statement of the problem can be formulated as follows:

"*Given a set of social network textual posts that are written in both Arabic and/or English, it is required to accurately indicate the appropriate class of each of them depending on a predefined corpus on a specific application domain (opinion mining, news analytics, etc.)*"

The process of indicating the appropriate class (classification process) for bilingual text documents face some challenges, mainly:

1- Limitations related to the data scraping process using the available social media sites' APIs for collecting real-time data to be used in building real-time systems.
2- The shortage of the existing studies and the available datasets cover the process of classifying bilingual textual data.
3- The lack of clarity, purity, and comprehensiveness of the available training bilingual datasets.
4- Limitations related to storing the scraped data as it is against most of the social network platforms' terms of service.
5- Problems with the automatic handling of the contents in both Arabic and English languages when present within the same document.
6- The extremely high dimensionality of text data, as the number of potential features (resulted from feature selection) often exceeds the number of training documents.
7- Limitations in the current feature extraction techniques used in building feature vectors for the categorization process.
8- The huge time required to perform feature engineering, build classification models, and obtain the classification results, especially when having new data instances for training.
9- Limitations with the existing performance evaluation techniques that are deployed to fully assess the performance of developed systems.

1.3 Dissertation Goals and Contributions

Based on the problem and the challenges discussed in the previous section, the followings are a set of objectives that we aim to achieve in this dissertation.

1- Develop a social network classification framework, for performing different text mining tasks, with the following capabilities:
   a. Handling textual data in both Arabic and English languages, without employing any language-translation services, while avoiding delays and limitations of the available machine translation APIs. In Addition, being able to be extended to cover data in different languages other than (Arabic/English), such as French and Spanish, and to be scalable to handle a huge amount of data.
   b. Selecting a set of the available meta-data features, accompanied with social network posts, besides the text-linguistic features, to enrich the feature vector with more information to enhance the classification results.
   c. Overcoming the problems of using the traditional TF-IDF-based techniques, by utilizing the novel feature extraction technique in weighting features in the feature vector.
   d. Performing data structuring and storing of the collected data, in practice, or the used datasets, in the implementation phases, in a relational database. This helps in saving the features obtained from the training process, in a structured way, to be used later when adding new instances to the training data. By only updating the values that correspond to the newly added or changed features in the feature vector, instead of retraining the model again with all the training data. It allows the benefit of being much faster, more robust, flexible, durable, reliable, and scalable by:
      i. Easing the process of handling concurrent connections or any sort of required data manipulation.
      ii. Keeping huge amounts of data structured and secured.
      iii. Ensuring the ACID (Atomicity, Consistency, Isolation, and Durability) compliance to guarantee that a database operation is completed in a timely manner.
      iv. Utilizing the benefits of the indexes in tables to ease and speed up the retrieval and data loading process.
      v. Easing the process of adding or updating weights of the used features stored in the database, unlike what would happen if the data was stored in traditional text files in different formats or even in JavaScript Object Notation (JSON) formats, where we have to have some process that opens

each Text/JSON file, adds the field, then saves it, as compared to the ease of using the ALTER TABLE Structured Query Language (SQL) data manipulation statement with a DEFAULT value for the field in databases.

2- Propose a new performance evaluation framework to overcome the limitations of existing performance evaluation frameworks, to be able to accurately assess the performance of different algorithms and find out the best one to use, based on the interest of the system's end-user.
3- As this work was under progress, COVID-19 showed up as a pandemic that critically threatens human life, with many confirmed cases that exceed 164 million and many confirmed deaths that exceed 3.4 million around the world and demands all humankind to stand together to fight against. Therefore, it becomes the normal choice to use COVID-19 as a case study, to show the capabilities of the proposed classification framework. Thus, the proposed framework is utilized to address the scarcity of the available bilingual (Arabic/English) datasets by building a multipurpose, bilingual, and multidialectal social media dataset for the detection of misleading information related to the COVID-19 outbreak.

Next, to illustrate its capabilities, the developed system is utilized to achieve the previously mentioned goals. The contributions of this dissertation are summarized as follows.

1- Introduce a generic categorization of the terms used in describing the shared misleading information on social network platforms into Disinformation, Misinformation, and Malinformation. This helps in better understanding and formalizing the problems that are related to the classification of misleading information and hence developing different efficient misleading information detection (MID) systems.
2- Introduce a data preparation technique that can be employed to handle bilingual (Arabic/English) data without using any of the existing machine translation techniques.
3- Introduce a novel technique for handling textual content of social network platforms and selecting hybrid features to enhance the overall classification results when applied in building fake news detection systems as a case study.
4- Introduce a novel feature extraction technique, to overcome the existing limitations of the traditional TF-IDF-based techniques, that is capable of giving more representative and discriminative weights to features in the feature vector.
5- Perform a critical review on different performance evaluation metrics and introduce a novel multidimensional, multifaceted performance evaluation metric that is valid regardless of the size of the used datasets even with skewed class distributions. Introduce

a new evaluation framework procedure to fully assess the performance of binary classification algorithms.

6- Introduce a bilingual (Arabic/English) Twitter dataset, for misleading information detection with the following features:
   a. It is unique in being to date and containing real data that covers hot topics.
   b. It is very huge compared with the largest currently available benchmark datasets and the annotation process of these datasets is automated, which means the labeling processes are not biased.
   c. An Exploratory Data Analysis (EDA) is performed, and baseline classification results are obtained using different machine/deep learning classifiers when deploying different feature extraction techniques, allowing this dataset to serve in optimizing different web mining and social network analysis tasks.
   d. It is publicly available for the research community to use in evaluating their produced models (https://github.com/mohaddad/COVID-FAKES). Also, it is available for researchers to perform a re-annotation process to label the data to serve in different application domains such as hate-speech detection, botnet detection, crisis management, authorship verification, etc.

1.4 Dissertation Methodology

This study aims to propose and implement a framework for processing, analyzing, and performing different text mining tasks on social network textual content written in both Arabic and English languages. To fulfill this ultimate goal, the following methodology has been followed.

1- Study state-of-the-art techniques and algorithms used for the analysis of textual content of social network platforms in different languages, to highlight the strength and discover the limitations and drawbacks of the currently used techniques.
2- Design and implement a technique for handling textual data in social media sites that has a mix of textual data written in both Arabic and English languages simultaneously, not based on traditional data translation-based techniques. Then, these techniques are used for building different social network analysis models for different application domains such as sentiment analysis and misleading information detection.
3- Study state-of-the-art terminologies, techniques, and algorithms used for handling textual data on social network platforms, in addition to the used datasets to build misleading information detection systems. Moreover, give a general categorization of the terminologies used in describing the consumed misleading information in social media sites.

4- Design and implement feature selection techniques to select features from textual news metadata to enhance the obtained detection results from fake news detection systems. This is done by handling the news document as one cleaned segment: by taking the union of all its segments. Moreover, select a hybrid set of features from both the news' textual content and its accompanying metadata.
5- Design and implement different feature extraction techniques to be used in weighting features in the feature vectors for building the fake news detection models.
6- Design and implement an effective performance evaluation metric to fully assess the performance of the introduced systems.
7- Scrape both Twitter, as one of the social network platforms, and the web to collect bilingual data and employ the proposed frameworks to automatically annotate it. In addition to making this data available for researchers to utilize in designing, building, and optimizing their social network analysis models. The dataset has the characteristics of being a multipurpose, multidialectal, bilingual, and automatically annotated social media benchmark dataset that contributes to covering the shortage of the available bilingual (Arabic/English) datasets.
8- Design and build an adaptive text mining framework that can automatically manage social media textual data in more than one language with multi-dialect at the same time to perform different social media analytical tasks.
9- Validate the proposed framework using a set of benchmark datasets and report the validation results when using different feature extraction techniques (TF and TF-IDF with character level, unigram, bigram, trigram, and n-gram word size, and word embedding) and various classification algorithms (DT, MNB, BNB, LR, kNN, Perceptron, NN, LSVM, ERF, and XGBoost).

1.5 Dissertation Outline

This dissertation is organized into six chapters, which are as follows. In Chapter 1, a general overview and motivation are discussed, the problem statement is also stated, the dissertation goals are indicated, and finally, the dissertation outline is shown. Chapter 2 delves into the background of web mining and social networks, as well as the work that has been done in this area. Chapter 3 introduces and addresses the proposed frameworks for text classification and for assessing the performance of classification systems. Chapter 4 presents the obtained results when applying the proposed frameworks on a set of benchmark datasets and discusses these findings. Deploying the proposed frameworks to a real-life case study is discussed in depth in Chapter 5. Finally, Chapter 6 contains conclusions and recommendations for future work.

Chapter 2

Background and Literature Survey

Section 2.1 of this chapter introduces a comprehensive background on web mining, social networks, and various social network analysis tasks. Then, in *Section 2.2,* some related work to the research issues covered in this dissertation is discussed. Applications relating to both sentiment analysis and misleading information detection are also discussed, in which different web mining techniques are employed to perform different analysis tasks on the textual content of social network platforms. Related existing work covers the handling of bilingual textual, preprocessing, feature selection and feature extraction techniques, datasets used for building and evaluating social network analysis, misleading information detection, sentiment analysis models, and evaluation metrics to assess the performance of the developed binary classification systems.

2.1 Background

2.1.1 Web Mining

Web mining deals with the automatic extraction and discovery of useful knowledge from diverse and heterogeneous web records, including web documents, hyperlinks between documents, usage logs of websites, etc., by employing a combination of techniques related to databases, statistics, machine learning, etc. [21] [22].

2.1.1.1 Web Mining Processes

Generally, to perform any mining tasks on social network data, it should pass through six important steps, as shown in Figure 2.1 [23].

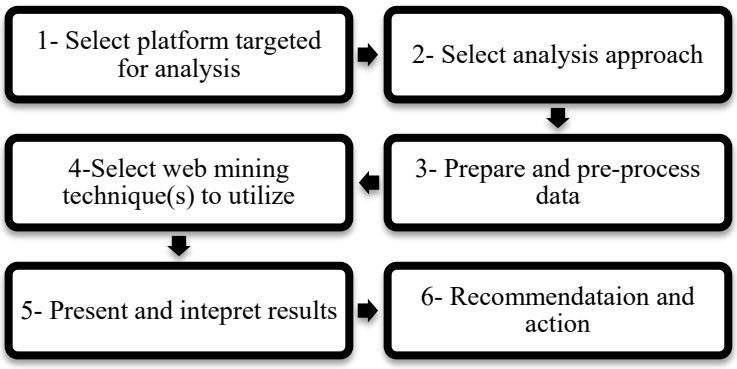


Figure 2.1 Web Mining Processes for Social Network Analysis [23].

The first step is the selection of the analysis target, that is, the social network source of data. Depending on the analysis goal(s), data can be retrieved from either one source or multiple ones.

Then comes the selection of what kind of social network analysis approach to follow, as discussed later in *Subsection 2.1.3*. Once we have the data and decided which analysis approach to follow, data is prepared and transformed into a suitable format for these tasks. In the data preparation and preprocessing step, the collected data, depending on the type, is cleaned, and transformed into a suitable format for different web mining techniques. Then, the next step is selecting and proceeding with the appropriate web mining technique, sometimes the collaboration of different types of web mining techniques is necessary depending on the required analysis task. After that, the selected techniques are applied to perform the analysis of the data that has been collected and prepared in the previous steps. Web mining categories and application domains are discussed in *Subsubsection 2.1.1.2,* and *Subsubsection 2.1.1.3*.

After performing the web mining task, the obtained results are analyzed, interpreted, and visualized either manually or automatically. This last step is optional, as the analysis processes may be completed after the analysis results have been generated. The recommendation and action step deals with the analysis results generated in the previous step. For example, if it is discovered that the collected news data is fake or one user usually posted fake news, some actions could be performed such as preventing this user from posting data. It could be remarked that, in some cases, modifications of these processes are necessary to suit the requirements of different social network analysis applications depending on the required analysis task.

2.1.1.2 Web Mining Categories

Web Mining is categorized into three main areas of interest based on which part of the web to mine: web content mining, web structure mining, and web usage mining [24] as shown in Figure 2.2.

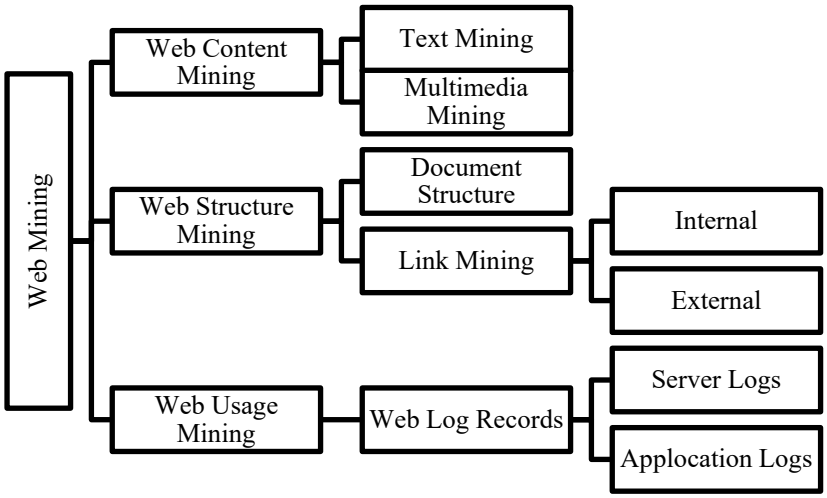


Figure 2.2 Web Mining Categories [24].

a) ***Web Content Mining*** is the application of data mining that aims at the discovery of useful information from the web contents; these contents could be in structured, semi-structured, or unstructured data formats [25]. Generally, web content mining is categorized into:

(1) *Text mining*, which focused on discovering patterns and trends from textual data. Text mining usually involves the process of some natural language processing techniques such as text parsing, along with some linguistic analysis and data cleaning processes [26].

(2) *Multimedia mining*, which focused on pattern discovery, rule, and knowledge extraction from multimedia data (images, video, audio, graphics/animations) [21].

Web content mining has many applications; it is used in categorizing and clustering web documents, search web pages from several different servers, and show or hide the documents based on some ranking on their content [27].

b) ***Web Structure Mining*** aims in discovering structural information from the web; it generates a structural summary about web pages and extracts some interesting web graph patterns like co-citation, social choice, complete bipartite graphs, etc. The focus of web structure mining is on link information [28]. Web structure mining has many applications such as web page ranking and web page similarity detection [27].

c) ***Web Usage Mining (Web Log Mining)*** is the application of data mining that aims to discover meaningful patterns (usage patterns) that reflects the usages of web resources from data generated by client-server transactions to understand and better serve the needs of web-based applications [29]. The main goal of web usage mining is to make contributions in improving the overall quality of information systems, to support designers during the design process, and to ensure ease of use to end-users [30]. Web Usage Mining applications are mainly targeting digital marketing and the building of recommender systems as it helps in delivering advertisements to specific groups of users based on their behavior on different web pages [31].

2.1.1.3 Web Mining Application Domains

There are many applications in different domains that utilize web mining techniques to ease and improve our quality of life [32], such as Decision Support [33] [34] [35], Marketing Intelligence, E-Commerce, and E-Business [36] [37], Digital Libraries and Information Retrieval [38] [39], E-Learning [40] [41] [42], E-Government and politically related applications [43] [44] [45], Security, Criminal, Radical and Extremist Investigations [46] [47] [48], etc.

2.1.2 Social Networks

Social network platforms are web-based and mobile-based internet applications that allow their users to create, edit, access, exchange and share different content (views, feelings, experiences, advice, news, etc.) easily and at no cost. There are many categories of social network platforms including, but not limited to, social networking (*Facebook*, *LinkedIn*), microblogging (*Twitter*), photo sharing

(*Instagram*), video sharing (*YouTube*), social search (*Google.com, Bing.com*), and instant messaging (*WhatsApp, Facebook Messenger, WeChat, Skype*). The number of social network platforms has been growing dramatically since 1994 when the first social network website "*Yahoo! GeoCities*" appeared.

2.1.2.1 Types and Categories of Data

The data on social network platforms could be of different types, but generally, there exist two main types in which social network users exchange their information, as shown in Figure 1.1. The consumed data could contain only textual data, which could be in structured/unstructured/semi-structured formats, or multimedia data that could be audio/video/images/graphics, or it may have a combination of these two formats. The categories of the available data on social network platforms could be divided into either *Historic data* or *Real-time feeds* [49].

It should be noted that, while the material consumed on social network platforms is a mix of text and multimedia data, the whole idea behind every social media post cannot be comprehended except through understanding its textual content. In this dissertation, we focus on the classification of the shared textual content on social network platforms.

2.1.2.2 Formats of Textual Data

The available APIs used for data scraping from social network platforms extract the textual content in Hypertext Markup Language (HTML), XML, JSON, or Comma-separated values (CSV) file formats [2] [50].

In general, the social network data used for research are those from the sites that allow programmable access to their content using their released APIs. Although many social network platforms provide APIs, not all sites (e.g., Bing, LinkedIn, and Skype) grant access to their private data and they have restrictions on programmable access to their publicly available content. As previously discussed in *Subsection 2.1.1*, Twitter is considered the most commonly used social network platform, where users can easily communicate with each other or share emotions, stories, and concerns, and provide better means to get quick responses and feedback on different global issues in the form of short blogs [4] [51]. Moreover, in 2014, it introduced an API, which allows research institutions to access Twitter's public and historic data to get insights from its massive set of data [3]. Hence, we briefly discuss the APIs provided by Twitter in the following Subsubsection.

2.1.2.3 Twitter Data and its API

The default configurations of Twitter accounts keep all the posted tweets public, although any post owner has the authority to make them accessible only by their approved followers or by certain group members. However, more than 90% of all Twitter accounts are public [2] [52]. These publicly available tweets, including user information, retweets, replies, and mentions, are available in JSON

format through Twitter's provided API. Twitter API enables a pre-authorized access to both historical and real-time feeds. As for the historical data, the Twitter search API allows the query for tweets containing specific keywords up to one week prior. For the streaming API, it allows the filtering of live streaming tweets by many identifiers such as user ID, keyword, geographic location, or random sampling [3].

For developers to employ this API, they must fill an application form on Twitter, to register their research project information, giving full details about the purpose and the expected output of the project. Then Twitter reviews this application and decides whether to accept or reject the application. If Twitter accepts the submitted application, it assigns a unique application ID to the project and produces a set of credentials to allow developers to access the API. With this access, developers can establish connections to query Twitter's historical database, as well as accessing its feeds in real-time. Then, after creating the connection and running the query through the API, results in JSON format are retrieved.

2.1.3 Social Network Analysis

Social network analysis is an interdisciplinary field that aims to process the content of different social network platforms to effectively represent, analyze, and extract meaningful information from them in an automated way [53]. This analysis helps in having a better understanding of the social network platforms' content. Besides, social media mining provides the necessary tools to analyze the dissemination of social networks' information, study how the shared information influences users, provide effective recommendations for users, and analyze different social behaviors of either users or a group of them in social network platforms [54].

For dealing with the content of social network platforms, various web content mining, text mining, or natural language processing techniques are being used. For example, web content mining can be used to classify the documents of the social network platform, especially for blog content analysis to classify the articles of blogs or the consumed news and events [55]. Furthermore, web content mining can also be used to analyze users' behavior, reading interests, such as the favorite contents of users [49]. For building a social network analytical system, researchers are utilizing different tools such as R-Programming [56] and Apache Spark [57], and programming languages such as C#, Java, and Python [54].

2.1.3.1 Social Network Analysis Approaches

In general, for performing the social network analysis, researchers are following either quantitative or qualitative approaches or a combination of both [2] [58]. The quantitative approaches aim in obtaining information based on the frequencies of the contained features in the datasets used to perform one or more analysis tasks, such as volume analysis, relationship analysis, correlation

analysis, classification, clustering, etc. The qualitative approaches aim in performing tasks related to group identification and analysis, thematic analysis, sentiment analysis, graphical media analysis, etc. To establish a wider contextual analysis and to obtain a much more efficient analysis of the social network platforms, a combination of qualitative and quantitative approaches should be employed [59].

2.1.3.2 Social Network Analysis Challenges

Social network analysis faces many challenges such as collecting enough samples for different tasks, the difficulty of detection and removal of the noise of the obtained data, and the problems related to the analysis and performance evaluation. This produces different challenges and opportunities for research in the validity, accuracy, and reliability of the consumed data on social networks. Due to the variety of used languages with different dialects for data exchange over the social network and the medium's noisy nature, conventional handling, preprocessing, and analysis technologies are inadequate. When the data source is a social network platform, and due to the huge amount of daily jotted data on it, all challenges related to Big Data become even more salient, and ensuring the quality of the data including privacy-preserving are still open research issues.

As a result, a careful research design is required with clear research objectives and questions and the appropriate selection of analytical techniques/tools. If initial findings are statistically significant then they should be verified using at least one more distinct additional dataset which has been collected at a different time, using different methods, or on a different platform. If not, it may be necessary to reduce the scope of a study or reframe its central hypothesis to address a more specific aspect of human behavior on a given dataset from a particular social network platform.

As with all research, any analytical limitations and considerations should be placed alongside findings to ensure research findings are not inappropriately used. The notion that today's "Big" Data poses new challenges is widely acknowledged in various fields. The key factors by which this new phenomenon differs from traditional analytics can be summarized as follows:

a) Volume: the storage space required.
b) Velocity: the speed of data creation coupled with the advantage gained from analyzing the data in real-time.
c) Variety: the fact that data takes many different forms. It is often unstructured, or its structure is specific to the data source, and
d) Validity: uncertainty especially regarding data quality.

For example, when more physical space the data takes, the harder it is to fit into memory, the slower many algorithms run. Moreover, with the need of performing real-time management and analysis of data, the data architectural and structural choices are directly influenced. Other challenges are the privacy and the availability of data. Yet the data's lack of accuracy, representativeness, and

context is affected by the chosen data source and data extraction methods. These issues fall under the broader definition of veracity. It has even been debated and explored if social network analysis can replace traditional and more expensive ways of data collection. But it was also criticized that there is a lack of tested standard procedures for data collection and a danger of data-driven, non-theoretical approaches.

We could sum up social network analysis challenges in one statement as follows: The consumed data on social network platforms is huge, unstructured, multilingual, multidialectal, linked, dynamic, redundant, noisy, and of hybrid content, which needs special techniques and methods to handle, preprocess and perform different analysis tasks on it accurately with no delay.

2.2 Literature Survey

Typically, for building and optimizing classification systems, data must go through different steps, including data preparation, feature engineering, model building and deployment, and performance evaluation [55], as shown in Figure 2.3.

Based on the end-user requirements and the obtained evaluation results, a decision of which technique(s) to use is being taken by the developer. Accordingly, in some cases, to obtain the best performance from the system, an ensemble classification model (Voting Ensemble Model) is deployed. Figure 2.4 shows the detailed flowchart for the main steps for building classification systems that deploys a hard voting mechanism.

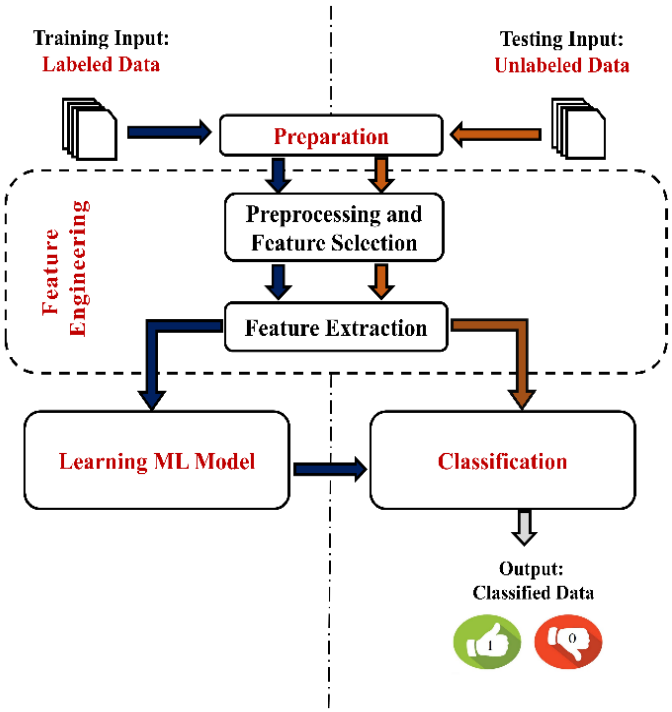


Figure 2.3 Block Diagram of a Typical Classification System.

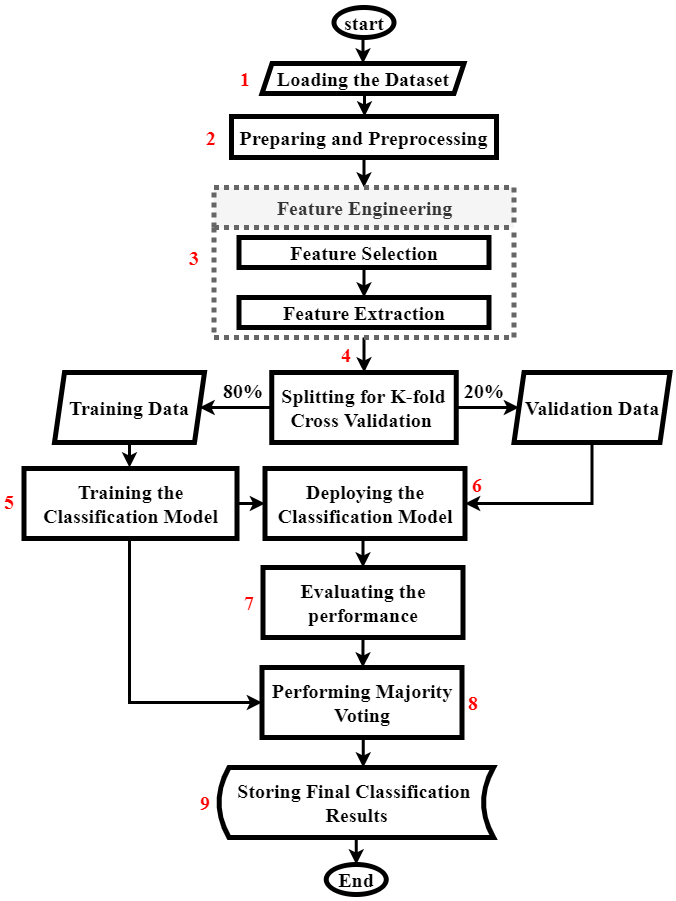


Figure 2.4 A Classification System Flowchart.

2.2.1 Data Collection

Datasets can be found in different formats whether they are unstructured, structured, or semi-structured, but approximately 80-90% of the available data is in an unstructured format [60]. Unstructured data is data that is not organized in a pre-defined structure. This data is usually dense in text and can come in various forms, including plain text, emails, social media posts, chats, IoT sensor data, and more [6]. This data may contain, in addition to text, dates, and numbers as well [61]. This makes it difficult to handle, especially using traditional programs, compared to data stored in structured datasets [62].

As for structured data, it is data that follows a predefined scheme, where this data conforms to a tabular format and is characterized by the presence of linking relationships not only between different rows and columns but also can be between data from different files [63]. There are many examples of representing structured data, among the most popular ones are Excel files and SQL databases. Each of them contains rows and columns organized in the form of tables that can be dealt

with and manipulated separately or in combination with data from other fields. This makes structured data very powerful with the ability to quickly merge data from different fields in the database [64].

As for semi-structured data (also known as self-describing structure [65]), it combines the characteristics of both types of data. They can be in the form of structured data, but they do not follow the formal structure of data models associated with relational databases or other forms of spreadsheets [66]. Semi-structured data contain tags or other markers to separate and distinguish semantic elements and to impose hierarchies of records and fields within the data. Common examples of semi-structured data formats include JSON and XML.

Depending on the application domain of interest and whether the application is targeting real-life/real-time data or targeting stored data, relevant datasets can be retrieved from different data repositories, including *kaggle.com* [67], *github.com* [68], *huggingface.co* [69], etc., or be scraped directly from online sources, such as social network platforms, websites, etc. Moreover, in some cases, the datasets can be loaded from in-memory, where they could be provided within tools or packages, such as WEKA [70], Python [71], R [72], MATLAB [73], etc.

2.2.1.1 Handling of Textual Data in Different Languages

Generally, from the approximately 7,000 languages spoken, only about 150 of them are used online [74] [75]. English is considered the most used language on the internet (25%) as shown in Figure 2.5 [76].

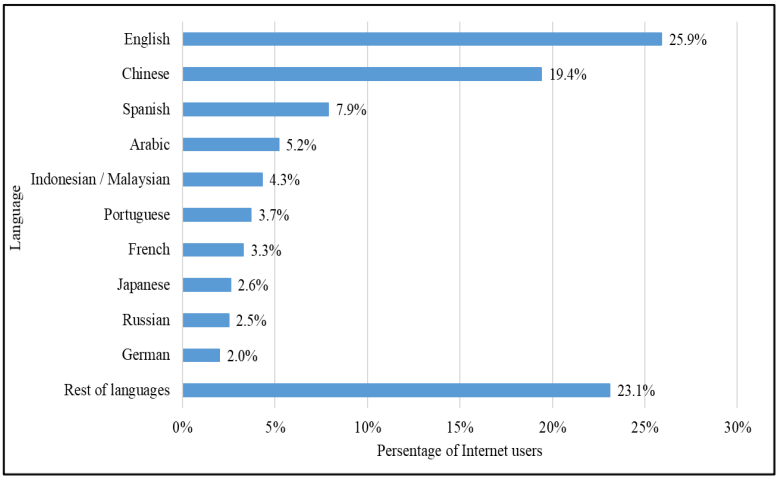


Figure 2.5 Most Common Languages Used on the Internet as of June 2020 [76].

Moreover, it is noteworthy that the Arabic language is spoken by more than 422 million people, and its speakers are distributed in the area known as the Arab World in addition to many other neighboring regions such as Ahvaz, Turkey, Chad, Senegal, Eritrea, and others. It is the fourth of the six official languages of the United Nations. This accentuates the need to develop techniques that could deal with textual content in different languages as clarified in *Section 1.1*.

The most common method to manage bilingual textual data is to translate it into a language that is supported by the used tools. There are three major approaches to obtain the translation textual content: Machine Translation, Knowledge-Based Methods using Machines Readable Dictionary, and Corpus-Based Methods using Parallel Corpus [77] [78].

In Machine Translation (MT), the text is translated, using translation services, into a single target language (most likely English), and then data mining systems are implemented in English [79]. Since social media posts are usually brief, grammatically invalid, and formulated in a few words, it provides little, or no, context for the correct translation of the Machine Translation scheme. Machine Translation most often substitutes original words with only one of their many possible synonymous translations in the target language [80], resulting in a mistranslation of Proper Nouns (PN) and expert terminology; consequently, a huge knowledge loss [81]. Moreover, the daily limits of using these translation services (e.g., Google Translate API has a default limit of 500,000 characters per month and to increase the limit to be between 500,000 to 1 billion characters per month it costs $20 per million characters [82]), making it impossible to translate the huge amount of social media posts that are needed to be processed. For English-speaking countries, Machine Translation can be considered the optimal solution, as there is only one official national language. However, when compared to other countries around the world (e.g., Canada, China, Spain, countries of the MENA region, etc.) it becomes clear that this solution is not their best one, as either there is no agreement on one common language, or they have more than one official language.

Multilingual text can be translated by substituting every word in the text for a list of all its potential interpretations as encoded in a Machine-Readable Dictionary [79]. However, this approach is ineffective mainly due to the translation ambiguity of polysemous terms (i.e., terms with multiple meanings), idioms, and acronyms [83]. These words might have several alternative translations carrying different meanings depending on both the source and the targeted translation languages. Translating text by including every possible translation of every word can greatly increase the set of possible meanings in the translated text, which by itself causes a great ambiguity, thus contributing to poor accuracy. In addition, the insufficient coverage of technical vocabulary and sentences is still a significant deficiency in such Machine-Readable Dictionary.

As an alternative to Machine-Readable Dictionary, a parallel corpus can be utilized. A parallel corpus is a collection of text pairs that share the same meaning but are written in different languages [84]. Corpus-Based text translation is based on building a multi-dimensional semantic space (Ontology) in which each term in the parallel corpus is represented as points, and all the equivalent translations are mapped to the same set of points that are used to describe a concept in that semantic space [79] [85]. Geometric relationships between terms within the semantic space are automatically inferred by analyzing co-occurrence statistics of terms across a parallel corpus. The translation process is achieved by substituting every term in text with its nearest geometric translation in the

semantic space, using various semantic similarity measures. The Corpus-Based approach is the most suitable when the translation process is on domain-specific data [86].

It is worth noting that, we did not find many publications that explicitly discuss the question of how to reduce the effort of multilingual system creation, but we did find some that explain the efforts of deploying machine translation techniques in translating a specific language to English. Steinberger [79] discussed how to minimize the effort in multiple language NLP implementations. Besides the core guidelines that support attempts to build complicated text mining applications for thousands of languages. Dashtipour *et al.* [87] presented a review on multilingual sentiment analysis showing an independent comparison of techniques and features used. Balahur *et al.* [88] dealt with the problem of sentiment detection in three different languages - French, German, and Spanish - using three distinct machine translation systems - Bing, Google, and Moses. They claim the superiority of their system, but not taking into consideration that the translation process is not accurate enough and it takes time. Baur [89] has developed a MarketMiner framework to improve the utilization of multi-source, multi-language social network platforms' content, which can be applied to areas such as social media analytics and business intelligence.

Vilares *et al.* [90] tackled the problem of performing multilingual classification of tweet's polarity for tweets written in both English and Spanish. They utilized the techniques introduced by Khaleghi *et al.* [91] and Balazs *et al.* [92] by fusing a set of pre-existing monolingual datasets to build their multilingual system. They compared the performance of three techniques (a multilingual model trained on their multilingual dataset, a dual monolingual model with perfect language detection on monolingual texts and, a monolingual model based on the decision provided by a language identification tool). Their results revealed that neither monolingual nor multilingual approaches based on language detection are optimal to deal with code-switching texts.

Salam *et al.* [93] developed a framework that works with news data in different languages, e.g., Spanish, Arabic, etc., using a universal dependency-based parser to handle data after performing the dictionary translation process. Ge *et al.* [94] introduced a distributed framework for multilevel-multilingual streaming analytics of social media data. They utilized different distributed open-source tools and deep learning architectures, deploying a cross-language and language translation to handle multilingual documents. Guibon *et al.* [95] worked on a dataset containing text from various websites in English and French, as well as automatic transcripts of YouTube French videos about vaccination which is a widely disseminated topic in false news data of each language. Their goal was to classify the textual content into Fake News, Trusted, or Satire. They handled the data from each language independently. Similarly, Adekotujo *et al.* [96] aimed to handle bilingual Twitter data to classify hate intent using the topical n-gram model. The preprocessing, feature engineering, and topic modeling steps are employed for each of the used datasets in different languages separately.

Our observation (backed up by the findings from several studies, e.g., [85], [97], [98], [99], and others) is that processing text in the source language (non-translated) is more accurate and reliable than the machine-translated ones. Moreover, we concluded that the main reason for the existence of a wide range of studies that deal with the English written data is the vast amount of available data and the huge number of English-speaking countries which make it easier to produce applications for them. Furthermore, producing applications that deal with other languages is suffering from the lack of available resources that hinder the development process. In other words, we can say that the required effort to build applications in N languages is N times the effort of developing applications for English written applications. In this dissertation, one of the goals is to deal with bilingual text within the same context, by detecting the language in which each part of the text is written with, without using languages detection services and APIs, and process it, in its source language, without being translated into English using any translation services or APIs.

2.2.2 Data Preparation and Preprocessing

Given that we are working with textual documents, the first challenge to solve is how to represent text and extract its features. A correct representation not only simplifies data manipulation and saves the time and memory required for processing such data, but also ensures that the relevant information is maintained without loss.

2.2.2.1 Data Preparation

Regardless of whether the used data is structured, unstructured, or semi-structured, data integrity and consistency are a must to keep in mind. The ultimate goal of preparing the used data is to ensure that the manipulation of it is correct and efficient. Data integrity requires a proper understanding of the format and the structure of the used data and is best conceived using established data governance practices and data management techniques. Depending on the data source and the application domain of interest, each instance of the used data must be put in a format that allows the automatic handling of it.

2.2.2.2 Preprocessing

The preprocessing step aims in easing the data manipulation of huge data [6]. This is done by utilizing some natural language processing (NLP) techniques, including text parsing, data cleaning, part of speech (PoS) tagging, stop words removal, and stemming. The use of these techniques limits the presence of noisy data, which is, in practice, the main cause of degradation in the performance of various classification algorithms [100].

2.2.2.3 Feature Engineering

Feature engineering is considered the most important task that affects the performance of any machine learning model [25], [101]. It enables the management of the huge amount of data that is

used in building different classification models by selecting and extracting the most representative feature vector [102].

In general, textual content on social networks are related to certain topics (e.g., sports, economics, politics, etc.), from a particular source (e.g., news websites, blogs, etc.), and have other metadata, such as location, publisher, author of news, publication date, etc. [103]. Accordingly, many feature selection and feature extraction techniques can be employed.

2.2.2.4 Feature Selection

Feature selection aims to speed up the used algorithms, improve learning accuracy, and enhance model comprehensibility, in addition, to find an optimal feature subset, for dimension reduction, by removing irrelevant and redundant information from the dataset [104]. It is also a knowledge discovery tool for providing insights into the problems through the interpretation of the most relevant features [105].

For this purpose, many feature selection techniques have been proposed. These techniques are either search-based or correlation-based [106]. For the search-based techniques, data is filtered, and a certain subset of features is obtained. For instance, in the case of dealing with tweets, many feature subsets could be obtained, including textual features (e.g., hashtags, user mentions, URLs, etc.), image and video-related features (e.g., image tags, video title, etc.), metadata-related features (e.g., geo-coordinates of a tweet, tweet creation time, language of a tweet, etc.), and network-related features (e.g., retweets count, reply count, number of followers, etc.) [107]. The correlation-based techniques are concerned with the correlation analysis between features (feature-feature correlation) or between features and classes (feature-class correlation), e.g., Principal Component Analysis (PCA), Chi-Square (CS), and Information Gain (IG) [60] [108].

Feature selection techniques are being applied to the available features in the textual documents [109]. Castillo *et al.* [110], Shu *et al.* [7], and Kursuncu *et al.* [111] gave a categorization for the types of features according to their perspectives. Depending on the application and type of data in the document, we believe the extracted features can be classified as content-based, context-based, or domain-based, as shown in Figure 2.6. The content-dependent features are meta-information that represent the raw content of a document including the author, editor, publisher, news title, the body, and any attached multimedia.

Therefore, depending on the content that is available in the data, different kinds of features can be selected. For instance, lexical and syntactic features, such as the number of Uniform Resource Locator (URLs), word length, hashtag, retweet, and term frequency, can be selected from the title and the body of the document. While from any attached multimedia, we could select visual and statistical features with information such as clarity score, coherence score, similarity distribution histogram, image ratio, multi-image ratio, etc. [112]. The author/editor/publisher of the document,

besides being a part of the news content, is also highly related to the news context. The context of the shared data is the social engagements of its consumption on the social media platform. This social engagement represents the data propagation over time and the group of users that engaged with it. Hence, we could extract from individuals, groups, and postings, social-based features such as number of followers, friends count, registration age, number of authored posts/tweets, related social groups, demographic information, user stance, average credibility scores, etc.

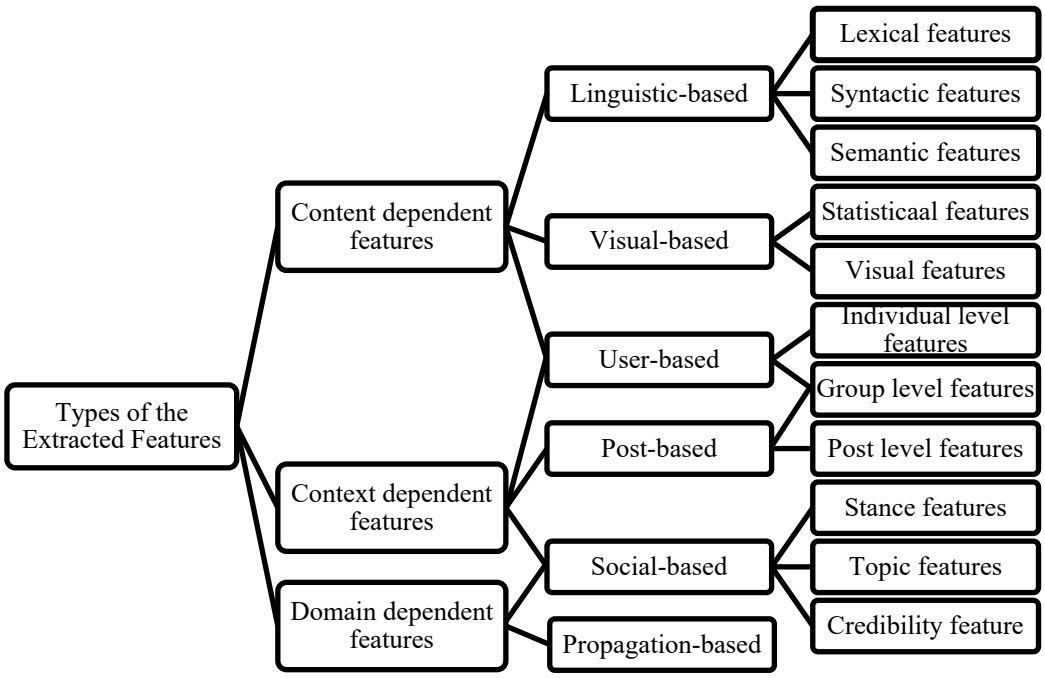


Figure 2.6. Types of Selected Features from Social Network Post.

Moreover, we could select features about the domain that the data belongs to, by selecting propagation features that consider characteristics related to the propagation tree, that can be built from the re-tweets/re-shares of data in a certain domain. These include features such as the depth of the re-tweet/re-share tree and the number of initial posts of a topic. Ruchansky *et al.* [113] introduced a model which incorporates the features related to news text, responses that the news document receives, and the user who source it. Potthast *et al.* [114] proposed a technique in which they extracted a set of features to capture writing style. These features are being used to assess the style similarity between different textual news document categories.

Khurana [115] explored the linguistic features that could be extracted using some NLP techniques from news statements and headlines and found that the use of n-grams as a feature, especially unigrams, plays a vital role in the discrimination of fake and real textual news documents. Ahmed *et al.* in [116] used n-gram analysis and machine learning techniques for detecting fake news in textual documents. They used both TF and TF-IDF as feature extraction techniques and selected

only the features ranging from the top 1,000 to 50,000. The best-obtained accuracy result was 92% on ISOT fake news dataset using LSVM, with unigram and the top 50,000 features. While in [117], the same group extended their experiments on another dataset related to the detection of fake reviews used in [118]. The best-obtained accuracy result was 90% using KNN, with bigram and the top 10,000 features.

Al Asaad *et al.* [119] introduced a technique for verifying the credibility of news articles depending on their characteristics. They combined several classification methods with text models. They used three text representation models (BoW, n-gram, and TF-IDF). They used in their experiments two different datasets and tested their model using Multinomial Naïve Bayes and LSVM algorithms. They obtained the best accuracy of 95.7% when using the Multinomial Naïve Bayes classifier with BoW as a text representation technique on the fake or real news dataset.

Khan *et al.* [120] presented a comparative analysis of the performance of existing methods by implementing each one on three different datasets. They observed that the performance of models is dependent on the dataset used and it is hard to obtain a unique superior model for all datasets. Moreover, from their experimental results, they found that the proper selection of features enhances the obtained accuracy. Also, they claimed that, for small datasets with less than 100k documents, Naïve Bayes (with n-gram) could achieve results similar to those obtained when using neural network-based models.

Bali *et al.* [121] proposed an approach for automatic detection of fake news. They extracted a set of features from both news headlines and news contents such as the n-grams count feature, sentiment polarity score, and some other linguistic features. They used three datasets to evaluate their model using seven different classification algorithms. They obtained the best accuracy when using Gradient Boosting (XGB) classification algorithm with three news datasets.

In summary, there are many works in the field of misleading information detection. To the best of our knowledge from existing literature, our work is the first to deal with documents without segmenting them. Additionally, we use a complex set of selected features from textual news metadata besides the linguistic features to enrich the generated feature vector with valuable features for building the information detection model.

2.2.2.5 Feature Extraction

Feature extraction aims to give representative weight to each feature in the used feature vector [122]. Many term weighting techniques can be used including Term Frequency (TF), Inverse Document Frequency (IDF), TF-IDF, Information Gain (IG), and Entropy, and others [123] [124]. In practice, more than 70% of the text-based classification systems use TF-IDF as a feature extraction technique [125]. Many researchers aim to introduce new term weighting techniques by either combining hybrid weighting techniques or modifying existing techniques to enhance classification performance.

a) ***Hybrid Techniques:*** Ren and Sohrab [126] introduced an automatic indexing approach (TF-IDF-ICSdF) that combines TF-IDF and Inverse Class Space Density Frequency (ICSdF) for extracting features from textual datasets. Sabbah *et al.* [127] introduced a hybrid term weighting technique that combines multiple feature extraction techniques (TF, DF, IDF, TF-IDF, Entropy, and Glasgow) into a single feature.

Elhadad *et al.* [60] introduced a technique to enrich the (TF-IDF)-based feature vector with some of the semantic features to enhance the overall classification performance. Wan *et al.* [128] proposed a text structure-based technique that aims in extracting composite features by measuring relevance category frequency (RCF).

Bhattacharjee *et al.* [129] proposed an approach for extracting features by reweighting the TF-IDF score, using the information of the classes that have fewer instances (minority class) in the training data. Bali *et al.* [121] extracted a set of features from both news title and news content, such as n-grams Count Feature, Sentiment Polarity Score, and some other linguistic features.

Dogan and Uysal [130] introduced two-term weighting techniques, Term Frequency-Inverse Gravity Moment (TF-IGMimp) and Square Root of Term Frequency-Inverse Gravity Moment (SQRT-TF-IGMimp), that combine the IGM and TF techniques. Moreover, they conducted an extensive analysis of the effect of using different term weighting techniques in enhancing the classification results in [131]. Alsmadi and Hoon [132] proposed an approach for building the feature vector based on a supervised term weighting scheme (SW) that combines the distribution of a term in the class and the whole dataset.

b) ***Modified Techniques:*** Sabbah *et al.* [133] proposed four weighting techniques, namely, Modified Term Frequency (mTF), Modified Term Frequency-Inverse Document Frequency (mTF-IDF), Term Frequency-Modified Inverse Document Frequency (TFm-IDF), and Modified Term Frequency-Modified Inverse Document Frequency (mTFm-IDF). These techniques consider the number of missing terms when calculating the weight of existing terms.

Chen [134] introduced a distance-based term weighting method to overcome the flaw of the traditional TF-IDF method that treats terms with lower TF as noise, by lowering its overall weight. This approach advocates that all documents should not contribute equally to the weighting of specific terms. Shaban *et al.* [11] proposed a weighted Logarithmic Term Frequency (wTF) technique. wTF is based on normalizing the count of each term by the average frequency of this term across all the days reported in the collected data.

Ghosh and Desarkar [135] proposed a modified TF-IDF score weighting technique (TF-IDFCNE) that incorporates class details by both entropy and term frequencies. Fan and Qin [136] introduced a modified TF-IDF weighting technique (TF-IDCRF) that considers the

relationships between classes to complete the text classification process. Supriyanto *et al.* [137] proposed a global weighting technique based on intra-/inter-class term distributions to be used on Bag-of-Visual Words (BoVW) based image classification.

Lakshmi and Baskar [138] proposed two-term weighting techniques: Term Frequency-Ranking of Term Frequency (TF-RTF) and Term Frequency-Ranking of Fuzzy logic with the Semantic relationship of Terms (TF-RFST). In these techniques, each term is weighted based on its frequency and the frequency of its semantically related terms.

To sum up, for each term in the used feature vector, a weight value must be assigned to indicate its importance. These weights are assigned according to a term weighting technique that assesses the relevance of each term within each data instance, regardless of the others or across the entire data. TF-IDF is the most used term weighting technique and throughout the literature, several studies have made modifications of them, as discussed earlier in this Section 2.2. The main limitation of these techniques is the assumption that the more frequent a term is throughout a document means the less discriminative they are. Although, when a term appears frequently within a particular class and rarely in others, this indicates that the term has high discriminatory power, and greatly assists in indicating that class.

2.2.3 Cross-validation

It is a statistical procedure used to estimate the classification ability of supervised learning models. This procedure has a single parameter, called *k,* that refers to the number of groups to which the dataset is split [139]. Practically, this procedure is known as k-fold cross-validation. When a certain value for k is chosen, it may be used in place of k in the reference to the model, such as k = 5 becoming 5-fold cross-validation. The general procedure is as follows.

a) Shuffle the dataset randomly.

b) Split the dataset into k groups with members equal to the number of samples that each group contains (each contains 20% of the whole dataset instances).

c) For each of the obtained unique groups, take one group as a validation dataset (validation fold), while keeping the remaining groups as a training dataset (training fold).

2.2.4 Classification Models Training

The extracted feature vector from the feature engineering stage can be fed into different classification algorithms. For instance, this could be done by utilizing any machine learning library in Python (e.g., Keras [140], Pytorch [141], and Scikit-Learn [142]), R (e.g., TensorFlow [143], TidyModels [144], and MLlib [145]), MATLAB's statistics and machine learning toolbox [146], etc., to build different classification models. All these tools provide a variety of supervised and unsupervised learning algorithms via a consistent interface. The previously split training and validation sets are now used

for the training of the classification model and the validation of the built model. In the following, we discuss some of the related work to both sentiment analysis and misleading information detection as application domains of social network analysis.

2.2.4.1 Sentiment Analysis

In general, sentiment analysis, also interchangeably referred to as, opinion mining (OM), is the computational treatment of opinions, dedicated to the exploration of subjective opinions, or feelings about a particular entity (subject, or object). Depending on the source of data under analysis and the application domain, this entity can be activities, events, individuals, organizations, decisions, etc.

Despite both the terms sentiment analysis and opinion mining express mutual meaning, they have slightly different notations. Opinion mining aims to extract and analyze the opinion about an entity while sentiment analysis first identifies the sentiment expressed in a text then performs full analysis on it. Therefore, opinion mining can be considered as the first step to perform sentiment analysis, followed by the identification of polarity scores of the text, then finally classifying text into either positive, negative, or neutral classes.

Sentiment analysis can be carried out using one of three techniques (lexicon-based, machine learning-based, or deep learning-based techniques) on three different levels of analysis (sentence-level, document-level, or aspect-level). The level of analysis and the deployed technique varies depending on the goal of the sentiment analysis task. Figure 2.7 shows a workflow for a typical sentiment analysis system [147].

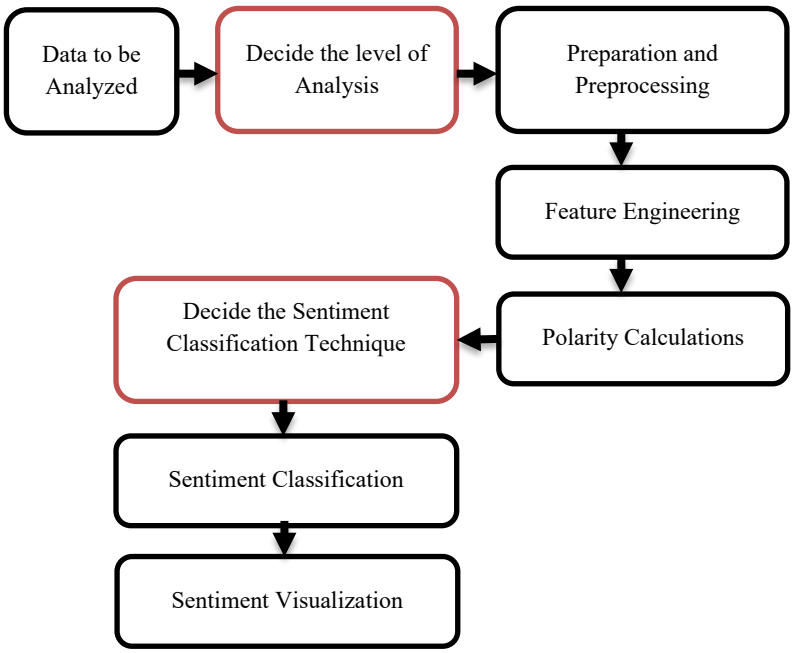


Figure 2.7 Typical Framework for Text Sentiment Analysis.

a) ***Sentiment Analysis Levels:*** For the sentiment analysis levels, both the sentence-level and the document-level analyses are relatively the same as a document can contain only one sentence. In both cases, the document is parsed into sentences. Next, the subjectivity of each sentence is examined, and a subjectivity score is calculated to determine whether a sentence is subjective or objective. Then, for the subjective sentences, the polarity score is calculated. Finally, based on the level of analysis and the obtained polarity score, the sentences/documents are assigned into either positive or negative class notations.

For illustration, assume that we have a document *D* which consists of five sentences {s_1, s_2, s_3, s_4, s_5}. Also, assume that using one of the sentiment analysis classification techniques, the obtained sentiment of each sentence is {s_1: positive, s_2: negative, s_3: positive, s_4: negative, s_5: negative}, and when treating all sentences as one big entity, the obtained sentiment of *D* is positive. Hence, for the sentence-level analysis, *D* is assigned to the negative class, as a result of the majority voting between the sentiment class obtained for each sentence. While, for the document level analysis, *D* is assigned to positive. One of the limitations of these two levels of analysis is that they do not provide the necessary detail needed on all aspects of the entity which is needed in many applications.

Unlike both the sentence and the document levels of analyses, aspect-level analysis, also interchangeably referred to as, entity-level or feature-level, is done to perform fine-grained sentiment analysis [148]. It can find what people like or dislike regarding different aspects of a topic [149]. For instance, customers can express their opinion towards different aspects of the same entity, as shown in this sentence "The sound of this car's engine is very low, but it is very expensive" where the customer gives a positive opinion towards the engine's operating sound but gives a negative opinion towards the car's stock price.

b) ***Sentiment Classification Techniques:*** There are many classifications of the techniques that can be used in sentiment analysis as introduced in [147], [150], [151], [152], [153], [154], [155], and others. Based on these classifications, it can be generalized that, sentiments can be classified mainly using either Handcrafted, Machine-generated features, or a combination of them (hybrid features). The techniques used with Handcrafted features are either lexicon-based or machine learning-based techniques [156], while the Machine-generated features are obtained from deploying different deep learning techniques [151]. Figure 2.8 summarizes different sentiment analysis classification techniques.

(1) ***Handcrafted Features:*** These features have been employed for decades, and are still widely used, especially when combined with traditional machine learning classifiers [157]. These features are obtained using different feature engineering techniques (feature selection and feature extraction) and used to enhance the performance of different machine learning techniques [109]. An example of the handcrafted features is

the linguistic features, including Part-of-Speech tags (PoS-Tags), hashtags, emotions, n-grams (unigram, bigram, trigram, etc.), negations, acronyms, and others [151].

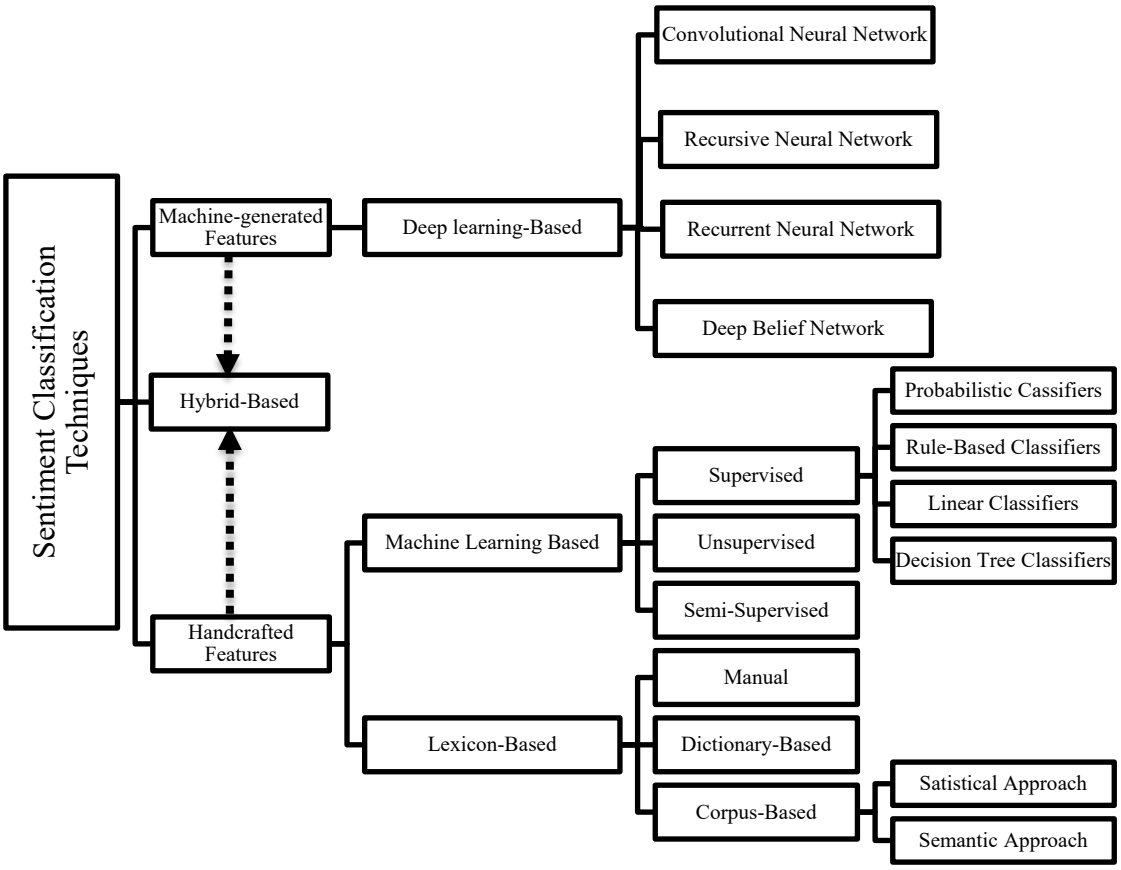


Figure 2.8 Sentiment Classification Techniques.

(a) ***Machine Learning Approach:*** Generally, machine learning is divided into supervised, unsupervised, and semi-supervised learning methods [60], As shown in Figure 2.8. For performing sentiment analysis, different machine learning techniques can be employed to detect the sentiment class of a document or set of documents using a set of syntactic features, linguistic features, or a combination of them.

(i) Supervised Learning Methods [158]*:* there is a need to train the used algorithm using pre-existing labeled training data. This training data must contain labeled instances highly related to the topic under analysis. As shown in Figure 2.8, the used algorithms for building supervised sentiment analysis classification systems can be categorized into:

- Probabilistic classifiers are also referred to as generative classifiers (such as Naïve Bayes Classifier (NB), Bayesian Network (BN), and Maximum Entropy Classifier (ME)).
- Linear classifiers (such as Support Vector Machines Classifiers (SVM), Logistic Regression (LR), Perceptron, and Multilayer Perceptron (also referred to as Neural Network (NN))).
- Decision Tree classifiers, in which the training data is recursively split based on a logical condition and provides a hierarchical decomposition of the training data space. This split could be a single attribute split, similarity-based multi-attribute split, or discriminant-based multi-attribute split.
- Rule-based classifiers, in which the training data is modeled into a set of rules based on measuring the support and the confidence of each of the obtained rules.

It should be remarked that both the Decision Tree classifiers and the Rule-based classifiers are modeling the data in form of a set of rules, but the Decision Tree tends to achieve this hierarchically.

(ii) *Unsupervised Learning Methods:* If the available labeled data is not sufficient to train supervised sentiment analysis classification models, unsupervised techniques are employed [156]. The goal of unsupervised learning is to group unlabeled data according to some patterns without any previous information of data and without any human interaction (Clustering). Therefore, it finds the unseen relationships between data instances [155]. Self-Organizing Maps, k-Means Clustering, and Expectation Maximization are some examples of unsupervised clustering algorithms [159].

(iii) *Semi-Supervised Learning Methods:* When the amount of the available labeled data is small, Semi-Supervised methods utilize a combination of labeled and unlabeled data (partially labeled training data) to minimize both the computational cost and effort for labeling data instances [160]. Semi-Supervised methods are categorized into transductive and inductive methods [159]. Latent semantic indexing (LSI), self-trained Logistic Model Trees (LMT), graph-based models, semi-supervised support vector machines are some examples of semi-supervised clustering algorithms [161].

(b) *Lexicon-Based Approach:* The lexicon-based approaches integrate a lexical resource with different text processing algorithms. The sentiment lexicons can be constructed manually, using word dictionaries, or based on a corpus related to some

topic. Opinion Lexicon, Subjectivity Lexicon, and Semantic Lexicon are examples of state-of-the-art lexical resources.

A small set of sentiment words are labeled manually, and this seed list is expanded with synonyms and antonyms from a thesaurus. This approach is very time-consuming and usually combined with the other two approaches as a final check to avoid the mistakes that resulted from the automated method. Both the dictionary-based and the corpus-based approaches are unable to find context-specific sentiment orientations.

We could summarize the limitations and the challenges of the existing approaches that employ the handcrafted features as follows:

- There is a shortage of available labeled data in many application domains. Also, there is a limited number of resources written in languages other than English.
- With the huge amount of available online data, obtaining sentiment labels is a very costly process. Moreover, the amount of available labeled data might not be sufficient to build and optimize different sentiment analysis systems.
- It is hard to build cross-domain sentiment analysis models, and almost all the available models are domain-specific.

(2) Machine-generated Features: Machine-generated features are obtained when deploying different deep learning techniques. They result in reducing the burden of feature design as when the network learns, it automatically creates the required features for the classification process. Moreover, it can capture non-linear and complex patterns in the data.

Deep Neural Networks (DNNs) are made up of artificial neural networks with multiple hidden layers between both the input and the output layers. Convolutional Neural Network (CNNs), Recursive Neural Network (Rec-NNs), Recurrent Neural Network (RNNs), and deep belief networks are examples of deep learning algorithms [151].

One of the main limitations of using deep learning techniques is choosing and tunning hyperparameters. The performance of the developed model is highly dependent on the values of the hyperparameters used in designing the network. Hence, discovering the optimal hyperparameter values is a challenging task. Also, with a large number of parameters, the huge amount of data, and how dense the designed network is, it requires high computational resources (e.g., fast GPUs and large RAM) to be able to perform the training efficiently [151].

Khan *et al.* [162] and Jindal *et al.* [163] gave a detailed and deep review of machine learning approaches, document representation techniques, and the datasets and techniques used for text-

documents classification. It is indicated from the review that the Vector Space Model (VSM) is the most used method for document representation for text in the classification process. For textual documents to be transformed into VSM each document is being represented in a compact set of words. Ravi *et al.* [164] introduced a survey that covers the different types of sentiment analysis, natural language processing techniques used, their applications, and categorization of some sentiment analysis algorithms and their originating references, as well as the datasets used. Also, Yang *et al.* [165] introduced the common sentiment analysis used techniques such as Support Vector Machines (SVM), Naive Bays (NB), Maximum Entropy, and Artificial Neural Network (ANN) methods. Furthermore, they discuss the performance assessment and difficulties.

Pang *et al.* [166] were the first to apply many machine learning techniques for sentiment mining on movie reviews corpus. Unigram and bag of words were utilized to obtain features. The ratio of accuracy differs according to their application. While, Das *et al.* [167], attempted to make financial decisions such as stock market prediction and the potential prices of a company's stock. Kim *et al.* [168] found that if features are highly dependent on other variables, Naïve Bayes works relatively well. This may be counterintuitive because Naïve Bayes uses features that are independent of each other. Niu *et al.* [169] used a new model where there were well-thought-out approaches used for feature selection, weight computation, and classification. This was based on the Bayesian algorithm in which the weights of the classifier are modified by using both representative and unique features. 'representative feature' deals with the information that represents a class and 'unique feature' deals with the information that aids in distinguishing classes from each other. Probabilities were calculated using this method on each classification. This helped improve the Bayesian algorithm.

Barbosa *et al.* [170] created a 2-step automatic sentiment analysis method for classifying tweets. A noisy training set was used to reduce the effort required to label in developing classifiers. They initially classified tweets into subjective and objective tweets. Next, subjective tweets were classified as positive and negative tweets. Celikyilmaz *et al.* [171] also created a pronunciation-based word clustering method that could normalize noisy tweets. In pronunciation-based word clustering, words with a similar pronunciation are grouped and labeled as common tokens. Additionally, they also assigned similar tokens for numbers, HTML links, user identifiers, and target organization names for normalization. Finally, after normalization, they used probabilistic models to identify polarity lexicons for features. Classification using the BoosTexter classifier with these polarity lexicons as features was performed and a reduced error rate (ERR) was obtained.

Nabil *et al.* [172] introduced ASTD, an Arabic social sentiment analysis dataset gathered from Twitter. It consists of about 10,000 tweets which are classified as objective, subjective positive, subjective negative, and subjective mixed. They presented the properties and the statistics of the dataset and run experiments using standard partitioning of the dataset. Guellil *et al.* [173] presented an approach to automatically classify sentiments of Arabic text which relies on Latin letters, numerals,

and punctuation rather than Arabic letters (Arabizi). By translating and transforming Latin into Arabic, then applying different machine learning algorithms. The automatic classification process of the sentiment is made with techniques, such as SVM and NB are used. The obtained results demonstrate the superior performance of the NB algorithm over all others. The highest achieved F1-Score is up to 78% and 76% for manually and automatically transliterated datasets, respectively.

Baly *et al.* in [174] conducted an analytical study to observe challenges to perform opinion mining in a language that is rich in morphology such as the Arabic language. They applied different models on the same dataset in [172], the obtained accuracy results were 49.5% when applying the SVM, and 58.5% when applying the Recursive Neural Tensor Networks (RNTN). While, Heikal *et al.* [175], applied different deep learning models that have not been applied to Arabic data, to improve the Arabic sentiment analysis accuracy. They used an ensemble model, combining Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) models, to predict the sentiment of Arabic tweets. The obtained result after applying their model on the Arabic Sentiment tweets Dataset (ASTD), achieves an accuracy of 64.3%, 64.75%, 65.05% after applying CNN, LSTM, and the ensemble models respectively [172].

Bhavitha *et al.* in [176] compared different classifiers for sentiment analysis. This was done on Twitter posts related to electronic products such as mobiles and laptops. They utilized various classifiers such as Naïve Bayes, SVM, Maximum Entropy, and Ensemble for text classification and compared their ACC, Precision (P), and Recall (R). A dataset of 1200 Twitter posts equally divided into positive and negative sentiment classes are used as training (1000) and test (200) sets. Results concluded that the Naïve Bayes classifier has the highest P but lacks ACC and R. Rane *et al.* [177] worked on a dataset comprising of tweets for 6 major US Airlines and performed a multi-class sentiment analysis. This approach starts with preprocessing techniques used to clean the tweets and then representing these tweets as vectors using a deep learning concept (Doc2vec) to do a phrase-level analysis. They applied different classification techniques and uses the P, R, and F1 to evaluate the performance of the classifiers.

We searched for the most used datasets related to the task of analyzing the sentiment in both Arabic and English languages for only the last three years (since 2017). The most used datasets are shown in Table 2.1. Among these, Sentiment140 and the Restaurants Reviews Dataset (RES) are the most popular ones. This could be due to the variety of available samples on them. Moreover, we concluded that most of the work depends on manually collected and annotated datasets and there is a lack of the availability of a standard dataset in Arabic languages.

It is noted that Arabic resources are not as rich as in the English language. i.e., the lack of Arabic datasets may be considered a challenge in the field of text mining.

Table 2.1 A Comparison Between Most Used Sentiment Datasets.

No.	Dataset	Lang.	Size	# of related citations (Since Jan. 2017 until May 2021)
1	Amazon Reviews for Sentiment Analysis [178]	En.	4,000,000	50
2	Sentiment140 [179]	En.	1,600,000	249
3	Sentiment Labelled Sentences Dataset [180]	En.	770,000	52
4	Tweets for Sentiment Analysis [181]	En.	43,679	138
5	Sentiment Analysis in Text [182]	En.	40,000	186
6	Airline Twitter Sentiment [183]	En.	14,872	79
7	Sentiment of Climate Change [184]	En.	6,090	2
8	RES: Restaurants Reviews Dataset [185]	Ar.	33,000	8
9	ASTD: an Arabic Social Sentiment Analysis [186]	Ar.	10,000	63
10	Twitter Dataset for Arabic Sentiment Analysis [187]	Ar.	2,000	20

2.2.4.2 News Analytics (Misleading Information Detection)

Misleading information has been widely used to indicate information pollution in the form of false news, hoaxes, propaganda, rumors, and junk news [188], but there are still no agreed definitions and categorizations for them [7]. In [6] we introduced a general categorization of misleading information into Disinformation, Misinformation, and Malinformation. Disinformation is defined as false information created and shared by people with harmful intent. Misinformation is defined as some kind of false information disseminated online by people who do not have ill intent, while Malinformation is defined as the sharing of genuine information with the intent to cause harm [189]. Table 2.2 shows a comparison of the different used terms to point to misleading information and the corresponding category or categories assigned to each term.

Table 2.2 Description of the Misleading Information Terminologies Used on Social Media.

Term	Definition	Category		
		Disinfo.	Misinfo.	Malinfo.
Fake/False News	Fabricated news articles that could be potentially or intentionally misleading for the readers, as they mimic traditional news content in form but not in the intent or the organizational process [190].	√	√	√
Hoax	A fiction intentionally fabricated to masquerade as the truth [191].	√	√	
Propaganda	News stories are created to influence the emotions, opinions, and actions of the target audiences through deception, selectively omitting or providing one-sided messages for political, ideological, or religious purposes [192].	√		√
Satire/Parody	A form of news that is written to entertain or criticize the readers, and it could mimic genuine news; this is harmful when shared out of context [192].			√
Rumors	Originated from a Latin word that means noise and has been identified by some scholars as a subset of propaganda [193]. As an unverified claim that did not originate from news events, it could spread from one user to another [194].	√	√	
Click-bait	Low-quality journalism is intended to attract traffic and benefit from advertising revenue [195].	√		
Junk news	More generic and aggregates several types of information; it usually refers to the overall content that pertains to a publisher rather than a single article [196].	√		√

The spread of such misleading information, using eWOM, on these social network platforms could potentially have a negative impact on society. In recent years, most countries were suffering from economic problems, political issues, wars, terrorism, and violent conflicts, as discussed in *Section 1.1*. The term fake news is very old, it gained a bad reputation and became a popular term during the 2016 US residential election campaign [197]. Active research has been ongoing for the development of automated, reliable, and accurate techniques for detecting fake news on social network platforms. The detection of fake news could be defined as the process of estimating whether a particular news article of any topic, from any domain, is being intentional or unintentionally misleading [198]. Most of the fake news detection systems deploy machine learning techniques to assist users in filtering the news they are viewing and detecting whether a particular news article is deceptive or not. This classification and analysis are done based on comparing a given news article with some pre-known news corpora that contain both misleading and truthful news articles [199].

Fake news detection is not simple but rather is a complicated problem [200]. The detection task requires several steps to classify a given set of news articles (text documents). The preprocessing of the collected news documents varies depending on the type of data and the language used in the documents as shown in Figure 1.1. Generally, most news articles contain textual data. Therefore, the documents must be transformed to another representation, to be able to have extracted feature vectors that contain enough information to ensure accurate classification and are suitable to be maintained by machines.

For deploying machine learning techniques (supervised learning) in building fake news detection systems, all news documents should pass through different stages. These stages aim in making news documents machine-processable, easing their manipulation, and reducing the required memory and time to process them. Feature engineering is considered the most important stage for building fake news detection systems [201]. It assists the handling of the huge amount of data that are used in building the detection model; by selecting and extracting the most representative feature vector [7]. Accordingly, news-related feature selection techniques are being applied based on feature types available in the news documents [109]. These features could be user-based, content-based, context-based, domain-based, etc. [6] [7], or a combination of them. In general, textual news documents are simply textual documents that are related to certain topics (sports, economics, politics, etc.), from a specific source (news websites, blogs, social network platforms, etc.), and have news specific characteristics (location, publisher, author of news, publication date, etc.) [202].

Some of the studies on text classification in general, used the weighted feature vectors to improve classification results, as discussed in detail in *Subsubsection 2.2.2.5*. The weight of each feature indicates the feature's importance thus enhancing the classification results. We observed from the investigated research that the main challenge facing the researchers in implementing new techniques for fake news detection is the poor quality of the existing data. The most used datasets

are shown in Table 2.3. Among these, the LIAR dataset is one of the most popular, probably due to the variety of feature types that could be extracted from it.

Table 2.3 A Comparison Between Most Used Misleading Information Datasets.

No.	Dataset	Purpose	Size	# of related citations (Since Jan. 2017 until May 2021)
1	FEVER [203]	Fact Extraction	185,000	102
2	KaggleFN [204]	Fake News Detection	13,000	95
3	FNC-1 [205]	Stance Detection	50,000	44
4	LIAR (PolitiFact) [206]	Fake News Detection	12,800	202
5	PHEME [207]	Fake News and Rumor Detection	6,425	37
6	Sina Weibo [208]	Fake News and Rumor Detection	4,664	100
7	FakeNewsNet (GossipCop) [209]	Fake News Detection	3,570	313
8	Buzzfeed news [210]	Fake News Detection	2,282	44
9	Twitter15 [211]	Fake News and Rumor Detection	1,490	114
10	Twitter16 [211]	Fake News and Rumor Detection	818	115

After an in-depth review, we could conclude that most of the available work focuses on linguistic features in only one specific language, especially, English written textual data. Other multilingual platforms that handle the data in more than one language deploy a language translation step to convert the data from any language into English data. Moreover, the handling of multilingual data simultaneously, either for News Analytics (misleading information detection) or sentiment analysis, has not been considered yet [212]. Moreover, it could be remarked that the content-dependent features are the most used since most of the available data is textual data in either structured, semi-structured, or unstructured formats. Almost 80 percent of the available textual data are in the unstructured format [60], and the supervised learning techniques are extensively used in building the detection models for them. Finally, the main challenge for building and testing the effectiveness of any detection system is the lack of a generalized and standard dataset for fake news detection [7] [200].

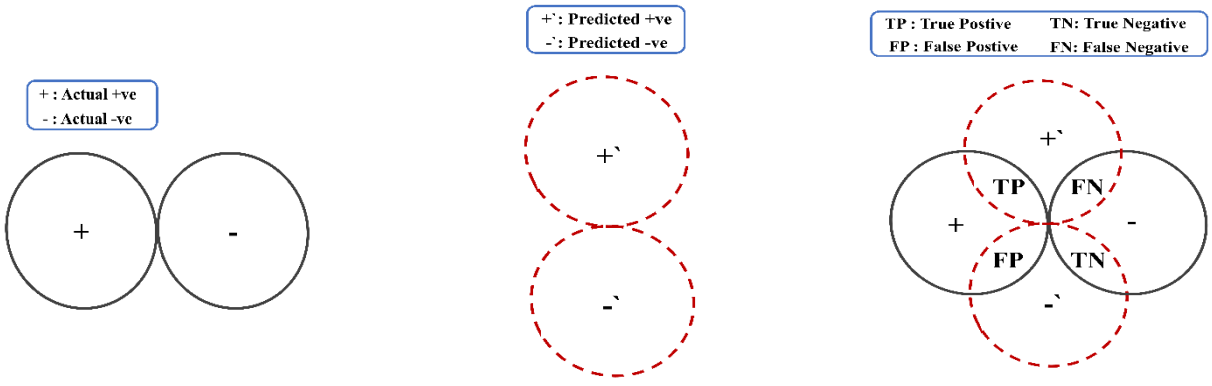
Additionally, many existing works focus on introducing different feature extraction techniques to build weighted feature vectors from textual data; most of them are (TF-IDF)-based techniques. TF-IDF considers that the term (t) can distinguish between classes if this term appears frequently in a document (d) and is infrequently found in other documents in the dataset (D). This technique gives the different weights to a term in the feature vector based on its frequency in its contained document. This means that the TF-IDF can make the distinction at the documents level. Although if a term appears frequently in a document of a class, it means that this term is more

frequent in this class than others and can effectively distinguish between the textual content of different classes. Hence, this potentially class-important term should be given a higher weight and treated unequally in different classes to overcome the deficiency of the calculation of TF-IDF weight.

2.2.5 Performance Evaluation

When building and optimizing any supervised learning model, measuring how accurately it can classify data is crucial, especially when the developer must choose between two or more algorithms. It is an easy question to ask but rather a problematic dilemma to answer what algorithm should be chosen if one of the used algorithms performs better on one class and the other on the other class [213]. In most cases, obtaining high classification accuracy results gives a misleading indication of the model's classification ability, especially when dealing with imbalanced datasets available in real life [214]. A dataset that contains two classes is said to be imbalanced when one of the classes is under-represented with respect to the other class. Overcoming this problem is particularly important in applications where misclassifying instances from the minority class is more costly [215]. Hence, the ability to evaluate classification models independently of the size of datasets and the distribution of data on their classes is pivotal to selecting the most appropriate model to employ [12]. It should be remarked that the selection of the most appropriate evaluation metric varies and is based on many factors, such as, the size of the dataset, the distribution of data on classes, and which class is more important to the end-user and thus to the developer [216].

In this dissertation, we are focusing on binary classification problems. For binary classification, each instance in a given dataset is mapped to either positive (+) or negative (-) classes [217], [218], as shown in Figure 2.9(a). A classification model is a mapping from instances to predicted classes. To distinguish between the actual class and the predicted class of an instance, we use the labels (+`) and (-`) for the classifications produced by a model for both the positive and negative classes, respectively, as shown in Figure 2.9(b). Figure 2.9(c) gives a visualization of the obtained TP, TN, FP, and FN.



(a) (b) (c)

Figure 2.9 Classification Results Visualization. (a) Original Dataset, (b) Classification Results, (c) The Obtained Confusion Matrix.

To judge the performance of any classification algorithm, it is required, first, to understand how to make this judgment and to define the criteria on which this judgment holds. A good classification algorithm is the one that has as large values of TP, and TN as possible ($the\ best\ is\ when\ TP + TN \rightarrow 100\%\ of\ the\ overall\ data$) while keeping both the values of both FP and FN as small as possible ($the\ best\ is\ when\ FP + FN \rightarrow 0\%\ of\ the\ overall\ data$), as shown in Figure 2.10(a). While on the contrary, the worst classification algorithm is one that has no TP, and TN (the worst when *TP+TN→0%* of the overall data), FP+FN→100% of the overall data, as shown in Figure 2.10(b).

The tradeoff made in some cases is when an algorithm can classify all instances of one class (i.e., TP = 100% of positive instances), while not being able to classify data from the other class (i.e., TN = 0% of negative instances), as shown in Figure 2.10(c), and hence, the obtained classification accuracy, in this case, is misleading.

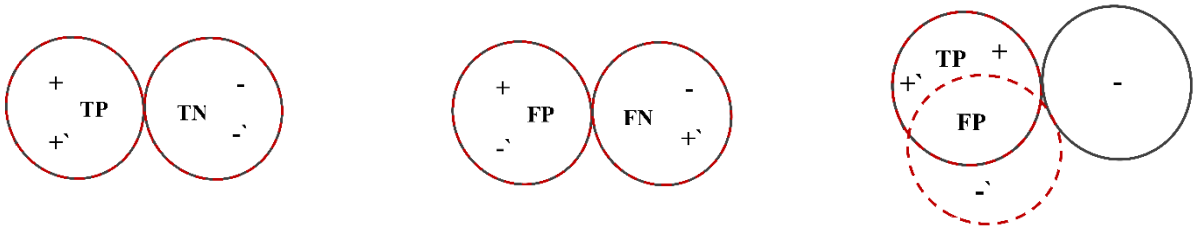


Figure 2.10 Visualization of Different Scenarios for Classification Performance. (a) Best-Case Scenario (TP + TN = 100%), (b) Worst-Case Scenario (FP + FN = 100%), (c) Tradeoff Case Scenario (TN = FN = 0%).

The process of evaluating the performance of designed machine learning models is crucial to ensure the efficiency and reliability of designed models [109]. When assessing these models, developers must keep in mind the ultimate goal of the system under development, and the purpose of the evaluation process. Hence, several quantitative measures are used to provide a single numerical representation of the model's quality, the representation of error, and misclassification rates. Hence, the performance evaluation is based on the TP, FN, FP, and TN scores specified in the 2x2 confusion matrix [219], [220]. The values that form the confusion matrix represent the count of correctly and incorrectly predicted instances by the model. It provides general insight into the model, which is not only related to its performance but also the behavior of the model, showing which classes are being predicted correctly and incorrectly. Moreover, it helps in showing what type of errors are being made in the classification process. For instance, in the case of a binary classification problem, we have two main types of error: FP (Type I error (False Alarm)) and FN (Type II error).

Based on the obtained classification confusion matrix for each of the used algorithms, many measures can be used to measure their classification performance. The measures used are highly

dependent on the size and the distribution of classes of the datasets, as discussed in detail in the rest of this *Section*. Figure 2.11 shows the typical performance evaluation workflow.

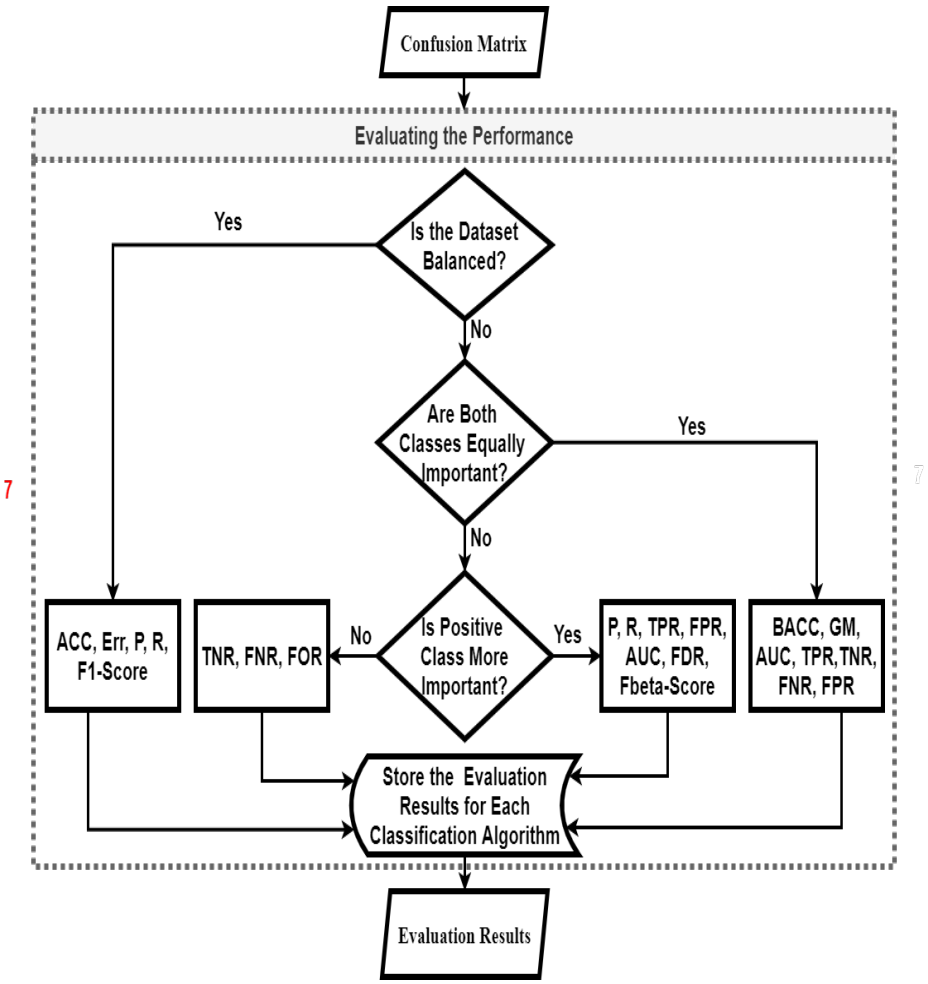


Figure 2.11 Typical Performance Evaluation Workflow.

The measure varies depending on the size of each class of the datasets, in addition to, which class is more important to analyze the performance of the classifier. The first step is to check whether the data set has a balanced or skewed distribution. Then, in the case of the skewed distribution, the metrics used are varied if the positive class is more important. All the used metrics for different purposes are discussed later within this *Section*. The limitation and power of each metric are discussed and clarified in detail.

From the detailed overview discussed in *Appendix A,* many researchers have grouped performance evaluation metrics used to evaluate supervised learning algorithms into three main groups from two points of view. Table 2.4 summarizes the most popular performance evaluation metrics and the corresponding group to which they belong from both points of view.

Table 2.4 Summary of the Most Used Evaluation Metrics (Ordered Alphabetically).

#	Metric	Equation #	1st Categorization			2nd Categorization		
			TM	RM	PM	FM	CM	GRM
1	Accuracy (ACC)	1	√			√		
2	Adjusted Geometric Mean (AGM)	23	√				√	
3	Area Under Lift (AUL)	35		√				√
4	Area Under the Curve (AUC)	33		√				√
5	Balanced Accuracy (BA), Macro-Averaged Accuracy (MAA), or Balanced Classification Rate (BCR)	11	√				√	
6	Balance Error Rate (BER)	15	√				√	
7	Cohen Kappa Metric (κ)	16			√		√	
8	Discriminant Power (DP)	18		√			√	
9	Diagnostic odds ratio (DOR)	29		√			√	
10	Error Rate (ERR)	2	√			√		
11	F1-Score (F1)	20	√				√	
12	Fall-Out (FO), or False Positive Rate (FPR)	10		√		√		
13	False Discovery Rate (FDR)	4		√		√		
14	False Omission Rate (FOR)	8		√		√		
15	Fbeta-Score (F_β)	19	√				√	
16	Fowlkes-Mallows Index (FMI)	21			√		√	
17	Geometric-Mean (GM)	22	√				√	
18	Gini Coefficient (GC)	34		√				√
19	Index of Balanced Accuracy (IBA_α(M))	13, 14	√				√	
20	Jaccard (J), Tanimoto Similarity Coefficient (TSC), or Critical Success Index (CSI)	24			√		√	
21	Markedness (MK)	25			√		√	
22	Matthews Correlation Coefficient (MCC)	26		√			√	
23	Mean-Class-Weighted Accuracy (MCW)	12	√				√	
24	Miss Rate (MR), or False Negative Rate (FNR)	6		√		√		
25	Negative Likelihood Ratio (LR (-))	27		√			√	
26	Negative Predictive Value (NPV)	7			√	√		
27	Optimized Precision (OP)	30	√				√	
28	Positive Likelihood Ratio (LR (+))	28		√			√	
29	Positive Predictive Value (PPV)	3			√	√		
30	Precision (P)	3	√			√		
31	Recall (R), True Positive Rate (TPR), or Sensitivity (SN)	5	√			√		
32	Specificity (SP), True Negative Rate (TNR), or Selectivity (SL)	9	√			√		
33	Youden's Index (γ), or Bookmaker Informedness (BM)	31, 32		√			√	

Among all these metrics, the most common and straightforward metrics are the ACC and the F1 [219]. However, we should note that these values can sometimes be misleading. It is not necessary that obtaining high values of ACC or F1 reflects the superiority of the classification algorithm. ACC and F1 do not fully consider the size of the dataset and the distribution of its classes in their final score computation. Therefore, to highlight the shortcomings of using ACC and F1, consider we have an imbalanced synthetic dataset D that contains two classes C_1 (+ve) and C_2 (-ve). C_1 has 1,000 instances, while C_2 has 50 instances. Assume that we used three classification algorithms CL_1, CL_2, and CL_3, and suppose also that we have, by mistake and we are not aware of it, some errors in designing and training CL_3 which make it always predicts positive. Suppose that after applying the classification process, the TP, TN, FP, and FN obtained results are shown in Table 2.5.

Table 2.5 The TP, TN, FP, and FN Results.

	TP	FN	TN	FP
CL_1	960	40	41	9
CL_2	950	50	48	2
CL_3	1000	0	0	50

From Table 2.5 and by substituting in eq. A.1, the ACC of the three classifiers is 95.33%, 95.05, and 95.24%, and by substituting in eq. A.20, the F1 is 97.51%, 97.34%, and 97.56%, respectively. It could be remarked that CL_1 gives the highest ACC, while CL_3 gives the highest F1. Despite the three classifiers give high scores, we have two problems. The first is not being able to decide which is the best, and the second is that these results are misleading; CL_3 was unable to correctly detect any of the C_2 instances. At the first glance, we could say that CL_1 is the best, in reality, the best one is CL_2; despite its obtained ACC and F1 are not the best and lower than other classifiers, it correctly detects 96 % of C_2.

To avoid being dragged into this debate and to overcome ambiguity, other performance evaluations come into play to help extract more meaning from the built model, such as Balanced Accuracy (BACC), Error (Err), Precision (P), Recall (R), True Positive Rate (TPR), True Negative Rate (TNR), False Discovery Rate (FDR), False Omission Rate (FOR), Area Under the Curve (AUC), Geometric Mean (GM), Fbeta-Score, and the Matthews Correlation Coefficient (MCC), and others [221] [222]. Many of the used metrics are not useful when the two classes are different in size.

The MCC is regarded as being the most informative single score to determine the quality of a binary classifier prediction in a confusion matrix context [223], [224]. The MCC is a correlation coefficient between the observed and predicted binary classifications. It returns a value between −1 and +1. A coefficient of +1 represents a perfect prediction, 0 represents a random prediction and −1 indicates total disagreement between prediction and observation.

In the example above and by substituting in eq. 26, the MCC for CL_1 and CL_2 is 62.255% and 66.619% respectively. While for CL_3, since TN and FN are equal to 0, the denominator becomes 0. Consequently, the result of MCC is undefined. Hence to solve this mathematical error, the assumption is made by arbitrarily set the denominator to one when any of the four sums in the denominator is zero. As a result, *MCC = 0*, which can be shown to be the correct limiting value. By checking the obtained MCC scores we could conclude that CL_2 is performing the best among other algorithms. Obviously, we should note that F1 depends, totally, on which is considered as the positive class. That is why the obtained F1 in the previous example is high because the majority class is defined as the positive class. Hence, assuming that the positive and negative classes are switched, the results are shown in Table 2.6.

Table 2.6 The TP, TN, FP, and FN Results (-ve Class is the Majority).

	TP	FN	TN	FP
CL_1	41	9	960	40
CL_2	48	2	950	50
CL_3	0	50	1000	0

From Table 2.6, the resulted ACC remains unchanged, whereas the F1 decreases dramatically to 62.595%, 64.865%, and 0%, for CL_1, CL_2, and CL_3, respectively. The MCC is unaffected by this switch, and the results remain the same. This shows the advantage of MCC over other evaluation metrics.

Despite the superiority of MCC compared with other metrics, it has a limitation of making arbitrary assumptions to overcome the divide-by-zero problem [225]. Moreover, it is unable to assess the class-based performance of classification algorithms [226]. Hence, there is a need to devise a more generic performance evaluation metric, in addition to the traditional ones, to give an overall insight into the performance of the supervised algorithms used, as introduced in *Section 3.2*. Moreover, it must have the ability to effectively distinguish between the performance of different classification algorithms, without making assumptions. Besides, it should be able to act independently of both the size of the used dataset and the distribution of instances over its classes.

2.2.6 Majority Voting

To obtain the final classification results, based on the obtained performance results of each of the classification techniques used, the best n-models are being loaded to assemble a voting ensemble classification model. Then, hard voting is carried out on all the obtained results to get the classification decision.

For example, by choosing n = 3, i.e., the ensemble result is based on voting amongst the top three algorithms' results (Alg1, Alg2, and Alg3). Given an instance, I, whose class can be predicted as either 0 or 1, assume that the resultant classification value from each model is 1 for Alg1 and 0

for Alg2 and Alg3. As a result, 0 will be the final ensemble decision because two out of the three classifiers predict class 0. Figure 2.12 shows the voting ensemble method.

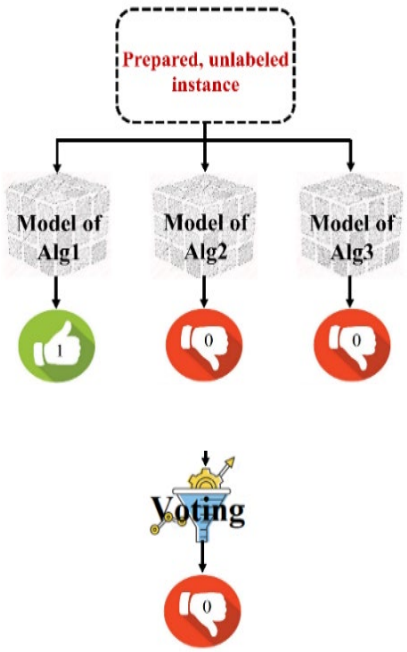


Figure 2.12 The Hard-Voting Ensemble Method.

2.3 Summary

To sum up, all the previously discussed steps aim to find the best configuration that achieves the optimal performance, taking into consideration which error type is most acceptable. In practice, this represents a major tradeoff consideration depending on the situation. A typical real-life example of this tradeoff dilemma is the diagnosis of COVID-19, where a positive diagnosis of a viral infection is based on some tests. FP means that someone has been informed they are infected while not being a carrier of the viral illness. FN, on the other hand, happens when a person is informed that they have tested negative despite being sick.

If we want to check whether a patient is to be quarantined or not, in this case, telling someone that they have tested positive, while they are negative, can lead to immense emotional distress, stress, additional medical costs, and waste of the available medical resources. On the contrary, failure to reveal positively infected patients leads to the uncontrolled spread of the infection. On some other non-critical applications, neither type of error is as serious as the case of COVID-19. For example, in detecting spam emails; missing important emails due to the restricted classification policy of the used spam classifiers is more critical than letting some spam get into your inbox.

Therefore, when choosing evaluation metrics, their drawbacks should be known and considered to be aware of possible limitations and weaknesses in the evaluation assessment [154]. This is of critical importance because the applied metric is the basis for all performance judgments in the respective task. Therefore, the used metrics should be informative, comparable, and concurrently give intuitive assessment for better interpretability. As a matter of fact, choosing an appropriate metric is generally a challenge in assessing supervised learning techniques but is more difficult when dealing with imbalanced datasets. Firstly, because most of the widely used standard metrics assume a balanced class distribution, which is not a practical real-life case. Secondly, not all prediction errors are equal for imbalanced classification. Hence, different fields could use different specific metrics for their designated goals. For example, sensitivity and specificity are the most used evaluation metrics in the medical domain, while in computer science, using precision, recall, and F1-Score are preferred.

Chapter 3

Proposed Classification and Performance Evaluation Frameworks

For performing classification of the textual content of social network platforms, each piece of textual content is transformed and represented by a set of words that expresses its global meaning. In traditional approaches, a document is represented by a group of words describing its contents [163]. This representation is done by transforming documents from the full-text version to a document vector. This transformation aims to ease the handling of these documents and reduce their complexity. The main problems when dealing with the huge amount of daily jotted textual content in social network platforms are not only the extremely high dimensionality of textual data so that the number of potential features often exceeds the number of training documents, but also the variety of the used languages with different dialects for data exchange over the social network and the medium's noisy nature. This makes the conventional handling, preprocessing and analysis technologies inadequate as discussed in *Subsubsection 2.1.3.2*.

This chapter is structured as follows: The proposed framework for analyzing and classifying the textual content of social network platforms, and the details of both the learning and the classification stages, are introduced in *Section 3.1.* Then, the proposed performance evaluation framework is presented in *Section 3.2*. In section 3.3. the limitations of the introduced frameworks are discussed.

3.1 Proposed Framework for the Classification of Social Network Textual Content

For social network textual content (within this dissertation it is also referred to as textual documents) to be classified, they must be prepared first to be suitable for processing, by passing through the document preparation stage as described in *Subsection 3.1.1*. Then, each text document must be transformed and represented by a set of words that expresses its content. The classification process goes through two phases: training and testing. Each of the phases contains different stages, as depicted in Figure 3.1.

In the training phase, the two stages are the feature engineering stage and the learning stage. While in the testing phase, the stages are the feature engineering stage (which is the same as in the training phase) and the classification stage. Feature engineering is described in *Subsection 3.1.2*, while the learning and the classification stages are presented in *Subsection 3.1.3*. The proposed

techniques result in better results when performing two individual social network analysis tasks, sentiment analysis, and fake news detection, as discussed later in *Subsection 4.3.1*.

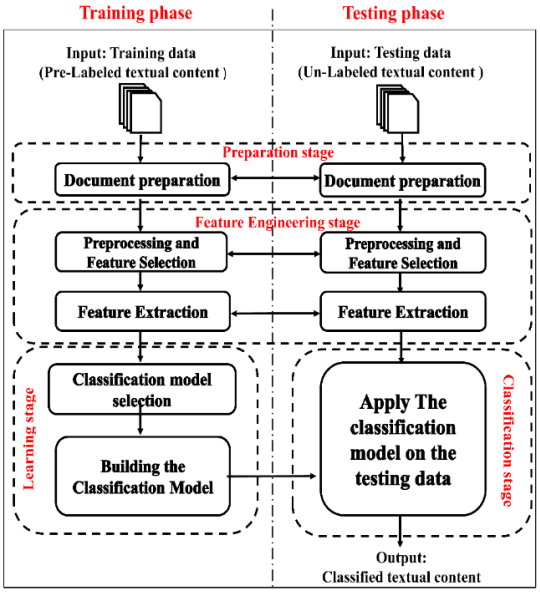


Figure 3.1 Functional Block Diagram of Building the Classification Model for Social Network Textual Content.

3.1.1 Document Preparation Stage

Usually, the extracted social network content is segmented into title, content, publisher, source, location, etc. Each textual document is transformed into a non-segmented format by grouping all its segments into a single segment. This segment contains a union of the original segments. For example, a sample text document we obtained from the FA-KES benchmark dataset [227], is shown in Figure 3.2.

Title:	Eight Civilians Killed and 14 Injured in Terrorist Bombing in Damascus
Content:	09-02-2016 Eight Civilians Killed and 14 Injured in Terrorist Bombing in Damascus. A terrorist car bomb rocked Masaken Barzeh neighborhood in Damascus city on Tuesday claiming eight and injuring 14 citizens according to Al-Manar reporter. SANA news agency mentioned that the attack took place near a fruit and vegetable market in the area and caused material damage to the nearby building of the General Establishment for Mills.
Source:	Al-Manar
Date:	2/9/2016
Location:	Damascus
Class:	Fake

Figure 3.2 Sample Training Textual Document.

As a result of the Initial Data Analysis (IDA) and the Exploratory Data Analysis (EDA) on the sample document and subsequently, to perform the transformation, the following actions are applied:

1- Remove repeated data from that content that is the same as in the title or in any other field.

2- Enforce the date into a standard format, for example, the date is transformed into (02 Sep 2016) from (2/9/2016).

3- Convert all numbers from numeric values into textual written numbers, e.g., the numeric value (14) is written as (fourteen).
4- Perform class encoding by assigning a numerical value to each of the dataset categorical classes, for example, the numeric value (0) is assigned to the categorical class value (Fake) and the numeric value (1) is assigned to the categorical class value (Real).

For illustration, we notice in Figure 3.2 that the content of the title section of the document is replicated at the beginning of the content section "*Eight Civilians Killed 14 Injures in Terrorist Bombing in Damascus*". Moreover, the date in the document is also repeated many times but in a different format *"09-02-2016"* and *"2/9/2016"*. Additionally, numbers are written in both the numeric and textual format "*Eight"* and "*14*". Hence, after passing through the document preparation stage, the repeated data is removed, and only one copy is kept. The dates are set to the "*DD MON YYYY*" format. All numbers are transformed into a textual format and then all segments are grouped into a single segment, as shown in Figure 3.3.

Text:	Eight Civilians Killed and fourteen Injured in Terrorist Bombing in Damascus. A terrorist car bomb rocked Masaken Barzeh neighborhood in Damascus city on Tuesday claiming eight and injuring fourteen citizens according to Al-Manar reporter. SANA news agency mentioned that the attack took place near a fruit and vegetable market in the area and caused material damage to the nearby building of the General Establishment for Mills, Al-Manar, 09 Feb 2016, Damascus.
Class:	0

Figure 3.3 The Prepared Textual Document Sample.

The purpose of this step is to simulate how some humans, in real life, are dealing with textual content on social media. Moreover, reducing the noise and eliminating the redundant information in our data ease its handling in the upcoming steps.

3.1.2 Feature Engineering Stage

This stage is composed of two steps: 1) Preprocessing and the proposed novel feature selection step as described in *Subsubsection 3.1.2.1. 2*) The proposed novel feature extraction step, as presented in *Subsubsection 3.1.2.2.*

3.1.2.1 Preprocessing and Feature Selection step

As our goal is to deal with the bilingual textual content of social network platforms, the first issue that needs to be addressed is how to represent this huge amount of textual content and select their corresponding features. Not only that a proper representation eases data manipulation and saves time and memory needed for processing such data, but it also maintains the necessary information without any loss and helps in overcoming some of the challenges discussed earlier in *Section 1.3.*

In this step, the goal is to transform and represent the textual documents using VSM. This transformation results in extracting the BoW that represents the textual content. This is done by extracting a set of words the text contains and their frequency regardless of their order [25]. In our approach, we use a method in which we apply two main sub-steps on both the training and the testing text documents to build the feature vector for mining tasks. The first is preprocessing, while the second is selecting a set of features from both textual content and its accompanying metadata (hybrid features).

Algorithm 1 (preprocessing and feature selection algorithm) and Table 3.1 include the definition of the basic elements of the algorithm, and the tasks as follows:

a) Preprocessing:

During the preprocessing step, the encoding of the processed text document is examined to detect the language. Depending on the detected language, each text document is being preprocessed and then undergone a procedure to select highly descriptive and representative features for the feature vector, as follows:

(1) NLP Parser: depending on the detected language, this step is responsible for processing text to detect sentences and tokens, by separating the words for analysis. For example, the English word "*won't*" should be split into two words "*will*" and "*not*" for further text analysis. PoS tagging is performed to tag the words in the text as corresponding to a particular part of speech, such as verbs, nouns, adjectives, etc. [228].

To illustrate the idea behind the NLP Parser, assume that the following text *"The Egyptian army is fighting terrorism* 💪✌, *and their efforts won't be neglected* 😇♡. *الجيش المصري يحارب الإرهاب، ولا يمكن لأحد أن ينكر جهودهم* " is fed as an input to the feature extraction phase. This text contains both Arabic and English words. First, this text is processed to detect the sentences in it, and results in a set of separated sentences as follows:

- The Egyptian army is fighting terrorism 💪✌.
- and their efforts won't be neglected 😇💔.
- الجيش المصري يحارب الإرهاب
- ولا يمكن لأحد أن ينكر جهودهم

After that, each of the resulting sentences is tokenized by separating the words for analysis, and results in a set of separated sentences as follows:

- The Egyptian army is fighting terrorism 💪✌.
- and their efforts will not be neglected 😇💔.
- الجيش المصري يحارب الإرهاب
- ولا يمكن لأحد أن ينكر جهودهم

Algorithm 1 Preprocessing and Feature Selection Algorithm.

```
Process (Document Ndoc, EngStopWords ESWF, AraStopWords ASWF) {
    /* Text Parsing */
    DB: = Parse (Ndoc)
    /* Text Tokenization */
    TDB: = Tokenize (DB)
    /*Data Cleaning*/
    ERE = "(@[A-Za-z0-9 \u0600-\u06FF] +) | ([^0-9A-Za-z \u0600-\u06FF \t]) | (\w+: \/\/\S+) | (RT)"  /*The regular expression utilized to keep only Arabic and English written data, in addition to numeric values*/
    ∀ TDB_i∈TDB {
        if (Encoding (TDB_i, (ISO-8859-1 || Windows-1252)) && (PatternMatcher (TDB_i, ERE) = True || IsCapital (TDB_i) = True)) /*Check the character encoding to detect English tokens in the processed text and deploy both the regular expression and the capital letters heuristic*/
            CTBD: = CTBD ∪ TDB_i
        elseif (Encoding (TDB_i, (ISO 8859-6 || Windows-1256)) && (PatternMatcher (TDB_i, ERE) = True) /*Check the character encoding to detect Arabic tokens in the processed text and deploy the regular expression)
            AraNormalize (TDB_i) /*Normalize Arabic written tokens*/
            CTBD: = CTBD ∪ TDB_i
    }
    /* Part of Speech Tagging*/
    TW= PoS_Tag (CTBD)
    ∀ TW_i∈TW {
        if (WordTag (TW_i, (("NN"||"NNP" ||"JJ" ||"VB")) = True && WordLength (TW_i)>2) /*Check PoS tags to keep only NN, NNP, JJ, and VB tagged words, besides applying the no-short heuristic*/
            TTBD: = TTBD ∪ TW_i
    }
    /* Chunking Text*/
    CW: = Chunk (TW)
    /* Extracting Named Entities*/
    NE: = ExractNE (CW)
    /* Initializing BoW*/
    IBoW = null
    /* Loading Stop Words File */
    LSW = ReadWords (ESWF, ASWF)
    /*Stop words removal*/
    ∀ TTBD_i ∈TTBD {
        ∀ LSW_j ∈ LSW {
            if (WordCompare (TTBD_i, LSW_j) = False)
                IBoW: = IBoW ∪ TTBD_i
        }
    }
    /* Stemming process*/
    ∀ IBoW _I ∈ IBoW {
        SBoW: = SBoW ∪ StemTerm (IBoW_i)
    }
Return (SBoW ∪ NE)
}
```

Table 3.1 Basic Elements of Algorithm 1.

#	Element	Definition
1	Ndoc	Textual document in an unstructured format.
2	ESWF	Stop words file that contains a list of English stop words, auxiliary verbs, adverbs, etc.
3	ASWF	Stop words file that contains a list of Arabic stop words, auxiliary verbs, adverbs, etc.
4	DB	The textual document parsed in a readable format suitable for the next steps.
5	TDB	Tokens of the parsed text document.
6	ERE	The regular expression is used to allow words that contain only Arabic and English letters, and numeric values.
7	CTBD	Clean tokens after removing non-English/non-Arabic and symbolic words.
8	TW	List of tagged words after applying Part of Speech tagging algorithm.
9	TTBD	List of nouns, verbs, and adjectives tagged words.
10	CW	List of chunks related to the tagged words after applying the chunking algorithm.
11	NE	The list of named entities (organizations, locations, date, time, persons, etc.).
12	IBoW	Initial Bag of Words.
13	LSW	List of Stop words.
14	SBoW	Stemmed Bag of Words.

(2) *Data Cleaning:* this step is responsible for cleaning noise. For English written data, the regular expression "*(@[A-Za-z0-9 \u0600-\u06FF] +) | ([^0-9A-Za-z \u0600-\u06FF \t]) | (\w+: \/\/\S+) | (RT)*" is used to remove non-English words and the words that contain symbolic characters, emojis, flags, punctuations, etc., by using a mechanism that detects only the words that match the expression and discards those which do not match. For Arabic written data, in addition to the use of the previously mentioned regular expression, a normalization process is done to ensure the same way of writing Arabic letters, e.g., ("إ"→"ا"), ("ى"→"ي"), ("ؤ"→"ء"), ("ئ"→"ء").

The result after applying the Data Cleaning task is cleaned words with only English, Arabic, and numeric data. The output is as follows:

- The Egyptian army is fighting terrorism
- and their efforts will not be neglected
- الجيش المصري يحارب الارهاب
- ولا يمكن لاحد ان ينكر جهودهم

For further illustration of the data cleaning step, only one English sentence and one Arabic sentence is examined as an example. Figure 3.4 provides a sample example of the parsing process, with the English sentence "*The Egyptian army is fighting terrorism.*". First, the words are marked as corresponding to a part of speech, through PoS tagging. The tagged components are:

- DT: Determiner.
- JJ: Adjective.
- NN: Common Noun.
- VBZ: Verb, 3rd person singular present.
- VBG: Verb, gerund, or present participle.

(3) Chunking: is used to divide the text into syntactically correlated parts of words. In this example the result is noun and verb phrases:

- NP: Noun Phrase
- VP: Verb Phrase

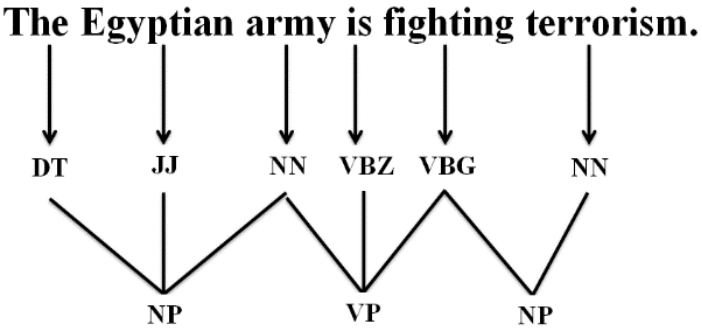


Figure 3.4 Sample Output of the English NLP Parser.

While for the Arabic sentence, Figure 3.5 provides a sample example of the parsing process for the sentence "الجيش *المصري يحارب الإرهاب*". First, the words are marked as corresponding to a part of speech, through PoS tagging, then Chunking results in dividing the sentence into noun and verb phrases

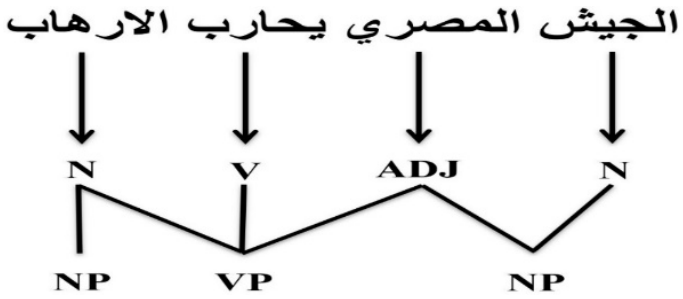


Figure 3.5 Sample Output of the Arabic NLP Parser.

(4) Stop words removal: in which removal of the stop words, such as (a, an, in, at, …, etc.), the auxiliary verbs, adverbs, etc., is done by searching for words in a pre-existing Stop words list [229] [230]. The same concept exists in other languages; in Arabic language, examples of stop word are امس, ولم, لها, في, …, etc.

(5) Stemming: It is fundamental in text analysis to avoid such patterns and redundancy. Thus, different equivalent morphological forms are replaced by their corresponding root word. Compound words are replaced by their morphological root. For example, the words "tester", "testing", and "tested", all share the same root word "test". Stemming is done by applying any of the stemming algorithms. It was noted that combining words with the same root may reduce indexing size by as much as 40-50% [109].

In this dissertation, for Arabic data, ISRI Arabic stemmer [231] from the NLTK Python library is used to perform the stemming process. SRI Arabic stemmer is based on an algorithm named "Arabic stemming without a root dictionary" that was developed by the Information Science Research Institute, University of Nevada, Las Vegas, USA. While for English data, we used Porter English stemmer [232] from the NLTK python library to perform the word stemming process. At the end of the preprocessing step, we obtained a set of the stemmed-cleaned bag of words that represents the original features that are used for the generation of the feature vector $d_i = \{t_1, t_2, t_3, \ldots, t_m\}$. Table 3.2 shows the results of stemming the set of words of the leftmost column.

Table 3.2 Sample Results of Applying Porter and ISRI Stemming Algorithms.

Words	Stemmed word
Fighting Fights	Fight
Concentrate Concentrates Concentrating	Concentr
يحارب (yuhaarib) (Fight)	حارب (haarab)
الجيش (aljaish) الجيوش (aljeuyoush) (Army)	جيش (jaish)
المصري (almisri) (Egyptian)	مصري (misri)
الإرهاب (al'iirhab) (Terrorism)	ارهاب ('iirhab)

b) The Proposed Hybrid Feature Selection:

In this step, the goal is to use a complex set of features from metadata accompanied with the document (language-independent features), in addition to the extracted linguistic features (language-dependent features). This aims to enrich the generated feature vector with valuable features for building the information classification model by applying the following steps:

(1) Applying capital letters heuristic to keep all words that begin with capital letters. As wherever there exists in the text a word that begins with a capital letter, it is an indication of its importance, and it should not be neglected.

(2) Applying a no-short heuristic to remove all words with the number of characters less than or equal to two.

(3) Considering only the words that are tagged as verbs, nouns, and adjectives to reduce the dimension of the extracted feature vector size, as these words are the most representative and descriptive parts in any text.

(4) Use TextBlob python library [233] for English text, and TextBlob_ar python library [234] for Arabic text to create a new feature for the text sentiment by calculating sentiment polarity which lies in the range of [-1, 1] where 1 means positive sentiment and -1 means negative sentiment.

(5) Selecting relevant information from data, such as location-based, user-based, and time-based features. Taking Twitter as an example, the selected features for each tweet are as follows:

- *UserCountry and UserCity:* the user-defined country and city stored in his/her profile. The extracted country and city names are processed and replaced with their corresponding standard name utilizing the *Pycountry Python API* [235].
- *UserTweetsRatio:* the ratio between the number of tweets (including retweets) issued by the user and the number of days since his/her account was created.
- *UserListedCountRatio:* the ratio between the number of public lists that this user is a member of, and the number of days since his/her account was created.
- *UserDescriptionLength:* the length of the text in the description field in the user's profile.
- *UserDescriptionWordCount:* the number of text words in the description field in the user's profile.
- *UserFollowerToFriendsRatio:* the ratio between the number of friends of a user and his/her followers.
- *TweetCountry:* the country that corresponds to where a tweet is originating from. The extracted country name is processed and replaced with its corresponding standard name utilizing the *Pycountry Python API* [235].
- *CountryMatch:* equals 1 if the values of both the *TweetCountry and UserCountry* match and 0 otherwise.
- *TweetSource:* the utility (application) used to post the tweet.
- *TweetLength:* the length of the tweet's text.
- *TweetWordCount:* the number of words in a tweet's text.

- *TweetCapitalCount:* the number of words starts with capital letters in the tweet's text.
- *TweetSentenceCount:* the count of sentences in a tweet's text.
- *TweetVerbCount:* the count of verbs in a tweet's text.
- *TweetNounCount:* the count of nouns in a tweet's text.
- *TweetAdjCount:* the count of adjectives in a tweet's text.
- *TweetStopWordsCount:* the count of stop words in a tweet's text.
- *TweetProperNamesCount:* the count of proper names found in a tweet's text.
- *TweetLocCount:* the count of location names mentioned in the tweet's text.
- *AvgCharWord:* the average number of characters in words of a tweet.
- *AvgCharSent:* the average number of characters in sentences of a tweet.
- *AvgWordSent:* the average number of words in sentences of a tweet.
- *AvgPuncSent:* the average number of punctuations in sentences of a tweet.
- *TweetLinkCount:* the count of URLs in the tweet's text.
- *TweetDate*, and *TweetDayHour* for both the date and at what time of the day the tweets have been posted. We get these by processing the associated timestamp with each tweet.
- PositiveWordsRatio: the ratio of positive words to the number of words in a tweet's text.
- NegativeWordsRatio: the ratio of negative words to the number of words in a tweet's text.

(6) Removing all links, HTML encodings (e.g., &amp, &lt, &gt, etc.), symbolic and non-English words.

(7) Select the source information related to the document's publisher. For example: "published on Twitter", "published on CNN website", etc., as the classification model could come up with some relation between the document's source as a feature, other selected features from both the text, metadata, and the label of the textual document.

At the end of this step, each document on our dataset is represented as a vector of words. Then, the whole data document could be represented as a vector of vectors of words, which can be represented as a sparse matrix. The rows of this sparse matrix represent the documents in the dataset we have, while the columns contain the words that each document contains.

3.1.2.2 The Proposed Feature Extraction Step

From the literature, most researchers are depending on TF-IDF or an enhanced version of it as a feature extraction technique, as discussed in *Subsubsection 2.2.2.5*. TF-IDF weight reflects the importance of a term (*t*) to each document (d) in the whole dataset documents (*D*) used for training. This gives the

weight value for *t* in each of the containing documents based on how frequent it is in this document, even if it should be more important for some classes than others. In other words, TF-IDF can be considered a document-level weight; as it computes the importance of *t* based on its frequency within a document (*d*) where it belongs, and the number of documents in the whole *D* in which *t* appears [236].

In our model, we propose a novel technique, "**T**erm **C**lass **I**mportance (**TCI**)", for assigning representative weight for each term $t \in$ document $d \in D$ regarding its importance in class $c \in C$ to which *d* belongs. This technique is considered a class-level weight that takes into account the distribution of documents on different dataset classes. It gives weights to each *t* based on their importance in the classes their contained documents belong to. This weight is affected by the distribution of documents in each class. In other words, the TCI feature extraction technique gives different weights to the same term according to the class its containing document belongs to. This leads to a more class-representative and discriminative feature vector. As a result, the overall performance of the detection model is enhanced by being able to effectively distinguish between documents with respect to different classes.

The proposed feature extraction technique is applied to calculate the importance of each term *t* with respect to each class $c \in$ *all the classes* C in the dataset *D*. This technique consists of two parts as follows:

1. **T**erm **C**lass **F**requency (**TCF**), which represents how many times the term *t* appears in the documents that belong to class $c \in C$. We defined it as,

$$\mathrm{TCF}(t, c) = fr(t, c) \tag{3.1}$$

where fr (*t, c*) is the frequency of *t* in *c*.

2. **I**nverse **C**lass-**D**ocuments **F**requency (**ICDF**), which represents the inverse relative class documents frequency of *t* with respect to other classes *c* in *D*. To calculate **ICDF**, we first calculate the term **C**lass **D**ocument **F**requency (**CDF**) which is the number of documents *d* in class *c* that the term *t* appears in, as shown in eq. 3.2.

$$\mathrm{CDF}(t, c) = |c \in C : d \rightarrow c : t \in d| \tag{3.2}$$

Then we defined ICDF(*t, c*) as,

$$\mathrm{ICDF}(t, c) = \log\left(\frac{n_c}{\mathrm{CDF}(t, c)}\right) \tag{3.3}$$

where n_c is the total number of class c documents; $n_c = |d \in D : d \rightarrow c|$

It could be remarked that some terms may not be present in any of the classes. This makes CDF(*t, c*) = 0, and accordingly, leads to the problem of *divide-by-zero*. Therefore, the denominator is adjusted to be $(1 + \mathrm{CDF}(t, c))$ to avoid the mathematical error. Then, the ICDF(*t, c*) formula becomes as we defined in eq. 3.4.

$$\text{ICDF}(t, c) = \log\left(\frac{n_c}{\text{CDF}(t, c) + 1}\right) \quad (3.4)$$

3. Finally, we defined TCI in eq. 3.5.

$$\text{TCI}(t, c) = \text{TCF}(t, c) \times \text{ICDF}(t, c) \quad (3.5)$$

The added adjustment value could have different impacts on the calculated ICDF values depending on the size of the used training dataset and the distribution of documents in its contained classes, according to one of the following cases:

a) Case1, large-balanced dataset:

(1) With a large CDF, the effect could be neglected as it does not make any significant changes to the ICDF value. As with large n_c and large CDF, the logarithm result is very small, then the overall TCI is small reflecting the unimportance of this term to its corresponding class.

(2) With a small CDF, the effect could also be neglected as it does not make any significant change on the ICDF. As with large n_c and small CDF, the logarithm result is large, then the overall TCI is large reflecting the importance of this term to its corresponding class.

b) Case2, large-unbalanced dataset:

(1) With large n_c and large CDF, the effect could be neglected, the same as clarified for Case1(1), and the overall TCI is small reflecting the unimportance of this term to its corresponding class.

(2) With large n_c and low CDF, the effect could be neglected, the same as clarified for Case1(2), and the overall TCI is large reflecting the importance of this term to its corresponding class.

(3) With small n_c and large CDF, the value of the CDF could be affected; as with small n_c, CDF is relatively small as ($\forall t \rightarrow \text{CDF} \leq n_c$), but their values are still not comparable to the added 1, then the overall TCI remains small as if we did not add the adjustment. The TCI value reflects the unimportance of this term to its corresponding class.

(4) With small n_c and small CDF, the effect could be the same as clarified for Case2(3), but TCI is larger than Case2(3), reflecting the importance of this term to its corresponding class.

c) Case3, small-(balanced/unbalanced) dataset:

(1) With a large or small CDF, the effect is significant which is the same as clarified for Case2(3), and Case2(4).

***d) Case4*, General case**:

Small/large-balanced/unbalanced dataset, a value of CDF = n_c, means that the term appears in each of the documents that represent the class, then this term is not important

in representing this class. The logarithm results in a negative value, which is set to zero, and the TCI, as a result, is also zero. The reason for this is that not all the classification algorithms could accept negative values in the data provided for building the classification model, such as the Multinomial Naïve Bayes classifier.

When the data sources are social network platforms, then the size of the collected data is typically large and thus is considered to be big data. Many studies have found that for applications with a large number of features and complex classification rules, the training sample size must be large enough [237]. Moreover, a large number of testing samples is essential for an accurate evaluation of the built classification models [238], [239]. Hence, only Cases1 and Case2 may occur. This means that the adjustment value has minimal impact on the calculated TCI values while it helps to avoid the divide-by-zero problem. The TCI value of a term is high, only if the term has a high TCF with low ICDF values, indicating the significance of this term in its corresponding class. For example, a term that appears multiple times in more documents in a certain category causes the value of ICDF to be very low, hence the total TCI is closer to 0.

To illustrate how TCI works, assume, for simplicity, consider having a dataset that consists of ten documents (d_1, d_2, …, d_{10}), classified into two classes (c_1 and c_2) and the resulted bag of words of the dataset are (t_1, t_2, …, t_5), as shown in Figure 3.6. The obtained sparse matrix that represents the dataset is as shown in Figure 3.7.

Documents	Vector of Words	Class
d_1	$\{t_1, t_2, t_3\}$	c_1
d_2	$\{t_1\}$	c_1
d_3	$\{t_3, t_5\}$	c_2
d_4	$\{t_2, t_3, t_5\}$	c_1
d_5	$\{t_1, t_3\}$	c_2
d_6	$\{t_2\}$	c_2
d_7	$\{\Phi\}$	c_1
d_8	$\{t_1, t_3\}$	c_1
d_9	$\{t_1, t_2, t_3\}$	c_1
d_{10}	$\{t_4, t_5\}$	c_2

Figure 3.6 Vector of Words for the Sample Dataset.

Documents	Words' Sparse Matrix					Class
d_1	t_1	t_2	t_3	0	0	c_1
d_2	t_1	0	0	0	0	c_1
d_3	0	0	t_3	0	t_5	c_2
d_4	0	t_2	t_3	0	t_5	c_1
d_5	t_1	0	t_3	0	0	c_2
d_6	0	t_2	0	0	0	c_2
d_7	0	0	0	0	0	c_1
d_8	t_1	0	t_3	0	0	c_1
d_9	t_1	t_2	t_3	0	0	c_1
d_{10}	0	0	0	t_4	t_5	c_2

Figure 3.7 Words' Sparse Matrix for the Sample Dataset.

This matrix passes through the feature extraction step to obtain well-represented and discriminated weights for the words in this matrix, to be used in training the classification model. The frequency of each term in the dataset documents is shown in Table 3.3.

Table 3.3 Frequencies in the Documents of the Dataset.

Dataset documents	Terms					Class
	t_1	t_2	t_3	t_4	t_5	
d_1	20	2	3	0	0	c_1
d_2	13	0	0	0	0	c_1
d_3	0	0	16	0	2	c_2
d_4	0	30	2	0	46	c_1
d_5	10	0	33	0	0	c_2
d_6	0	20	0	0	0	c_2
d_7	0	0	0	0	0	c_1
d_8	15	0	4	0	0	c_1
d_9	11	1	3	0	0	c_1
d_{10}	0	0	0	16	1	c_2

From Table 3.3,

- Class c_1 has six documents (d_1, d_2, d_4, d_7, d_8, d_9)
- Class c_2 has four documents (d_3, d_5, d_6, d_{10}).

If TCI is calculated for t_3 with respect to c_1 utilizing equations 1, 2, 4, and 5.

From eq. 3.1, TCF(t_3, c_1) = 12

From eq. 3.2, CDF(t_3, c_1) = 4

From eq. 3.4, ICDF(t_3, c_1) = $log\left(\frac{6}{4+1}\right) = 0.08$

Finally, from eq. 3.5, TCI(t_3, c_1) = 0.95

By repeating the previous steps on all the terms in the dataset, we have a TCI-based feature vector that effectively discriminates between the term weights for each available class in the given dataset as shown in Table 3.4.

Table 3.4 Calculated TCI.

Dataset documents	TCI					Class
	t_1	t_2	t_3	t_4	t_5	
d_1	4.67	5.81	0.95	0.00	0.00	c_1
d_2	4.67	0.00	0.00	0.00	0.00	c_1
d_3	0.00	0.00	6.12	0.00	0.37	c_2
d_4	0.00	5.81	0.95	0.00	21.94	c_1
d_5	3.01	0.00	6.12	0.00	0.00	c_2
d_6	0.00	6.02	0.00	0.00	0.00	c_2
d_7	0.00	0.00	0.00	0.00	0.00	c_1
d_8	4.67	0.00	0.95	0.00	0.00	c_1
d_9	4.67	5.81	0.95	0.00	0.00	c_1
d_{10}	0.00	0.00	0.00	4.81	0.37	c_2

In Table 3.4, the TCI weight for each term in the given sample dataset differs from one class to another. This difference represents how important each term is to its corresponding class. For example, the assigned weight for t5 is very low for c2, (TCI(t5, c2) = 0.37), is compared to its weight in c1, (TCI(t5, c1) = 21.94). This indicates that t5 is more important in describing the documents that belong to c1 rather than c2. Moreover, it could be remarked that the assigned weights are the same for terms in the same class even if the term frequencies in their containing documents are different.

For example, even though t1 has frequencies of (20, 13, 15, and 11) in (d1, d2, d8, and d9) ∈ c1, respectively, it has the same TCI weight of 4.67 in all of its containing documents belong to c1. This is in contrast to TF-IDF which gives different weights to the same term based on its frequency in each document, giving more than one weight for the same term in the same class. TF-IDF of t1 has the values of (6.02, 3.91, 3.01, 4.52, and 3.31) in (d1, d2, d5, d8, and d9) using eq. 3.6 [240]. This hinders the process of building effective classification models due to potential overfitting and misclassification. On the other hand, TCI can distinguish between classes based on the documents they contain, and not between documents based on what terms they have.

$$\text{TF-IDF}(t, d) = \text{TF}(t, d) \times \log\left(\frac{N}{\text{DF}}\right) \quad (3.6)$$

where, TF(t, d) is the frequency of t in d, $DF = |\mathrm{d} \in D : t \in d|$ is how many documents in the dataset that contains t, and $N = |D|$ is the total number of documents in D.

3.1.3 Learning and Classification Stages

The extracted feature vector from the feature engineering stage is fed into different well-known classification algorithms. This is done by deploying the Scikit-learn machine learning library in Python [142] to build different classification models. Scikit-learn provides a variety of supervised and unsupervised learning algorithms via a consistent interface, such as Logistic Regression (LR), Decision Trees (DT), SVM, etc. The data is split into training and testing sets of 5 folds. Training of the classification model is being done on the training set. As for testing the classification algorithm corresponding to the built model, we used the test set. The classification results are obtained as presented in *Subsection 4.3.1*.

It should be noted that our proposed classification framework has the following limitations:

1. It is targeting binary classification problems and has not been tested with multiclass classification problems.
2. It is neglecting the effect of different emojis when measuring the sentiment of textual posts from social networks.
3. It is not considering the semantic relations either between social networks textual posts or within the post itself.
4. It can handle only the Gregorian and the Islamic Hijri calendars.

3.2 Proposed Framework for Performance Evaluation

When assessing the performance of different binary classification algorithms, for most applications, the main objective is to find the algorithm that achieves as high as possible of correctly detected instances, and as low as possible of incorrectly detected instances, for all the classes. In other words, the main objective is to choose the system that has the maximum true positive rate (TPR) and true negative rate (TNR), or the maximum sensitivity and specificity values. Most often, the TPR is inversely proportional to TNR; the increase in one of these values leads to a decrease in the other value [241]. For example, to investigate the quality of products in a manufacturing line, it is crucial to minimize both the FP and FN as much as possible, to avoid rejecting good products and accepting defective ones. In contrast, certain applications may aim to achieve the lowest possible FP or FN. For example, when identifying patients who have tested positive for any infectious disease, it is critical to avoid misdiagnosis, that is, the system should have FN as low as possible.

From the literature, the most used performance evaluation metrics are Accuracy (ACC) and F1-Score (F1). However, this can often lead to sub-optimal decisions and giving a misleading indication of the model's classification ability, especially when dealing with real-life imbalanced datasets, as discussed earlier in *Subsection 2.2.5.* Hence, the use of other measures, among them the Matthews Correlation Coefficient (MCC) which is the most widely used measure, became the solution to this problem. Despite the superiority of the MCC compared with other metrics, it has some limitations, as illustrated in *Subsection 2.2.5*, besides not considering the effect of the failure rates in assessing the performance of the classification models.

To meet these needs, we introduce a performance evaluation framework based on a new evaluation metric we name "Multidimensional Classification Assessment Score (MCAS)". MCAS is used to evaluate the performance of learning algorithms by measuring how good is the classification algorithm in the presence of errors. This evaluation metric overcomes the limitations of the existing ones as it works independently regardless of the size of the datasets and the distribution of samples in its classes. The MCAS is a score that measures how efficient is the classification algorithm for dealing with binary classification problems in the presence of errors, as introduced in the following subsections.

3.2.1 Methodology

The MCAS aims in determining the ability of the classification algorithm to achieve high detection rates for each of the dataset classes (referred to as Critical Success Score (CSS)) in presence of the average unsuccessful detection rates (referred to as Critical Failure Score (CFS)) for both the dataset classes.

The CSS represents the total number of correctly classified instances divided by the total number of instances of a relative class (the total number of positive instances or negative instances) plus the number of misclassified instances for that class. It is calculated for both the positive class (CSS_{+ve}) and the negative class (CSS_{-ve}), as we defined in eq. 3.7, and eq. 3.8.

$$\mathrm{CSS}_{+ve} = \frac{TP}{TP + FP + FN} \tag{3.7}$$

$$\mathrm{CSS}_{-ve} = \frac{TN}{TN + FP + FN} \tag{3.8}$$

The CFS represents the average of the total number of positively misclassified instances divided by the summation of the total number of correctly classified instances and the number of positively misclassified instances, and the total number of negatively misclassified instances divided by the summation of the total number of correctly classified instances and the number of negatively misclassified instances), as we defined in eq. 3.9.

$$CFS = \frac{1}{2} \times \left(\frac{FP}{TP + TN + FP} + \frac{FN}{TP + TN + FN} \right) \tag{3.9}$$

The MCAS is obtained as in eq. 3.10 and eq. 3.11. The MCAS is independent of the size of the datasets and the distribution of class samples. In addition, it takes into consideration the effect of falsely detected instances on the correctly detected ones.

$$MCAS = \frac{1}{\lambda_1 + \lambda_2} \times \left(\lambda_1 \times (\mathrm{CSS}_{+ve} - CFS) + \lambda_2 \times (\mathrm{CSS}_{-ve} - CFS) \right) \tag{3.10}$$

$$\begin{aligned} MCAS = \frac{1}{\lambda_1 + \lambda_2} \times \Bigg(& \lambda_1 \times \left(\frac{TP}{TP + FP + FN} - \frac{1}{2} \times \left(\frac{FP}{TP + TN + FP} + \frac{FN}{TP + TN + FN} \right) \right) \\ & + \lambda_2 \times \left(\frac{TN}{TN + FP + FN} - \frac{1}{2} \times \left(\frac{FP}{TP + TN + FP} + \frac{FN}{TP + TN + FN} \right) \right) \Bigg) \end{aligned} \tag{3.11}$$

3.2.2 MCAS Calculation

The values λ_1 and λ_2 , in eq. 3.11, are initially set to 1 when the goal is to assess the classification algorithm when both classes are equally important. Whereas, if the positive class is more important the full weight is set towards the positive component of the equation; hence, λ_1 is set to 2, and λ_2 is set to 0. Similarly, when the negative class is more important, λ_1 is set to 0, and λ_2 is set to 2. The MCAS measures the rate of success of both classes relative to the actual count of each class in addition to the falsely detected instances for that class and subtracts the relative errors for both classes. λ_1 and λ_1 could be in one of the three cases as follows:

a) ***Case #1 (MCAS):*** If both the +ve and the -ve classes are important; then, $\lambda_1 = \lambda_2 = 1$.

b) ***Case #2 ($MCAS_{+ve}$):*** If the +ve class is more important; then, $\lambda_1 = 2,\ \lambda_2 = 0$.

c) ***Case #3 ($MCAS_{-ve}$):*** If the -ve class is more important; then, $\lambda_1 = 0,\ \lambda_2 = 2$.

The range of the possible values of the MCAS is between [-1, 1]. The maximum value of 1 is obtained when the classification system is capable of correctly detecting all the positive and the negative instances. The higher the MCAS value the better is the classification algorithm's performance. While the lowest value -1 is obtained when the system is unable to correctly classifying any of the samples. Table 3.5 shows the MCAS value in both the best-case and the worst-case scenarios illustrated above. Figure 3.8 shows the new performance evaluation framework when deploying MCAS.

Table 3.5 MCAS for Best/Worst-Case Scenarios.

Value / Scenario	Confusion Matrix				MCAS		
	TP (%)	FN (%)	TN (%)	FP (%)	Case #1	Case #2	Case #3
Best-Case	100	0	100	0	1	1	1
Worst-Case	0	100	0	100	-1	-1	-1

As shown in Figure 3.8, depending on the end-user requirements and what assessment is required to be done, independent of the size of the used dataset, values of both λ_1 and λ_1 are set to calculate the MCAS. Then, the results are stored for each of the used classification algorithms for the next step to choose the algorithm with the highest MCAS value. It should be remarked that if the goal is to design an ensemble model, then n-algorithms are chosen, where n is the number of algorithms used in building the ensemble model (n = 3, 5, 7, etc.). For example, if n = 3, the result of the ensemble model is determined based on a vote between the individual outputs of three algorithms with the best MCAS value.

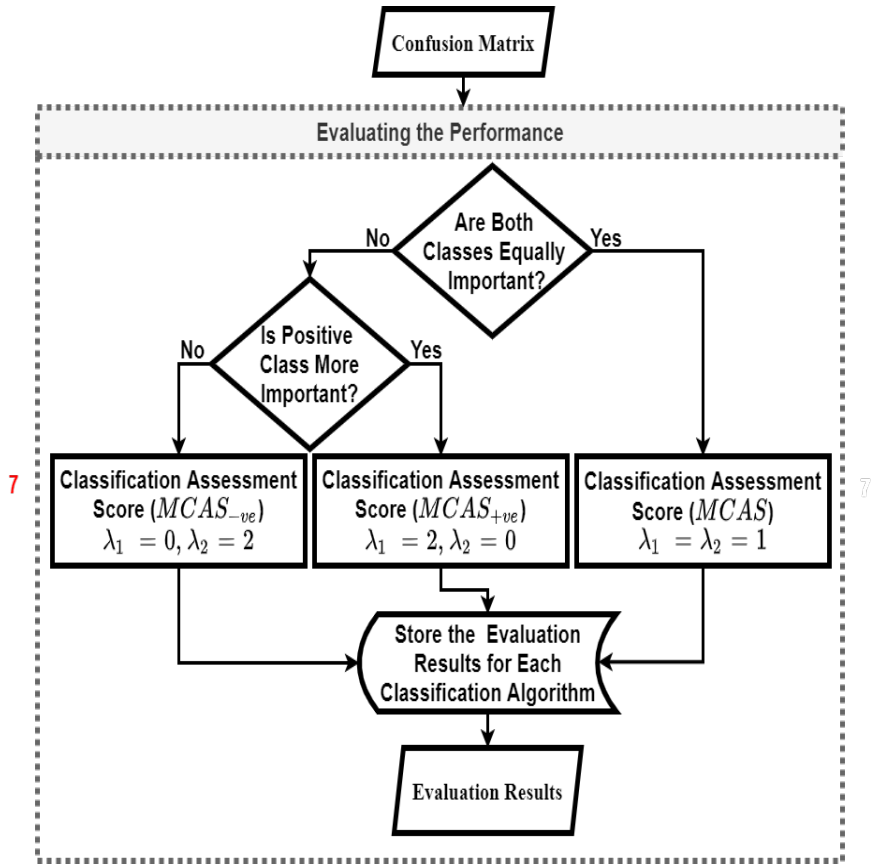


Figure 3.8 Classification Performance Evaluation Framework When Deploying MCAS.

3.2.3 An Empirical Example

As an empirical example to illustrate the benefits and advantages of MCAS, assume that we have two synthetic datasets D_1 and D_2 which represent an imbalanced, and balanced dataset, respectively. Assume that we used seven classification algorithms (C_1, C_2, C_3, C_4, C_5, C_6, and C_7) with these datasets, and the synthetic TP, TN, FN, and FP values are as shown in Table 3.6. It should be noted that the size of datasets and the obtained results were randomly set, in order to show some of the cases that we might find when dealing with real-life datasets and to emphasize the role of the MCAS in overcoming the limitation of existing techniques.

Table 3.6 Assumed TP, TN, FN, and FP Values for D_1 and D_2 Synthetic Datasets.

Dataset	D_1 (P = 1,000, N = 90)				D_2 (P = 1,000,000, N = 900,000)			
Metric / Classifier	TP	FN	TN	FP	TP	FN	TN	FP
C_1	900	100	72	18	900,000	100,000	720,000	180,000
C_2	700	300	63	27	0	1,000,000	630,000	270,000
C_3	400	600	45	45	400,000	600,000	450,000	450,000
C_4	600	400	54	36	600,000	400,000	540,000	360,000
C_5	0	1,000	81	9	900,000	100000	810,000	90,000
C_6	1,000	0	36	54	500,000	500,000	360,000	540,000
C_7	1,000	0	0	90	1,000,000	0	900,000	0

Next, using the assumed TP, TN, FN, and FP values, we calculate the corresponding MCC and MCAS values for each of the obtained results, as shown in Table 3.7.

Table 3.7 Calculated MCC, and MCAS of the Classification Algorithms and the Corresponding Rank for Classifiers (When Both Classes Have the Same Importance ($\lambda_1 = \lambda_2$)).

Dataset	D_1 (P = 1,000, N = 90)				D_2 (P = 1,000,000, N = 900,000)			
Metric / Classifier	MCC (%)	MCAS (%)	MCC- based rank	MCAS- based rank	MCC (%)	MCAS (%)	MCC- based rank	MCAS- based rank
C_1	52.85	57.58	2	2	70.57	66.23	3	3
C_2	23.36	26.34	3	4	-42.90	-29.10	7	7
C_3	-5.60	-10.90	6	6	-10.04	-9.20	6	6
C_4	11.17	12.88	4	5	19.97	17.84	4	4
C_5	-30.42	-47.54	7	7	79.96	76.52	2	2
C_6	61.60	64.96	1	1	-10.03	-8.58	5	5
C_7	0.00	41.74	5	3	100.00	100.00	1	1

To highlight the advantage of using the MCAS, it could be remarked from Table 3.7 that for D_2, both the MCC and MCAS measures give the same ranking of the classification algorithms. While for the case of D_1 the obtained rank of the MCAS is more representative. The best performance is clearly for C_6 with the highest TP and TN, and C_7 is better than C_2 and C_4. It is noticed that the performance of C_7 is better than others in terms of correct classification rates and the corresponding classification errors, even with its inability to correctly detect any negative instances.

When calculating MCC for C_7, because $(TN + FN) = 0$, the denominator is zero causing a mathematical error. As previously discussed, to overcome this mathematical error, the value of the denominator is arbitrarily set to 1 and the total MCC value is 0, which leads to giving C_7 a lower rank and accordingly excluded from the competition. On the contrary, MCAS was able to logically rank the corresponding classifier, with C_7 giving an ACC of 91.74% and TPR at 100%, thus, it does not make sense to exclude it from the competition. Subsequently, the role and importance of conducting a class-

based assessment to fully evaluate and compare the performance of different classification techniques became evident.

Table 3.8 shows the MCAS results for assessing the performance of the classifiers employed with respect to each of the positive and negative classes. These results give a detailed assessment of the performance of the used classification algorithms. This aids in demonstrating to end-users the capabilities of each classification technique in classifying instances of individual classes, complementing the overall assessment by the MCAS.

Table 3.8 Calculated MCAS of the Classification Algorithms and the Corresponding Rank for Classifiers (When Both Classes Have Different Importance ($\lambda_1 \neq \lambda_2$)).

Dataset	D_1 (P = 1,000, N = 90)				D_2 (P = 1,000,000, N = 900,000)			
Relative Class / Classifier	$MCAS_{(+ve)}$	$MCAS_{(-ve)}$	+ve-based rank	-ve-based rank	$MCAS_{(+ve)}$	$MCAS_{(-ve)}$	+ve-based rank	-ve-based rank
C_1	82.84	32.32	3	2	68.36	64.09	3	3
C_2	52.34	0.33	4	3	-45.67	-12.52	7	7
C_3	4.98	-26.78	6	6	-10.41	-8.00	6	5
C_4	36.33	-10.56	5	5	19.13	16.55	4	4
C_5	-51.25	-43.82	7	7	77.31	75.74	2	2
C_6	92.40	37.52	1	1	-5.20	-11.95	5	6
C_7	87.61	-4.13	2	4	100.00	100.00	1	1

From the reported results in Table 3.7 and Table 3.8, The MCAS is able to clarify the ability of C_7 when the positive class is more important, and C_7's inability of performing well when the negative class is more important.

To ensure that the imbalanced distribution of instances does not affect MCAS, the negative and positive classes for our two synthetic datasets D_1 and D_2 are switched, yielding D_1` and $D_2$` datasets. The new TP, TN, FN, and FP classification values are shown in Table 3.9.

Table 3.9 New TP, TN, FN, and FP Values for D_1` and $D_2$` Synthetic Datasets.

Dataset	D_1` (P = 90, N = 1,000)				D_2` (P = 900,000, N = 1,00,000)			
Metric / Classifier	TP	FN	TN	FP	TP	FN	TN	FP
C_1	72	18	900	100	720,000	180,000	900,000	100,000
C_2	63	27	700	300	630,000	270,000	0	1,000,000
C_3	45	45	400	600	450,000	450,000	400,000	600,000
C_4	54	36	600	400	540,000	360,000	600,000	400,000
C_5	81	9	0	1,000	810,000	90,000	900,000	100000
C_6	36	54	1,000	0	360,000	540,000	500,000	500,000
C_7	0	90	1,000	0	900,000	0	1,000,000	0

Using the new TP, TN, FN, and FP values of D_1` and $D_2$` synthetic datasets, the corresponding MCC and MCAS values are calculated for each of the obtained results, as shown in Table 3.10.

Table 3.10 Calculated MCC and MCAS of the Classification Algorithms and the Corresponding Rank for Classifiers (When Both Classes Have the Same Importance ($\lambda_1 = \lambda_2$)).

Dataset	D_1` (P = 90, N = 1,000)				D_2` (P = 900,000, N = 1,00,000)			
Metric / Classifier	MCC (%)	MCAS (%)	MCC- based rank	MCAS- based rank	MCC (%)	MCAS (%)	MCC- based rank	MCAS- based rank
C_1	52.85	57.58	2	2	70.57	66.23	3	3
C_2	23.36	26.34	3	4	-42.90	-29.10	7	7
C_3	-5.60	-10.90	6	6	-10.04	-9.20	6	6
C_4	11.17	12.88	4	5	19.97	17.84	4	4
C_5	-30.42	-47.54	7	7	79.96	76.52	2	2
C_6	61.60	64.96	1	1	-10.03	-8.58	5	5
C_7	0.00	41.74	5	3	100.00	100.00	1	1

Table 3.11 shows the MCAS results for assessing the performance of the employed classifiers with respect to both the positive and negative classes. It could be noticed that the obtained performance results of the MCAS did not change when switching over the classes of the used dataset. This ascertains its stability and ability to perform independently of the distribution and the size of the datasets. In *Subsection 4.3.2*, the MCAS is examined using real-life datasets from both the sentiment analysis and misleading information detection application domains.

Table 3.11 Calculated MCAS of the Classification Algorithms and the Corresponding Rank for Classifiers (When Both Classes Have Different Importance ($\lambda_1 \neq \lambda_2$)).

Dataset	D_1` (P = 90, N = 1,000)				D_2` (P = 900,000, N = 1,00,000)			
Relative Class / Classifier	$MCAS_{(+ve)}$	$MCAS_{(-ve)}$	+ve-based rank	-ve-based rank	$MCAS_{(+ve)}$	$MCAS_{(-ve)}$	+ve-based rank	-ve-based rank
C_1	32.32	82.84		3	64.09	68.36	3	3
C_2	0.33	52.34	3	4	-12.52	-45.67	7	7
C_3	-26.78	4.98	6	6	-8.00	-10.41	5	6
C_4	-10.56	36.33	5	5	16.55	19.13	4	4
C_5	-43.82	-51.25	7	7	75.74	77.31	2	2
C_6	37.52	92.40	1	1	-11.95	-5.20	6	5
C_7	-4.13	87.61	4	2	100.00	100.00	1	1

Chapter 4

Validation and Discussion of the Proposed Frameworks

In this Chapter, we discuss the validation datasets, criteria, procedures, and results, for evaluating the effectiveness of our proposed frameworks. The discussed results in this Chapter correspond to the application domains of sentiment analysis and misleading information detection application domains.

4.1 Benchmark Datasets Description

4.1.1 Sentiment Analysis Datasets

1- ***DAT-01:*** *Restaurant Reviews Dataset (RES)* [242], contains 11.2K Arabic customer reviews on restaurants. These reviews were retrieved from two sources. The first is from "Qaym4" with 8.6K reviews, while the second is from "TripAdvisor" with 2.6K reviews. These reviews are the feedbacks of 3K users on 4.5K restaurants.

2- ***DAT-02****: Arabic Sentiment Tweets Dataset (ASTD)* [172], contains 10K Arabic tweets. These tweets are retrieved over two stages. The first stage involves utilizing SocialBakers [243] to identify the 30 most active Egyptian Twitter accounts. Then, all of the tweets included in these accounts are downloaded. The second stage involves utilizing EgyptTrends [244] to obtain the top 2,500 trending hashtags in Egypt in November 2013. Then, using these hashtags, Twitter is crawled, and the retrieved tweets are downloaded. The gathered tweets from both stages are manually annotated, into four classes (subjective positive, subjective negative, subjective mixed, and objective), by employing Amazon Mechanical Turk (AMT) service [245] through Boto API [246].

3- ***DAT-03****: Stanford Twitter Sentiment Corpus (Stanford)* [247], contains 1.6M tweets automatically annotated as 4 and 0 that represent positive and negative classes based on emotions, respectively. The tweets with positive emoticons, like ":)", were annotated as positive, while the tweets with negative emotions, like ":(", were annotated as negative [247].

4- ***DAT-04****: Twitter US Airline Sentiment Dataset (US Airline)* [248], contains 14.485K instances, derived from Twitter and online reviews from Skytrax for the top 10 US-based airline carriers in February 2015. Data is classified as positive, negative, and neutral tweets. The negatively labeled tweets are followed by a reason, such as "late flight", or "rude service".

5- ***DAT-05****: Uber Ride Reviews Dataset (Uber)* [249], contains 1.344K Uber ride reviews that were collected between 2014-2017. Data is classified into positive, negative, and neutral tweets.

4.1.2 Misleading Information Detection Datasets

1- ***DAT-06****: ISOT Fake News Dataset* [116], contains 25.2K textual news documents related to both fake and real news. These news documents were collected from real-world sources covering the period 2016-2017. The fake news was collected from different unreliable sources that were flagged by *politifact.com* (a fact-checking organization in the USA) and Wikipedia sources. As for the real news, they were collected from *reuters.com*. The dataset covers different topics, and each news document is described by title, text, type, and news publishing date.

2- ***DAT-07****: LIAR Dataset* [250], contains 12.8K, manually labeled, textual news documents. These news documents were collected from politifact.com and contain six labels of truthfulness ratings: pants-fire, false, barely true, half-true, mostly true, and true. The dataset provides some additional meta-data like subject, speaker, job, state, party, context, and history. It could be remarked that all this metadata may not always be available in real-life scenarios.

3- ***DAT-08****: FA-KES Dataset* [227], contains 804 textual news documents about the Syrian war. These news documents were collected from several news organizations and have been labeled using a semi-supervised fact-checking approach. It has both fake and real labels and is the first dataset that presents fake news surrounding the conflict in Syria.

4- ***DAT-09****: Twitter15, and* ***DAT-10****: Twitter16 datasets* [251], are broadly adopted as benchmark datasets for building and optimizing rumor detection applications. The Twitter15 dataset contains 1.49K tweets while Twitter16 contains 818 tweets. Each tweet is labeled into one of four classes: non-rumor, false-rumor, true-rumor, and unverified-rumor.

5- ***DAT-11****: Fake vs Satire Dataset* [252], contains a total of 486 manually annotated news articles (283 fake news stories and 203 satirical stories). These articles are collected from a diverse set of sources. Every article focuses on American politics and was posted between January 2016 and October 2017. Each fake news article is paired with a rebutting article from a reliable source that rebuts the fake source.

In this dissertation, the focus is on binary classification problems. Therefore, for DAT-02, we considered the instances that were labeled as subjective positive or objective as positive instances. While for those that were labeled as subjective negative or subjective mixed, we considered them as negative instances. Additionally, for all other sentiment analysis datasets, we considered neutral and positive labeled instances as positive. For the misleading information datasets, we followed the same technique introduced in [120] to transform the six labels of the DAT-07 dataset into only two labels. This was done by considering the documents labeled with pants-fire, barely true, and false as fake while considering the documents labeled with half-true, mostly-true, and truly as true. Also, we

transformed the four labels of both the DAT-09 and DAT-10 datasets into only two labels rumor and non-rumor, by considering "false-rumor", "true-rumor", and "unverified-rumor" labeled tweets as "rumor". The descriptive characteristics of the used datasets are summarized in Table 4.1.

Table 4.1 Details of the Datasets Used for Validation.

Application Domain	Name	Abbrev.	Size	Covered Topics	Language/ Dialect	Source	Ref.
Sentiment Analysis	Restaurant Reviews (RES)	DAT-01	4.5K	Restaurants	Standard Arabic/ Dialectal Arabic	(qaym4, and tripadvisor).com	[242]
	Arabic Sentiment Tweets (ASTD)	DAT-02	10K	Politics	Arabic/ Egyptian Dialect	twitter.com	[172]
	Stanford Twitter Sentiment Corpus (Stanford)	DAT-03	1.6M	Mixed (Including, Politics, Economy, etc.)	Modern English	twitter.com	[247]
	The Twitter US Airline Sentiment (US Airline)	DAT-04	14.485K	Transportation	Modern English	twitter.com	[248]
	Uber Ride Reviews (Uber)	DAT-05	1.344K	Transportation	Modern English	consumeraffairs. com	[249]
Misleading Information Detection	ISOT Fake News	DAT-06	25.2K	Mixed (Including, Politics, Economy, etc.)	Modern English/ Formal English	(wikipedia, and reuters.) com	[116]
	LIAR	DAT-07	12.8K	Politics	Modern English	politifact.com	[250]
	FA-KES	DAT-08	804	Politics, and Terrorism	Formal English	News Websites	[227]
	Twitter15	DAT-09	1.49K	Mixed (Including, Politics, Economy, Terrorism, etc.)	Modern English	twitter.com, snopes.comandemergent.info	[251]
	Twitter16	DAT-10	818	Mixed (Including, Politics, Economy, Terrorism, etc.)	Modern English	(twitter, and snopes).com	[251]
	Fake vs Satire	DAT-11	486	Mixed (Including, Politics, Terrorism, etc.)	Modern English/ Formal English	Fake News and Satire Websites	[252]

4.2 Performance Evaluation Criteria

There exist many evaluation metrics for measuring how good a classification algorithm is, as discussed in detail in *Subsection 2.2.5*. For testing the performance of our proposed classification framework, we utilize the same performance metrics used in the references that we are comparing. Finally, to examine the superiority of MCAS, validation results are reported based on 15 performance metrics: Accuracy (ACC), Precision (P), Recall (R), F1-Score (F1), True Positive Rate (TPR), True

Negative Rate (TNR), False Positive Rate (FPR), False Negative Rate (FNR), Positive Predictive Value (PPV), Negative Predictive Value (NPV), False Omission Rate (FOR), False Discovery Rate (FDR), Geometric-Mean (GM), Area Under the Curve (AUC), and Geometric-Mean (MCC). These metrics are the most commonly used measures to assess the performance of classification techniques [12]. Each measure suggests the best classifier, which in many cases can be misleading. For instance, the best classifier based on the obtained TPR is the one with the highest value, which is not an appropriate way to analyze this classifier's performance; this classifier may have a low TNR, which suggests that it is not thoroughly assessed, and the conclusion of its superiority is deceptive. The obtained results demonstrate MCAS's superiority over other measures, as well as its ability to provide class-based assessments for each of the classification methods.

4.3 Results and Discussion

To examine the effectiveness of the proposed classification framework, it is employed in two social network analysis application domains: sentiment analysis and misleading information detection. The results are verified in *Subsection 4.3.1*. Then, to show the effectiveness of our proposed performance evaluation framework, we extended our verification and reassessed the results in *Subsection 4.3.1* by utilizing various performance evaluation metrics in addition to the newly introduced Multidimensional Classification Assessment Score (MCAS), as shown in *Subsection 4.3.2*.

4.3.1 Classification Framework

4.3.1.1 Sentiment Analysis

To evaluate the proposed classification framework in the field of sentiment analysis, datasets of both Arabic and English instances are used, as described in *Subsection 4.1.1*. The performance of seven classification algorithms: Decision Tree (DT), k-Nearest Neighbors (kNN), Logistic Regression (LR), Multinomial Naïve Bayes (MNB), Bernoulli Naïve Bayes (BNB), Perceptron, and Neural Network (NN), are analyzed. Validation results reported in this section are based on four performance evaluation metrics: ACC, P, R, and F1, as shown in *Subsection 2.2.5*. 5-folds cross-validation is used for the evaluation. Each dataset is randomly split into two sets: 70% of the documents as a training set, and the rest is the testing set. TF-IDF is used as a term weighting technique.

Table 4.2 shows the ACCs, P, R, and F1 for the seven classifiers on the five Arabic/English datasets. As seen in Figure 4.1 and Figure 4.2, for DAT-01, the MP classification algorithm is the most performant with an accuracy of 81%. Also, in terms of TPR, the best results are obtained by the same classifier with an R of 82%. For DAT-02, the LR classification algorithm is the most performant with an accuracy of 68 %, and in terms of TPR, the best results are obtained by the same classifier with an R of 68%.

Table 4.2 Results Based on Various Datasets Run Among Several Classifiers.

Algorithm	Metrics	Datasets							
		Arabic Dataset				English Dataset			
		DAT-01	DAT-02			DAT-04		DAT-05	DAT-03
			Rerults from [172]	Rerults from [174]	Our results	Rerults from [177]	Our results		
Decision Tree	ACC	74	N/A	N/A	59	N/A	65	75	68
	P	0.73	N/A	N/A	0.56	0.63	0.65	0.73	0.69
	R	0.73	N/A	N/A	0.58	0.645	0.65	0.73	0.69
	F1	0.73	N/A	N/A	0.57	0.645	0.65	0.73	0.69
Multinomial Naïve Bayes	ACC	79	48.4	N/A	67	N/A	70	80	78
	P	0.62	N/A	N/A	0.6	0.642	0.73	0.84	0.78
	R	0.79	N/A	N/A	0.67	0.647	0.70	0.80	0.78
	F1	0.69	0.485	N/A	0.55	0.646	0.63	0.72	0.78
Bernoulli Naïve Bayes	ACC	77	25.3	N/A	66	N/A	76	84	77
	P	0.72	N/A	N/A	0.59	0.642	0.77	0.84	0.77
	R	0.77	N/A	N/A	0.66	0.647	0.76	0.84	0.77
	F1	0.74	0.1.7	N/A	0.61	0.646	0.77	0.84	0.77
Logistic Regression	ACC	79	45.1	N/A	68	N/A	74	80	79
	P	0.72	N/A	N/A	0.64	0.81	0.74	0.64	0.79
	R	0.79	N/A	N/A	0.68	0.816	0.74	0.80	0.79
	F1	0.70	0.449	N/A	0.56	0.819	0.71	0.71	0.79
K-Nearest Neighbors	ACC	74	40.9	N/A	66	N/A	69	84	N/A
	P	0.74	N/A	N/A	0.48	0.59	0.68	0.82	N/A
	R	0.74	N/A	N/A	0.66	0.592	0.69	0.84	N/A
	F1	0.74	0.409	N/A	0.54	0.593	0.68	0.81	N/A
Perceptron	ACC	79	44	N/A	61	N/A	73	80	73
	P	0.76	N/A	N/A	0.59	N/A	0.72	0.78	0.73
	R	0.79	N/A	N/A	0.61	N/A	0.73	0.80	0.73
	F1	0.77	0.439	N/A	0.60	N/A	0.72	0.79	0.73
Multilayer Perceptron	ACC	81	N/A	58.5	63	N/A	74	83	N/A
	P	0.78	N/A	N/A	0.59	N/A	0.73	0.80	N/A
	R	0.82	N/A	N/A	0.61	N/A	0.74	0.83	N/A
	F1	0.78	N/A	0.536	0.60	N/A	0.73	0.80	N/A

For DAT-03, the LR classifier achieves the best ACC and TPR results of 79% each. For DAT-04, the BNB classifier outperforms all others with an ACC of 76%, but in terms of TPR, the LR classifier gives the best results with a value of 74%. For DAT-05, the BNB classifier performs the best with an ACC and TPR of 84% each.

Moreover, the following points are noted for the kNN classifier:

1- The kNN classifier assigns equal weights to each feature in the feature vector. When there are numerous irrelevant features in the data, this may result in increased classification errors.
2- The traditional kNN classifier measures the distance between the documents using different distance measures, such as Euclidean, Manhattan, Chebyshev, and Minkowski, while

ignoring the semantic relations between them. This may lead to misclassification of some of the tested data.

3- When comparing our results with the results from [172], [174], and [177], we find the superiority of our proposed model with much better results in all the classifiers used as shown in Table 4.2.

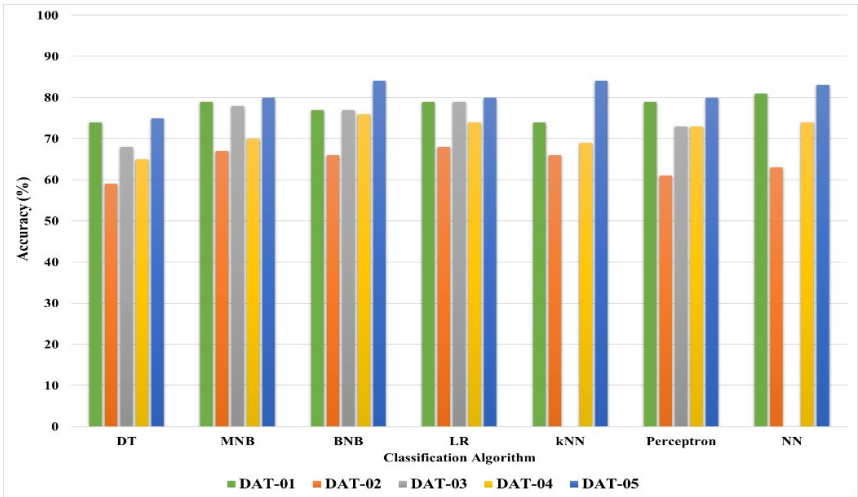


Figure 4.1 Accuracies of Various Classifiers Using the Proposed Classification Framework on Sentiment Datasets.

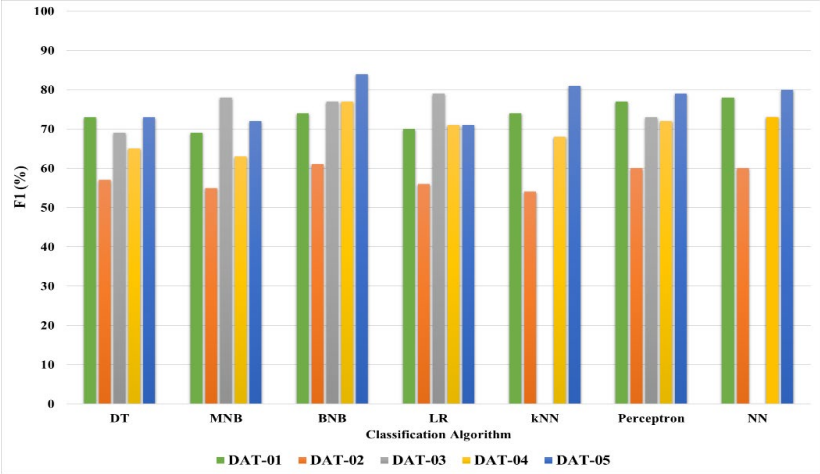


Figure 4.2 F1-Score of Various Classifiers Using the Proposed Classification Framework on Sentiment Datasets.

4.3.1.2 Misleading Information Detection

To test the proposed classification framework in classifying misleading information, we conducted a two-part experiment on the datasets shown in *Subsubsection 4.1.2*, as follows:

1- Part 1:

This part aims to test the effectiveness of the proposed feature selection technique, compared to the traditional TF-IDF feature extraction technique. The performance of eight classification

algorithms (DT, kNN, LR, MNB, BNB, Perceptron, LSVM, and NN) are analyzed. News documents, from each dataset, were randomly split into 80% for training and 20% for testing. All news documents for training and testing pass through the stages as discussed in *Section 3.1*. Validation results reported in this section are based on ACC, P, R, and F1 using the equations shown in Table 2.4. 5-folds cross-validation was used for the evaluation and TF-IDF was used as a feature extraction method. Table 4.3 and Table 4.4 show the classification results in detail compared with the results from [120] and [116] on DAT-06, and DAT-07 respectively. Table 4.5 shows the obtained results when applying our proposed model on DAT-08 that was recently published in [227].

Table 4.3 Accuracy Comparison of Our Approach Using the Traditional TF-IDF on DAT-06.

Algorithm / n-gram		DT	kNN	LR	SVM	BNB	MNB	LSVM	Perceptron	NN
Results from [116]	N = 1	89	83	89	86	N/A	N/A	92	N/A	N/A
	N = 2	85	68	88	78	N/A	N/A	89	N/A	N/A
	N = 3	87	73	88	71	N/A	N/A	87	N/A	N/A
	N = 4	74	69	81	55	N/A	N/A	81	N/A	N/A
Our Results	N = 1	100	88	98	99	97	95	100	99	99
	N = 2	96	89	98	99	98	97	99	98	99
	N = 3	91	88	98	98	98	98	99	97	99
	N = 4	87	52	95	95	94	96	95	93	96

Table 4.4 Performance Comparison of our Approach Using the Traditional TF-IDF on DAT-07.

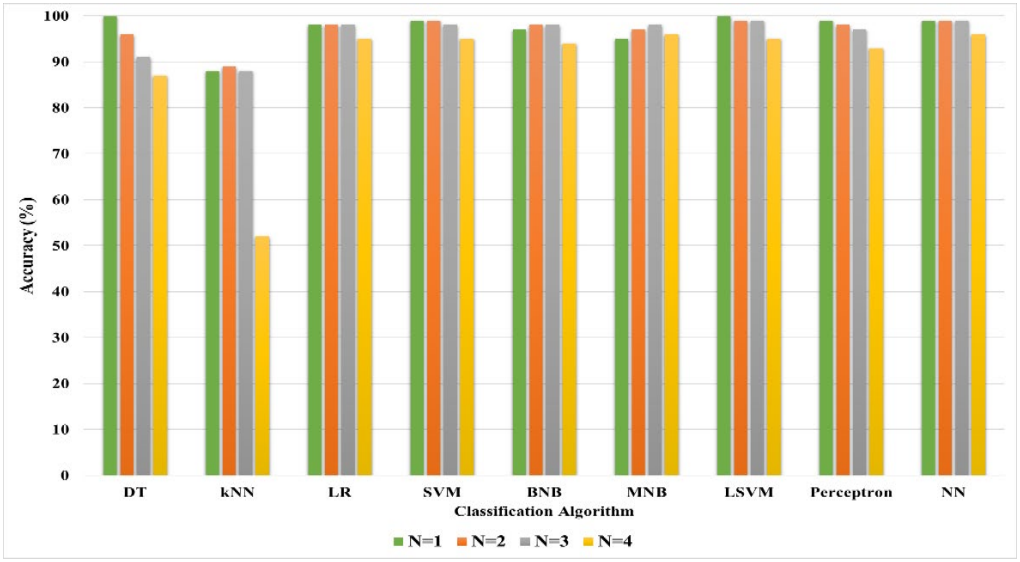
Algorithm / Metric (%)		DT	kNN	LR	SVM	Naïve Bayes with n-gram		LSVM	Perceptron	NN
						N = 1	N = 2			
Results from [120]	ACC	51	53	56	56	60	60	N/A	N/A	N/A
	P	51	53	56	57	60	59	N/A	N/A	N/A
	R	51	53	56	56	60	60	N/A	N/A	N/A
	F1	51	53	51	48	57	59	N/A	N/A	N/A
Our Results	ACC	55	58	62	62	62	62	60	59	58
	P	56	58	62	62	62	61	60	59	58
	R	56	58	62	62	62	62	60	59	59
	F1	56	58	61	62	62	61	60	59	59

Table 4.5 Obtained Results Using the Traditional TF-IDF on DAT-08.

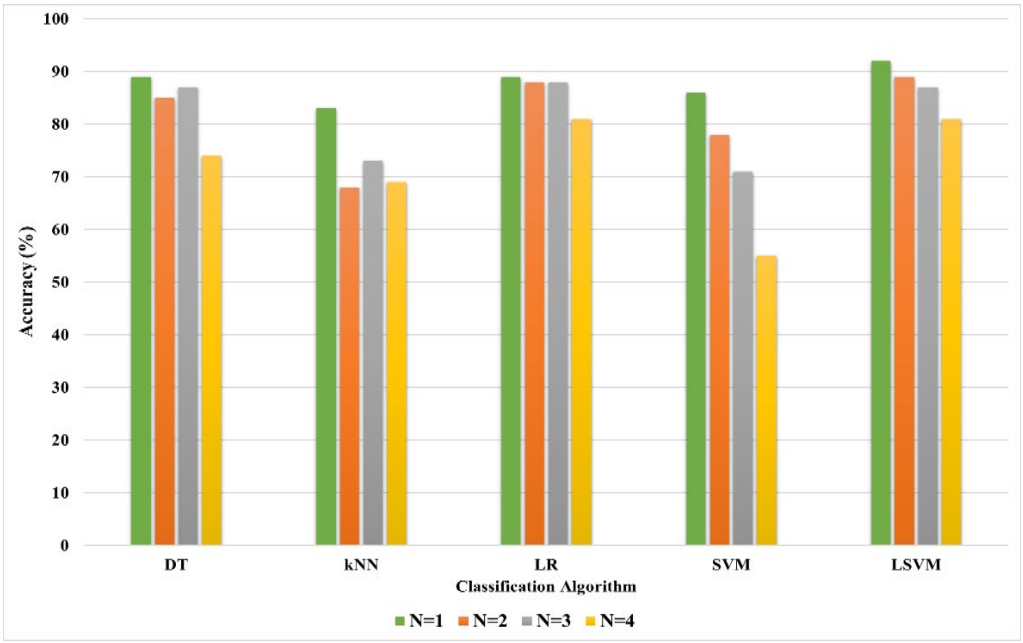
Algorithm / Metric (%)	DT	kNN	LR	SVM	BNB	MNB	LSVM	Perceptron	NN
ACC	51.07	51.14	52.13	50.14	45.15	58.09	57.24	52.08	54.08
P	50	50	50	48	45	63	57	51	54
R	50	51	52	50	45	58	57	52	55
F1	50	50	47	48	45	50	56	51	54

From the obtained results, our proposed model together with our novel hybrid feature selection technique compares favorably with other reviewed models.

As seen in Figure 4.3, the obtained enhancement in ACC results was between 4% to 40% when employing our proposed classification framework on DAT-06.



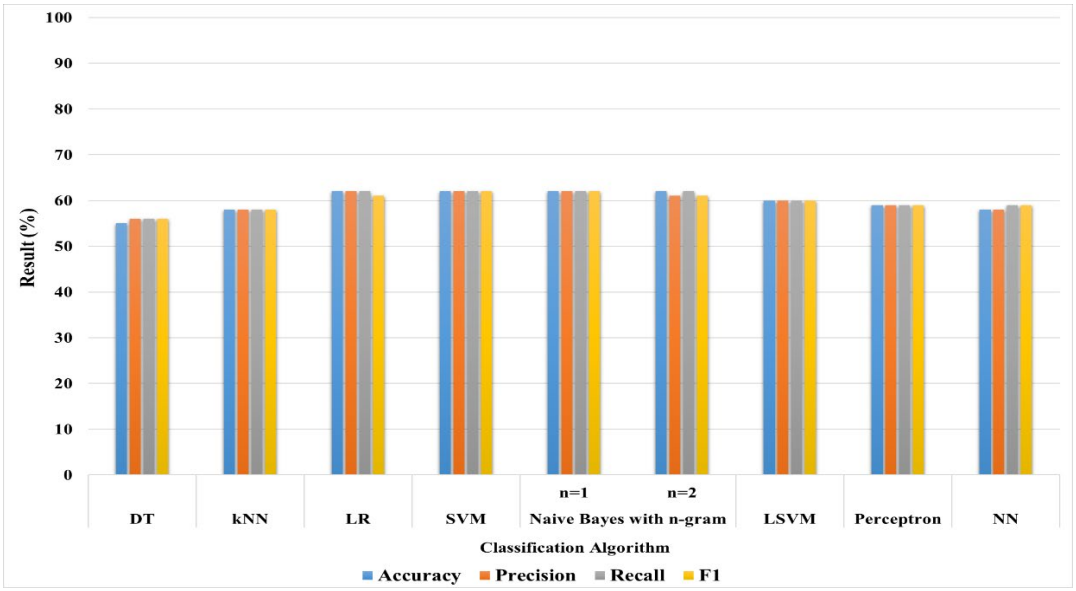
(a)



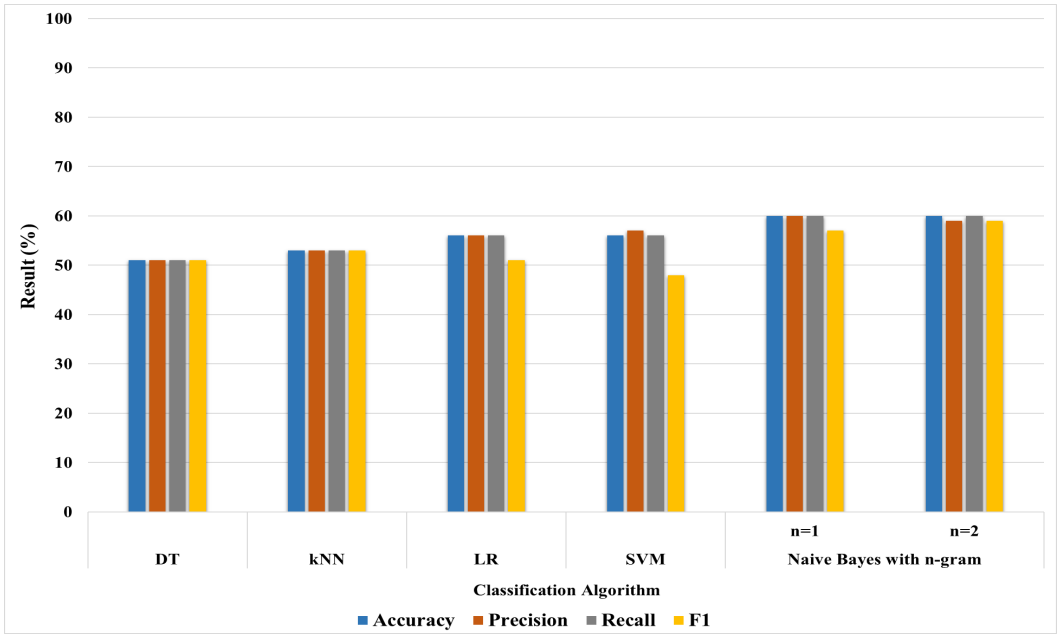
(b)

Figure 4.3 Accuracy of Various Classifiers Using the Proposed Classification Framework Results on DAT-06: (a) Our Performance Results, (b) Results from [116]

As seen in Figure 4.4, the best enhancement was almost 6% with accuracy results of 62% when applying both SVM and logistic regression classifiers on the DAT-07 dataset.



(a)



(b)

Figure 4.4 Performance Results of Various Classifiers Using the Proposed Classification Framework Results on DAT-07: (a) Our Performance Results, (b) Results from [120]

Finally, with DAT-08, the best accuracy is almost 58% when using the Multinomial Naïve Bayes Classifier. It could be noticed that performing proper preparation, cleaning, and feature selection have a positive impact on the performance of the detection process.

2- Part 2:

This part aims to test the effectiveness of the proposed feature selection technique when using the proposed novel feature extraction technique (TCI). The performance of eight classification algorithms (DT, kNN, LR, MNB, BNB, Perceptron, LSVM, and NN) are analyzed on two news

datasets, DAT-07 and DAT-08, that were described in *Subsection 4.1.2*. For the evaluation process, we used a 5-fold cross-validation, and each dataset was randomly split into two sets (80% of documents as a training set, and the rest is the testing set). Table 4.6 shows the obtained ACC, P, R, and F1 results when using our proposed TCI technique compared to the TF-IDF techniques introduced in [109] and [120].

Table 4.6 Performance Comparison Between the TCI Approach and the Traditional TF-IDF.

Classification Algorithm		Metric (%)	DAT-07			DAT-08	
			Results from [120]	Results from [109]	Our results	Results from [109]	Our results
Decision Tree		ACC	51	55	99.01	51.07	99.05
		P	51	56	99	50	98
		R	51	56	99	50	98
		F1	51	56	99	50	98
kNN		ACC	53	58	62.02	51.14	52.11
		P	53	58	74	50	55
		R	53	58	62	51	52
		F1	53	58	67	50	53
Logistic Regression		ACC	56	62	64.01	52.13	60.1
		P	56	62	64	50	60
		R	56	62	64	52	60
		F1	51	61	64	47	60
Linear SVM		ACC	56	62	66.06	57.24	63.13
		P	57	62	66	57	63
		R	56	62	66	57	63
		F1	48	62	66	56	63
Naïve Bayes	Multinomial	ACC	60	62	69.05	58.09	61
		P	60	62	69	63	61
		R	60	62	69	58	61
		F1	57	62	69	50	61
	Bernoulli	ACC	N/A	N/A	60.02	45.15	55.03
		P	N/A	N/A	61	45	55
		R	N/A	N/A	60	45	55
		F1	N/A	N/A	60	45	55
Perceptron		ACC	N/A	59	62.04	52.08	56.12
		P	N/A	59	63	51	57
		R	N/A	59	62	52	57
		F1	N/A	59	62	51	57
Neural Networks		ACC	N/A	58	67.04	54.08	60.11
		P	N/A	58	66	54	53
		R	N/A	59	66	55	53
		F1	N/A	59	66	54	53

From Table 4.6, we could notice the effectiveness of our model based on the TCI feature extraction technique, compared to the TF-IDF features extraction-based models proposed in [109] and [120], for all the tested classifiers. Moreover, the highest obtained accuracy results are 99.01% and 95.05% when using the DT classifier with our proposed technique, with a marked improvement of 44% and 48% on DAT-07 and DAT-08, respectively.

Hence, we could conclude that our proposed TCI extracted features are more representative and discriminative than TF-IDF extracted features. TCI enhanced the performance of the detection system, as shown in the eight machine learning algorithms on DAT-07 and DAT-08 datasets. Moreover, as the process of detecting whether a news document is fake or not is a binary classification problem, the TCI could facilitate the building of any binary classification-based models, such as sentiment analysis applications.

In addition, the obtained results of the DT classifier show the effectiveness of our extracted features and their ability to discriminate between different classes. It leads to the building of a well-discriminative tree that avoids overfitting problems and has an outstanding ability to distinguish between different classes.

4.3.2 Performance Evaluation Framework

We utilized 7 of the most commonly used classification techniques (Decision Tree (DT), Multinomial Naïve Bayes (MNB), Bernoulli Naïve Bayes (BNB), Logistic Regression (LR), k-Nearest Neighbors (kNN), Perceptron, and Multilayer Perceptron (MP)), and used the15 performance metrics as described in *Section 4.2*, to evaluate the performance of each classification technique on each of the datasets used. Table 4.7 shows the classification results, using the 15 performance metrics in addition to the MCAS on the sentiment analysis datasets introduced in *Subsection 4.1.1*. Table 4.8 shows the classification results when testing the performance of classification algorithms on the misleading information datasets introduced in *Subsection 4.1.2*. It should be noted that we consider both classes have the same importance when testing the performance of classification algorithms. Moreover, the highlighted values in these tables are the best-obtained results for each performance evaluation metric.

From the results presented in Table 4.7 and Table 4.8, some observations can be made. In general, as expected, ACC and F1 give a misleading indication about the performance of some classifiers, e.g., for DAT-01 and DAT-11 datasets with the MP classifier. As for DAT-01, the best values for ACC and F1 were 91.30% and 95.32%, respectively. While for DAT-02, the best values for ACC and F1 were 68.37% and 77.37%, respectively. After performing an in-depth analysis, we notice that for DAT-01 and DAT-02, the DT and BNB classifiers give the best GM and AUC results.

Table 4.7 Validation Results for Sentiment Analysis Datasets.

Dataset	Metric / Classifier	ACC	P	R	F1	TPR	TNR	FNR	FPR	PPV	NPV	FDR	FOR	GM	AUC	MCC	MCAS
DAT-01	DT	88.85	**93.86**	93.66	93.76	93.66	**48.21**	06.34	**51.79**	**93.86**	47.37	**06.14**	52.63	**67.20**	**70.94**	**41.55**	**53.92**
	MNB	89.41	89.41	**100.00**	94.41	**100.00**	00.00	**00.00**	100.00	89.41	00.00	10.59	100.00	00.00	50.00	00.00	39.41
	BNB	88.47	92.56	94.71	93.62	94.71	35.71	05.29	64.29	92.56	44.44	07.44	55.56	58.16	65.21	33.56	50.25
	LR	89.41	89.41	**100.00**	94.41	**100.00**	00.00	**00.00**	100.00	89.41	00.00	10.59	100.00	00.00	50.00	00.00	39.41
	kNN	90.74	91.57	98.73	95.02	98.73	23.21	01.27	76.79	91.57	68.42	08.43	31.58	47.87	60.97	36.28	51.01
	Perceptron	89.41	92.81	95.56	94.16	95.56	37.50	04.44	62.50	92.81	50.00	07.19	50.00	59.86	66.53	37.62	52.55
	MP	**91.30**	91.78	99.15	**95.32**	99.15	25.00	00.85	75.00	91.78	**77.78**	08.22	**22.22**	49.79	62.08	40.99	52.79
DAT-02	DT	75.92	84.94	86.17	85.55	86.17	26.88	13.83	73.12	84.94	28.88	15.06	71.12	48.13	56.53	13.43	31.78
	MNB	**82.82**	82.80	**100.00**	**90.59**	**100.00**	00.58	**00.00**	99.42	82.80	**100.00**	17.20	**00.00**	07.62	50.29	06.92	33.10
	BNB	81.62	**86.06**	92.81	89.31	92.81	28.03	07.19	71.97	**86.06**	44.91	**13.94**	55.09	51.00	**60.42**	**25.41**	**40.77**
	LR	82.77	82.79	99.94	90.56	99.94	00.58	00.06	99.42	82.79	66.67	17.21	33.33	07.61	50.26	05.06	33.04
	kNN	82.72	82.72	**100.00**	90.54	**100.00**	00.00	**00.00**	100.00	82.72	00.00	17.28	100.00	00.00	50.00	00.00	32.72
	Perceptron	75.82	85.52	85.21	85.36	85.21	**30.92**	14.79	**69.08**	85.52	30.40	14.48	69.60	**51.33**	58.07	16.02	32.53
	MP	79.17	85.54	90.04	87.73	90.04	27.17	09.96	72.83	85.54	36.29	14.46	63.71	49.46	58.61	19.38	36.70
DAT-03	DT	70.32	70.21	70.66	70.43	70.66	69.98	29.34	30.02	70.21	70.43	29.79	29.57	70.32	70.32	40.64	36.80
	MNB	76.36	77.35	74.59	75.94	74.59	**78.13**	25.41	**21.87**	77.35	75.43	22.65	24.57	76.34	76.36	52.75	48.35
	BNB	77.00	77.15	76.76	76.95	76.76	77.23	23.24	22.77	77.15	76.84	22.85	23.16	76.99	77.00	53.99	49.60
	LR	**78.84**	**78.07**	**80.27**	**79.15**	**80.27**	77.42	**19.73**	22.58	**78.07**	**79.67**	**21.93**	**20.33**	**78.83**	**78.85**	**57.71**	**53.25**
	kNN	59.48	58.34	66.47	62.14	66.47	52.48	33.53	47.52	58.34	60.98	41.66	39.02	59.06	59.48	19.13	16.91
	Perceptron	71.08	71.24	70.76	71.00	70.76	71.40	29.24	28.60	71.24	70.92	28.76	29.08	71.08	71.08	42.16	38.23
	MP	73.89	72.78	76.38	74.54	76.38	71.40	23.62	28.60	72.78	75.12	27.22	24.88	73.85	73.89	47.84	43.58
DAT-04	DT	69.71	60.88	64.00	62.40	64.00	73.40	36.00	26.60	60.88	75.92	39.12	24.08	68.54	68.70	37.09	34.60
	MNB	77.66	**86.80**	50.87	64.15	50.87	**94.99**	49.13	**05.01**	**86.80**	74.93	**13.20**	25.07	69.51	72.93	53.21	47.82
	BNB	**81.52**	75.96	**77.48**	**76.71**	**77.48**	84.14	**22.52**	15.86	75.96	**85.24**	24.04	**14.76**	**80.74**	**80.81**	**61.41**	**57.65**
	LR	80.67	82.81	64.09	72.26	64.09	91.39	35.91	08.61	82.81	79.74	17.19	20.26	76.53	77.74	58.91	54.88
	kNN	74.39	66.08	71.48	68.67	71.48	76.27	28.52	23.73	66.08	80.52	33.92	19.48	73.84	73.88	47.17	43.68
	Perceptron	76.23	69.91	69.30	69.60	69.30	80.71	30.70	19.29	69.91	80.26	30.09	19.74	74.79	75.01	50.09	46.87
	MP	76.57	70.86	68.52	69.67	68.52	81.78	31.48	18.22	70.86	80.07	29.14	19.93	74.86	75.15	50.61	47.44
DAT-05	DT	75.46	38.00	35.19	36.54	35.19	85.58	64.81	14.42	38.00	84.02	62.00	15.98	54.88	60.39	21.38	34.00
	MNB	79.93	00.00	00.00	00.00	00.00	**100.00**	100.00	**00.00**	00.00	79.93	100.00	20.07	00.00	50.00	00.00	29.93
	BNB	80.67	56.25	16.67	25.72	16.67	96.74	83.33	03.26	56.25	82.21	43.75	17.79	40.16	56.71	22.71	37.23
	LR	79.93	00.00	00.00	00.00	00.00	**100.00**	100.00	**00.00**	00.00	79.93	100.00	20.07	00.00	50.00	00.00	29.93
	kNN	80.67	56.25	16.67	25.72	16.67	96.74	83.33	03.26	56.25	82.21	43.75	17.79	40.16	56.71	22.71	37.23
	Perceptron	79.18	47.62	37.04	41.67	37.04	89.77	62.96	10.23	47.62	85.02	52.38	14.98	57.66	63.41	29.58	40.35
	MP	**84.76**	**70.97**	**40.74**	**51.76**	**40.74**	95.81	**59.26**	04.19	**70.97**	**86.55**	**29.03**	**13.45**	**62.48**	**68.28**	**45.86**	**51.11**

Table 4.8 Validation Results for Misleading Information Datasets.

Dataset	Metric / Classifier	ACC	P	R	F1	TPR	TNR	FNR	FPR	PPV	NPV	FDR	FOR	GM	AUC	MCC	MCAS
DAT-06	DT	**99.62**	**99.66**	**99.56**	**99.61**	**99.56**	**99.68**	**00.44**	**00.32**	**99.66**	**99.59**	**00.34**	**00.41**	**99.62**	**99.62**	**99.24**	**99.06**
	MNB	94.60	94.32	94.56	94.44	94.56	94.64	05.44	05.36	94.32	94.87	05.68	05.13	94.60	94.60	89.20	86.98
	BNB	97.25	96.32	98.07	97.19	98.07	96.48	01.93	03.52	96.32	98.15	03.68	01.85	97.27	97.28	94.52	93.26
	LR	98.45	98.17	98.65	98.41	98.65	98.27	01.35	01.73	98.17	98.72	01.83	01.28	98.46	98.46	96.90	96.17
	kNN	87.51	86.62	87.81	87.21	87.81	87.24	12.19	12.76	86.62	88.38	13.38	11.62	87.52	87.53	75.02	71.13
	Perceptron	98.78	98.45	99.04	98.74	99.04	98.53	00.96	01.47	98.45	99.09	01.55	00.91	98.78	98.79	97.55	96.96
	MP	98.83	98.43	99.17	98.80	99.17	98.51	00.83	01.49	98.43	99.22	01.57	00.78	98.84	98.84	97.66	97.10
DAT-07	DT	55.00	59.66	59.12	59.39	59.12	49.82	40.88	50.18	59.66	49.26	40.34	50.74	54.27	54.47	08.93	08.55
	MNB	60.30	60.35	**83.59**	**70.09**	**83.59**	31.07	**16.41**	68.93	60.35	**60.14**	39.65	**39.86**	50.96	57.33	17.33	16.47
	BNB	**62.06**	**65.72**	66.55	66.13	66.55	**56.43**	33.45	**43.57**	**65.72**	57.33	**34.28**	42.67	**61.28**	**61.49**	**23.02**	**21.16**
	LR	**62.06**	63.51	74.82	68.70	74.82	46.04	25.18	53.96	63.51	59.30	36.49	40.70	58.69	60.43	21.81	20.53
	kNN	58.12	61.54	65.99	63.69	65.99	48.24	34.01	51.76	61.54	53.05	38.46	46.95	56.42	57.12	14.41	13.82
	Perceptron	57.46	62.41	59.26	60.79	59.26	55.19	40.74	44.81	62.41	51.90	37.59	48.10	57.19	57.23	14.38	13.10
	MP	59.09	63.00	64.24	63.61	64.24	52.64	35.76	47.36	63.00	53.97	37.00	46.03	58.15	58.44	16.92	15.78
DAT-08	DT	52.80	56.18	57.47	56.82	57.47	47.30	42.53	52.70	56.18	48.61	43.82	51.39	52.14	52.39	04.78	04.72
	MNB	59.01	57.45	**93.10**	**71.05**	**93.10**	18.92	**06.90**	81.08	57.45	**70.00**	42.55	**30.00**	41.97	56.01	18.17	13.98
	BNB	53.42	56.12	63.22	59.46	63.22	41.89	36.78	58.11	56.12	49.21	43.88	50.79	51.46	52.56	05.22	05.55
	LR	59.01	58.27	85.06	69.16	85.06	28.38	14.94	71.62	58.27	61.76	41.73	38.24	49.13	56.72	16.41	14.57
	kNN	59.63	60.19	74.71	66.67	74.71	41.89	25.29	58.11	60.19	58.49	39.81	41.51	55.94	58.30	17.61	16.36
	Perceptron	49.07	53.16	48.28	50.60	48.28	**50.00**	51.72	**50.00**	53.16	45.12	46.84	54.88	49.13	49.14	-01.72	-01.61
	MP	**60.25**	**61.17**	72.41	66.32	72.41	45.95	27.59	54.05	**61.17**	58.62	**38.83**	41.38	**57.68**	**59.18**	**19.06**	**17.63**
DAT-09	DT	71.48	79.82	82.35	81.07	82.35	**40.26**	17.65	**59.74**	79.82	44.29	20.18	55.71	**57.58**	61.31	23.35	30.83
	MNB	73.83	74.24	99.10	84.89	99.10	01.30	00.90	98.70	74.24	33.33	25.76	66.67	11.35	50.20	01.73	24.21
	BNB	69.46	76.00	85.97	80.68	85.97	22.08	14.03	77.92	76.00	35.42	24.00	64.58	43.57	54.03	09.59	23.93
	LR	74.16	74.16	**100.00**	**85.16**	**100.00**	00.00	**00.00**	100.00	74.16	00.00	25.84	100.00	00.00	50.00	00.00	24.16
	kNN	69.46	75.79	86.43	80.76	86.43	20.78	13.57	79.22	75.79	34.78	24.21	65.22	42.38	53.61	08.73	23.63
	Perceptron	69.80	78.11	82.35	80.17	82.35	33.77	17.65	66.23	78.11	40.00	21.89	60.00	52.73	58.06	17.09	26.92
	MP	**75.17**	**80.25**	88.24	84.06	88.24	37.66	11.76	62.34	**80.25**	**52.73**	**19.75**	**47.27**	**57.65**	**62.95**	**29.22**	**36.30**
DAT-10	DT	65.85	81.74	72.87	77.05	72.87	**40.00**	27.13	**60.00**	81.74	28.57	18.26	71.43	53.99	56.44	11.52	20.96
	MNB	78.66	78.66	**100.00**	88.06	**100.00**	00.00	**00.00**	100.00	78.66	00.00	21.34	100.00	00.00	50.00	00.00	28.66
	BNB	73.17	79.31	89.15	83.94	89.15	14.29	10.85	85.71	79.31	26.32	20.69	73.68	35.69	51.72	04.39	26.04
	LR	78.66	78.66	**100.00**	88.06	**100.00**	00.00	**00.00**	100.00	78.66	00.00	21.34	100.00	00.00	50.00	00.00	28.66
	kNN	74.39	80.42	89.15	84.56	89.15	20.00	10.85	80.00	80.42	33.33	19.58	66.67	42.23	54.58	11.22	29.29
	Perceptron	71.95	81.68	82.95	82.31	82.95	31.43	17.05	68.57	81.68	33.33	18.32	66.67	51.06	57.19	14.69	28.31
	MP	**81.71**	**83.67**	95.35	**89.13**	95.35	31.43	04.65	68.57	**83.67**	**64.71**	**16.33**	**35.29**	**54.74**	**63.39**	**35.99**	**43.87**
DAT-11	DT	61.22	67.80	67.80	67.80	67.80	51.28	32.20	48.72	67.80	51.28	32.20	48.72	58.96	59.54	19.08	18.83
	MNB	60.20	60.20	**100.00**	75.16	**100.00**	00.00	**00.00**	100.00	60.20	00.00	39.80	100.00	00.00	50.00	00.00	10.20
	BNB	**68.37**	**85.00**	57.63	68.69	57.63	**84.62**	42.37	**15.38**	**85.00**	56.90	**15.00**	43.10	**69.83**	**71.13**	**42.07**	**34.24**
	LR	61.22	60.82	**100.00**	75.64	**100.00**	02.56	**00.00**	97.44	60.82	**100.00**	39.18	**00.00**	16.00	51.28	12.49	12.31
	kNN	62.24	64.86	81.36	72.18	81.36	33.33	18.64	66.67	64.86	54.17	35.14	45.83	52.07	57.35	16.72	18.65
	Perceptron	63.27	67.69	74.58	70.97	74.58	46.15	25.42	53.85	67.69	54.55	32.31	45.45	58.67	60.37	21.47	21.78
	MP	**68.37**	67.95	89.83	**77.37**	89.83	35.90	10.17	64.10	67.95	70.00	32.05	30.00	56.79	62.87	31.25	29.41

In general, these two metrics were recommended by many studies to act as good discriminators and to be utilized in comparing the performance of different classification algorithms for binary classification problems [253]. Moreover, we notice that MCAS most often chooses the classification algorithms with the highest GM, AUC, and MCC. The rest of the measures are usually affected by high TNR and NPV, thus undervaluing the relevance of TPR and PPV.

Consequently, to assess the behavior of the used classification algorithms with respect to each of the positive and negative classes, we calculated the class-based values of the MCAS, as shown in Table 4.9.

Table 4.9 Class-Based Validation Results.

Metric	Dataset / Classifier	Sentiment analysis datasets					Misleading information datasets					
		DAT-01	DAT-02	DAT-03	DAT-04	DAT-05	DAT-06	DAT-07	DAT-08	DAT-09	DAT-10	DAT-11
$MCAS_{(+ve)}$	DT	82.34	61.07	36.94	27.51	08.38	**99.03**	13.20	08.79	51.55	42.29	27.23
	MNB	84.12	**74.21**	47.82	35.38	-10.04	86.69	**30.57**	**32.78**	60.45	67.99	40.31
	BNB	81.91	70.68	49.54	**52.04**	04.60	93.14	25.99	12.08	49.87	57.10	34.61
	LR	84.12	74.12	**53.67**	46.08	-10.04	96.09	29.21	28.93	**61.24**	67.99	41.44
	kNN	85.78	74.08	19.81	37.63	04.60	70.66	20.28	25.21	50.02	58.77	33.89
	Perceptron	83.41	60.71	38.14	39.89	14.75	96.90	16.67	-00.22	49.17	53.63	32.61
	MP	**86.66**	66.57	44.42	40.20	**26.87**	97.04	20.93	25.09	58.47	**70.65**	**45.40**
$MCAS_{(-ve)}$	DT	**25.49**	02.49	36.66	41.70	59.62	**99.08**	03.89	00.64	10.11	-00.38	10.43
	MNB	-05.29	-08.01	48.88	60.25	69.89	87.26	02.37	-04.83	-12.02	-10.67	-19.90
	BNB	18.58	**10.85**	49.66	63.26	69.85	93.37	**16.33**	-00.98	-02.01	-05.02	**33.87**
	LR	-05.29	-08.05	**52.82**	**63.68**	69.89	96.26	11.86	00.21	-12.92	-10.67	-16.82
	kNN	16.24	-08.64	14.02	49.73	69.85	71.60	07.37	07.50	-02.76	-00.19	03.42
	Perceptron	21.70	04.36	38.33	53.85	65.95	97.03	09.52	-03.00	04.67	02.99	10.94
	MP	18.92	06.82	42.75	54.68	**75.35**	97.16	10.62	**10.18**	**14.13**	**17.09**	13.41

Focusing on each particular dataset (from DAT-01 to DAT-11), based on the obtained results shown in Table 4.8, and Table 4.9, some comments can be noted as follows.

1- ***For DAT-01:*** the MP classifier gives the best ACC and F1 of 91.30% and 95.32%, respectively. However, MCAS suggests DT because its performances in both classes are more balanced. In terms of the classifiers' ability to classify positive class instances, MCAS suggests the MP classifier despite that both MNB and LR achieve 100% of TPR. The reason is that the MP has higher PPV and lower FPR. On the other hand, in terms of the classifiers' ability to classify negative class instances, MCAS suggests the DT classifier because it has the highest TNR and the lowest FDR. Figure 4.5 shows the resulted confusion matrix corresponding to each of the used classifiers on DAT-01.

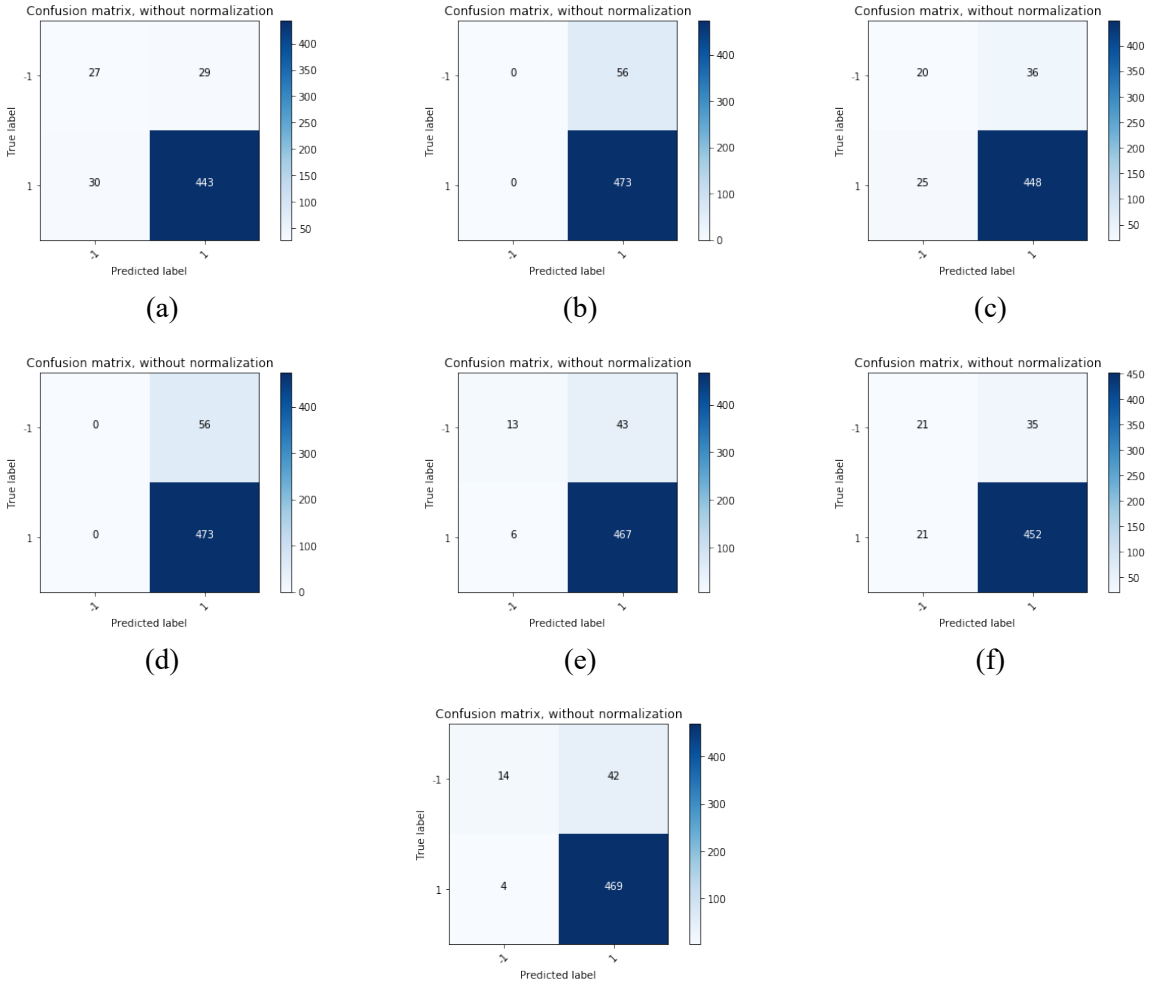


Figure 4.5 Classification Confusion Matrix on DAT-01. (a) DT, (b) MNB, (c) BNB, (d) LR, (e) kNN, (f) Perceptron, (g) MP.

2- For DAT-02: the MNB classifier gives the best ACC and F1 of 82.82% and 90.59%, respectively. However, the MCAS suggests the BNB classifier. In terms of the classifiers' ability to classify positive class instances, MCAS suggests the MNB classifier despite that both MNB and kNN achieve 100% of TPR. The reason is that the MNB has higher PPV and lower FPR. On the other hand, in terms of the classifiers' ability to classify negative class instances, MCAS suggests the BNB classifier despite its lower TNR than the Perceptron classifier. It has a higher NPV with lower FOR. Figure 4.6 shows the resulting confusion matrix corresponding to each of the seven classifiers on DAT-02.

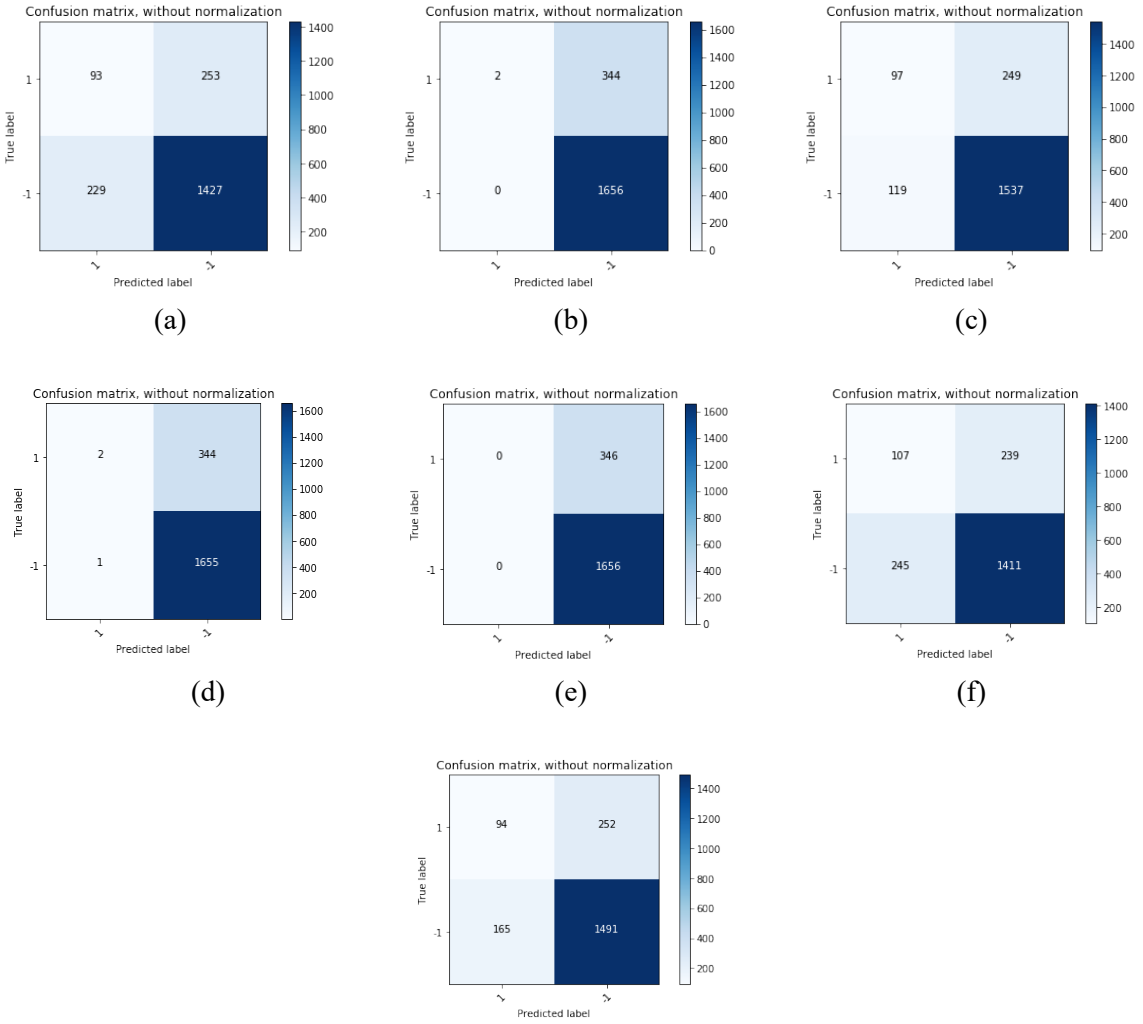


Figure 4.6 Classification Confusion Matrix on DAT-02. (a) DT, (b) MNB, (c) BNB, (d) LR, (e) kNN, (f) Perceptron, (g) MP.

3- ***For DAT-03:*** LR gives the best performance compared to other classification techniques in all metrics except for the TNR and FPR values. The GM's highest overall performance maximizes TPR and TNR while keeping both rates relatively balanced. Accordingly, MCAS suggests DT as the classifier which gives the best performance among the other classifiers when classifying the positive and negative instances. Figure 4.7 shows the resulting confusion matrix corresponding to each of the seven classifiers on DAT-03.

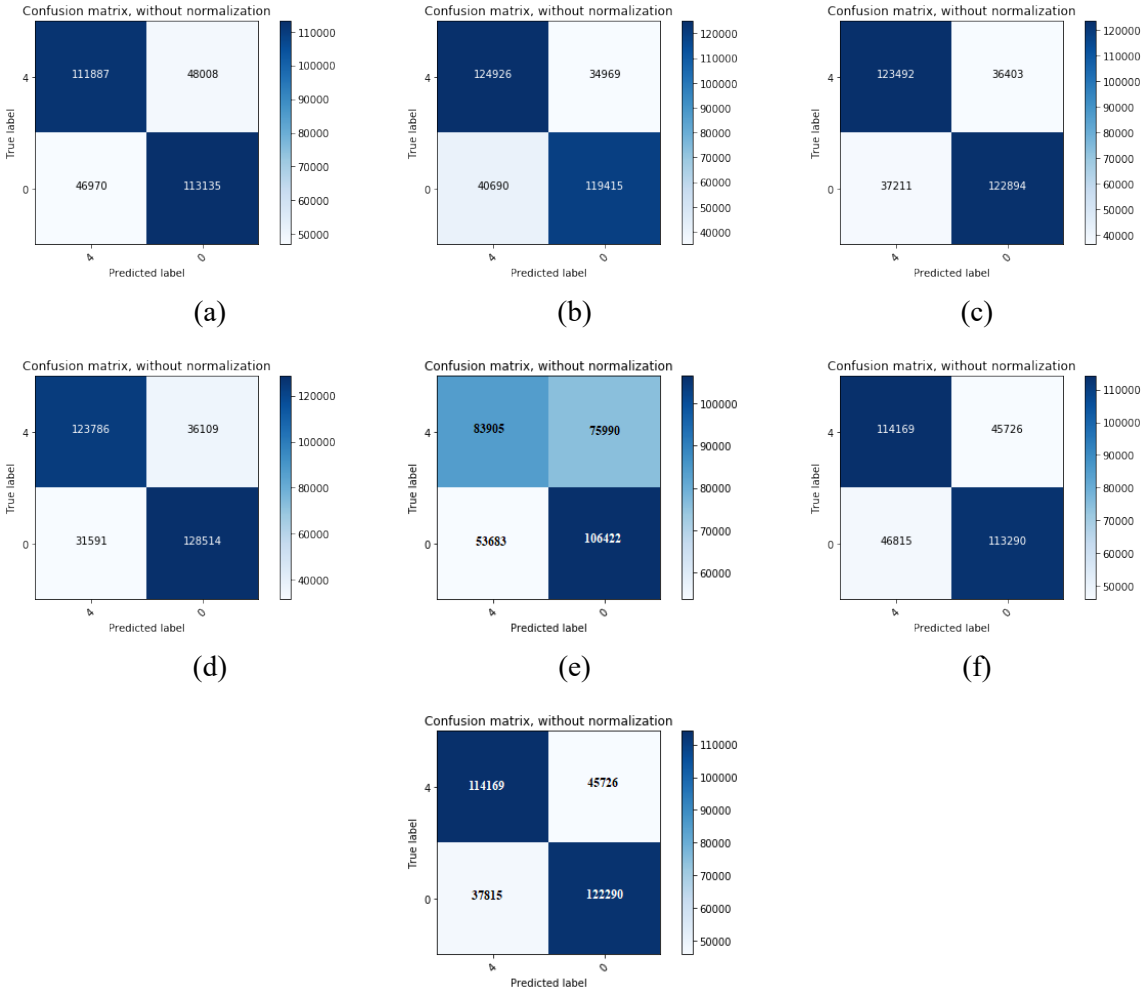


Figure 4.7 Classification Confusion Matrix on DAT-03. (a) DT, (b) MNB, (c) BNB, (d) LR, (e) kNN, (f) Perceptron, (g) MP.

4- For DAT-04: BNB gives the best ACC and F1 with values of 81.52% and 76.71%, respectively. It achieves the highest GM which maximizes TPR and TNR while keeping both rates relatively balanced. Moreover, it achieves the highest AUC. Accordingly, MCAS suggests BNB be used when both classes have the same level of importance or when the positive is more important. While if the negative class is more important, MCAS suggests LR as it achieves the highest TNR and NPV compared to other classification techniques. Figure 4.8 shows the resulting confusion matrix corresponding to each of the seven classifiers on DAT-04.

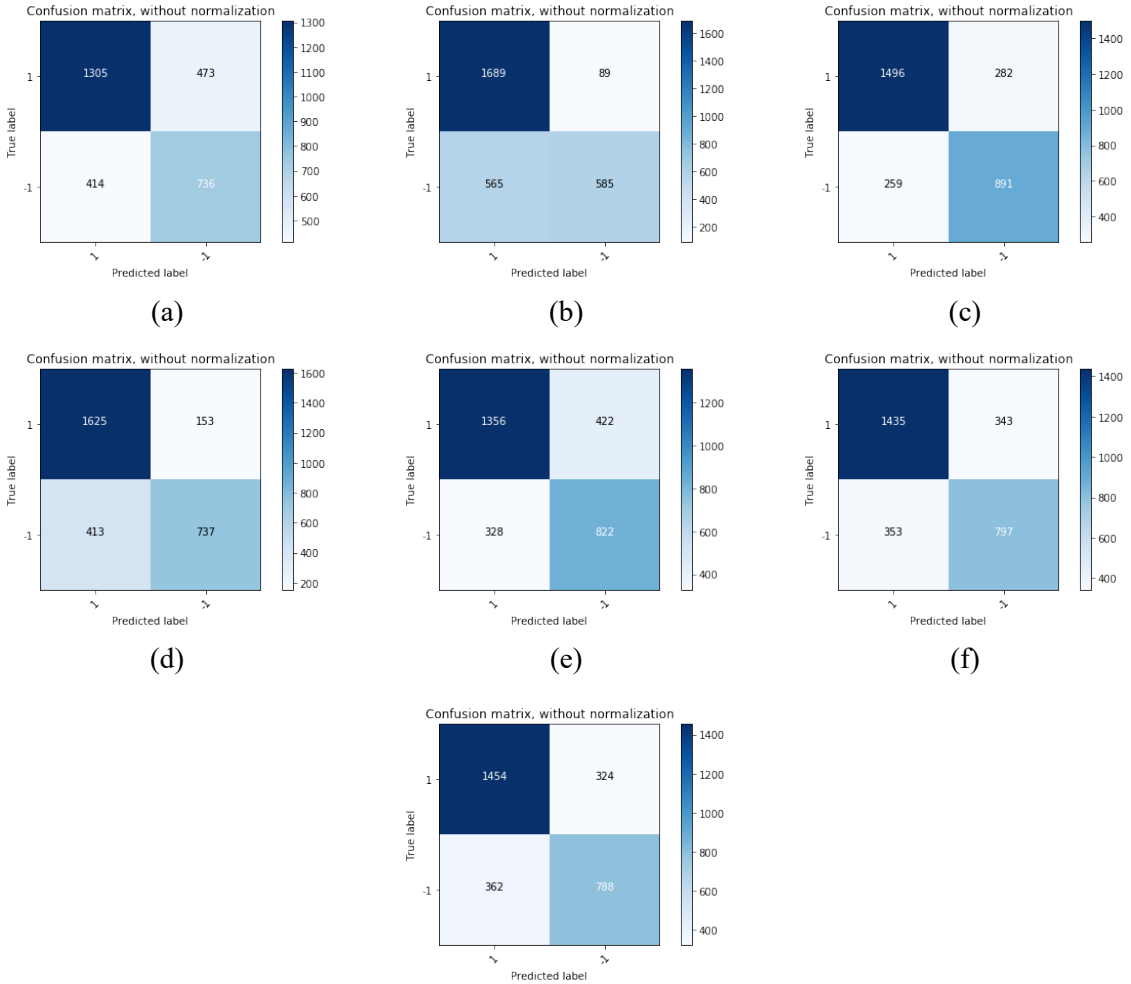


Figure 4.8 Classification Confusion Matrix on DAT-04. (a) DT, (b) MNB, (c) BNB, (d) LR, (e) kNN, (f) Perceptron, (g) MP.

5- ***For DAT-05:*** MP gives the best performance in all the measures except for TNR and FPR. On the other hand, it achieves the best NPV with the lowest FOR. Also, it achieves the highest GM of TPR and TNR. Accordingly, MCAS suggests MP be used when both classes have the same level of importance and when either the positive or the negative class is more important. Figure 4.9 shows the resulting confusion matrix corresponding to each of the seven classifiers on DAT-05.

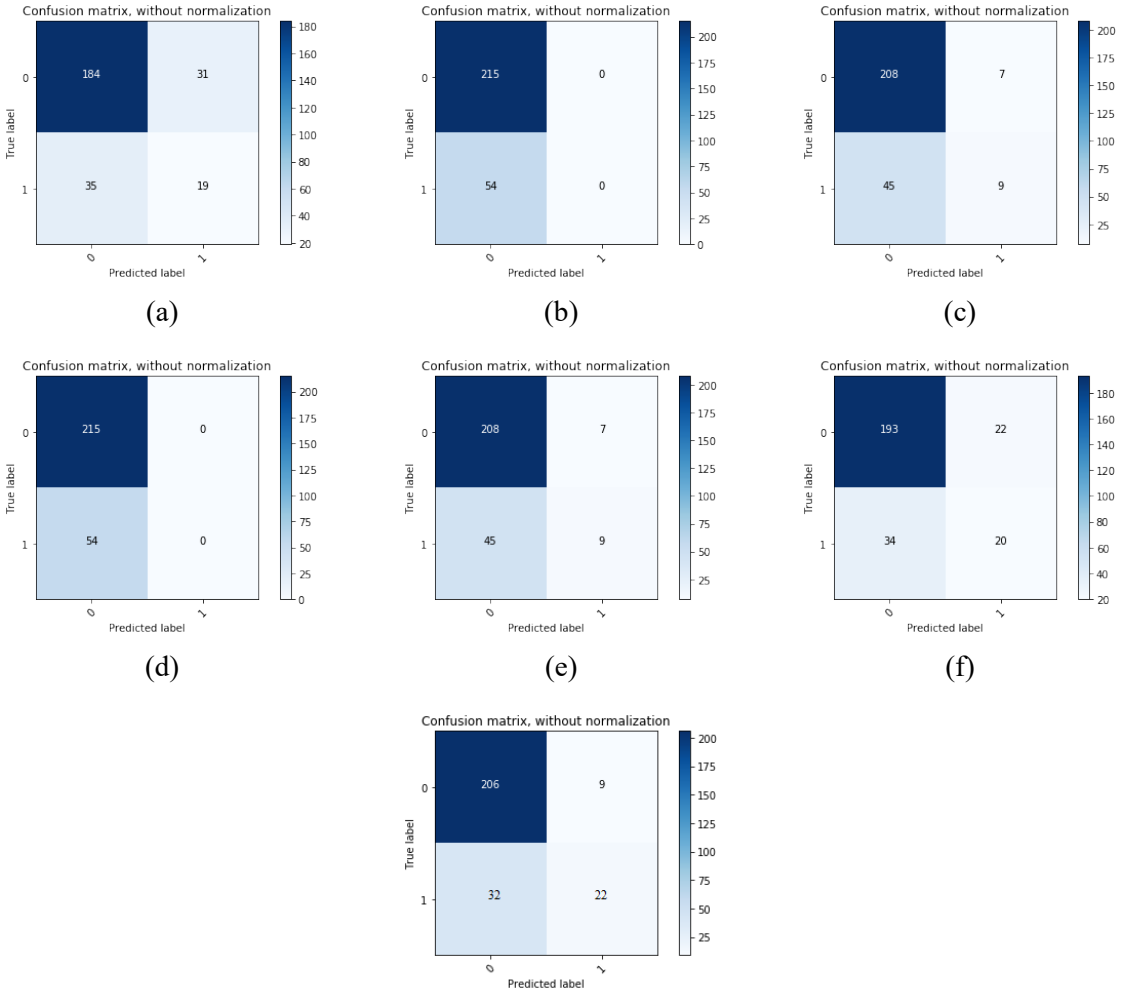


Figure 4.9 Classification Confusion Matrix on DAT-05. (a) DT, (b) MNB, (c) BNB, (d) LR, (e) kNN, (f) Perceptron, (g) MP.

6- ***For DAT-06:*** This is a straightforward case as all the metrics suggest DT being the best classifier. The reason for getting these extremely good results is the way DAT-06 is built. As described in *Subsection 4.1.2*, the fake news documents for DAT-06 were collected from both the *politifact.com* fact-checking website and *Wikipedia*, while the real news documents were collected from the Reuters website (regarded as the world's largest international multimedia news provider). The data from each source has its own writing style, especially since the collected news documents from Reuters are professionally written. Hence, after building the classification model, the classification problem shifted from being to classify between real and fake classes to classify whether the news documents were a Reuters document or not. Figure 4.10 shows the resulting confusion matrix corresponding to each of the seven classifiers on DAT-06.

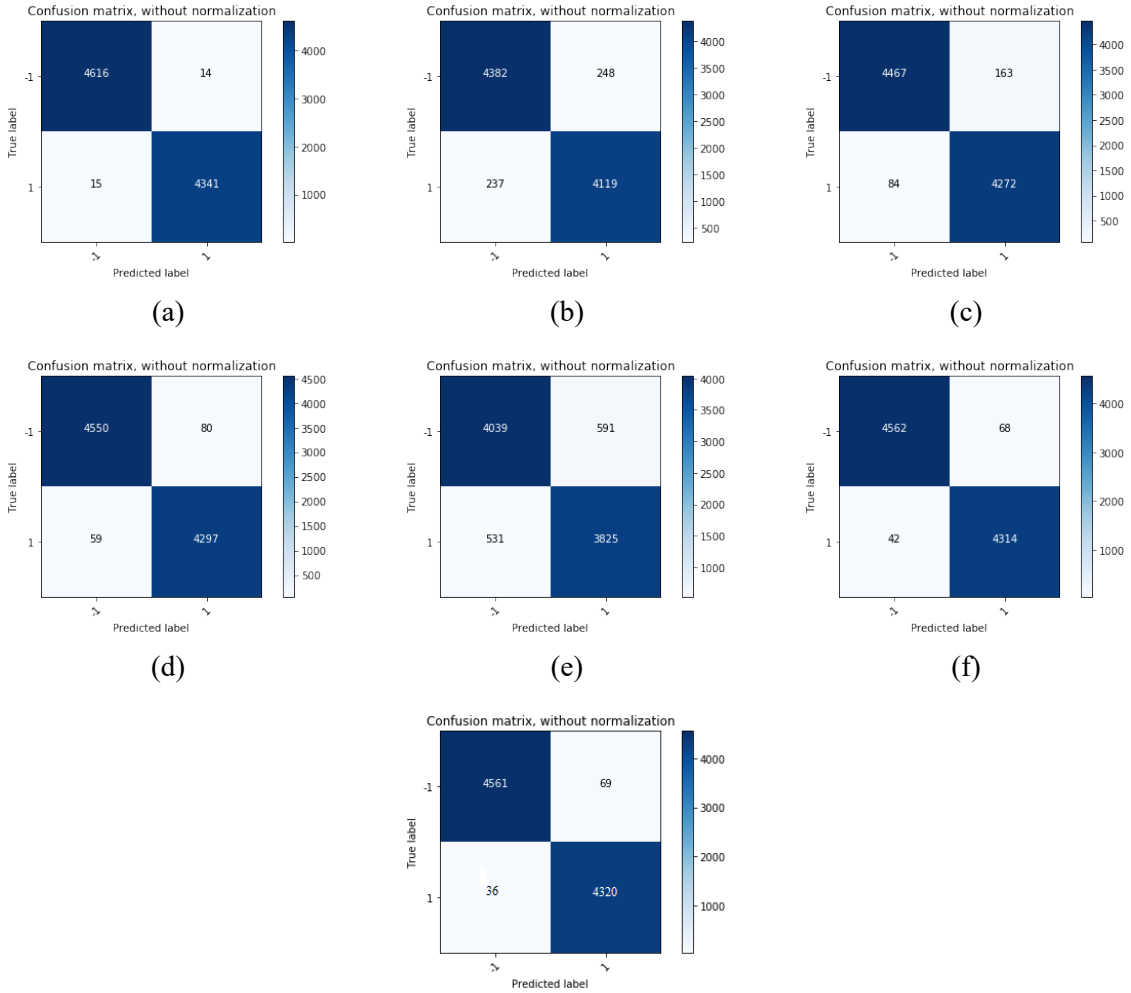


Figure 4.10 Classification Confusion Matrix on DAT-06. (a) DT, (b) MNB, (c) BNB, (d) LR, (e) kNN, (f) Perceptron, (g) MP.

7- ***For DAT-07:*** it is hard to decide which classifier is the best. On one hand, LR and BNB give the best ACC of 62.06%. On the other hand, MNB gives the best harmonic mean between P and R. Based on the calculated AUC and GM values, BNB gives the best performance which means that it can effectively classify both positive and negative instances with the lowest relative positive and negative errors. Therefore, MCAS suggests BNB as the best classifier. Figure 4.11 shows the resulting confusion matrix corresponding to each of the seven classifiers on DAT-07.

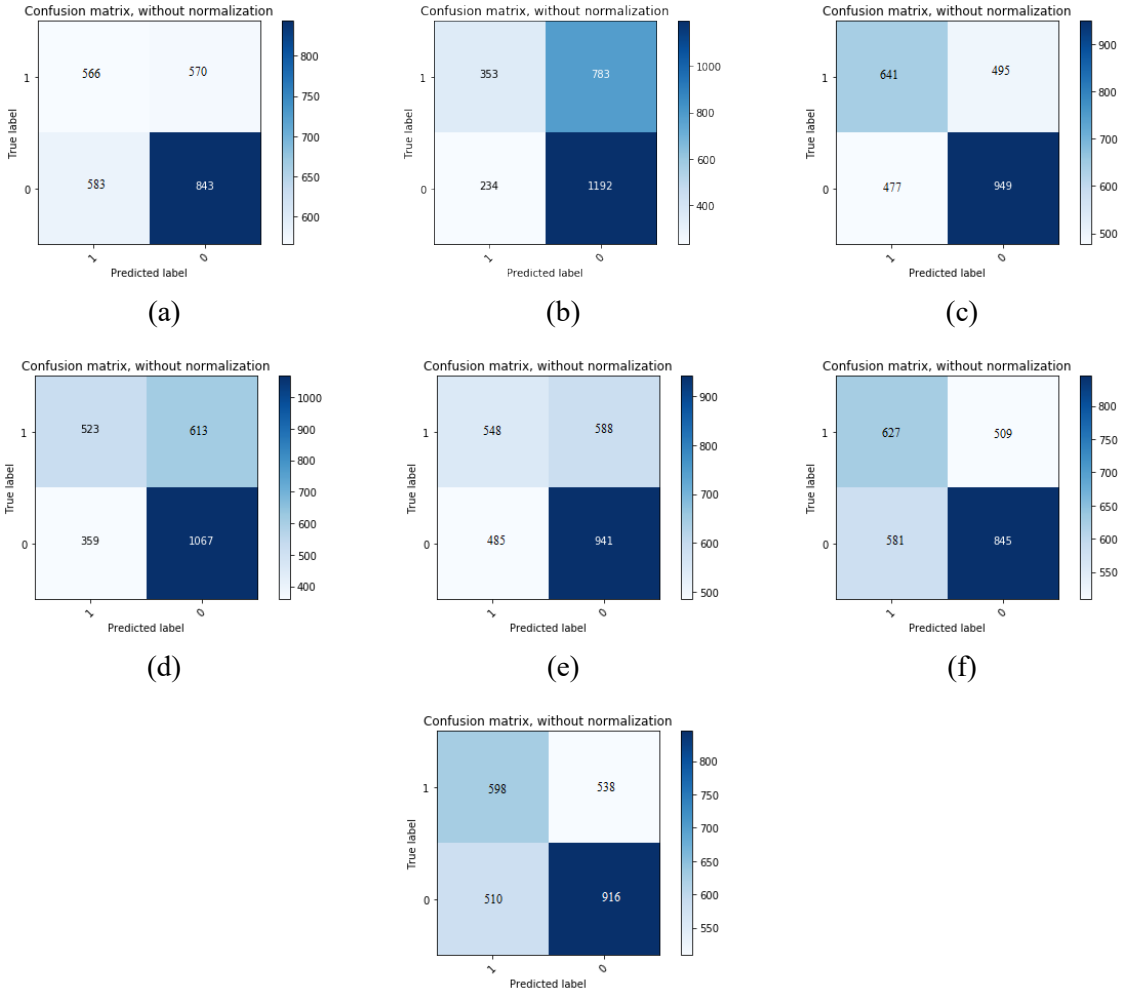


Figure 4.11 Classification Confusion Matrix on DAT-07. (a) DT, (b) MNB, (c) BNB, (d) LR, (e) kNN, (f) Perceptron, (g) MP.

8- ***For DAT-08:*** MP gives the best ACC of 60.25% and has the highest GM and AUC values of 57.68% and 59.18%, respectively. Accordingly, MCAS suggests MP as the best classifier when both classes have the same importance, and when the negative class is more important. MNB achieves the best harmonic mean (F1) of 71.05% between P and R, in addition to the best TPR value of 93.10% and the lowest FOR of 30.00%. Hence, MCAS suggests MNB as the best classifier when the positive class is more important. Figure 4.12 shows the resulting confusion matrix corresponding to each of the seven classifiers on DAT-08.

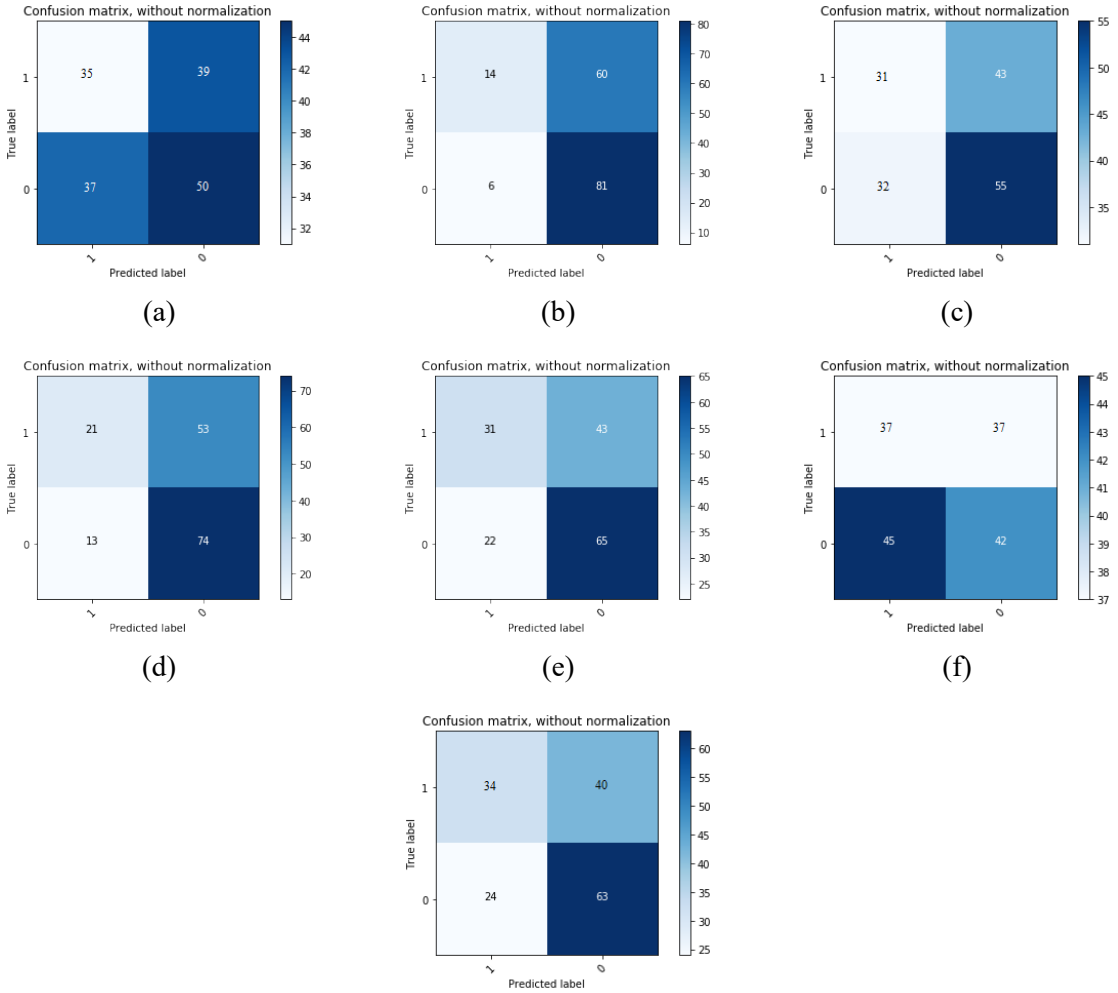


Figure 4.12 Classification Confusion Matrix on DAT-08. (a) DT, (b) MNB, (c) BNB, (d) LR, (e) kNN, (f) Perceptron, (g) MP.

9- ***For DAT-09:*** MP gives the best ACC of 75.17% and has the highest GM and AUC values of 57.65% and 62.95%, respectively. Therefore, MCAS suggests MP as the best classifier when both classes have the same importance, or when the negative class is more important. LR achieves the best harmonic mean (F1) of 85.16% between P and R, in addition to the best TPR value of 100% and the lowest misclassification rate of positively detected instances. It should be remarked that for LR, because of having $TP = TN = 0$, the denominator is arbitrarily set to 1 when calculating the corresponding MCC value and accordingly the overall value becomes 0. This leads to the exclusion of LR from consideration even though it performs the best when dealing with positive class instances. Therefore, MCAS suggests LR as the best classifier when the positive class is more important. Figure 4.13 shows the resulting confusion matrix corresponding to each of the seven classifiers on DAT-09.

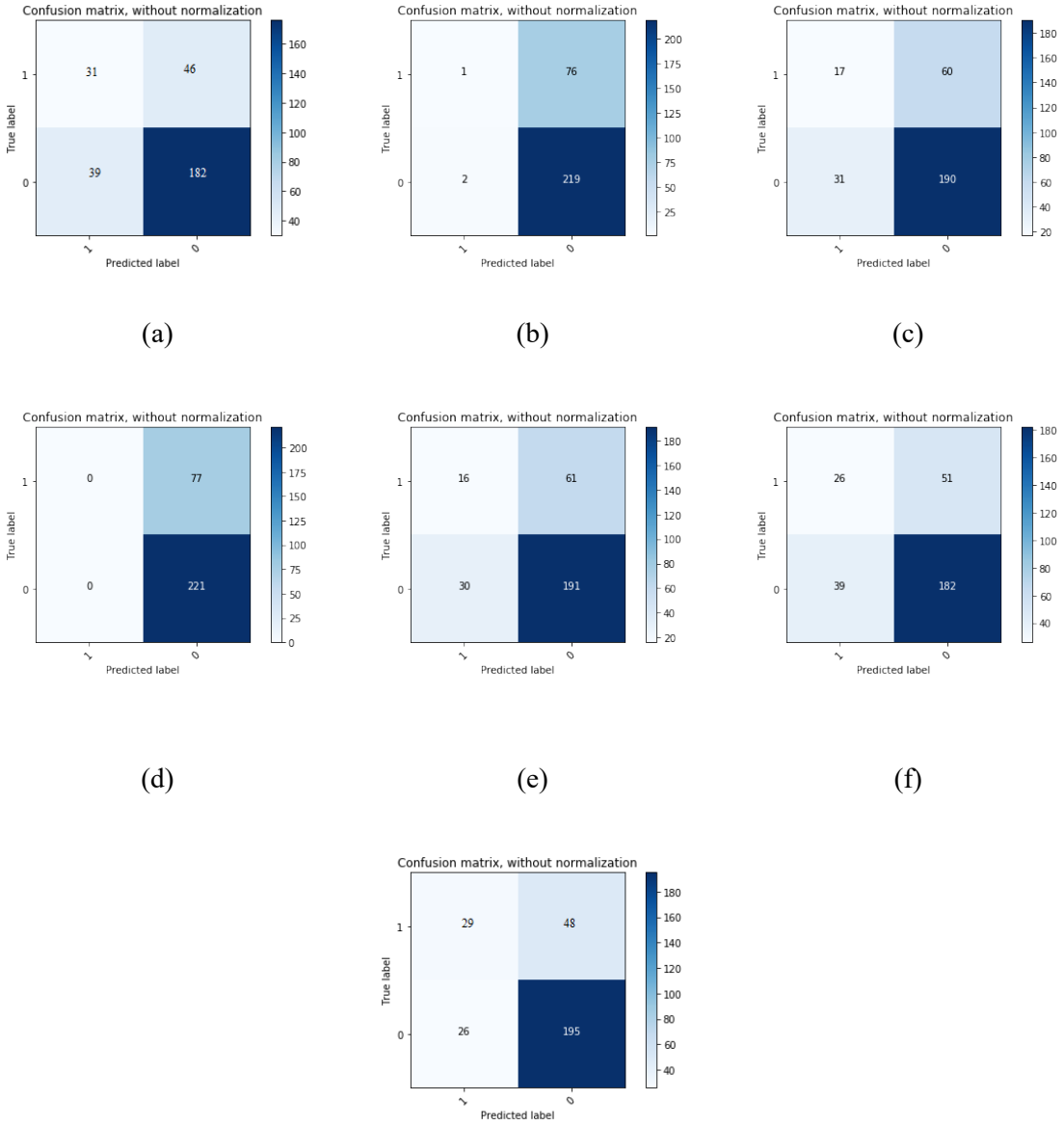


Figure 4.13 Classification Confusion Matrix on DAT-09. (a) DT, (b) MNB, (c) BNB, (d) LR, (e) kNN, (f) Perceptron, (g) MP.

10- For DAT-10: MP gives the best ACC and F1 values of 81.71% and 89.13%, respectively. Moreover, it has the highest GM and AUC values of 54.74% and 63.39%, respectively. Therefore, MCAS suggests MP as the best classifier when both classes have the same importance, and when either the positive class or the negative class is more important. Figure 4.14 shows the resulting confusion matrix corresponding to each of the seven classifiers on DAT-10.

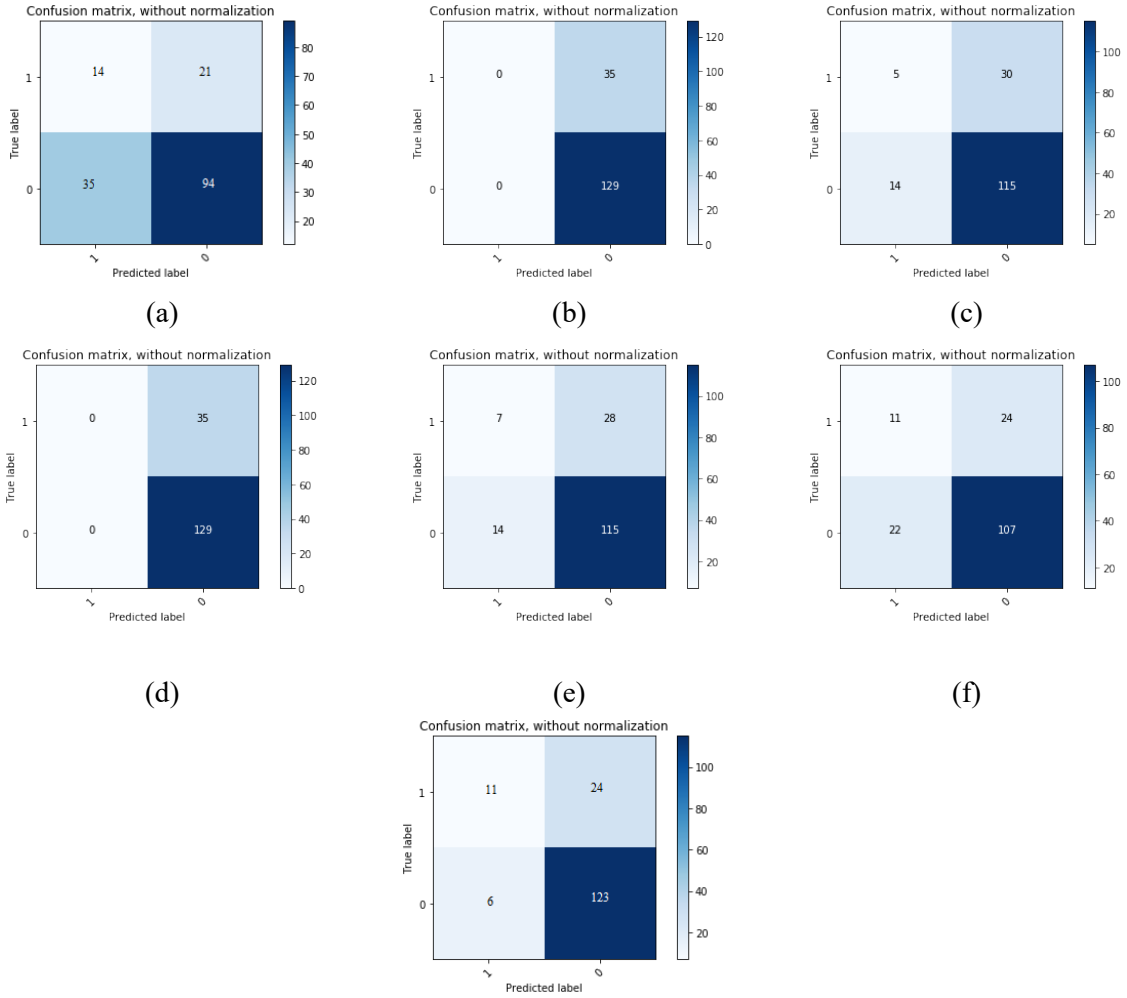


Figure 4.14 Classification Confusion Matrix on DAT-10. (a) DT, (b) MNB, (c) BNB, (d) LR, (e) kNN, (f) Perceptron, (g) MP.

11- For DAT-11: it is very ambiguous to decide which classifier is the best. On one hand, both the BNB and MP classifiers give the best ACC of 68.37%. On the other hand, only the MP gives the best harmonic mean between P and R of 77.37%. Moreover, both MNB and LR give SN of 100%. Therefore, based on the techniques' ability to correctly classify instances from both positive and negative classes in the presence of misclassification errors, MCAS suggests MNB has the best performance among the classifiers when both classes have the same importance and when the negative class is the most important. Also, MCAS suggests MP as the best when the positive class is the most important. Figure 4.15 shows the resulting confusion matrix corresponding to each of the seven classifiers on DAT-11.

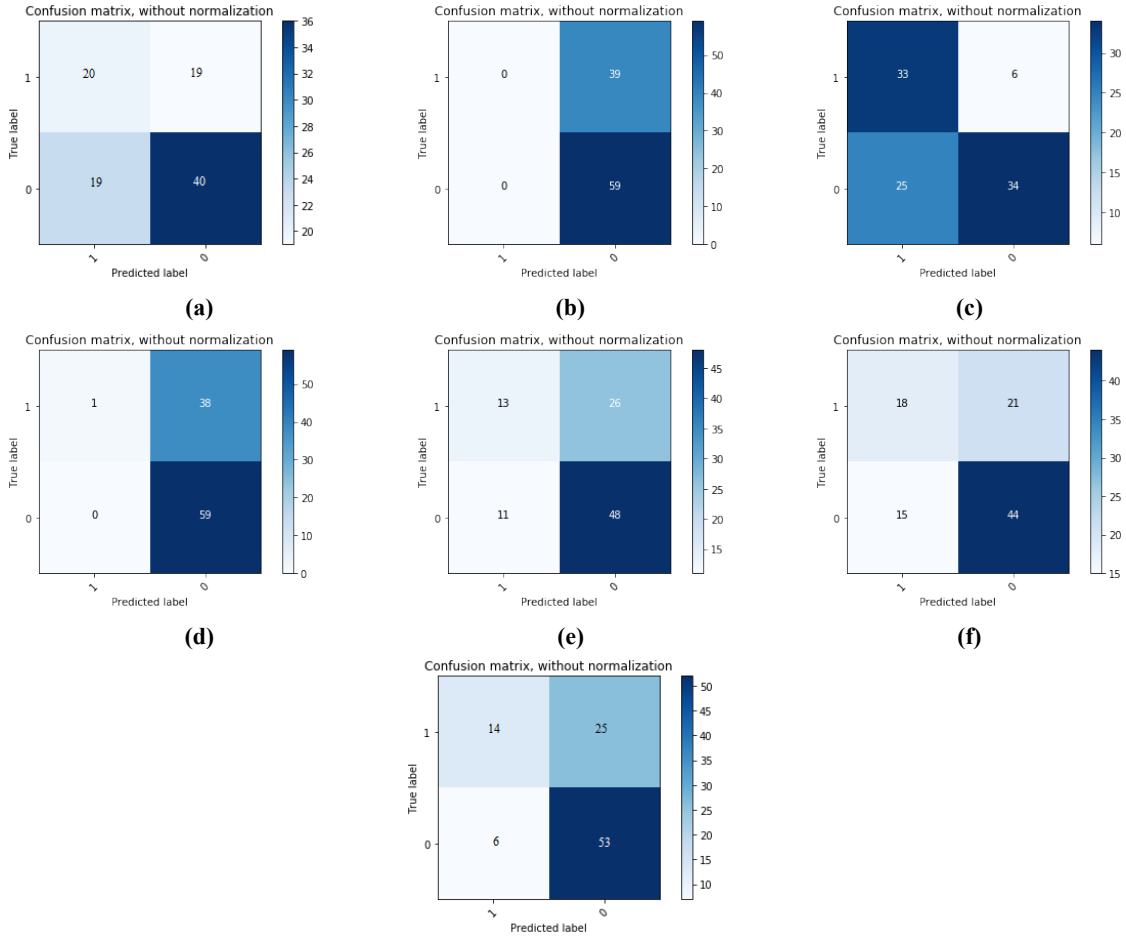


Figure 4.15 Classification Confusion Matrix on DAT-11. (a) DT, (b) MNB, (c) BNB, (d) LR, (e) kNN, (f) Perceptron, (g) MP.

Now that an empirical analysis of both the proposed frameworks has been carried out, the next chapter examines their effectiveness in real-life problem.

Chapter 5

Case Study: COVID-19 Misleading Information Detection

While this study was in progress, COVID-19 emerged as a pandemic that poses a serious threat to human life. With more than 17.5 million confirmed cases and more than 676 thousand confirmed fatalities worldwide in August 2020 (these numbers after a year, in August 2021, climbed dramatically to reach more than 200 million confirmed cases and more than 4.25 million confirmed fatalities) it demands all of humanity to band together to fight it. As a result, using COVID-19 as a case study to demonstrate the capabilities of the proposed classification framework becomes the logical choice. Thus, we utilized the proposed classification framework to build a novel misleading information detection model to help in the fight against the spread of misleading information surrounding the COVID-19 outbreak pandemic. We employed the obtained detection model to address the scarcity of the available bilingual datasets by building a multipurpose, bilingual, and multidialectal social media dataset for the detection of misleading information related to the COVID-19. It should be noted that the reported work related to the fight against COVID-19 misleading information in this chapter covers the efforts that have been done up to September 2020.

5.1 Overview

At the end of December 2019, the World Health Organization (WHO) was informed of a cluster of pneumonia cases of unknown cause that were detected in the city of Wuhan, Hubei Province, China. Initially, these patients were diagnosed as having acute pneumonia. Most of them worked in a wet market in Wuhan and showed common symptoms of fever, dry cough, tiredness, and in more severe cases breathing difficulty. However, these symptoms were not of acute pneumonia as was first thought. With the increasing number of cases, China informed the WHO of the situation and its unknown cause in early January 2020 [254].

The WHO named the virus "Severe Acute Respiratory Syndrome Coronavirus 2 (SARS-CoV-2)" and the disease "Coronavirus Disease (COVID-19)". COVID-19 is a global health problem that requires extreme caution, strict maintenance of personal and general hygiene, and the cleanliness of all places. These practices help in avoiding the occurrence of mutations so that the virus can be controlled and contained. All reports issued by the WHO indicate that the epidemiological situation (since the beginning of January 2020) is very critical, and scientists are frantically working to develop a vaccine to eradicate the virus. Many studies expected that an effective vaccine is to be available to the public between December 2020 and June 2021 [255]. As of March 2021, 308 vaccine candidates were in various stages of development. As of June 2021, 2.15 billion doses of the COVID-19 vaccine have been administered worldwide based on official reports from national health agencies [256].

Transportation means and social network platforms render the world a small village. As far as transportation is concerned, it has become easy to transport people from one place to another. This promotes the circulation of COVID-19 very quickly and makes it a pandemic [255]. As for social network platforms, they play a vital and effective role not only in spreading misleading information related to COVID-19 but in all matters of our daily lives as well as the various crises and conflicts around the world. With the presence of a new virus whose characteristics and details are not fully known yet, and with a state of fear and panic among the general public, the spread and circulation of misleading information about this virus and its impact are ubiquitous.

The misleading information may be intended to disrupt the economy of countries, reduce people's confidence in their governments, or promote a specific product to achieve enormous profits. This has already happened with COVID-19. In 2020, the shared misleading information about lockdowns, vaccinations, and death statistics, have fueled the panic of purchasing groceries, sanitizers, masks, and paper products. This led to shortages that disrupted the supply chain and exacerbated demand-supply gaps and food insecurity. Moreover, it has caused a sharp decline in the international economy, severe losses in the value of crude oil, and the collapse of the world's stock markets [257] [258]. Additionally, some people have lost faith in their governments as in Italy and Iran, due to the spread of COVID-19 and the shortage of medical protection products all over the world. In early 2020, all these lead the world into an economic recession [258] [259].

The WHO has issued numerous data, directives, and warnings that are not only related to COVID-19 but also the "Infodemic" [260]. Infodemic is like a disease that spreads and circulates in the form of misleading information. It is very challenging to verify the validity, credibility, and correctness of the shared information, especially if it is related to a horrific disease that is a threat to humanity [261]. The WHO has asked popular search engines, such as Google, Yahoo, and Bing, and many social network platforms to display its officially issued reports and information as top hits of any search that is related to COVID-19 [262]. It is evident from this WHO request that utmost care and caution must be exercised when selecting sources of information. We should not rely on what is promoted on social networks but rather on reliable and unbiased information sources such as the WHO, global scientific research bodies, and NGOs. Hence, there was an urgent need to provide a tool for the public to verify the trustworthiness of information related to COVID-19 [254].

To date, coronavirus disease (COVID-19) has considerably impacted our lives. Although the significant role of various digital technologies and social network platforms in fighting against COVID-19 is apparent, it has also offered a ground for the exploitation of many social behavior vulnerabilities (e.g., the spread of different kinds of misinformation (fake news, propaganda, hoaxes, etc.), stigma, hatred, racism, and Cybercrimes) [263]. Some organizations may profit from the spread of such misleading information [264]. Whereas, once the misleading information is published, it becomes a rumor that attracts many users. Later, when this information becomes a trending topic, it

is used by advertising organizations and companies to promote products or ideas and gain huge financial profits [265]. Therefore, the fight against the spread of any kind of misleading information, and the need to find a system that assists in verifying the integrity of the shared information surrounding COVID-19 arise. In this context, the WHO is doing a great effort in fighting Infodemic; it is working closely with different technology companies and social network platforms such as Twitter, Facebook, YouTube, Google, and Microsoft, to endorse critical updates from reliable sources, and to point out the shared misleading information on their platforms [265].

Developing an automated Misleading Information Detection system is very challenging, especially with the lack of available standards and information related to the virus, which makes the consequences of wrong decisions dire [266]. All misleading information detection systems utilize Artificial Intelligence (AI) [267], machine learning and deep learning, and Natural Language Processing techniques (NLP) [268]. These techniques are used to assist users in filtering the information they are viewing [109]. Moreover, they help in classifying whether a piece of information is misleading or not. This is done by comparing a piece of given information with some pre-known dataset that contains both misleading and truthful information [6].

In this Chapter, we introduce a model to detect misleading information in both the Arabic and the English languages, with the COVID-19 pandemic as our case study. For the ground-truth data, we decided to gather COVID-19 related information from international, and what we perceived as reliable and unbiased, institutions. We also collected facts from different fact-checking websites in addition to the information found in official reports and news related to the pandemic from the WHO, UNICEF, and the UN official websites [254] [269] [270]. Our detection process is based on the ensembled learning of ten machine learning classifiers that are built on the collected ground-truth data. Then, we propose a voting ensemble deep learning model for detecting misleading information in the English language, related to COVID-19. The final decision is based on the ensemble result from 6 deep learning techniques: Sequential model, Convolutional Neural Network (CNN), Recurrent Neural Network (RNN-LSTM, and RNN-GRU), Bidirectional Recurrent Neural Network (BiRNN-GRU), and Recurrent Convolutional Neural Network (RCNN). We deploy Word Embedding based on a pre-trained word embedding list in addition to the existing word impeding in the input layer of the used techniques.

Afterward, we deploy this model in building an automatically annotated, bilingual (Arabic/English) COVID-19 Twitter dataset (COVID-19-FAKES). This dataset has been continuously collected since February 04, 2020, four days after the outbreak was declared a Public Health Emergency of International Concern, by the WHO, on January 30, 2020, till March 10, 2020, one day before the outbreak was declared a Pandemic. For performing the automated annotation task, we collect a set of ground-truth data, related to COVID-19, by scraping the shared information, in both Arabic and English languages, from the official websites and the official Twitter accounts of

the WHO, UNICEF, and UN as sources of reliable information. Besides, we collect COVID-19 pre-checked facts from different fact-checking websites, e.g., "poynter.org", "snopes.com", factcheck.org", etc. The ground-truth data is used to build detection models of 13 different machine learning algorithms, employing 7 different feature extraction techniques with each. Then, we use these models to automatically annotate our dataset into either Real or Misleading.

The rest of this chapter is organized as follows. *Section 5.2* shows the related work, done before September 2020, of using machine/deep learning for building misleading information detection systems related to COVID-19, and the related work about the available COVID-19 datasets. *Section 5.3* introduces the proposed misleading information detection system, with the details of the 4-stage process using both machine learning and deep learning algorithms, while the model's performance evaluation results are discussed in *Section 5.4*.

5.2 Related Work to the Fight Against COVID-19 Misleading Information

Many available misleading information detection websites could be used to search for pre-checked data, e.g., *Snopes.com, PolitiFact.com, Factcheck.org*, etc. However, these websites are mostly human-based, where the analysis of data is carried out manually. This analysis is performed by expert analysts who are intimately familiar with the subject context. The manual approach is slow, expensive, highly subjective, biased, and has become impractical due to the huge volume of available data on social networks [16]. Hence, the process of automated classification of data represents an exciting and productive area of study.

Recently, numerous studies are focusing on the analysis of shared information related to COVID-19 on different online platforms. In 2020 and early 2021, there were many attempts for deploying different ML and DL techniques for the detection and diagnosis of COVID-19 as a disease ((e.g., [271], [272], [273], [274], [275], and others). Some other researchers are focusing on the treatment discovery (e.g., [276], [277], and others). In contrast, in 2020, only a few machine learning-based attempts existed to develop misleading information detection systems around COVID-19. All these researchers' goal is to analyze the data collected from different online sources using a list of hashtags for a given period. However, there are still no available benchmark datasets to test and validate different developed models, especially those related to misinformation detection systems. All the existing systems and datasets study and analyze the human and social behavior, and information consumption surrounding COVID-19, e.g., [278], [279], [280], [281], [282], [283], and [284].

We are not able to find any existing work that deploys ensemble deep learning techniques for building misleading information detection systems. This is partly due to the lack of available

benchmark datasets. Most of the current efforts focus on building datasets related to COVID-19. To the best of our knowledge, our work is the first that introduces an ensemble model based on deep learning techniques to detect misinformation related to the emergence of the COVID-19 disease.

To enable COVID-19 related research, several studies have targeted the shared COVID-19 related information on different social network platforms and the collected datasets to be used in building, optimizing, and testing different machine/deep learning systems. To the best of our knowledge, all these works depend on a list of hashtags related to COVID-19 and focus on a given period. Moreover, all the available datasets are general-purpose ones, with no clear assigned annotation to the data. Also, in early 2021, all the publicly available datasets are used to study and analyze the human and social behavior, and information consumption surrounding COVID-19.

Chen *et al.* [278] collected a multilingual coronavirus dataset of 67M million English tweets and 101M non-English tweets intending to study online conversation dynamics. They used Twitter's streaming API [285] for collecting tweets from January 22 to April 23, 2020. Also, Lopez *et al.* [279] collected around 6.5M multilingual dataset to identify public responses to the pandemic and analyze the information related to it. They used Twitter API for collecting their data from January 22 to March 13, 2020. Singh *et al.* [281] collected around 2.8M tweets in multiple languages to investigate the amount of shared information and discussions on social network platforms, specifically Twitter, related to COVID-19, myths shared about the virus, and how much of it is connected to other high and low-quality information on the Internet through shared URL links. They used Twitter API for collecting their data from January 16 to March 15, 2020. Sharma *et al.* [280] collected 30.8M tweets in multiple languages to design a dashboard for visualizing discussions around Coronavirus and identifying the quality of its related information shared on Twitter. They used Twitter API for collecting their data from March 1 to March 30, 2020.

Alqurashi *et al.* [282] collected nearly 4M Arabic language tweets on COVID-19 to study the pandemic from a social perspective and analyzed human behavior, and information spread with special consideration to Arabic-speaking countries. They used Hydrator [286] and TWARC [287] tools to retrieve the full objects of the tweet, covering the period between March 1, 2020, to March 30, 2020. Haouari *et al.* [288] collected 748k Arabic language tweets in addition to propagation networks of a subset of 65k tweets to enable research related to natural language processing, information retrieval, and social network analysis. They used Twitter search API to retrieve the data daily, covering the period from January 27, 2020, to March 31, 2020, and deployed the detection and classification techniques introduced in [12].

Zarei *et al.* [283] collected social media content from Instagram using hashtags related to COVID-19. They collected 5.3K posts, 18.5K comments, and 329K likes with the aim of identifying and analyzing social behavior on their collected data. They used the official Instagram API [289] for collecting their data from January 5 to March 30, 2020. Cul *et al.* [284] collected 1,896 news, 183,564

related user engagements, 516 social platform posts about COVID-19 to call out for public attention to the spread of misinformation related to COVID-19 and to assist ongoing research to develop misinformation-detection systems.

5.3 COVID-19 Misleading Information Detection Model

To build a detection system for misleading pandemic news, we must first decide on how to judge COVID-19 related information, and what sources that we can rely on for evaluating each data instance. Figure 5.1 shows the block diagram of our proposed misleading information detection framework. The process for detecting misleading information is divided into four main stages: Information-Fusion, Information-Filtering, Model-Building, and Detection.

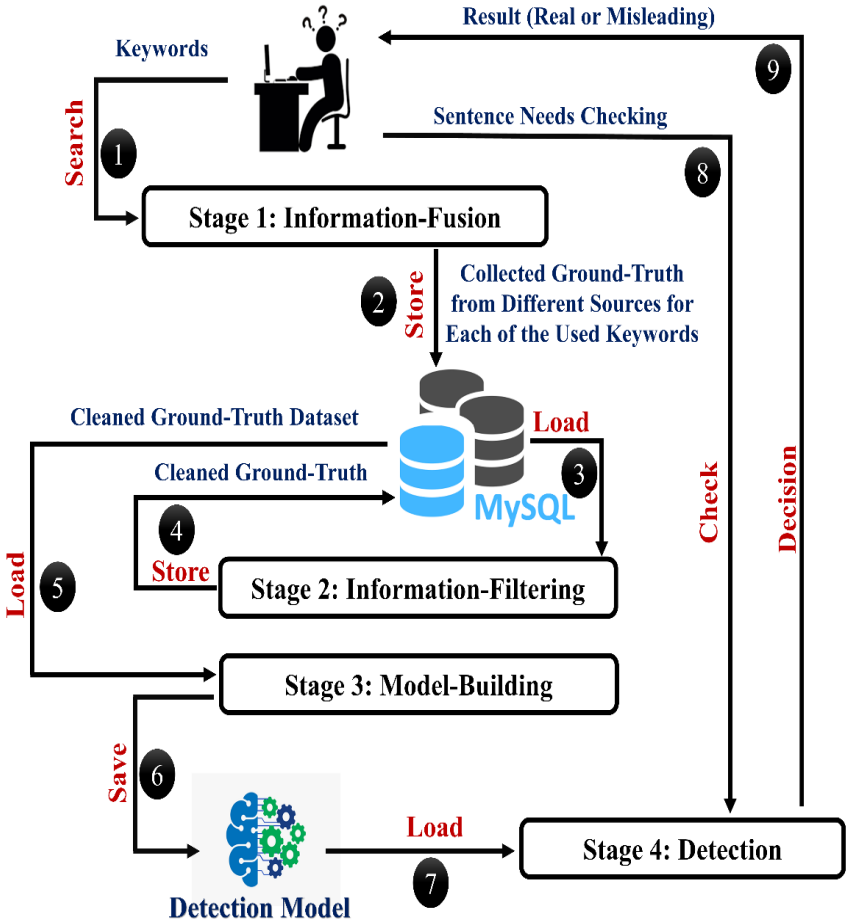


Figure 5.1 The Proposed Misleading Information Detection Framework.

5.3.1 Information-Fusion Stage

The accuracy of any detection system is highly affected by the quality of data used in building the detection model, the machine learning algorithm employed, and how these data describe the facts related to the topic of interest. Hence, when the topic of interest is critical, it is essential to ensure the

accuracy and reliability of information sources and not to be drawn into shared information and news from unreliable entities. Therefore, reliance on perceptions and feelings should be avoided.

We must rely only on documented information and facts without making any modifications. As the COVID-19 pandemic is more than a purely medical event and is of concern to all people, it is necessary to depend on reliable and authoritative sources to get our information. With more scrutiny, we should be able to find medical and other organizations that try not to spread fear and terror to the public. Moreover, they should be impartial and objective in their handling of information and news of the COVID-19 outbreak crisis.

For all the previously mentioned reasons, we decided to get our COVID-19 ground-truth mainly by scraping the websites of the WHO and its regional branches, as well as UNICEF [335] and its affiliated bodies, and of course the UN [290]. We extracted all the information related to the COVID-19 outbreak from these organizations' daily situation reports [255], the briefing of the WHO Director-General on COVID-19 [291], in addition to the news published on their websites' newsroom [292].

Moreover, we utilized the Google Fact Check Tools API [293], which allows users to browse and search for facts from different fact-checking websites around the world, including *opensecrets.org, snopes.com, factcheck.afp.com, washingtonpost.com/news/fact-checker, factcheck.org, politifact.com*, etc. We did not employ information published by the official accounts of the health ministries in various countries or any of the organizations and research centers affiliated with a single country. Rather, there is a reliance on international organizations to avoid biased and inaccurate statements and information.

For querying the fact-checking websites, we used the following search keywords which are related to the coronavirus disease (COVID-19):

- "Coronavirus"
- "Corona_virus"
- "Corona-virus"
- "Novel_Coronavirus"
- "2019-nCoV"
- "Novel-Coronavirus"
- "NovelCoronavirus"
- "2019_nCoV"
- "nCoV"
- "COVID-19"
- "SARS-CoV-2"
- "covid19"
- "كورونا"
- "كوفيد19"
- "كوفيد-19"
- "كوفيد_19"
- "كورونا-المستجد"
- "كورونا_المستجد"
- "فيروس-كورونا"
- "فيروس_كورونا"

At the end of this stage, we stored the collected data into our MySQL-Server, with data from each source in a different table. It should be remarked that the collected data are different in structure, and the ones from the fact-checking websites are labeled in various forms to describe real and misleading data. For example, the real data may be labeled as True, Real, Correct Attribution, Benar, Verdadeiro, Gerçek, Verdadero, etc., while the misleading ones could be labeled as False, Fake, Misleading, Falso, Faux, Engañoso, False Connection, False Context, False Content, C'est faux, etc. Hence, the data from different sources must be organized in a uniform format, and the labels need to be binarized to either Real or Misleading, as shown in the next stage.

Moreover, the published data in both the fact-checking websites and the official websites of international organizations are continuously increasing. Consequently, the amount of collected data is expected to change continuously. To build a near real-time detection system, we should continuously update our collected ground-truth to accommodate frequent updates from these organizations.

5.3.2 Information-Filtering Stage

As we are interested in detecting misleading information that is written in Arabic and English, the first step is to filter the collected data from different sources and select only written Arabic and English data. The following steps are then carried out for standardizing our data and integrating them into a uniform ground-truth dataset.

5.3.2.1 Duplicate Removal

We checked the collected data from the information-fusion stage and eliminated the redundant ones. This was done by removing the data that had the same content and originated from the same source and keeping only one copy of them.

5.3.2.2 Data Standardization and Label Binarization

We ensured the consistency of the data regardless of their source by making the data fit in a standard structure that contains the following fields:

- *Data_Publishing_Date (the date when the text was published at its source).*
- *Fact_Publishing_Date (the date when the text was checked and published on the fact-checking websites).*
- *Fact_text.*
- *Data_Origin (e.g., Facebook, Twitter, news website, blog, WHO, UNICEF, UN, etc.).*
- *Fact_publisher (e.g., politifact.com, snopes.com, factcheck.org, factual.afp.com, opensecrets.org, colombiacheck.com, truthorfiction.com, who.int, etc.).*
- *Label (e.g., Real = 1, Misleading = 0).*
- *Language (e.g., Arabic* = 'ar', Spanish = 'es', French = 'fr', etc. In the current implementation we are only interested in both the Arabic and English written tweets).

All date fields were reformatted to a standard date format (i.e., *YYYY-MM-DD*). Moreover, each fact was given a unique Fact_ID to be used for indexing purposes.

5.3.2.3 Data Integration

After the data from different sources were indexed and standardized, they were inserted into the newly generated facts table.

5.3.2.4 Exploratory Data Analysis (EDA) on Ground-Truth Data

EDA provides an in-depth understanding of the data. Moreover, the visual representation of text documents is considered one of the most important tasks in the field of text mining [294]. Its aim is not only exploring the content of documents from different aspects and at different levels of detail, but also summarizing a single document, showing the words and topics, detecting events, and creating storylines [295].

However, there are still some gaps between visualizing unstructured textual data and structured data. For example, many text visualizations do not represent the text directly, rather, they represent an output of a language model (word count, character length, word sequences, etc.). This section provides insights into the collected tweets, not only through studying them but also by visualizing quantitative and category data in them. This visualization is done by employing both Plotly's Python graphing library [296] and the Bokeh visualization library [297].

The collected data were stored in our MySQL-Server, and then we performed information filtering as described in *Subsection 5.3.2*. The resulting data from the Information-Filtering stage were labeled as Real or Misleading. The size of the collected ground-truth data is 7,486 instances.

We performed Exploratory Data Analysis to get some general insights on the collected ground-truth data. Figure 5.2 shows the word cloud of the top-100 words in them while Figure 5.3 shows the distribution of ground-truth data classes.

Figure 5.2 Word Cloud.

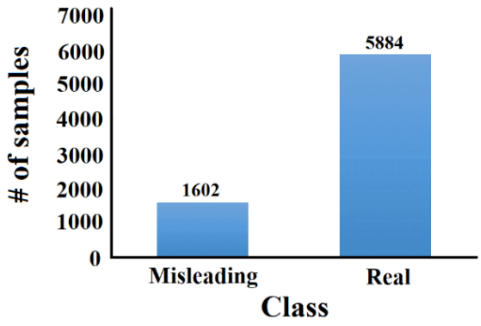
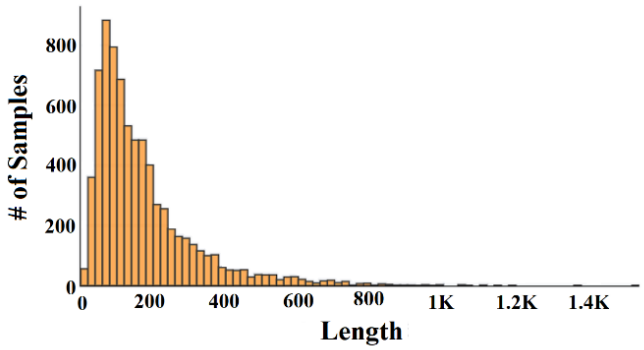
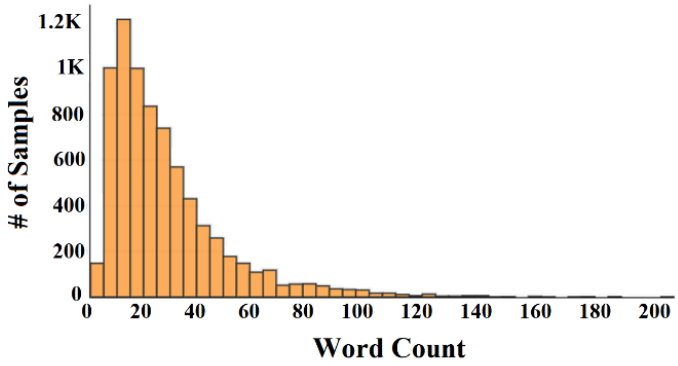


Figure 5.3 Distribution of Ground-Truth Data.

The Distribution of length and word count of the ground-truth data is shown in Figure 5.4.



(a)



(b)

Figure 5.4 Distribution of Ground-Truth Sample Length and Word Count. (a) Samples' Length Distribution, (b) Samples' Word Count Distribution.

We noticed from Figure 5.4 that about 75% of the samples have less than or equal to 200 characters and less than or equal to 30 words. Figure 5.5 shows the top-10 repeated unigrams in the ground-truth data.

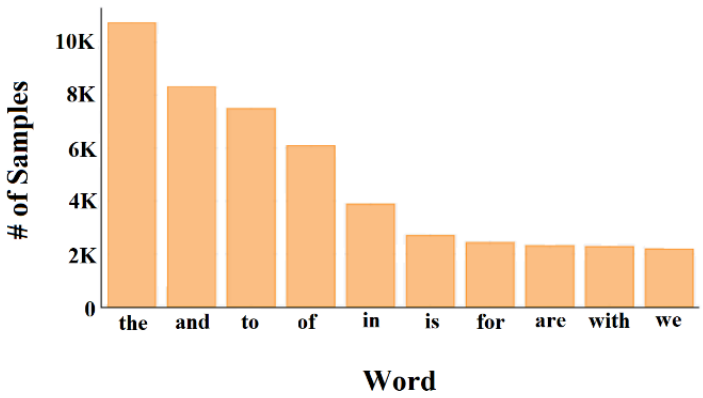


Figure 5.5 Distribution of Top-10 Unigrams.

From Figure 5.5 we noticed that all the top-10 repeated unigrams in the ground-truth data are stop words and have relatively high frequencies. These stop words are useless when processing our data. This indicates that the data needs to be preprocessed to remove noisy and unimportant contents.

Figure 5.6 shows the top-10 unigrams after removing the stop words. After performing the preparation and the preprocessing step, we were able to minimize the indexing size by around 75-80%.

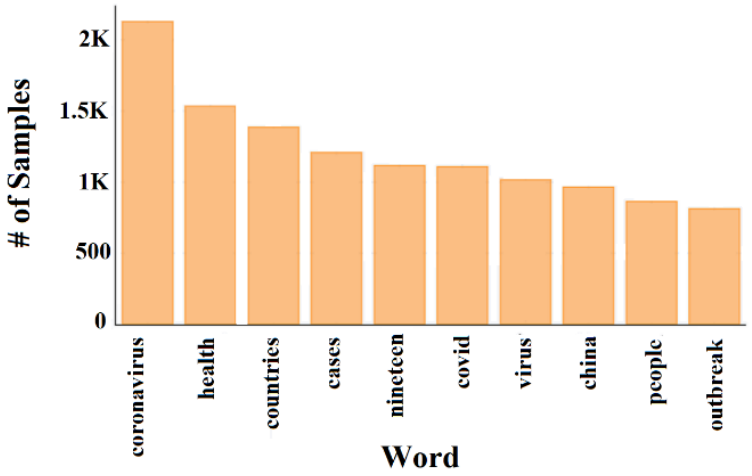


Figure 5.6 Distribution of Top-10 Unigrams After Removing the Stop Words.

Figure 5.7 shows the distribution of top-10 Part of Speech (PoS) tags and their description [298]. we noticed that the frequent words are mostly nouns, verbs, and adjectives. Hence, for dimension reduction of the extracted feature vector, we could consider only the words with these most frequent tags and neglecting the words with other tags.

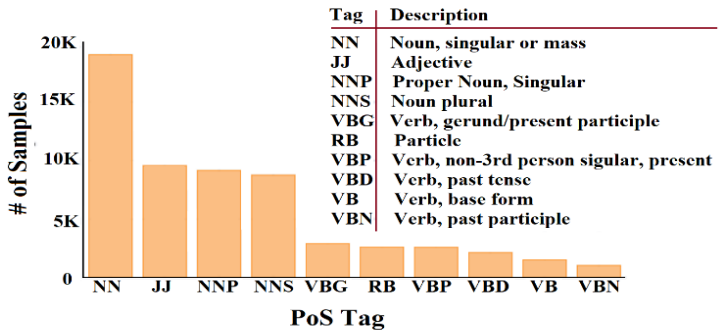


Figure 5.7 Distribution of Top-10 PoS Tags.

5.3.3 Model-Building Stage

To build the misleading information detection model, all the collected ground-truth must be prepared first and then passed through feature engineering and learning stages as shown in Figure 5.8.

5.3.3.1 Preparation

To build the detection model, the ground-truth data must be prepared first. This is done by utilizing the introduced technique in [109]. Each instance of the ground-truth data is represented by three fields: "Fact_Data", "Label", and "Language". The "Fact_Data" field is obtained by the union of the original segments: Fact_text, Data_Publishing_Date, Fact_Publishing_Date, Data_Origin, and Fact_publisher. The "Label" field contains the label assigned to the ground-truth instance, while the "Language" field indicates the language in which the instance is written.

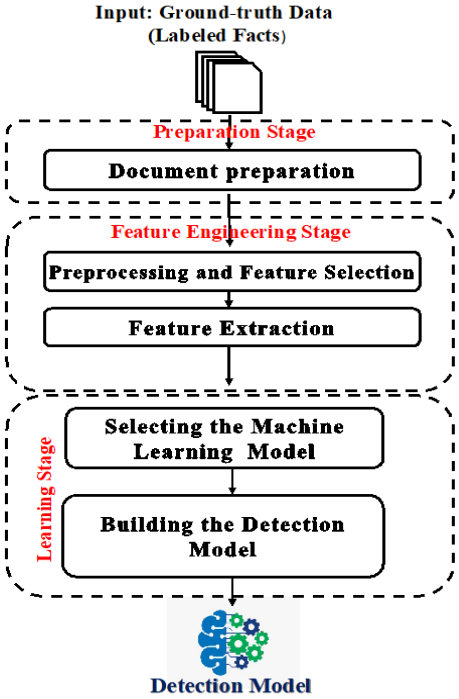


Figure 5.8 Block Diagram of the Detection Model Building Stage.

5.3.3.2 Feature Engineering

This stage is composed of two steps: a) Preprocessing and feature selection, and b) Feature extraction, as follows.

a) ***Preprocessing and Feature Selection*** [109]**,** this step aims to facilitate data manipulation, reduce memory space needed, and shorten the processing of huge amounts of data. This was done by following the same steps as previously discussed in *Subsubsection 3.1.2.1*.

At the end of the preprocessing and feature selection phase, we obtained a set of stemmed bag of words (BoW) which represents the original feature vector that would be used for the feature extraction phase.

b) ***Feature Extraction,*** machine learning algorithms, such as kNN, Support Vector Machine (SVM), etc., do not take the ordering of the features inside data samples into account. It relies

on either the use of n-gram vector representation or a bag-of-words approach for representing the feature vector. Then, we utilized TF-IDF as a feature extraction technique as in eq. 3.6.

In contrast, in deep learning algorithms, the sequence of the used data is mandatory. For some applications, word order is critical to ensure the high accuracy of the results and any change to this order affects the overall process. For example, the sentences, "COVID-19 is a critical disease that affects all of us. Our life changed completely." can be understood only when reading them in order. Models such as Convolutional Neural Networks (CNN)s can infer meaning from the order of words in a sample. Hence, for extracting features from our data to be suitable for build deep learning models, the sequence representation of features to preserve their order is employed.

The sequence of the textual data can be either a sequence of characters or a sequence of words. Character-level representation is used mainly if the textual data have a lot of typos, which is not in our case, as our data is collected from the official websites of reputable international organizations. Therefore, we used the word-level representation in our system. For example, consider that we have the following two sentences "COVID-19 disease spreads fast", and "COVID-19 badly impacted our lives", the index assigned for every word is {"COVID-19": 1, "fast": 2, "spreads": 3, "disease": 4, "badly": 5, "impacted": 6, "our": 7, "lives": 8}. Then, the sequence of word indexes of the sentence "COVID-19 disease spreads fast." is {1, 4, 3, 2}.

5.3.3.3 Learning

The extracted feature vectors, that represented each document in the training data, from the feature engineering stage were fed into different machine/deep learning classification algorithms.

a) ***Machine Learning,*** is done by utilizing the Scikit-learn machine learning library in Python [87]. First, we needed to validate the collected data. The ground-truth data were split into 80% training and 20% testing sets of 5-fold for cross-validation purposes. We used the training set to build detection models using various classification algorithms. As for the validation set, we passed it to the built detection models, and the validation results are presented in Subsection 5.3.3. Then, we used the whole collected ground-truth data as training data to build our misleading information detection models using various machine learning algorithms.

b) ***Deep Learning,*** for building our model, TensorFlow backend [299] with Keras API [140] is employed to implement deep learning algorithms: Sequential Model, Convolutional Neural Network (CNN), Recurrent Neural Network with Long Short Term Memory (RNN-LSTM), Recurrent Neural Network with Gated Recurrent Units (RNN-GRU), Bi-Directional Recurrent Neural Network with Gated Recurrent Units (BiRNN-GRU), and Recurrent Convolutional Neural Network (RCNN). The wiki-news-300d-1M (GloVe) [300] pre-

trained word embedding is used in the first layer of our model. Figure 5.9 [301] shows the typical layers of a Keras Sequential Model as an example.

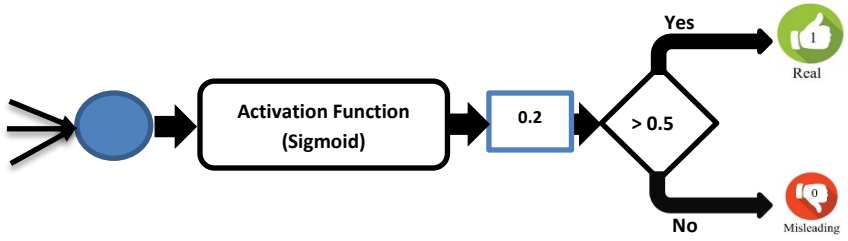


Figure 5.9 Misinformation Detection System's Output Layer [301].

We used the sigmoid function as an activation function in all our models in the output layer. For the training model, we used the binary cross-entropy and Adam's optimizer with a learning rate of 1e-3 with 100 as the number of epochs and a batch size of 64. The early stopping option was set to 1 to automatically terminate the learning once the accuracy stops to change between consecutive epochs. Figure 5.10 shows the functionality of our model's output layer.

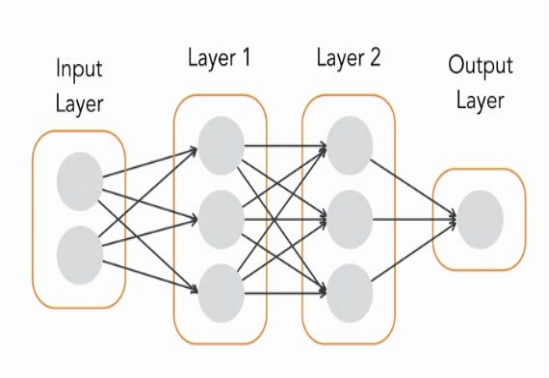


Figure 5.10 Keras Sequential Model.

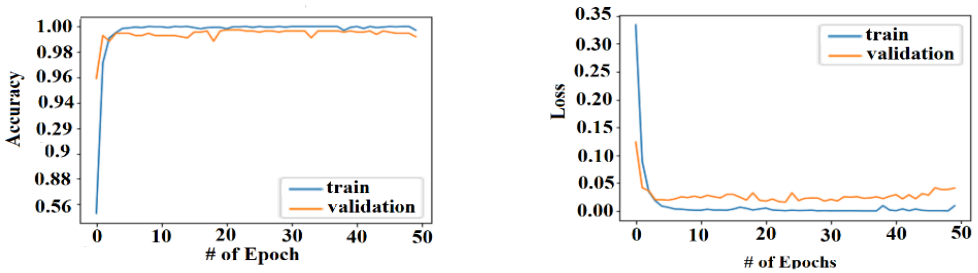
Now all the layers and functions have been constructed in our model architecture. We configured six deep learning models, Sequential, CNN, RNN-LSTM, RNN-GRU, BiRNN-GRU, and RCNN. We trained the models using 80% of the collected data as training data. We used the rest to perform validation to ensure the validity of both our developed system and our collected data. Figure 5.11 shows a summary of the sequential model as a sample for other algorithms. Figure 5.12 shows a sample of the obtained accuracy and loss per epoch for our CNN model.

The optimum number of epochs should give good performance while avoiding both overfitting and underfitting. From Figure 5.12, the number of epochs that corresponds to the intersection between the train and validation Accuracy and Loss lines is 5 epochs. The performance evaluation results are reported using fourteen performance measures

(Accuracy, Error Rate, Loss, Precision, Recall, F1-Score, Area Under the Curve, Geometric-Mean, Specificity, Miss Rate, Fall-Out Rate, False-Discovery Rate, False-Omission Rate, and the Total Training Time). We use these metrics to evaluate the performance of the various detection models from different perspectives without the bias of depending on only a single measure.

```
Model: "sequential_33"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_29 (Embedding)     (None, 224, 200)          938400
_________________________________________________________________
dropout_66 (Dropout)         (None, 224, 200)          0
_________________________________________________________________
separable_conv1d_116 (Separa (None, 224, 64)           13464
_________________________________________________________________
separable_conv1d_117 (Separa (None, 224, 64)           4352
_________________________________________________________________
max_pooling1d_29 (MaxPooling (None, 74, 64)            0
_________________________________________________________________
separable_conv1d_118 (Separa (None, 74, 128)           8512
_________________________________________________________________
separable_conv1d_119 (Separa (None, 74, 128)           16896
_________________________________________________________________
global_average_pooling1d_29  (None, 128)               0
_________________________________________________________________
dropout_67 (Dropout)         (None, 128)               0
_________________________________________________________________
dense_37 (Dense)             (None, 1)                 129
=================================================================
Total params: 981,753
Trainable params: 981,753
Non-trainable params: 0
```

Figure 5.11 Model Summary of the Sequential Technique.



(a) (b)

Figure 5.12 Sample of the Obtained Accuracy and Loss per Epoch. (a) Samples of Accuracy per Epoch, (b) Samples of Loss per Epoch.

5.3.4 Detection Stage

5.3.4.1 Collecting Tweets

After establishing the connection, as discussed in *Subsubsection 2.1.2.3*, we used the search keywords as described in *Subsection 5.3.1*, which are the trends related to the coronavirus disease

(COVID-19), to collect the corresponding shared tweets. We used the streaming options of the API to collect real-time data.

For our current release of the COVID-19-FAKES dataset, we started our streaming process on February 04 and ended on March 10, 2020. We collected 5,224,912 tweets in 66 different languages, in addition to all the metadata associated with these tweets. We stored the collected data in real-time in our MySQL database. For the current work, we only consider the collected tweets in both Arabic and English languages with a total of 3,263,464 tweets. Figure 5.13 shows the distribution of tweets over the top-10 collected tweets' languages.

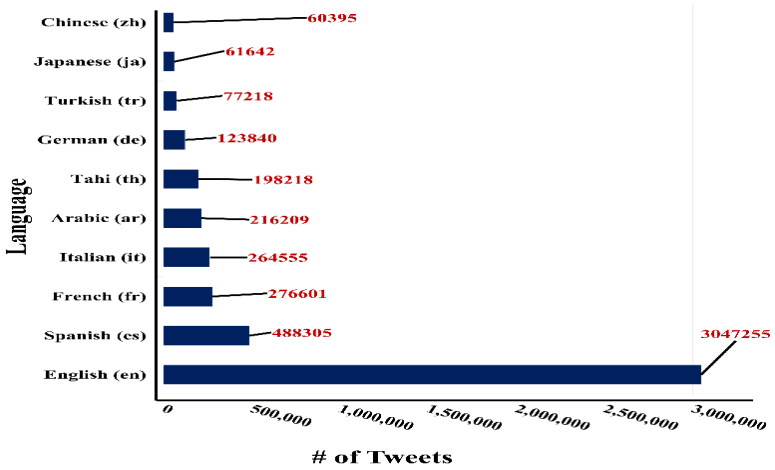


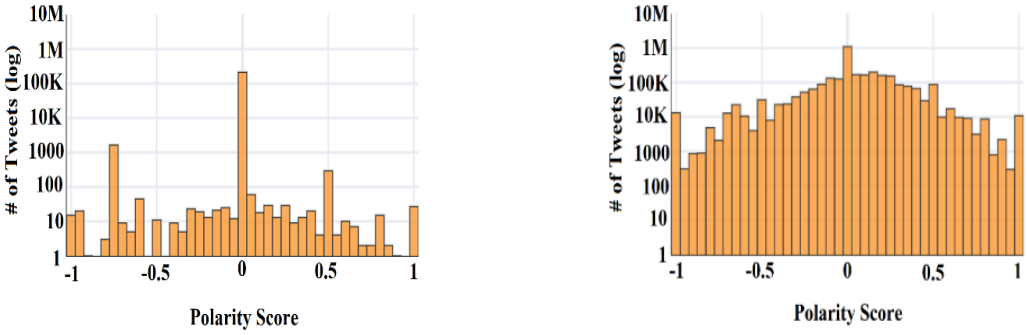
Figure 5.13 Distribution of Top-10 Collected Tweets' Languages.

It could be remarked that Twitter API allows only the return of 3,200 tweets per query at most. Moreover, due to some technical issues (connection errors, Internet problems, or power issues, etc.), we missed collecting some tweets during sometime periods for a few days.

5.3.4.2 Exploratory Data Analysis (EDA) on the Collected Tweets

For the collected English tweets, we have 3,047,026 tweets. These tweets were published by 993,320 users. Only around 2.2% (21,867 users) of these users are verified users on Twitter. Almost 32.523% (32,3058 users) of these users have incomplete profile information. The complete data indicate they are from 285 countries, and 118,247 locations, in addition to 64,259 undefined locations. While for the collected Arabic tweets, we found that we have 276,774 tweets. These tweets were published by 112,340 users. Only around 20.7% (23,241 users) of these users are verified users on Twitter. Almost 53.604% (60,219 users) of these users have incomplete profile information. The complete data indicate they are from 307 countries and 20,225 different locations, in addition to 9,787 undefined locations. It should be remarked that the country and location information contain some which are not real names, e.g., in the space, in my dreams, inside the car, zombie land, etc.

After calculating the sentiment polarity score of the collected tweets, we found that they are mostly neutral and deviated to the positive for both the Arabic and the English tweets, as shown in Figure 5.14. Most of the sentiment polarities are greater than or equal to 0, which means most of the tweets are positive. To investigate the daily rate of tweets, Figure 5.15 shows the tweet's publishing distribution over the collection period.



(a) (b)

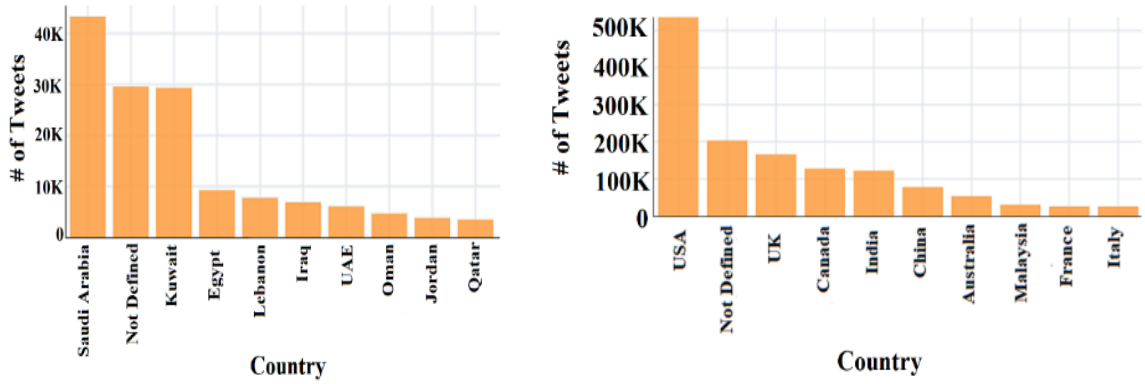
Figure 5.14 Tweet's Polarity Distribution. (a) Arabic Tweets, (b) English Tweets.

(a)

(b)

Figure 5.15 Tweet's Daily Rate. (a) Arabic Tweets, (b) English Tweets.

Figure 5.16 shows the distribution of tweets over the top-10 countries that engaged in publishing tweets in both languages.

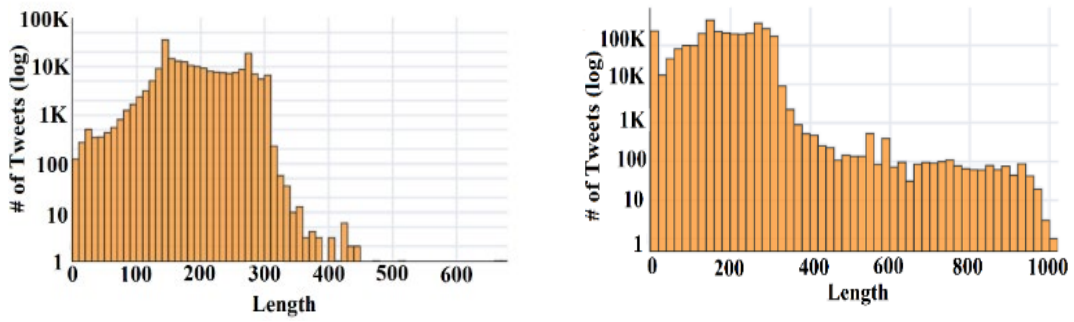


(a) (b)

Figure 5.16 Tweets' Country Distribution. (a) Arabic Tweets, (b) English Tweets.

It could be noticed that the most active tweeters are located in the United States of America and Saudi Arabia, while the next active group for both Arabic and English tweets comes from undefined countries (e.g., "The Kingdom of God", "???", "the planet of Kashyyyk", "Somewhere in this world", "Nowhere", "أم الدنيا", "أرض الله الواسعة", "فوق السحب", "انت عايزها فين", etc.).

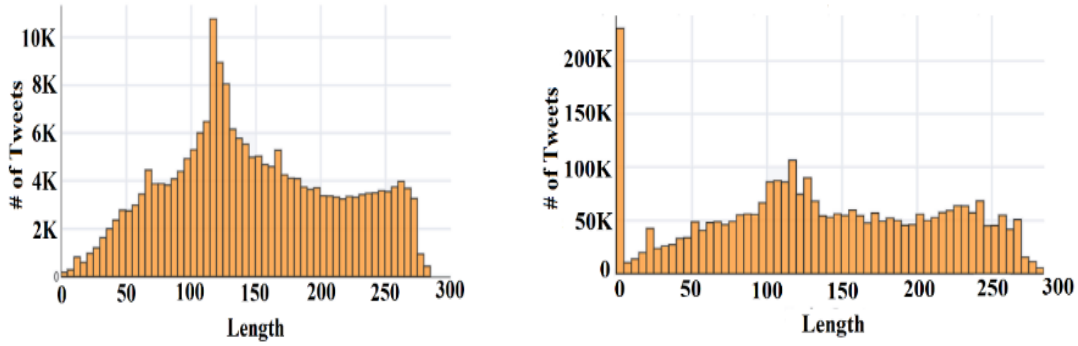
Figure 5.17 shows the text length distribution for both Arabic and English tweets. Despite the character limit for a tweet is 280 characters as imposed by Twitter, the calculated tweet length has shown many tweets exceed this limit. After further investigation of the collected tweets, we found that the URLs and HTML encoding (e.g., &amp, &lt, &gt, etc.) are affecting the character count although they are excluded from the tweets' length limit. This data needs cleaning too; in order to remove all the noise from the collected tweets.



(a) (b)

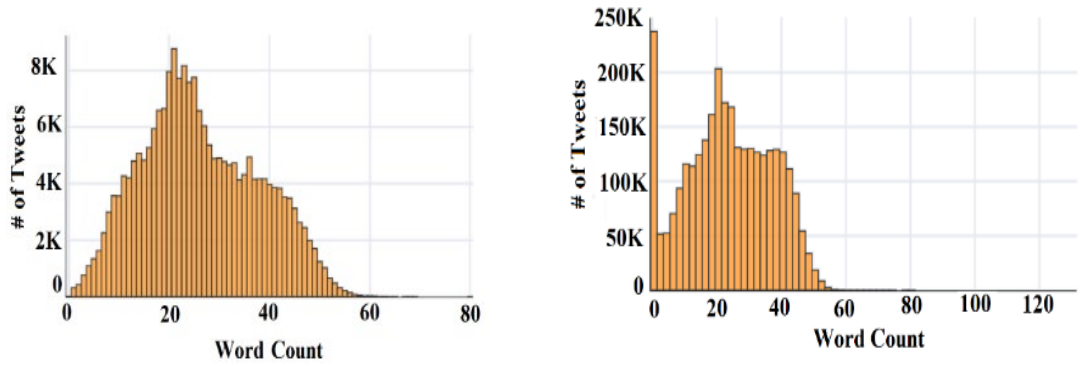
Figure 5.17 Tweets' Length Distribution. (a) Arabic Tweets, (b) English Tweets.

Figure 5.18 shows the tweet's text length distribution for both Arabic and English tweets after noise removal, while Figure 5.19 shows the distribution of the tweet's word count.



(a) (b)

Figure 5.18 Cleaned Tweets' Length Distribution. (a) Arabic Tweets, (b) English Tweets.

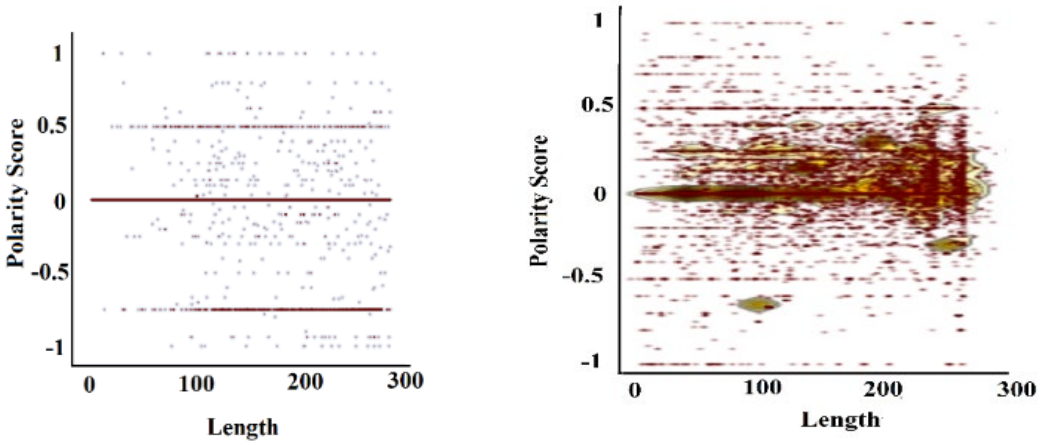


(a) (b)

Figure 5.19 Tweets' Word Count Distribution. (a) Arabic Tweets, (b) English Tweets.

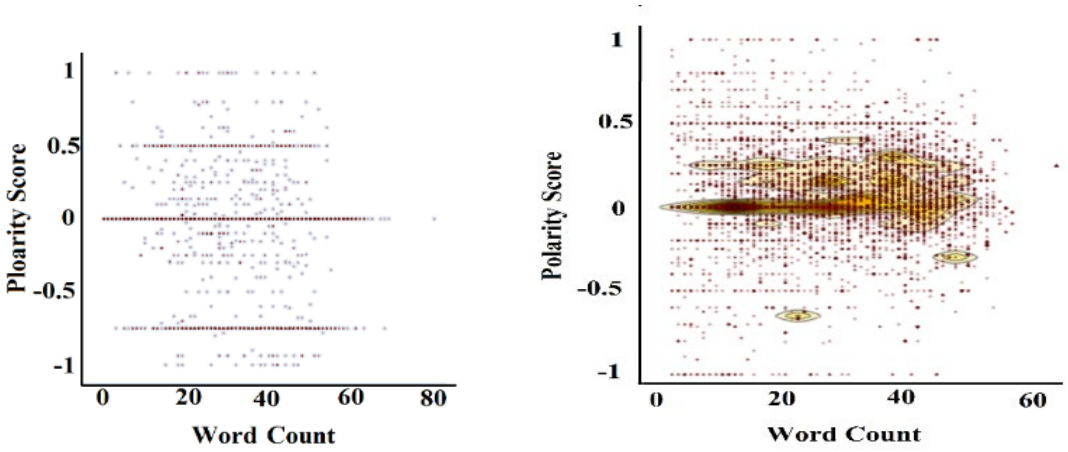
From Figure 5.18(a) and Figure 5.19(a), the length of most Arabic tweets are 100 to 180 characters in length, and 20 to 42 words. While from Figure 5.18(b) and Figure 5.19(b), the length of most English tweets is 90 to 120 characters in length, and 22 to 44 words. This means that some users like to leave long tweets, but most of the tweets are short ones.

We then investigated the relationship between the tweet's sentiment polarity and its text length as shown in Figure 5.20. Figure 5.21 shows the relation between the tweet's sentiment polarity and their word count.



(a) (b)

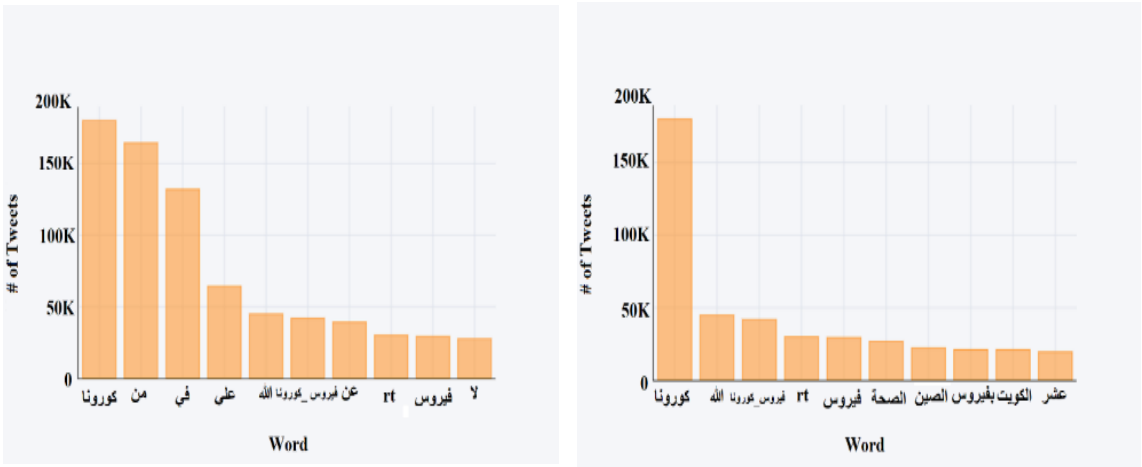
Figure 5.20 Distribution of Sentiment Polarity Score by Tweets' Text. (a) Arabic Tweets, (b) English Tweets.



(a) (b)

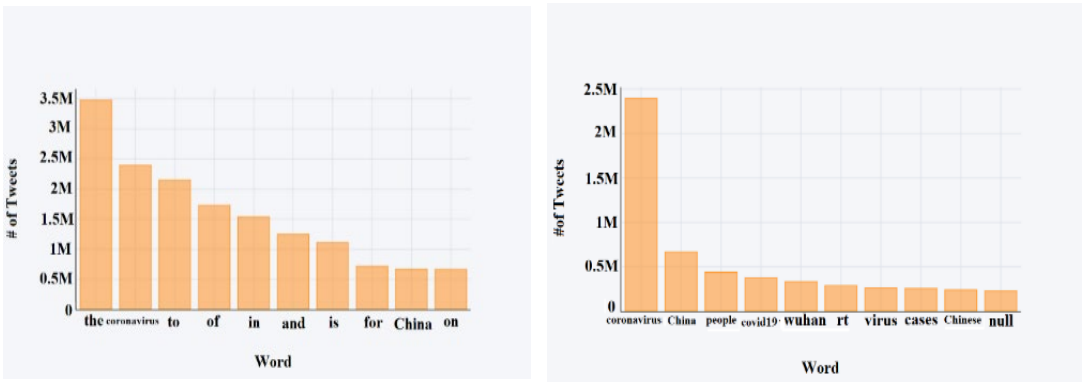
Figure 5.21 Distribution of Sentiment Polarity Score by Tweets vs Word Count. (a) Arabic Tweets, (b) English Tweets.

There are relatively few documents that are very positive or very negative. Tweets that have neutral to positive scores are more likely to be with text length greater than 50 and with a word count of more than 26 words. With this number of words, users can probably give a good impression. For further analysis of the tweets, we investigated the top-10 Unigrams, Bigrams, and Trigrams before and after removing the stop words for both Arabic and English data. Figure 5.22 and Figure 5.23 show the Unigram analysis, as a sample of our syntactic analysis on the collected data.



(a) (b)

Figure 5.22 Top-10 Arabic Unigrams. (a) Before Removing the Stop Words, (b) After Removing the Stop Words.



(a) (b)

Figure 5.23 Top-10 English Unigrams. (a) Before Removing the Stop Words, (b) After Removing the Stop Words.

We can notice from Figure 5.22 and Figure 5.23 that, many stop words have high frequencies, which may have a negative impact on detection results and increase the size of the extracted feature vector. Hence, stop words removal and data cleaning task are mandatory for that reason. This syntactic analysis gives a good indication of what are the most frequent words and the effect of removing unnecessary words from the feature vector to attain the goal of dimensionality reduction.

5.3.4.3 Detection Process

To carry out the detection process, we used the detection models obtained in the Model-Building stage, to assemble an ensemble prediction model (Voting Ensemble). Then, we passed the query strings through the ensemble model and obtained the results of each model. Finally, we performed hard voting on all the results to get the detection decision. For example, suppose that we are using these 3 classification algorithms (Alg1, Alg2, and Alg3) and our data belong to two classes

(Misleading and Real). We use the collected ground-truth data in building the detection models corresponding to each of Alg1, Alg2, and Alg3. Suppose that we need to predict the class of a query string (*Q*) as Real or Misleading. Assume that, after passing *Q* to these detection models, the resulting predictions from each model are as follows:

- Alg1 predicts class Misleading.
- Alg2 predicts class Real.
- Alg3 predicts class Real.

Two out of three classifiers predict class Real, so Real is the ensemble decision. Figure 5.24 shows a diagram of the employed voting ensemble method. For the query string to be classified as Real or Misleading, it must pass through the document preparation and the feature engineering stages as previously discussed in *Subsection 5.3.4*. It is then submitted to the voting ensemble model for the class assignment process.

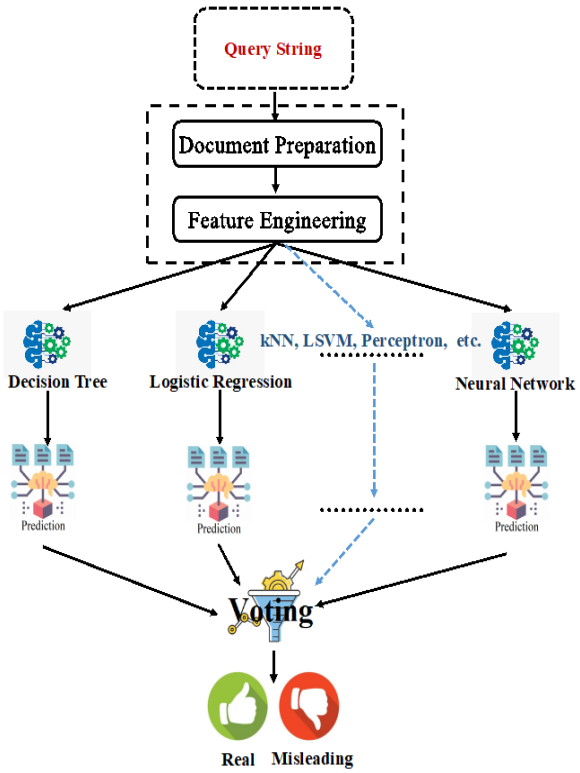


Figure 5.24 The Voting Ensemble Method.

After deploying the ensemble system, we used all the collected data to train the models for the best three classifiers. Then, we passed the query strings with unclassified data through the ensemble model and obtained the class label result. For the data to be classified as Real or Misleading, they must pass through all the steps as previously discussed in the following section.

5.4 Validation Results of the Detection Model on Ground-Truth Data

5.4.1 Utilizing Machine Learning Algorithms

To test the validity of the ground-truth data, a 5-fold cross-validation technique was used with the ground-truth data randomly split into two sets (80% of the documents as a training set, and the rest is the testing set). Table 5.1 shows the Accuracy (ACC), Error Rate (ERR), and the Area Under the Curve (AUC) of the obtained results from the ten classification algorithms (DT, MNB, BNB, LR, kNN, Perceptron, NN, LSVM, ERF, and XGBoost) when using TF and TF-IDF with character level, Unigram, Bigram, Trigram, and N-gram word size, and word embedding as feature extraction techniques. The best ACC, ERR, and AUC evaluations are from the NN classifier, between 93.75% to 99.68%, 0.32% to 6.25%, and 89.46% to 99.47%, respectively. The ACC and the ERR measures, despite being easy to compute with less complexity, have limitations in the evaluation of a classifier and discrimination process.

Table 5.1 Accuracy, Error Rate, and Area Under Curve of the Validation Results.

			Classification Algorithm									
Metric	Feature Extraction		DT	MNB	BNB	LR	kNN	Perceptron	NN	LSVM	ERF	XGBoost
Accuracy (%)	TF		99.04	98.29	95.09	99.36	96.42	98.93	99.68	99.52	99.52	99.15
	TF-IDF	Unigram	98.93	98.01	93.96	98.18	98.67	99.57	99.63	99.52	99.15	99.25
		Bigram	98.02	97.92	90.55	97.49	97.06	98.72	98.99	98.99	98.56	96.80
		Trigram	93.00	92.57	92.68	92.47	90.92	93.54	93.75	93.91	93.43	91.45
		N-gram (n = 2:3)	98.56	98.08	90.97	97.81	96.64	98.99	99.57	99.57	98.72	97.44
		Characters Level	99.20	97.97	91.35	98.08	99.36	99.52	99.63	99.63	99.09	99.41
	Word Embeddings		97.33	55.45	64.80	80.61	89.16	72.12	93.86	65.99	98.72	99.52
Error Rate (%)	TF		0.96	1.71	4.92	0.64	3.58	1.07	0.32	0.48	0.48	0.86
	TF-IDF	Unigram	1.07	1.99	6.04	1.82	1.34	0.43	0.37	0.48	0.86	0.75
		Bigram	1.98	2.08	9.46	2.51	2.94	1.28	1.02	1.02	1.44	3.21
		Trigram	7.00	7.43	7.32	7.53	9.08	6.46	6.25	6.09	6.57	8.55
		N-gram (n = 2:3)	1.44	1.92	9.03	2.19	3.37	1.02	0.43	0.43	1.28	2.56
		Characters Level	0.80	2.03	8.65	1.92	0.64	0.48	0.37	0.37	0.91	0.59
	Word Embeddings		2.67	44.55	35.20	19.39	10.84	27.89	6.14	34.01	1.28	0.48
Area Under the Curve (%)	TF		97.91	96.87	90.58	98.74	93.45	98.83	99.41	99.03	99.03	98.33
	TF-IDF	Unigram	97.83	97.36	88.75	97.75	98.65	99.16	99.28	99.12	98.14	98.58
		Bigram	97.09	97.62	84.60	98.13	97.75	98.00	99.07	99.07	98.06	96.75
		Trigram	92.86	94.16	92.29	94.98	88.41	95.06	95.06	95.70	94.34	93.26
		N-gram (n = 2:3)	97.15	97.67	84.66	98.43	96.91	97.96	99.34	99.34	97.68	97.24
		Characters Level	98.19	97.23	85.12	97.30	99.20	99.03	99.47	99.28	98.20	98.78
	Word Embeddings		94.68	63.05	65.02	70.54	82.48	60.20	89.46	57.59	97.85	99.21

One of the main limitations of ACC is that it produces less distinctive and less discriminable values. Consequently, its ability in selecting and determining the best classification algorithm is diminished. Besides, ACC is also less informative and biased towards minority class instances [222]. While for the AUC measure, it has been proven theoretically and empirically better than the ACC metric for evaluating a classifier's performance and discriminating an optimal solution during classification training [302]. It should be remarked that although the performance of AUC is excellent for evaluation and discrimination, its computational cost is high especially when dealing with large datasets [222].

Table 5.2 shows single evaluation measures (either positive or negative class): the Precision, Recall/True Positive Rate/Sensitivity, and Specificity/True Negative Rate. In terms of measuring the positive patterns that are correctly predicted from the total predicted patterns in a positive class, and the fraction of negative patterns that are correctly classified, the best results are 99.93% and 99.74%, respectively, for the DT classifier. In terms of the fraction of positive patterns that are correctly classified, the best result is 99.87% when using the LR classification algorithm. Therefore, based on the evaluation results and what is the most important measure desired, a user can decide which classification algorithm to use for a specific purpose.

Table 5.2 Precision, Sensitivity, and Specificity of the Validation Results.

			Classification Algorithm									
Metric	Feature Extraction		DT	MNB	BNB	LR	kNN	Perceptron	NN	LSVM	ERF	XGBoost
Precision (%)	TF		99.86	99.32	99.36	99.80	98.77	99.00	99.87	99.87	99.87	99.73
	TF-IDF	Unigram	99.73	98.46	99.43	98.47	98.67	99.87	99.87	99.80	99.66	99.73
		Bigram	98.71	98.12	99.85	97.10	96.65	99.25	98.92	98.92	98.92	96.82
		Trigram	93.07	91.90	92.88	91.48	92.28	92.84	93.13	93.09	93.00	90.76
		N-gram (n = 2:3)	99.59	98.34	99.70	97.45	96.48	99.73	99.73	99.73	99.46	97.56
		Characters Level	99.93	98.47	99.33	98.60	99.47	99.87	99.73	99.87	99.73	99.87
	Word Embeddings		99.45	95.68	94.73	81.85	95.80	84.87	97.51	85.63	99.33	99.73
Recall/True Positive Rate/Sensitivity (%)	TF		98.93	98.52	94.43	99.40	96.71	99.66	99.73	99.53	99.53	99.20
	TF-IDF	Unigram	98.99	99.06	92.95	99.26	99.66	99.60	99.66	99.60	99.26	99.33
		Bigram	98.78	99.25	88.13	99.80	99.73	99.12	99.80	99.80	99.25	99.19
		Trigram	98.44	99.32	98.24	99.73	96.54	99.46	99.39	99.66	99.12	99.25
		N-gram (n = 2:3)	98.59	99.26	88.93	99.87	99.40	98.99	99.73	99.73	98.93	99.26
		Characters Level	99.06	98.99	89.73	98.99	99.73	99.53	99.80	99.66	99.13	99.40
	Word Embeddings		97.18	46.11	59.06	97.18	90.34	79.06	94.70	69.12	99.06	99.66
Specificity /True Negative Rate (%)	TF		99.48	97.38	97.64	99.22	95.29	96.07	99.48	99.48	99.48	98.95
	TF-IDF	Unigram	98.58	93.98	97.91	93.98	94.76	99.48	99.48	99.22	98.69	98.95
		Bigram	95.23	92.97	99.50	88.95	87.19	97.24	95.98	95.98	95.98	87.94
		Trigram	72.86	67.51	72.11	65.58	70.10	71.61	72.86	72.61	72.36	62.56
		N-gram (n = 2:3)	98.43	93.46	98.95	89.79	85.86	98.95	98.95	98.95	97.91	90.31
		Characters Level	99.74	93.98	97.64	94.50	97.91	99.48	98.95	99.48	98.95	99.48
	Word Embeddings		97.91	91.89	87.17	15.97	84.56	45.03	90.58	53.40	97.38	98.95

Table 5.3 shows the F1-Score and Geometric-Mean validation results. The best results are 99.89% and 99.60% for both metrics F1-Score and Geometric-Mean when using the NN classifier. In general, these two metrics are considered as good discriminators and perform better than other metrics in optimizing classifiers, but only for binary classification problems and not for multiclass classification problems [253].

Table 5.3 F1-Score and Geometric-Mean of the Validation Results.

			Classification Algorithm									
Metric	Feature Extraction		DT	MNB	BNB	LR	kNN	Perceptron	NN	LSVM	ERF	XGBoost
F1-Score (%)	TF		99.39	98.92	96.83	99.60	97.73	99.33	99.80	99.70	99.70	99.46
	TF-IDF	Unigram	99.36	98.76	96.08	98.86	99.17	99.73	99.77	99.70	99.46	99.53
		Bigram	98.75	98.69	93.62	98.43	98.16	99.19	99.36	99.36	99.09	97.99
		Trigram	95.68	95.47	95.48	95.42	94.36	96.04	96.16	96.27	95.96	94.82
		N-gram (n = 2:3)	99.09	98.80	94.01	98.64	97.92	99.36	99.73	99.73	99.19	98.40
		Characters Level	99.49	98.73	94.29	98.79	99.60	99.70	99.77	99.77	99.43	99.63
	Word Embeddings		98.30	62.23	72.76	88.86	92.99	81.86	96.08	76.50	99.19	99.70
Geometric-Mean (%)	TF		99.20	97.95	96.02	99.31	96.00	97.85	99.60	99.50	99.50	99.07
	TF-IDF	Unigram	98.79	96.48	95.40	96.58	97.18	99.54	99.57	99.41	98.98	99.14
		Bigram	96.99	96.06	93.64	94.21	93.25	98.17	97.87	97.87	97.60	93.39
		Trigram	84.69	81.88	84.17	80.87	82.27	84.39	85.10	85.07	84.69	78.80
		N-gram (n = 2:3)	98.51	96.32	93.81	94.70	92.38	98.97	99.34	99.34	98.42	94.68
		Characters Level	99.40	96.45	93.60	96.72	98.82	99.50	99.38	99.57	99.04	99.44
	Word Embeddings		97.54	65.09	71.75	39.39	87.40	59.66	92.61	60.76	98.22	99.31

It should be remarked that Geometric-Mean aggregates both sensitivity and specificity measures for better discrimination between classes. As the objective of specificity usually conflicts with the objective of sensitivity, typically, the main goal of any classification algorithm is to improve the sensitivity, without sacrificing the specificity [253].

Finally, Table 5.4 shows different misclassification measures (Miss Rate, Fall-Out Rate, False Discovery Rate, and the False Omission Rate) for all the classification algorithms. These measures could help in choosing which algorithm to use in building a detection model. This choice is based on which measures that we want to keep as minimum as possible. For example, if we wanted to choose the detection model that had the lowest probability of false alarm (i.e., reducing the possibility of classifying a Real document as Misleading), we could choose the model that gives the lowest Fall-Out Rate. Whereas, if we wanted to reduce the rate of incorrectly classified Misleading documents as Real, we could choose the model that gives the lowest Miss Rate.

In terms of Miss Rate, which represents the False Negative Rate (FNR), the best result is 0.13 % when using the LR classifier. In terms of the Fall-Out Rate, which represents the False Positive Rate

(FPR) (also called False Alarm Rate (FAR)), the best result is 0.26% when using the DT classifier. It should be remarked that both the Miss Rate and the Fall-Out Rate are not sensitive to changes in data distributions and hence both metrics can be used with imbalanced data [253]. Additionally, the best obtained FDR is 0.07% when using the DT classifier, while the best obtained False Omission Rate result is 0.58% when using the LR classification algorithm.

Table 5.4 Miss Rate, Fall-Out Rate, False Discovery Rate, and False Omission Rate of the Validation Results.

			Classification Algorithm									
Metric	Feature Extraction		DT	MNB	BNB	LR	kNN	Perceptron	NN	LSVM	ERF	XGBoost
Miss Rate (%)	TF		1.07	1.48	5.57	0.60	3.29	0.34	0.27	0.47	0.47	0.81
	TF-IDF	Unigram	1.01	0.95	7.05	0.74	0.34	0.40	0.34	0.40	0.74	0.67
		Bigram	1.22	0.75	11.87	0.20	0.27	0.88	0.20	0.20	0.75	0.81
		Trigram	1.56	0.68	1.76	0.27	3.46	0.54	0.61	0.34	0.88	0.75
		N-gram (n = 2:3)	1.41	0.74	11.07	0.13	0.60	1.01	0.27	0.27	1.07	0.74
		Characters Level	0.94	1.01	10.27	1.01	0.27	0.47	0.20	0.34	0.87	0.60
	Word Embeddings		2.82	53.89	40.9	2.82	9.66	20.94	5.30	30.88	0.94	0.34
Fall-Out Rate (%)	TF		0.52	2.62	2.36	0.79	4.71	3.93	0.52	0.52	1.31	1.05
	TF-IDF	Unigram	1.42	6.02	2.09	6.02	5.24	0.52	0.52	0.79	1.31	1.05
		Bigram	4.77	7.04	0.50	11.06	12.81	2.76	4.02	4.02	4.02	12.06
		Trigram	27.14	32.49	27.89	34.42	29.90	28.39	27.14	27.39	27.64	37.44
		N-gram (n = 2:3)	1.57	6.55	1.05	10.21	14.14	1.05	1.05	1.05	2.09	9.69
		Characters Level	0.26	6.02	2.36	5.50	2.09	0.52	1.05	0.52	1.05	0.52
	Word Embeddings		2.09	8.12	12.83	84.03	15.45	54.97	9.42	46.60	2.62	1.05
False Discovery Rate (%)	TF		0.14	0.68	0.64	0.20	1.23	1.00	0.13	0.14	0.34	0.27
	TF-IDF	Unigram	0.27	1.54	0.57	1.53	1.33	0.14	0.13	0.20	0.34	0.27
		Bigram	1.29	1.88	0.15	2.90	3.35	0.75	1.08	1.08	1.08	3.18
		Trigram	6.93	8.10	7.12	8.53	7.72	7.16	6.87	6.91	7.00	9.24
		N-gram (n = 2:3)	0.41	1.66	0.30	2.55	3.52	0.27	0.27	0.27	0.54	2.44
		Characters Level	0.07	1.54	0.67	1.40	0.54	0.14	0.27	0.13	0.27	0.14
	Word Embeddings		0.55	4.32	5.27	18.15	4.20	15.13	2.49	14.37	0.67	0.27
False Omission Rate (%)	TF		4.04	5.58	18.20	2.32	11.86	1.34	1.04	1.81	2.84	3.08
	TF-IDF	Unigram	5.12	3.75	21.92	2.97	1.36	1.55	1.30	1.56	2.84	2.58
		Bigram	4.53	2.89	30.65	0.84	1.14	3.25	0.78	0.78	2.80	3.32
		Trigram	7.35	3.60	8.31	1.51	15.46	2.73	3.01	1.70	4.32	4.23
		N-gram (n = 2:3)	5.29	2.99	30.39	0.58	2.67	3.82	1.05	1.05	4.10	3.09
		Characters Level	3.54	4.01	29.09	3.99	1.06	1.81	0.79	1.30	3.33	2.31

From all the obtained results, it should be remarked that despite the NN, DT, and LR classifiers giving the best performance from different perspectives, all the results are satisfactory and indicate the validity of the collected ground-truth data. Hence, to get the benefits of different classification algorithms, we deploy the voting ensemble classifier in our detection model.

5.4.2 Utilizing Deep Learning Algorithms

To ensure the validity of the collected data, and to ensure that the model is not affected by data order, we shuffled the data first and then we split the samples into 80% as training and 20% for validation. We conducted two experiments, the first using 1 epoch and batch size of 100, while the second using 10 epochs with a batch size of 64. Table 5.5 shows the results for the first experiment, while Table 5.6 for the second one.

Table 5.5 Obtained Results for (Epochs = 1, Batch Size = 100).

Algorithm / Performance Metric	Sequential	CNN	RCNN	RNN		
				LSTM	GRU	Bidirectional
Total Training Time (sec)	51	4	4	41	35	74
Accuracy (%)	80.507	99.830	99.840	99.830	99.720	99.870
Error Rate (%)	21.910	0.170	0.160	0.170	0.280	0.130
Loss (%)	50.130	25.370	24.270	25.170	31.890	23.620
Recall (%)	99.940	98.220	99.730	99.250	79.060	94.810
Precision (%)	78.130	89.770	89.640	89.430	99.480	95.400
F1-Score (%)	86.690	93.810	94.410	94.080	88.100	95.110
Specificity (%)	0.080	99.850	99.840	99.840	99.990	99.940
Miss Rate (%)	0.060	1.780	0.270	0.750	20.940	5.180
Fall-Out Rate (%)	99.920	0.150	0.160	0.160	0.010	0.060
False-Discovery Rate (%)	21.880	10.230	10.360	10.570	0.520	4.600
False-Omission Rate (%)	75.000	0.020	0.000	0.010	0.280	0.070
Geometric-Mean (%)	70.692	99.030	99.790	99.550	88.910	97.340
Area Under the Curve (%)	50.01	99.030	99.790	99.550	89.530	97.340

Table 5.6 Obtained Results for (Epochs = 10, Batch Size = 64).

Algorithm / Performance Metric	Sequential	CNN	RCNN	RNN		
				LSTM	GRU	Bidirectional
Total Training Time (sec)	394	65	77	680	671	1077
Accuracy (%)	99.800	99.999	99.997	99.994	99.988	99.99
Error Rate (%)	0.200	0.001	0.003	0.006	0.012	0.01
Loss (%)	1.600	0.543	0.551	1.871	2.31	2.271
Recall (%)	99.830	100	99.932	99.727	99.591	99.523
Precision (%)	99.920	99.932	99.864	99.795	99.523	99.727
F1-Score (%)	99.880	99.966	99.898	99.761	99.557	99.624
Specificity (%)	99.660	99.999	99.998	99.997	99.994	99.996
Miss Rate (%)	0.170	0	0.068	0.273	0.409	0.477
Fall-Out Rate (%)	0.340	0.001	0.002	0.003	0.006	0.004
False-Discovery Rate (%)	0.080	0.068	0.136	0.205	0.477	0.273
False-Omission Rate (%)	0.680	0	0.001	0.004	0.006	0.006
Geometric-Mean (%)	99.750	100	99.965	99.862	99.792	99.759
Area Under the Curve (%)	99.790	100	99.965	99.862	99.792	99.759

It could be remarked from both Table 5.5 and Table 5.6 that all the results are of high quality. These results give a good indication that our model is capable to effectively distinguish between the Real and the Misleading samples. Moreover, the collected data, despite being unbalanced, gives very good results for all the evaluation metrics in both the conducted experiments except for the sequential model. Further investigation and analysis need to be done on the data to reach the best configuration for the system, by fine-tuning the hyperparameters (e.g., dropout rate, learning rate, embedding dimensions, etc.), and getting the optimum number of epochs and batch size for maximum gain from the system, that is, the best results in the lowest possible time.

It should be noted that from September 2020 until June 2021 (the completion of writing this detestation), many efforts are being done by researchers to fight against COVID-19 Infodemic. For example, [303], [304], [305], [306], [307], [308], and many others.

Chapter 6

Summary, Conclusions, and Future Work

6.1 Summary and Conclusions

In this dissertation, two novel frameworks were introduced. The first is to manage, process, and classify the bilingual textual content of social networks. The second is to fully assess the performance of binary classifiers. The proposed frameworks were utilized in detecting misleading health-related information (applied to COVID-19 as a real-life case study). These frameworks can be utilized also in detecting misleading information on any future global health issues, such as analyzing the information on the anticipated coming waves of Coronavirus, COVID-19 vaccines, and the vaccination process, at the time of writing this dissertation. Moreover, these frameworks can be used in analyzing shared information related to any topic of interest, by changing the sources of information that are deemed unbiased and reliable, e.g., instead of the WHO, UNICEF, and UN in our case study.

6.1.1 The Classification Framework

In this framework, the encoding of the processed text document is examined. Depending on the language, each text document is being pre-processed and then undergoes a procedure to select highly descriptive and representative features for the feature vector. Next, a novel approach for selecting hybrid features from online news textual metadata was introduced. This technique is based on dealing with the textual news document as a block without segmentation and selecting a set of user-based, post-based, social-based, and propagation-based features from the accompanying metadata in the documents, in addition to the linguistic features, to enrich the extracted feature vector. To examine the effectiveness of this feature selection technique, different classifiers are used, and performance is reported using various metrics: Accuracy, Precision, Recall, and F1-measure. The obtained results show noticeable improvement when using our technique in handling, selecting, and building the feature vector that is used in building detection models. From these reported results in this work, we could conclude that due to the application of this framework:

1- For the ISOT Fake News Dataset, the best accuracy of 100% was obtained when using Decision Trees and LSVM models with an enhancement of 8% compared with the results from [116].
2- For the LIAR dataset, the best accuracy of 62% was obtained when using Logistic Regression, SVM, and Naïve Bayes models with an enhancement of 6%, 6%, 2%, respectively, compared with the results from [116].

3- For the FA-KES dataset, the best result of 58% was obtained when applying the MNB model.

It could be remarked that, when using the ISOT Fake News Dataset, the obtained results are better than from the other two datasets. This could be as a result that ISOT's real news is collected from the Reuters website. So, after building the classification model, the decision is binary to classify whether the news document is a Reuters or Not-Reuters document.

Then, a novel approach for extracting more representative and discriminative features from textual data was introduced. We used this approach in the feature engineering stage in our model for fake news detection. This technique aims in determining the importance of each term in the data with respect to the classes that each document in the dataset that contains this term belongs to; this is done in four steps. First, the frequency of each term in different classes (TCF) is computed. Second, the number of documents in which the term appears per each class (CDF) is counted. Third, the inverse class-documents frequency (ICDF) that measures how important the term is in the document that belongs to a certain class with respect to documents in other classes is calculated. Finally, the term class importance (TCI) is obtained by multiplying TCF and ICDF; this is done for all the terms to generate a numerical feature vector for the detection tasks.

After building the feature vector, classification models using eight classification algorithms were built to examine the effectiveness of our proposed feature extraction technique against the traditional TF-IDF feature extraction technique. The performance is reported using well-known metrics: Accuracy, Precision, Recall, and F1-measure. Experimental results reported in this work show noticeable improvement when using our feature extraction technique in building the feature vector for fake news detection models. From the experimental results, we could conclude that our proposed technique gives better accuracy for all the classification algorithms used. Among these results, the best-obtained accuracies were 99.01% and 95.05% when using the DT classifier with our proposed technique. This is an improvement of 44% and 48%, as compared with the results from [120] and [109] on the LAIR and FA-KES datasets.

6.1.2 The Performance Evaluation Framework

For this second framework, a novel performance metric (MCAS) for evaluating the performance of binary classification algorithms was introduced. This metric aims in determining the ability of the classification algorithm to achieve higher detection rates for both dataset classes (Critical Success Score (CSS)) in the presence of the average of the unsuccessful detection rates (Critical Failure Score (CFS)) for both classes.

The CSS represents the average of the total number of correctly classified instances divided by the sum of the total number of instances of a relative class and the number of misclassified instances for that class. The CFS represents the total number of misclassified positive instances

divided by the sum of the total number of correctly classified instances and the number of misclassified positive instances, added to the total number of misclassified negative instances divided by the total number of correctly classified instances plus the number of misclassified negative instances. The range of the possible values of the MCAS is between [-1, 1]. The maximum value of 1 is obtained when the classification system is capable of correctly detecting all the positive and negative instances; the higher the MCAS value the better is the classification algorithm's performance. While the lowest -1 is obtained when the system is unable to correctly classify any of the samples.

The MCAS works independently regardless of the size of the used datasets and the distribution of class samples. It can differentiate between the performance of different supervised machine learning techniques, as illustrated by the numerical example in Subsection 3.2.3. Following the selection of the classification algorithm with the best MCAS value, the system undergoes a comprehensive performance evaluation to get all additional performance indicators.

6.1.3 The Case Study

For our case study, the proposed classification framework was employed to detect misleading information related to the COVID-19 outbreak. To train the detection model and due to the lack of available annotated COVID-19 misleading data, a set of ground-truth data from different reliable sources was collected. This collection was done by getting the ground-truth data from internationally reliable and independent institutions, such as WHO, UNICEF, and UN websites. The collected ground-truth data include all the textual data from written speeches, reports, and published news related to the COVID-19 outbreak from February 4, 2020, to March 10, 2020. Additionally, the Google Fact Check Tools API was utilized to collect the available prechecked facts from different fact-checking websites. These collected ground-truth data were employed to build detection models for various classification algorithms. Furthermore, a validation on the collected ground-truth data was done, to ensure its suitability in building a detection system. A 5-fold cross-validation was carried out on the ground-truth data using ten classification algorithms and seven feature extraction techniques and the validation results were reported for twelve evaluation measures.

The validation findings ascertained the validity of the acquired ground-truth data and provided useful insights into the performance of several classification algorithms on it. The Neural Network, Decision Tree, and Logistic Regression classifiers produced the best results. The Logistic Regression worked well in binary classification tasks and might be thought of as a one-layer Neural Network. Furthermore, because the Logistic Regression is a Perceptron with a sigmoid function, the findings of the Logistic Regression and the Perceptron were comparable. The classification algorithms that produced the best three results were used to form the ensemble detection model in the final configuration of the detection system.

The proposed detection system was employed to annotate 3.263M Arabic and English COVID-19 related tweets and made them publicly available to the research community (https://github.com/mohaddad/COVID-FAKES). COVID-19-FAKES is an automatically annotated misleading Information Twitter dataset about COVID-19. The full dataset characteristics, data collection steps, and how the annotation procedure were all introduced. The conducted Exploratory Data Analysis shows the main features of the COVID-19-FAKES dataset. This work could help researchers in understanding the dynamics behind the COVID-19 outbreak on Twitter. Furthermore, it could help in studies related to sentiment analysis, the propagation of misleading information related to this outbreak, users' behavior during the crisis, the detection of botnets, the performance of different classification algorithms with various feature extraction techniques that are used in text mining.

Additionally, we proposed an ensemble deep learning system for detecting misleading information related to COVID-19. We introduced the detailed steps for building our system. To improve the performance of the proposed ensemble detection system, we followed the same data preparation and preprocessing step, along with a features engineering step. We deployed word embedding based on a pre-trained word embedding list in addition to the existing word impeding in the input layer of the used techniques. We conducted two experiments with a different number of epoch and different batch sizes. Results were evaluated using fourteen performance measures. The obtained results are promising and indicate the quality and validity of the trusted information collected for building misleading-information detection models.

6.2 Future Work

This section discusses some recommendations that can either belong to the same direction of the current work, or to a new area of work that belongs to web text document classification in general.

6.2.1 For the Proposed Classification and Performance Evaluation Frameworks

1- Utilize the WordNet ontology for building the extracted feature vector to improve the classification accuracy.
2- Apply further dimension reduction techniques to minimize the size of the used feature vector without affecting the overall system performance.
3- Extend the proposed work to handle tweets that are written in languages other than Arabic and English, such as French, Spanish, German, Chinese, Arabizi, etc.

4- Implement the web text document classification system using parallel platforms, such as Apache Spark, to handle a large amount of data in less time without affecting the overall system performance.
5- Combine the decisions of machine learning and deep learning algorithms to improve the resultant accuracies.
6- Perform further investigation and analysis on the FA-KES fake news dataset.
7- Deploy syntactic similarity measures and semantic similarity measures to enrich the extracted feature vector with some syntactic and semantic features.
8- Perform further investigation to examine the effectiveness of the TCI feature extraction technique on other textual datasets other than news datasets as well as multi-class datasets.
9- Combine two or more feature extraction techniques, e.g., TF-IDF with TCI, for obtaining further enhancement in the classification results.
10- Perform further investigation to extend the proposed evaluation metric to cover multi-class datasets problems.

6.2.2 For the Case Study

1- Extend our proposed framework to include other trusted information sources such as the "International Committee of the Red Cross (ICRC)" [309].
2- Enrich the collected ground-truth data by including published information from the Twitter official accounts of the WHO [310], UNICEF [311], UN [312], and ICRC [313].
3- Enhance the web scraping process to eliminate 'irrelevant' data from the collected ground-truth data, for example, removing the contact-us information, the organization’s location, the descriptions that are associated with images, etc.
4- Extend the proposed framework to cover data written in other languages than English, to overcome the shortage of available detection systems, for example, cover the data written in French, Spanish, Chinese, etc.
5- Use the proposed framework for detecting misleading information, shared or re-twitted on Twitter in a near real-time manner.
6- Perform further investigation to provide a hybrid system that deploys both machine learning and deep learning techniques for building robust misleading information detection systems.

Bibliography

[1] S. Aghaei, M. A. Nematbakhsh, and H. K. Farsani, "Evolution of the World Wide Web: From WEB 1.0 TO WEB 4.0," *Int. J. Web Semantic Tech. (IJWesT),* vol. 3, no. 1, pp. 1-10, Jan. 2012. doi: https://doi.org/10.5121/ijwest.2012.3101.

[2] B. Batrinca, and P. C. Treleaven, "Social Media Analytics: A Survey of Techniques, Tools, and Platforms," *AI Society,* vol. 30, no. 1, pp. 89-116, Feb. 2015. doi: https://doi.org/10.1007/s00146-014-0549-4.

[3] R. Krikorian, "Introducing Twitter Data Grants," Twitter, 5 Feb. 2014. Accessed: 18 Nov. 2020. [Online]. Available: https://blog.twitter.com/engineering/en_us/a/2014/introducing-twitter-data-grants.html.

[4] K. Gligorić, A. Anderson, and R. West, "How Constraints Affect Content: The Case of Twitter's Switch from 140 to 280 Characters," in *Proc. 12th Int. AAAI Conf. Web Social Media (ICWSM 2018)*, Palo Alto, California, USA, Jun. 2018. pp. 596-599, doi: https://ojs.aaai.org/index.php/ICWSM/article/view/15079.

[5] M. Sikandar, "100 Social Media Statistics for 2021," Statusbrew Blog, 03 Mar. 2021. Accessed: 18 Mar. 2021. [Online]. Available: https://statusbrew.com/insights/social-media-statistics/.

[6] M. K. Elhadad, K. F. Li, and F. Gebali, "Fake News Detection on Social Media: A Systematic Survey," in *Proc. 2019 IEEE Pacific Rim Conf. Comm., Comp., Signal Processing*, Victoria, BC, Canada, Aug. 2019. pp. 1-8, doi: https://doi.org/10.1109/PACRIM47961.2019.8985062.

[7] K. Shu, A. Sliva, S. Wang, J. Tang, and H. Liu, "Fake News Detection on Social Media: A Data Mining Perspective," *SIGKDD Explor. Newsl.,* vol. 19, no. 1, pp. 22-36, Sep. 2017. doi: https://doi.org/10.1145/3137597.3137600.

[8] S. Kumar, and N. Shah, "False Information on Web and Social Media: A Survey," *arXiv:1804.08559,* Apr. 2018. doi: https://arxiv.org/abs/1804.08559.

[9] T. Zeitzoff, "How Social Media is Changing Conflict," *J. Conflict Resol.,* vol. 61, no. 9, pp. 1970-1991, Aug. 2017. doi: https://doi.org/10.1177%2F0022002717721392.

[10] O. Nicoara, and D. White, "A Contextual Model Of The Secessionist Rebellion in Eastern Ukraine," *arXiv preprint arXiv:1606.02748,* Jun. 2016.

[11] T. A. Shaban, L. Hexter, and J. D. Choi, "Event Analysis on the 2016 US Presidential Election Using Social Media," in *Proc. Int. conf. social informatics*, Oxford, UK, Sep. 2017. pp. 201-217, doi: https://doi.org/10.1007/978-3-319-67217-5_13.

[12] M. K. Elhadad, K. F. Li, and F. Gebali, "Detecting Misleading Information on COVID-19," *IEEE Access,* vol. 8, pp. 165201 - 165215, Sep. 2020. doi: https://doi.org/10.1109/ACCESS.2020.3022867.

[13] J. D. O'Cain, Language Ideologies, Multilingualism, and Social Media, London: Palgrave Macmillan, 2017, pp. 23-60.

[14] M. Y. Damanhouri, "Language Use in Computer-Mediated Communication and Users' Social Identity," *English Linguistics Research,* vol. 7, no. 3, pp. 16-25, Sep. 2018. doi: https://doi.org/10.5430/elr.v7n3p16.

[15] N. Kordzadeh, and D. K. Young, "How Social Media Analytics can Inform Content Strategies," *J. Comp. Info. Sys.,* pp. 1-14, Apr. 2020. doi: https://doi.org/10.1080/08874417.2020.1736691.

[16] A. Shastri, and M. Deshpande, "A Review of Big Data and Its Applications in Healthcare and Public Sector," *Big Data Analytics in Healthcare,* vol. 66, pp. 55-66, Oct. 2019. doi: https://doi.org/10.1007/978-3-030-31672-3_4.

[17] M. K Elhadad, K. F. Li, F. Gebali, "An Ensemble Deep Learning Technique to Detect COVID-19 Misleading Information," in *Proc. NBiS 2020*, BC, Canada, Aug. 2020. pp. 163-175, doi: https://doi.org/10.1007/978-3-030-57811-4_16.

[18] V. L. Rubin, Y. Chen and N. J. Conroy, "Deception Detection for News: Three Types of Fakes," *Proc. Assoc, Info. Sci. Tech.,* vol. 52, no. 1, pp. 1-4, Feb. 2016. doi: https://doi.org/10.1002/pra2.2015.145052010083.

[19] P. Ristoski, and H. Paulheim, "Semantic Web in Data Mining and Knowledge Discovery: A Comprehensive Survey," *J. Web Semant.,* vol. 36, pp. 1-22, Jan. 2016. doi: https://doi.org/10.1016/j.websem.2016.01.001.

[20] K. N. Singh, H. M. Devi, and A. K. Mahanta, "Document Representation Techniques and their Effect on the Document Clustering and Classification: A Review," *Int. J. Adv. Res. Comp. Sci.,* vol. 8, no. 5, pp. 1780-1784, Jun. 2017. doi: https://doi.org/10.26483/ijarcs.v8i5.3822.

[21] M. Ravi, M. E. Naidu, and G. Narsimha, "Extracting Multimedia Information and Knowledge Discovery Using Web Mining: Challenges and Research Directions," *Int. J. Applied Eng. Research,* vol. 14, no. 12, pp. 2830-2836, Dec. 2019.

[22] M. O. Samuel, A. I. Tolulope, and O. O. Oyejoke, "A Systematic Review of Current Trends in Web Content Mining," in *Proc. 3rd Int. Conf. Sci. Sustainable Develop. (ICSSD 2019)*, Ota, Nigeria, Aug. 2019. pp. 1-14, doi: https://doi.org/10.1088/1742-6596/1299/1/012040.

[23] S. Dhawan, K. Singh, and V. Khanchi, "Critical Analysis of Social Networks with Web Data Mining," *IJITKM Special Issue (ICFTEM-2014),* pp. 107-111, May. 2014, [Online].

Available: https://silo.tips/download/critical-analysis-of-social-networks-with-web-data-mining.

[24] S. Sharma, D. Soni, and A. K. Sharma, "Explorative Study of Web Data Mining Techniques and Tools: A Review," *Int. J. Comp. Sci. Tech.,* vol. 8, no. 1, pp. 43-47, Mar. 2017. doi: https://doi.org/10.17148/IJARCCE.2016.5535.

[25] M. K. Elhadad, K. M. Badran, G. I. Salama, "Towards Ontology-Based Web Text Document Classification," in *Proc. Int. Conf. Aerospace Sci. Aviation Tech. (ASAT)*, Cairo, Egypt, Apr. 2017. pp. Article 61: 1-8, doi: https://dx.doi.org/10.21608/asat.2017.22749.

[26] S. A. Salloum, M. Al-Emran, A. Abdel Monem, and K. Shaalan, "Using Text Mining Techniques for Extracting Information from Research Articles," *Studies Comput. Intell.,* vol. 740, pp. 373-397, Nov. 2017. doi: https://doi.org/10.1007/978-3-319-67056-0_18.

[27] G. S. Naganath, and M. S. Pralhad, "Web Mining-Types, Applications, Challenges, and Tools," *Int. J. of Adv. Res. Comp. Eng. Tech.,* vol. 4, no. 5, pp. 2013-2015, May 2015.

[28] S. Vijiyarani, and E. Suganya, "Research Issues in Web Mining," *Int. J. Comp. Aided Tech.,* vol. 2, no. 3, pp. 55-64, Jul. 2015. doi: https://doi.org/10.5121/ijcax.2015.2305.

[29] J. B. Upadhyay, and S. V. Patel, "A Review Analysis of Preprocessing Techniques in Web Usage Mining," *Int. J. Eng. Research Tech.,* vol. 4, no. 4, pp. 1160-1166, Apr. 2015. doi: http://dx.doi.org/10.17577/IJERTV4IS041348.

[30] C. Rana, "A Study of Web Usage Mining Research Tools," *Int. J. Adv. Net. App.,* vol. 3, no. 6, pp. 1422-1429, Mar. 2012.

[31] A. Rosyidah, I. Surjandari, and Zulkarnain, "Mining Web Log Data for Personalized Recommendation System," in *Proc. 6th Int. Conf. Info. Comm. Tech.*, Bandung, Indonesia, May 2018. pp. 441-446, doi: https://doi.org/10.1109/ICoICT.2018.8528799.

[32] S. Yadav, K. Ahmad, and J. Shekar, "Analysis of Web Mining Applications and Beneficial Areas," *IIUM Eng. J.,* vol. 12, no. 2, pp. 185-195, Oct. 2011. doi: https://doi.org/10.31436/iiumej.v12i2.141.

[33] V. Rao M, and V. Murthy G, "DSS for Web Mining Using Recommendation System," *Web Data Mining Dev. Knowl. Based DSS,* pp. 22-34, 2017. doi: https://doi.org/10.4018/978-1-5225-1877-8.ch003.

[34] G. Sreedhar, and A. A. Chari, "Development of Efficient Decision Support System Using Web Data Mining," *Web Data Mining Dev. Knowl. Based DSS,* pp. 1-11, 2017. doi: https://doi.org/10.4018/978-1-5225-1877-8.ch001.

[35] V. García-Díaz, J. P. Espada, R. G. Crespo, B. C. P. G-Bustelo, and J. M. C. Lovelle, "An Approach to Improve the Accuracy of Probabilistic Classifiers for Decision Support

Systems in Sentiment Analysis," *Applied Soft Comput.,* vol. 67, pp. 822-833, Jun. 2018. doi: https://doi.org/10.1016/j.asoc.2017.05.038.

[36] N. A. K. Dam, T. L. Dinh, and W. Menvielle, "Marketing Intelligence from Data Mining Perspective - A Literature Review," *Int. J. Innov. Manag. Tech.,* vol. 10, no. 5, pp. 184-190, Oct. 2019. doi: https://doi.org/10.18178/ijimt.2019.10.5.859.

[37] K. Kasemsap, "Electronic Commerce and Decision Support Systems: Theories and Applications," *Improv. E-Commerce Web App. Through Business Intell. Tech.,* pp. 251-270, 2018. doi: https://doi.org/10.4018/978-1-5225-3646-8.ch011.

[38] Y. Wang, L. Wang, M. Rastegar-Mojarad, S. Moon, F. Shen et al., "Clinical Information Extraction Applications: A Literature Review," *J. Biomed. Informat.,* vol. 77, pp. 34-49, Jan. 2018. doi: https://doi.org/10.1016/j.jbi.2017.11.011.

[39] W. Sun, Z. Cai, Y. Li, F. Liu et al., "Data Processing and Text Mining Technologies on Electronic Medical Records: A Review," *J. Healthcare Eng.,* vol. 2018, pp. 1-9, Apr. 2018. doi: https://doi.org/10.1155/2018/4302425.

[40] I. Azzi, A. Jeghal, A. Radouane, and H. Tairi, "Personalized E-Learning Systems Based On Automatic Approach," in *Proc. 2019 Int. Conf. Wireless Tech. Embedded Intell. Sys. (WITS)*, Fez, Morocco, Apr. 2019. pp. 1-6, doi: https://doi.org/10.1109/WITS.2019.8723847.

[41] K. Chaudhary, and N. Gupta, "E-Learning Recommender System for Learners: A Machine Learning-Based Approach," *Int. J. Math. Eng. Manag. Sci.,* vol. 4, no. 4, pp. 957-967, 2019. doi: https://doi.org/10.33889/ijmems.2019.4.4-076.

[42] V. Sathiyamoorthi, "An Intelligent System for Predicting a User Access to a Web-Based E-Learning System Using Web Mining," *Int. J. Inf. Tech. Web Eng. (IJITWE),* vol. 15, no. 1, pp. 75-94, Jan. 2020. doi: https://doi.org/10.4018/IJITWE.2020010106.

[43] Q. Al-Maatouk, M. S. Bin Othman, M. E. Rana, and W. M. Al-Rahmi, "A Cloud-Based Framework for E-Government Implementation in Developing Countries," *Int. J. Eng. Tech.,* vol. 7, no. 4, pp. 3018-3021, 2018. doi: https://doi.org/10.14419/ijet.v7i4.14740.

[44] N. Gigi, and A.Kaur, "Sentimental Analysis On Social Feeds to Predict the Elections," in *Proc. 2018 1st Int. Conf. Secure Cyber Comput. Comm. (ICSCCC)*, Jalandhar, India, Dec. 2018. pp. 514-517, doi: https://doi.org/10.1109/ICSCCC.2018.8703347.

[45] J. V. Chen, M. A. Elakhdary and Q. Ha, "The Continuance Use of Social Network Sites for Political Participation: Evidence from Arab Countries," *J. Glob. Info. Tech. Manag.,* vol. 22, no. 3, pp. 156-178, 2019. doi: https://doi.org/10.1080/1097198X.2019.1642021.

[46] F. Ali, F. H. Khan, S. Bashir, and U. Ahmad, "Counter-Terrorism on Online Social Networks Using Web Mining Techniques," in *Proc. Int. Conf. Intell. Tech. App.*,

Bahawalpur, Pakistan, Oct. 2018. pp. 240-250, doi: https://doi.org/10.1007/978-981-13-6052-7_21.

[47] M. Farsi, A. Daneshkhah, A. H. Far, O. Chatrabgoun, and R. Montasari, "Crime Data Mining, Threat Analysis, and Prediction," *Adv. Sci. Tech. Security App.,* pp. 183-202, Nov. 2018. doi: https://doi.org/10.1007/978-3-319-97181-0_9.

[48] S. H. Mokhtar, G. Muruti, Z. Ibrahim, F. Abdul Rahim, and H. Kasim, "A Review of Evidence Extraction Techniques in Big Data Environment," in *Proc. 2018 Int. Conf. Smart Comp. Electr. Enterprise (ICSCEE)*, Shah Alam, Malaysia, Jul. 2018. pp. 1-7, doi: https://doi.org/10.1109/ICSCEE.2018.8538437.

[49] S. Singh, P. Arya, A. Patel, and A. K. Tiwari, "Social Media Analysis through Big Data Analytics: A Survey," in *Proc. 2nd Int. Conf. Adv. Comp. SW Eng. (ICACSE-2019)*, Sultanpur, India, Apr. 2019. pp. 77-80, doi: https://dx.doi.org/10.2139/ssrn.3349561.

[50] A. S. Hadi, and E. M. Abdulshaheed, "Twitter Information Representation Using Resource Description Framework," *Int. J. Eng. Tech.,* vol. 8, no. 1.5, pp. 497-502, 2019.

[51] A. Zubiaga, A. Aker, K. Bontcheva, M. Liakata, and R. Procter, "Detection and Resolution of Rumours in Social Media: A Survey," *ACM Comput. Surveys (CSUR),* vol. 52, no. 2, p. 32, Feb. 2018. doi: https://doi.org/10.1145/3161603.

[52] C. De Maio, G. Fenza, V. Loia, and F. Orciuoli, "Unfolding Social Content Evolution Along with Time and Semantics," *Future Gen. Comp. Sys.,* vol. 66, pp. 146-159, Jan. 2017. doi: https://doi.org/10.1016/j.future.2016.05.039.

[53] R. Zafarani, M. A. Abbasi, and H. Liu, Social media mining: an introduction, USA: Cambridge University Press, 2014.

[54] A. Sapountzi, and K. E. Psannis, "Social Networking Data Analysis Tools and Challenges," *Future Generation Comp. Sys.,* vol. 86, pp. 893-913, Sep. 2018. doi: https://doi.org/10.1016/j.future.2016.10.019.

[55] N. V. Sailaja, L. Padmasree, and N. Mangathayaru, "Survey of Text Mining Techniques, Challenges, and their Applications," *Int. J. Comp. App. (IJCA),* vol. 146, no. 11, pp. 30-35, Jul. 2016.

[56] R. D. Peng, R Programming for Data Science, Lean Publishing, Dec. 2019.

[57] M. Podhoranyi, and L. Vojacek, "Social Media Data Processing Infrastructure by Using Apache Spark Big Data Platform: Twitter Data Analysis," in *Proc. 2019 4th Int. Conf. Cloud Comp. IoT*, Tokyo, Japan, 2019. pp. 1-6, doi: https://doi.org/10.1145/3361821.3361825.

[58] K. K. Kapoor, K.Tamilmani, N. P. Rana, P. Patil et al., "Advances in Social Media Research: Past, Present, and Future," *Info. Sys. Front.,* vol. 20, pp. 531-558, Jun. 2018. doi: https://doi.org/10.1007/s10796-017-9810-y.

[59] F. Vis, "Studies in Social Data: How Industry Uses Social Media for Communications and Research," 24 Mar. 2017. Accessed: 26 Mar. 2021. [Online]. Available: http://eprints.lse.ac.uk/id/eprint/70621.

[60] M. K. Elhadad, K. M. Badran, and G. I. Salama, "A Novel Approach for Ontology-Based Feature Vector Generation for Web Text Document Classification," *Int. J. SW Innov.,* vol. 6, no. 1, pp. 1-10, Jan. 2018. doi: https://doi.org/10.4018/IJSI.2018010101.

[61] S. N. Saleh, and Y. El-Sonbaty, "A Feature Selection Algorithm with Redundancy Reduction for Text Classification," in *Proc. 22nd Int. Symp. Comp. Inf. Sci.*, Ankara, Turkey, Nov. 2007. pp. 1-6, doi: 10.1109/ISCIS.2007.4456849.

[62] R. Blumberg, and S. Atre, "The Problem with Unstructured Data," *DM Review,* vol. 13, pp. 42-46, Feb. 2003.

[63] G. D. S. Martino, and A. Sperduti, "Mining Structured Data," *IEEE Comput. Intell. Magazine,* vol. 5, no. 1, pp. 72-49, Feb. 2010. doi: https://doi.org/10.1109/MCI.2009.935308.

[64] J. Hendler, "Data Integration for Heterogenous Datasets," *Big Data,* vol. 2, no. 4, pp. 205-215, Dec. 2014. doi: https://doi.org/10.1089/big.2014.0068.

[65] M. Bărbulescu, R. Grigoriu, I. Halcu, G. Neculoiu et al, "Integrating of Structured, Semi-Structured and Unstructured Data in Natural and Build Environmental Engineering," in *Proc. 2013 11th RoEduNet Int. Conf.*, Sinaia, Romania, Jan. 2013. pp. 1-4, doi: https://doi.org/10.1109/RoEduNet.2013.6511738.

[66] O. Rusu, I. Halcu, O. Grigoriu, G. Neculoiu et al., "Converting Unstructured and Semi-Structured Data into Knowledge," in *Proc. 2013 11th RoEduNet Int. Conf.*, Sinaia, Romania, Jan. 2013. pp. 1-4, doi: https://doi.org/10.1109/RoEduNet.2013.6511736.

[67] "Kaggle Datasets," Accessed: 14 Jul. 2020. [Online]. Available: https://www.kaggle.com/search?q=datasets.

[68] "GitHub Datasets," Accessed: 28 Jun. 2020. [Online]. Available: https://github.com/search?q=datasets.

[69] "Hugging Face," Accessed: 23 Oct. 2020. [Online]. Available: https://huggingface.co/datasets.

[70] M. Hall, E. Frank, G. Holmes, B. Pfahringer et al., "The WEKA Data Mining Software: An Update," *ACM SIGKDD Explorations Newsletter,* vol. 11, no. 1, pp. 10-18, Nov. 2009. doi: https://doi.org/10.1145/1656274.1656278.

[71] W. McKinney, Python for data analysis: Data wrangling with Pandas, NumPy, and IPython (1st edt.), California, USA: O'Reilly Media, Inc., Oct. 2012.

[72] D. Li, "Chapter 4: Packages and Datasets," in *Basic R Guide for NSC Statistics*, Bookdown org., Oct. 2020. [Online]. Available: https://bookdown.org/dli/rguide/packages-and-datasets.html.

[73] "MATLAB R2020b Sample Data Sets," MathWorks, Accessed: 21 Sep. 2020. [Online]. Available: https://www.mathworks.com/help/stats/sample-data-sets.html.

[74] "Languages Used on the Internet," Wikipedia, 2019. Accessed: 16 Jan. 2020. [Online]. Available: https://en.wikipedia.org/wiki/Languages_used_on_the_Internet.

[75] A. Kornai, "Digital Language Death," *PLOS ONE,* vol. 8, no. 10, p. e77056, Oct. 2013. doi: https://doi.org/10.1371/journal.pone.0077056.

[76] "Most Common Languages Used on the Internet as of January 2020, by Share of Internet Users," Statista, 07 Jan. 2020. Accessed: 30 Apr. 2021. [Online]. Available: https://www.statista.com/statistics/262946/share-of-the-most-common-languages-on-the-internet/.

[77] R. Chau, and CH. Yeh, "A Multilingual Text Mining Approach to Web Cross-Lingual Text Retrieval," *Knowledge-Based Sys.,* vol. 17, pp. 219-227, May 2004. doi: https://doi.org/10.1016/j.knosys.2004.04.001.

[78] N. Vanetik, and M. Litvak, Multilingual Text Analysis: Challenges, Models, And Approaches, USA: World Scientific, Feb. 2019.

[79] R. Steinberger, "A Survey of Methods to Ease the Development of Highly Multilingual Text Mining Applications," *Lang. Resources Eval.,* vol. 46, pp. 155-176, Jun. 2012. doi: https://doi.org/10.1007/s10579-011-9165-9.

[80] D. B. Carlo, M. Souza, C. C. Xavier, and L. Oliveira, "Multilingual Open Information Extraction: Challenges and Opportunities," *Info.,* vol. 10, no. 7, pp. 1-25, Jul. 2019. doi: https://doi.org/10.3390/info10070228.

[81] M. O. Ibrahim, and I. Bud, "Translated vs Non-Translated Method for Multilingual Hate Speech Identification in Twitter," *Int. J. Advan. Sci. Eng. Info. Tech.,* vol. 9, no. 4, pp. 1116-1123, Jan 2019.

[82] "Cloud Translation pricing," Accessed: 29 Jul. 2021. [Online]. Available: https://cloud.google.com/translate/pricing#charged-characters.

[83] L. A. Ballesteros, Resolving ambiguity for cross-language information retrieval: A dictionary approach, USA: The University of Massachusetts at Amherst, Sep. 2001.

[84] M. Rogati, J. S. McCarley, and Y. Yang, "Unsupervised Learning of Arabic Stemming Using a Parallel Corpus," in *Proc. 41st annual meeting Assoc. Comput. Lingu.*, Sapporo, Japan, Jul. 2003. pp. 391-398, doi: https://doi.org/10.3115/1075096.1075146.

[85] G. de Melo, and S. Siersdorfer, "Multilingual Text Classification Using Ontologies," *Advan. Info. Retrieval,* vol. 4425, pp. 541-548, Apr. 2007. doi: https://doi.org/10.1007/978-3-540-71496-5_49.

[86] M. Braschler, and P. Schäuble, "Using Corpus-Based Approaches in a System for Multilingual Information Retrieval," *Info. Retrieval,* vol. 3, pp. 273-284, Oct. 2000. doi: https://doi.org/10.1023/A:1026525127581.

[87] K. Dashtipour, S. Poria, A. Hussain, E. Cambria, et al., "Multilingual Sentiment Analysis: State of the Art and Independent Comparison of Techniques," *Cognitive Comput.,* vol. 8, no. 4, pp. 757-771, 2016. doi: http://dx.doi.org/10.1007/s12559-016-9421-9.

[88] A. Balahur, and M. Turchi, "Multilingual Sentiment Analysis using Machine Translation?," in *Proc. 3rd Workshop on Comput. Appr. to Subj. SA*, Korea, Jul. 2012. pp. 52-60, doi: https://dl.acm.org/doi/abs/10.5555/2392963.2392976.

[89] A. W. Baur, "Harnessing the Social Web to Enhance Insights into People's Opinions in Business, Government and Public Administration," *Info. Sys. Frontiers,* vol. 19, no. 2, pp. 231-251, Apr. 2017. doi: https://doi.org/10.1007/s10796-016-9681-7.

[90] D. Vilares, M. A. Alonso, and C. G. Rodríguez, "Supervised Sentiment Analysis in Multilingual Environments," *Info. process. Manag.,* vol. 53, no. 3, pp. 595-607, May 2017. doi: https://doi.org/10.1016/j.ipm.2017.01.004.

[91] B. Khaleghi, A. Khamis, F. O. Karray, and S. N. Razavi, "Multisensor Data Fusion: A Review of the State-of-the-Art," *Info. Fusion,* vol. 14, no. 1, pp. 28-44, Jan. 2013. doi: https://doi.org/10.1016/j.inffus.2011.08.001.

[92] J. A. Balazs, and J. D. Velásquez, "Opinion Mining and Information Fusion: A Survey," *Info. Fusion,* vol. 27, pp. 95-110, Jan. 2016. doi: https://doi.org/10.1016/j.inffus.2015.06.002.

[93] S. Salam, P. Brandt, J. Holmes, and L. Khan, "Distributed Framework for Political Event Coding in Real-Time," in *Proc. 2018 2nd Europ. Conf. Elec. Eng. Comp. Sci. (EECS)*, Bern, Switzerland, Dec. 2018. pp. 266-273, doi: https://doi.org/10.1109/EECS.2018.00057.

[94] S. Ge, H. Isah, F. Zulkernine, and S. Khan, "A Scalable Framework for Multilevel Streaming Data Analytics using Deep Learning," in *Proc. 2019 IEEE 43rd Annual Comp. SW App. Conf. (COMPSAC)*, Milwaukee, WI, USA, Jul. 2019. pp. 189-194, doi: https://doi.org/10.1109/COMPSAC.2019.10205.

[95] G. Guibon, L. Ermakova, H. Seffih, A. Firsov et al., "Multilingual Fake News Detection with Satire," in *Proc. CICLing: Int. Conf. Comput. Linguistics Intell. Text Process.*, La Rochelle, France, Apr. 2019. pp. 1-11, doi: https://halshs.archives-ouvertes.fr/halshs-02391141.

[96] A. S. Adekotujo, J. Y. Lee, A. O. Enikuomehin, M. Mazzara et al., "Bi-lingual Intent Classification of Twitter Posts: A Roadmap," *Advances Intell. Sys. Comput.,* vol. 925, pp. 1-9, Mar. 2019. doi: https://doi.org/10.1007/978-3-030-14687-0_1.

[97] CH. Lee, and HC. Yang, "A Multilingual Text Mining Approach Based on Self-Organizing Maps," *Appl. Intell.,* vol. 18, pp. 295-310, May 2003. doi: https://doi.org/10.1023/A:1023250105036.

[98] L. S. Larkey, F. Feng, M. Connell, and V. Lavrenko, "Language-Specific Models in Multilingual Topic Tracking," in *Proc. 27th annual int. ACM SIGIR conf. Research develop. info. retrieval*, Sheffield, UK, Jul. 2004. pp. 402-409, doi: https://doi.org/10.1145/1008992.1009061.

[99] R. Steinberger, "Challenges and Methods for Multilingual Text Mining," in *Proc. 7th Int. Conf. Lang. Resources Eval. (LREC 2010)*, Paris, France, May 2010. pp. 19-21, doi: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.167.4724.

[100] S. Kashef, H. Nezamabadi-pour, and B. Nikpour, "Multi-Label Feature Selection: A Comprehensive Review and Guiding Experiments," *WIREs Data Mining Knowl. Discov.,* vol. 8, no. 2, pp. 1-29, Jan. 2018. doi: https://doi.org/10.1002/widm.1240.

[101] A. Naresh, and S. Sreepada, "Automatic Classification of Bing Answers User Verbatim Feedback," in *Proc. AICC 2018*, Hyderabad, India, Nov. 2018. pp. 449-458, doi: https://doi.org/10.1007/978-981-13-1580-0_43.

[102] G. Dong, and H. Liu, Feature engineering for machine learning and data analytics, Florida, USA: CRC Press Inc., Apr. 2018.

[103] K. Shu, D. Mahudeswaran, and H. Liu, "FakeNewsTracker: A Tool for Fake News Collection, Detection, and Visualization," *Comp. Math. Org. Th.,* vol. 25, no. 1, pp. 60-71, Oct. 2019. doi: https://doi.org/10.1007/s10588-018-09280-3.

[104] Y. Li, and T. Li, "Feature Selection and Evaluation," in *Feature Engineering for Machine Learning and Data Analytics*, Florida, USA, CRC Press, Mar. 2018, pp. 191-220.

[105] G. Chandrashekar, and F. Sahin, "A Survey on Feature Selection Methods," *Comp. Elec. Eng.,* vol. 40, no. 1, pp. 16-28, Jan. 2014. doi: https://doi.org/10.1016/j.compeleceng.2013.11.024.

[106] Y. Li, T. Li, and H. Liu, "Recent Advances in Feature Selection and its Applications," *Knowl. Inf. Sys.,* vol. 53, pp. 551-577, Dec. 2017. doi: https://doi.org/10.1007/s10115-017-1059-8.

[107] S. Wijeratne, A. Sheth, S. Bhatt, L. Balasuriya et al., "Feature Engineering for Twitter-based Applications," in *Feature Engineering for Machine Learning and Data Analytics*, Florida, USA, CRC Press, Mar. 2018, pp. 359-393.

[108] M. Iqbal, M. M. Abid, M. N. Khalid, and A. Manzoor, "Review of Feature Selection Methods for Text Classification," *Int. J. Adv. Comp. Res. (IJACR),* vol. 10, no. 49, pp. 138-152, Jul. 2020. doi: https://doi.org/10.19101/IJACR.2020.1048037.

[109] M. K. Elhadad, K. F. Li, and F. Gebali, "A Novel Approach for Selecting Hybrid Features from Online News Textual Metadata for Fake News Detection," in *Proc. 3PGCIC*, Antwerp, Belgium, Nov. 2019. pp. 914-925, doi: https://doi.org/10.1007/978-3-030-33509-0_86.

[110] C. Castillo, M. Mendoza, and B. Poblete, "Information Credibility on Twitter," in *Proc. 20th Int. Conf. WWW*, Hyderabad, India, Mar. 2011. pp. 675-684, doi: https://doi.org/10.1145/1963405.1963500.

[111] U. Kursuncu, M. Gaur, U. Lokala, K. Thirunarayan, A. Sheth, and I. B. Arpinar, "Predictive Analysis on Twitter: Techniques and Applications," *Emer. Res. Challen. Opport. Comput. SNA and Mining,* pp. 67-104, Sep. 2019. doi: https://doi.org/10.1007/978-3-319-94105-9_4.

[112] N. A. Ghani, S. Hamid, I. A. T. Hashem, and E. Ahmed, "Social Media Big Data Analytics: A Survey," *Comp. Human Behavior,* vol. 101, pp. 417-428, Dec. 2019. doi: https://doi.org/10.1016/j.chb.2018.08.039.

[113] N. Ruchansky, S. Seo, and Y. Liu, "CSI: A Hybrid Deep Model for Fake News Detection," in *Proc. 2017 ACM Conf. Info. Knowledge Manag.*, Singapore, Singapore, Nov. 2017. pp. 797-806, doi: https://doi.org/10.1145/3132847.3132877.

[114] M. Potthast, J. Kiesel, K. Reinartz, J. Bevendorff et al., "A Stylometric Inquiry into Hyperpartisan and Fake News," in *Proc. 56th Annual Meet. Assoc. Comput. Lingu. (Volume 1: Long Papers)*, Melbourne, Australia, Jul. 2018. pp. 231-240, doi: http://dx.doi.org/10.18653/v1/P18-1022.

[115] U. Khurana, "The Linguistic Features of Fake News Headlines and Statements," *Diss. Master's thesis, University of Amsterdam,* 2017.

[116] H. Ahmed, I. Traore, and S. Saad, "Detection of Online Fake News Using N-Gram Analysis and Machine Learning Techniques," in *Proc. Int. Conf. Intell. Secure Dependable Sys. Dist. Cloud Environ.*, Vancouver, BC, Canada, Oct. 2017. pp. 127-138, doi: https://doi.org/10.1007/978-3-319-69155-8_9.

[117] H. Ahmed, I. Traore, and S. Saad, "Detecting Opinion Spams and Fake News Using Text Classification," *Sec. Priv.,* vol. 1, no. 1, pp. 1-15, Dec. 2018. doi: https://doi.org/10.1002/spy2.9.

[118] M. Ott, Y. Choi, C. Cardie, and J. T. Hancock, "Finding Deceptive Opinion Spam by any Stretch of the Imagination," in *Proc. 49th Annual Meet. Assoc. Comput. Lingu. Human Lang. Tech.*, Oregon, USA, Jun. 2011. pp. 309-319, doi: https://dl.acm.org/doi/10.5555/2002472.2002512.

[119] B. Al Asaad, and M. Erascu, "A Tool for Fake News Detection," in *Proc. 2018 20th Int. Symp. Symb. Num. Algo. Sci. Comput. (SYNASC)*, Timisoara, Romania, Sep. 2018. pp. 379-386, doi: https://doi.org/10.1109/SYNASC.2018.00064.

[120] J. Y. Khan, M. T. I. Khondaker, A. Iqbal, and S. Afroz, "A Benchmark Study on Machine Learning Methods for Fake News Detection," *arXiv preprint, arXiv:1905.04749,* May 2019.

[121] A. P. S. Bali, M. Fernandes, S. Choubey, and M. Goel, "Comparative Performance of Machine Learning Algorithms for Fake News Detection," in *Proc. ICACDS*, Ghaziabad, India, Jul. 2019. pp. 420-430, doi: https://doi.org/10.1007/978-981-13-9942-8_40.

[122] I. Guyon, and A. Elisseeff, "An Introduction to Feature Extraction," in *Feature Extraction, Foundations, and Applications*, Berlin, Heidelberg, Springer, Feb. 2009, pp. 1-25.

[123] S. Khalid, T. Khalil, and S. Nasreen, "A Survey of Feature Selection and Feature Extraction Techniques in Machine Learning," in *Proc. 2014 Sci. Inf. Conf.*, London, UK, Aug. 2014. pp. 372-378, doi: https://doi.org/10.1109/SAI.2014.6918213.

[124] C. Geigle, Q. Mei, and C. X. Zhai, "Feature Engineering for Text Data," in *Feature engineering for machine learning and data analytics*, Florida, USA, CRC Press, Mar. 2018, pp. 15-54.

[125] J. Beel, B. Gipp, S. Langer, and C. Breitinger, "Paper Recommender Systems: A Literature Survey," *Int. J. Digit. Libr.,* vol. 17, no. 4, pp. 305-338, Nov. 2016. doi: https://doi.org/10.1007/s00799-015-0156-0.

[126] F. Ren, and M. G. Sohrab, "Class-Indexing-Based Term Weighting for Automatic Text Classification," *Info. Sci.,* vol. 239, pp. 109-125, Feb. 2013. doi: https://doi.org/10.1016/j.ins.2013.02.029.

[127] T. Sabbah, A. Selamat, M. H. Selamat, R. Ibrahim, and H. Fujita, "Hybridized Term-Weighting Method for Dark Web Classification," *Neurocomputing,* vol. 173, no. 3, pp. 1908-1926, Jan. 2016. doi: https://doi.org/10.1016/j.neucom.2015.09.063.

[128] C. Wan, Y. Wang, Y. Liu, J. Ji, and G. Feng, "Composite Feature Extraction and Selection for Text Classification," *IEEE Access,* vol. 7, pp. 35208-35219, Mar. 2019. doi: https://doi.org/10.1109/ACCESS.2019.2904602.

[129] U. Bhattacharjee, P. K. Srijith, and M. S. Desarkar, "Term Specific TF-IDF Boosting for Detection of Rumours in Social Networks," in *Proc. 2019 COMSNETS*, Bengaluru, India, May 2019. pp. 726-731, doi: https://doi.org/10.1109/COMSNETS.2019.8711427.

[130] T. Dogan, and A. K. Uysal, "Improved Inverse Gravity Moment Term Weighting for Text Classification," *Expert Syst. Appl.,* vol. 130, pp. 45-59, Sep. 2019. doi: https://doi.org/10.1016/j.eswa.2019.04.015.

[131] T. Dogan, and A. K. Uysal, "On Term Frequency Factor in Supervised Term Weighting Schemes for Text Classification," *Arab J. Sci. Eng.,* vol. 44, pp. 9545-9560, May 2019. doi: https://doi.org/10.1007/s13369-019-03920-9.

[132] I. Alsmadi, and G. K. Hoon, "Term Weighting Scheme for Short-Text Classification: Twitter Corpuses," *Neural Comput. Applic.,* vol. 31, no. 8, pp. 3819-3831, Aug. 2019. doi: https://doi.org/10.1007/s00521-017-3298-8.

[133] T. Sabbah, A. Selamat, M. H. Selamat, F. S. Al-Anzi et al., "Modified Frequency-Based Term Weighting Schemes for Text Classification," *Appl. Soft Comput.,* vol. 58, pp. 193-206, Sep. 2017. doi: https://doi.org/10.1016/j.asoc.2017.04.069.

[134] C. H. Chen, "Improved TFIDF in Big News Retrieval: An Empirical Study," *Pattern Recognit. Lett.,* vol. 93, pp. 113-122, Jul. 2017. doi: https://doi.org/10.1016/j.patrec.2016.11.004.

[135] S. Ghosh, and M. S. Desarkar, "Class-Specific TF-IDF Boosting for Short-Text Classification: Application to Short-Texts Generated During Disasters," in *Proc. WWW '18*, Lyon, France, Apr. 2018. pp. 1629-1637, doi: https://doi.org/10.1145/3184558.3191621.

[136] H. Fan, and Y. Qin, "Research on Text Classification Based on Improved TF-IDF Algorithm," in *Proc. NCCE*, Chongqing, China, May 2018. pp. 501-506, doi: https://doi.org/10.2991/ncce-18.2018.79.

[137] C. Supriyanto, H. A. Nugroho, and T. B. Adji, "A Global Weighting Scheme Based on Intra-Class and Inter-Class Term Distributions in Bag-of-Visual Words Image Classification," *IAENG Int. J. Comp. Sci.,* vol. 45, no. 2, pp. 228-236, Jun. 2018.

[138] R. Lakshmi, and S. Baskar, "Novel Term Weighting Schemes for Document Representation Based on Ranking of Terms and Fuzzy Logic with Semantic Relationship of Terms," *Expert Syst. Appl.,* vol. 137, pp. 493-503, Dec. 2019. doi: https://doi.org/10.1016/j.eswa.2019.07.022.

[139] G. James, D. Witten, T. Hastie, and R. Tibshirani, "Resampling Methods," in *An Introduction to Statistical Learning. Springer Texts in Statistics*, New York, USA, Springer, Apr. 2013, pp. 175-201.

[140] "Keras," Open source, Accessed: 01 Jul. 2020. [Online]. Available: https://keras.io/.

[141] "Pytorch," Open source, Accessed: 01 Jul. 2020. [Online]. Available: https://pytorch.org/.

[142] "Scikit-Learn: Machine Learning in Python," Open source, Accessed: 01 Jul. 2020. [Online]. Available: https://scikit-learn.org/stable/index.html.

[143] "R Interface to Tensorflow," RStudio, Accessed: 28 Jun. 2020. [Online]. Available: https://tensorflow.rstudio.com/.

[144] "TIDYMODELS," RStudio, Accessed: 28 Jun. 2020. [Online]. Available: https://www.tidymodels.org/.

[145] "Spark Machine Learning Library (MLlib)," RStudio, Accessed: 28 Jun. 2020. [Online]. Available: https://spark.rstudio.com/mlib/.

[146] "Statistics and Machine Learning Toolbox," MathWorks, Accessed: 12 Jul. 2020. [Online]. Available: https://www.mathworks.com/products/statistics.html.

[147] D. Sharma, and A. Kumar, "Levels and Classification Techniques for Sentiment Analysis: A Review," *Adv. Comm. Comp. Tech.,* vol. 668, pp. 333-345, Aug. 2020. doi: https://doi.org/10.1007/978-981-15-5341-7_27.

[148] A. Mittal, and S. Patidar, "Sentiment Analysis on Twitter Data: A Survey," in *Proc. 2019 7th Int. Conf. Comp. Comm. Manag.*, Bangkok, Thailand, Jul. 2019. pp. 91-95, doi: https://doi.org/10.1145/3348445.3348466.

[149] K. Schouten, and F. Frasincar, "Survey on Aspect-Level Sentiment Analysis," *IEEE Trans. Knowl. Data Eng.,* vol. 28, no. 3, pp. 813-830, Mar. 2016. doi: https://doi.org/10.1109/TKDE.2015.2485209.

[150] B. Verma, and R. S. Thakur, "Sentiment Analysis Using Lexicon and Machine Learning-Based Approaches: A Survey," in *Proc. Int. Conf. Recent Advan. Comp.Comm.*, Bhopal, India, Apr. 2018. pp. 441-447, doi: https://doi.org/10.1007/978-981-10-8198-9_46.

[151] A. Yadav, D. K. Vishwakarma, "Sentiment Analysis Using Deep Learning Architectures: A Review," *AI Review,* vol. 53, pp. 4335-4385, Dec. 2019. doi: https://doi.org/10.1007/s10462-019-09794-5.

[152] F. Hemmatian, and M. K. Sohrabi, "A Survey on Classification Techniques for Opinion Mining, and Sentiment Analysis," *AI Rev.,* vol. 52, pp. 1495-1545, Dec. 2019. doi: https://doi.org/10.1007/s10462-017-9599-6.

[153] L. Chen, C. Lee, and M. Chen, "Exploration of Social Media for Sentiment Analysis Using Deep Learning," *Soft Comp.,* vol. 24, pp. 8187-8197, Oct. 2019. doi: https://doi.org/10.1007/s00500-019-04402-8.

[154] M. K. Elhadad, K. F. Li, and F. Gebali, "Sentiment Analysis of Arabic and English Tweets," in *Proc. WAINA 2019*, Matsue, Japan, Mar. 2019. pp. 334-348, doi: https://doi.org/10.1007/978-3-030-15035-8_32.

[155] H. Soong, N. B. A. Jalil, R. K. Ayyasamy, and R. Akbar, "The Essential of Sentiment Analysis and Opinion Mining in Social Media: Introduction and Survey of the Recent Approaches and Techniques," in *Proc. 2019 IEEE 9th Symp. Comp. App. Indust. Elect. (ISCAIE)*, Malaysia, Apr. 2019. pp. 272-277, doi: https://doi.org/10.1109/ISCAIE.2019.8743799.

[156] W. Medhat, A. Hassan, and H. Korashy, "Sentiment Analysis Algorithms and Applications: A Survey," *Ain Shams Eng. J.,* vol. 5, pp. 1093-1113, May 2014. doi: http://dx.doi.org/10.1016/j.asej.2014.04.011.

[157] H. Liang, X. Sun, Y. Sun, and Y. Gao, "Text Feature Extraction Based on Deep Learning: A Review," *EURASIP J. Wireless Comm. Net.,* vol. 211, pp. 1-12, Dec. 2017. doi: https://doi.org/10.1186/s13638-017-0993-1.

[158] I. Muhammad, and Z. Yan, "Supervised Machine Learning Approaches: A Survey," *ICTACT J. Soft Comp.,* vol. 5, no. 3, pp. 946-952, Apr. 2015. doi: https://doi.org/10.21917/IJSC.2015.0133.

[159] F. H. Khan, U. Qamar, and S. Bashir, "A Semi-Supervised Approach to Sentiment Analysis Using Revised Sentiment Strength Based on SentiWordNet," *Knowl. Info. Sys.,* vol. 51, pp. 851-872, Jun. 2017. doi: https://doi.org/10.1007/s10115-016-0993-1.

[160] V. L. S. Lee, K. H. Gan, T. P. Tan, and R. Abdullah, "Semi-Supervised Learning for Sentiment Classification Using Small Number of Labeled Data," *Procedia Comp. Sci.,* vol. 161, pp. 577-584, Jan. 2019. doi: https://doi.org/10.1016/j.procs.2019.11.159.

[161] X. Zhu, and A. B. Goldberg, "Introduction to Semi-Supervised Learning," *Synthesis lec. AI Mach.Learn.,* vol. 3, no. 1, pp. 1-130, Jun. 2009. doi: https://doi.org/10.2200/S00196ED1V01Y200906AIM006.

[162] M. S. Neethu, and R. Rajasree, "Sentiment Analysis in Twitter Using Machine Learning Techniques," in *Proc. IEEE 4th Int. Conf. Comput. Comm. Network. Tech. (ICCCNT)*, Tiruchengode, India, Jul. 2013. pp. 1-5, doi: https://doi.org/10.1109/ICCCNT.2013.6726818.

[163] R. Jindal, R. Malhotra, and A. Jain, "Techniques for Text Classification: Literature Review and Current Trends," *Webology,* vol. 12, no. 2, pp. 1-28, Dec. 2015.

[164] K. Ravi, and V. Ravi, "A Survey on Opinion Mining and Sentiment Analysis: Tasks, Approaches, and Applications," *Knowledge-Based Sys.,* vol. 89, pp. 14-46, Nov. 2015. doi: https://doi.org/10.1016/j.knosys.2015.06.015.

[165] P. Yang, and Y. Chen, "A Survey on Sentiment Analysis by Using Machine Learning Methods," in *Proc. 2017 IEEE 2nd Info. Tech. Net. Elect. Automat. Contr. Conf. (ITNEC)*, Chengdu, China, Dec. 2017. pp. 117-121, doi: https://doi.org/10.1109/ITNEC.2017.8284920.

[166] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up?: Sentiment Classification Using Machine Learning Techniques," in *Proc. ACL-02 Conf. Empirical Meth. NLP (EMNLP)*, Philadelphia, USA, Jul. 2002. pp. 79-86, doi: https://doi.org/10.3115/1118693.1118704.

[167] S. Das, R. K. Behera, and S. K. Rath, "Real-Time Sentiment Analysis of Twitter Streaming Data for Stock Prediction," *Procedia Comp. Sci.,* vol. 132, pp. 956-964, Jun. 2018. doi: https://doi.org/10.1016/j.procs.2018.05.111.

[168] SB. Kim, KS. Han, HC. Rim, and S. H. Myaeng, "Some Effective Techniques for Naive Bayes Text Classification," *IEEE Trans. Knowl. Data Eng.,* vol. 18, no. 11, pp. 1457 - 1466, Nov. 2006. doi: https://doi.org/10.1109/TKDE.2006.180.

[169] Z. Niu, Z. Yin, and X. Kong, "Sentiment Classification for Microblog by Machine Learning," in *Proc. 2012 4th Int. Conf. Comput. Info. Sci.*, Chongqing, China, Aug. 2012. pp. 286-289, doi: https://doi.org/10.1109/ICCIS.2012.276.

[170] L. Barbosa, and J. Feng, "Robust Sentiment Detection on Twitter from Biased and Noisy Data," in *Proc. 23rd Int. Conf. Comput. Lingu. Posters*, Beijing, China, Aug. 2010. pp. 36-44,

[171] A. Celikyilmaz, D. Hakkani-Tür, and J. Feng, "Probabilistic Model-Based Sentiment Analysis of Twitter Messages," in *Proc. 2010 IEEE Spoken Lang. Tech.Workshop*, Berkeley, CA, USA, 2011. pp. 79-84, doi: https://doi.org/10.1109/SLT.2010.5700826.

[172] M. Nabil, M. Aly, and A. Atiya, "Astd: Arabic Sentiment Tweets Dataset," in *Proc. 2015 Conf. Empirical Methods NLP*, Lisbon, Portugal, Sep. 2015. pp. 2515-2519, doi: http://dx.doi.org/10.18653/v1/D15-1299.

[173] I. Guellil, A. Adeel, F. Azouaou, A. Hachani et al., "Arabizi Sentiment Analysis Based on Transliteration and Automatic Corpus Annotation," in *Proc. 9th Workshop Comput. Appro. Subject. Sent. Soc. Media Anal.*, Brussels, Belgium, Oct. 2018. pp. 335-341, doi: https://doi.org/10.18653/v1/P17.

[174] R. Baly, G. Badaro, G. El-Khoury, R. Moukalled et al., "A Characterization Study of Arabic Twitter Data with a Benchmarking for State-of-the-Art Opinion Mining Models," in *Proc. 3rd Arabic NLP Workshop (WANLP)*, Valencia, Spain, Apr. 2017. pp. 110-118, doi: https://doi.org/10.18653/v1/W17-1314.

[175] M. Heikal, M. Torki, and N. El-Makky., "Sentiment Analysis of Arabic Tweets Using Deep Learning," *Procedia Comp. Sci.,* vol. 142, pp. 114-122, Nov. 2018. doi: https://doi.org/10.1016/j.procs.2018.10.466.

[176] B. K. Bhavitha, A. P. Rodrigues, and N. N. Chiplunkar, "Comparative Study of Machine Learning Techniques in Sentimental Analysis," in *Proc. 2017 Int. Conf. Inventive Comm. Comput. Tech. (ICICCT)*, Coimbatore, India, Jul. 2017. pp. 216-221, doi: https://doi.org/10.1109/ICICCT.2017.7975191.

[177] A. Rane, and K. Anand, "Sentiment Classification System of Twitter Data for US Airline Service Analysis," in *Proc. 2018 IEEE 42nd Annual Comp. SW App. Conf. (COMPSAC)*, Tokyo, Japan, Jul. 2018. pp. 769-773, doi: https://doi.org/10.1109/COMPSAC.2018.00114.

[178] A. Bittlingmayer, "Dataset: Amazon Reviews for Sentiment Analysis," 2017. Accessed: 28 Jan. 2020. [Online]. Available: https://www.kaggle.com/bittlingmayer/amazonreviews.

[179] M. Michailidis, "Dataset: Sentiment140 Dataset with 1.6 Million Tweets," Kaggle, 2017. Accessed: 28 Jan. 2020. [Online]. Available: https://www.kaggle.com/kazanova/sentiment140.

[180] M. Kaghazgarian, "Dataset: Sentiment Labelled Sentences Dataset," Kaggle, 2018. Accessed: 28 Jan. 2020. [Online]. Available: https://www.kaggle.com/marklvl/sentiment-labelled-sentences-data-set.

[181] H. Obie, "Dataset: Tweets for sentiment analysis," 2015. Accessed: 29 Jan. 2020. [Online]. Available: https://figshare.com/articles/Tweets_for_sentiment_analysis/1579265/1.

[182] "Dataset: Sentiment Analysis in Text," CrowdFlower, 2017. Accessed: 14 Jan. 2021. [Online]. Available: https://data.world/crowdflower/sentiment-analysis-in-text.

[183] "Dataset: Airline Twitter Sentiment," CrowdFlower, 2015. Accessed: 29 Jan. 2020. [Online]. Available: https://data.world/crowdflower/airline-twitter-sentiment.

[184] "Database: Sentiment of Climate Change," CrowdFlower, 2013. Accessed: 29 Jan. 2020. [Online]. Available: https://data.world/crowdflower/sentiment-of-climate-change.

[185] H. Elsahar, "Dataset: Large Arabic Resources For Sentiment Analysis," 2015. Accessed: 29 Jan. 2020. [Online]. Available: https://github.com/hadyelsahar/large-arabic-sentiment-analysis-resouces.

[186] M. Nabil, "ASTD: Arabic Sentiment Tweets Dataset," 2015. Accessed: 29 Jan. 2020. [Online]. Available: https://github.com/mahmoudnabil/ASTD.

[187] N. A. Abdulla, "Dataset: Twitter Dataset for Arabic Sentiment Analysis," 2013. Accessed: 29 Jan. 2020. [Online]. Available: https://data.world/uci/twitter-data-set-for-arabic-sentiment-analysis.

[188] A. Bondielli, and F. Marcelloni, "A Survey on Fake News and Rumour Detection Techniques," *Info. Sci.,* vol. 497, pp. 38-55, Sept. 2019. doi: https://doi.org/10.1016/j.ins.2019.05.035.

[189] M. F. Noordin, R. Othman, and A. H. R. Rassa, "Social Media and Knowledge Management Disruptive Technology," in *Proc. Knowledge Manag. Int. Conf.*, Miri Sarawak, Malaysia, Jul. 2018. pp. 6-11,

[190] F. Pierri, and S. Ceri, "False News On Social Media: A Data-Driven Survey," *ACM Sigmod Record,* vol. 48, no. 2, pp. 18-27, Dec. 2019. doi: https://doi.org/10.1145/3377330.3377334.

[191] C. D. MacDougall, Hoaxes, vol. 465, Dover Publ., 1958.

[192] E. C. Tandoc Jr, Z. W. Lim, and R. Ling, "Defining “Fake News”: A Typology of Scholarly Definitions," *Digi. J.,* vol. 6, no. 2, pp. 137-153, Feb. 2018. doi: https://doi.org/10.1080/21670811.2017.1360143.

[193] JN. Kapferer, Rumors: Uses, interpretations, and images, USA: Transaction Publishers, 2013.

[194] S. M. Alzanin, and A. M. Azmi, "Detecting Rumors in Social Media: A Survey," *Procedia Comp. Sci.,* vol. 142, pp. 294-300, Nov. 2018. doi: https://doi.org/10.1016/j.procs.2018.10.495.

[195] S. Volkova, K. Shaffer, J. Y. Jang, and N. Hodas, "Separating Facts from Fiction: Linguistic Models to Classify Suspicious, and Trusted News Posts on Twitter," in *Proc. 55th Annual Meet. Assoc. Comput. Lingu. (Volume 2: Short Papers)*, Vancouver, BC, Canada, Aug. 2017. pp. 647-653, doi: https://doi.org/10.18653/v1/P17-2102.

[196] S. C. Woolley, and P. N. Howard, Computational Propaganda: Political Parties, Politicians, and Political Manipulation on Social Media, Oxford University Press, 2018.

[197] I. Vogel, and P. Jiang, "Fake News Detection with the New German Dataset “GermanFakeNC”," in *Proc. Int. Conf. Theory Practice Digi. Lib.*, Oslo, Norway, Aug. 2019. pp. 288-295, doi: https://doi.org/10.1007/978-3-030-30760-8_25.

[198] S. Li, K. Ma, X. Niu, Y. Wang et al., "Stacking-Based Ensemble Learning on Low Dimensional Features for Fake News Detection," in *Proc. 2019 IEEE 21st Int. Conf. High Perform. Comput. Comm.*, Zhangjiajie, China, Aug. 2019. pp. 2730-2735, doi: https://doi.org/10.1109/HPCC/SmartCity/DSS.2019.00383.

[199] C. D. da Silva, F. Vieira, R. Garcia, and A. Cristina, "Can Machines Learn to Detect Fake News? A Survey Focused on Social Media," in *Proc. 52nd Hawaii Int. Conf. Sys. Sci.*, Grand Wailea, Maui, Jan. 2019. pp. 2763-2770, doi: https://doi.org/10.24251/HICSS.2019.332.

[200] S. Ghosh and C. Shah, "Towards Automatic Fake News Classification," *Proc. of the Association for Info. Sci. and Tech.,* vol. 55, no. 1, pp. 805-807, 2018.

[201] G. Jardaneh, H. Abdelhaq, M. Buzz, and D. Johnson, "Classifying Arabic Tweets Based on Credibility Using Content and User Features," in *Proc. 2019 IEEE Jordan Int. Joint Conf. Elec. Eng. Info. Tech. (JEEIT)*, Amman, Jordan, Apr. 2019. pp. 596-601, doi: https://doi.org/10.1109/JEEIT.2019.8717386.

[202] I. Santoso, I. Yohansen, Nealson, H. L. H. S. Warnars, and K. Hashimoto, "Early Investigation of Proposed Hoax Detection for Decreasing Hoax in Social Media," in *Proc. 2017 IEEE Int. Conf. Cybernetics Comput. Intell. (CyberneticsCom)*, Phuket, Thailand, Nov. 2017. pp. 175-179, doi: https://doi.org/10.1109/CYBERNETICSCOM.2017.8311705.

[203] "Dataset: FEVER," 2018. Accessed: 15 Jan. 2020. [Online]. Available: https://sheffieldnlp.github.io/fever/resources.html.

[204] M. Risdal, "Dataset: KaggleFN," 2016. Accessed: 15 Jan. 2020. [Online]. Available: https://www.kaggle.com/mrisdal/fake-news.

[205] E. Misback, and C. Pfeifer, "Dataset: FNC-1," 2017. Accessed: 15 Jan. 2020. [Online]. Available: https://github.com/FakeNewsChallenge/fnc-1.

[206] W. Y. Wang, "Dataset: Liar," 2017. Accessed: 15 Jan. 2020. [Online]. Available: https://www.cs.ucsb.edu/~william/data/liar_dataset.zip.

[207] A. Zubiaga, "Dataset: PHEME," 2014. Accessed: 15 Jan. 2020. [Online]. Available: https://www.zubiaga.org/datasets/.

[208] 车, 尚锟, "Dataset: The Sina Weibo Dataset of Double 11 Shopping Day from 2013 to 2017," 2018. Accessed: 15 Jan. 2020. [Online]. Available: https://opendata.pku.edu.cn/dataset.xhtml?persistentId=doi:10.18170/DVN/EC0G0E&language=en.

[209] D. Mahudeswaran, "Dataset: FakeNewsNet," 2017. Accessed: 15 Jan. 2020. [Online]. Available: https://www.kaggle.com/mdepak/fakenewsnet.

[210] J. Singer-Vine, "Dataset: 2016-10-Facebook-Fact-Check," 2016. Accessed: 15 Jan. 2020. [Online]. Available: https://github.com/BuzzFeedNews/2016-10-facebook-fact-check/tree/master/data.

[211] "Dataset: Twitter15, and Twitter16," 2016. Accessed: 15 Jan. 2020. [Online]. Available: https://www.dropbox.com/s/7ewzdrbelpmrnxu/rumdetect2017.zip?dl=0&file_subpath=%2Frumor_detection_acl2017.

[212] P. Meel, and D. K. Vishwakarma, "Fake News, Rumor, Information Pollution in Social Media and Web: A Contemporary Survey of State-of-the-Arts, Challenges, and Opportunities," *Expert Sys. App.,* vol. 153, p. 112986, Sep. 2020. doi: https://doi.org/10.1016/j.eswa.2019.112986.

[213] M. Sokolova, N. Japkowicz, and S. Szpakowicz, "Beyond Accuracy, F-Score and ROC: A Family of Discriminant Measures," in *Proc. Australasian Joint Conf. AI*, Hobart, Australia, Dec. 2006. pp. 1015-1021, doi: https://doi.org/10.1007/11941439_114.

[214] A. Fernández, S. García, M. Galar, R. C. Prati et al., Learning from Imbalanced Data Sets, Berlin: Springer, Oct. 2018.

[215] N. Japkowicz, and S. Stephen, "The Class Imbalance Problem: A Systematic Study," *Intell. Data Anal.,* vol. 6, no. 5, pp. 429-449, Nov. 2002. doi: 10.3233/IDA-2002-6504.

[216] S. Kotsiantis, D. Kanellopoulos, and P. Pintelas, "Handling Imbalanced Datasets: A Review," *GESTS Int. Trans. Comp. Sci. Eng.,* vol. 30, no. 1, pp. 25-36, Dec. 2006.

[217] R. Kumari, and S. K. Srivastava, "Machine Learning: A Review on Binary Classification," *Int. J. Comp. Appl.,* vol. 160, no. 7, pp. 11-15, Feb. 2017. doi: 10.5120/ijca2017913083.

[218] F. Provost, and T. Fawcett, "Robust Classification for Imprecise Environments," *Mach. Learn.,* vol. 42, no. 3, pp. 203-231, Mar. 2001.

[219] M. Bramer, Principles of data mining., vol. 180, London, UK: Springer, Mar. 2007, pp. 333-530. [Online]. Available: https://link.springer.com/content/pdf/10.1007/978-1-4471-7493-6.pdf.

[220] J. Han, J. Pei, and M. Kamber, Data Mining: Concepts and Techniques, Massachusetts, USA: Elsevier, 2011, ch. 6, sec. 12.1, pp. 360-362.

[221] M. Fatourechi, R. K. Ward, S. G. Mason, J. Huggins et al., "Comparison of Evaluation Metrics in Classification Applications with Imbalanced Datasets," in *2008 7th Int. Conf. Mach. Learn. App.*, San Diego, CA, USA, Dec. 2008. pp. 777-782, doi: https://doi.org/10.1109/ICMLA.2008.34.

[222] M. Hossin, and M. N. Sulaiman, "A Review on Evaluation Metrics for Data Classification Evaluations," *Int. J. Data Min. Knowl. Manag. Proc.,* vol. 5, no. 2, pp. 1-11, Mar. 2015. doi: https://doi.org/10.5121/ijdkp.2015.5201.

[223] Y. Jiao, and P. Du, "Performance Measures in Evaluating Machine Learning-Based Bioinformatics Predictors for Classifications," *Quant. Biol.,* vol. 4, no. 4, pp. 320-330, Dec. 2016. doi: https://doi.org/10.1007/s40484-016-0081-2.

[224] D. Chicco, and G. Jurman, "The Advantages of the Matthews Correlation Coefficient (MCC) Over F1 Score and Accuracy in Binary Classification Evaluation," *BMC Genomics,* vol. 21, no. 6, pp. 1-13, Jan. 2020. doi: https://doi.org/10.1186/s12864-019-6413-7.

[225] B. W. Matthews, "Comparison of the Predicted and Observed Secondary Structure of T4 Phage Lysozyme," *Biochim. Biophys. Acta. (BAA), Protein Struct.,* vol. 405, no. 2, pp. 442 - 451, Oct. 1975. doi: https://doi.org/10.1016/0005-2795(75)90109-9.

[226] S. Straube, and M. M. Krell, "How to Evaluate an Agent's Behavior to Infrequent Events?—Reliable Performance Estimation Insensitive to Class Distribution," *Front. Comput. Neurosci.,* vol. 8, pp. 1-6, Apr. 2014. doi: https://doi.org/10.3389/fncom.2014.00043.

[227] F. K. Abu Salem, R. Al-Feel, S. Elbassuoni, M. Jaber et al., "FA-KES: A Fake News Dataset Around the Syrian War," in *Proc. 13th Int. AAAI Conf. Web Soc. Media*, Munich, Germany, Jun. 2019. pp. 573-582, doi: https://ojs.aaai.org/index.php/ICWSM/article/view/3254.

[228] "Penn Part of Speech Tags," Computer Science Department at New York University, Accessed: 4 Apr. 2017. [Online]. Available: https://cs.nyu.edu/grishman/jet/guide/PennPOS.html.

[229] M. T. Alrefaie, "Arabic-Stop-Words," May 2019. Accessed: 29 Jan. 2020. [Online]. Available: https://github.com/mohataher/arabic-stop-words.

[230] "Onix Text Retrieval Toolkit," Lextek International, Accessed: 7 Apr. 2017. [Online]. Available: http://www.lextek.com/manuals/onix/stopwords1.html.

[231] K. Taghva, R. Elkhoury, and J. Coombs, "Arabic Stemming Without a Root Dictionary," in *Proc. Int. Conf. Inf. Tech. Coding Comput. (ITCC'05) - Volume II*, Las Vegas, USA, Apr. 2005. pp. 152-157, doi: https://doi.org/10.1109/ITCC.2005.90.

[232] P. Willett, "The Porter Stemming Algorithm: Then and Now," *Prog. Elec. Lib.,* vol. 40, no. 3, pp. 219-223, Jul. 2006. doi: https://doi.org/10.1108/00330330610681295.

[233] "TextBlob: Simplified Text Processing," 2020. Accessed: 21 Mar. 2020. [Online]. Available: https://textblob.readthedocs.io/en/dev/.

[234] "TextBlob-ar: Arabic Support for Textblob," 2020. Accessed : 21 Mar. 2020. [Online]. Available: https://github.com/adhaamehab/textblob-ar.

[235] C. Theune, "Pycountry," 03 Jul. 2020. Accessed: 20 Aug. 2020. [Online]. Available: https://pypi.org/project/pycountry/#description.

[236] H. M. Zin, N. Mustapha, M. A. A. Murad, and N. M. Sharef, "Term Weighting Scheme Effect in Sentiment Analysis of Online Movie Reviews," *Adv. Sci. Lett.,* vol. 24, no. 2, pp. 933-937, Feb. 2018. doi: https://doi.org/10.1166/asl.2018.10661.

[237] S. J. Raudys, and A. K. Jain, "Small Sample Size Effects in Statistical Pattern Recognition: Recommendations for Practitioners," *IEEE PAMI,* vol. 13, no. 3, pp. 252-264, Mar. 1991. doi: 10.1109/34.75512.

[238] A. Alwosheel, S. V. Cranenburgh, and C. G. Chorus, "Is Your Dataset Big Enough? Sample Size Requirements When Using Artificial Neural Networks for Discrete Choice Analysis," *J. Choice Model.,* vol. 28, pp. 167-182, Sep. 2018. doi: https://doi.org/10.1016/j.jocm.2018.07.002.

[239] G. Afendras, and M. Markatou, "Optimality of Training/Test Size and Resampling Effectiveness in Cross-Validation," *J. Stat. Plan. Inference,* vol. 199, pp. 286-301, Mar. 2019. doi: https://doi.org/10.1016/j.jspi.2018.07.005.

[240] M. Bramer, Principles of data mining., vol. 180, London, UK: Springer, 2007, pp. 333-530. [Online]. Available: https://link.springer.com/content/pdf/10.1007/978-1-4471-7493-6.pdf.

[241] R. Ranawana, and V. Palade, "Optimized Precision - A New Measure for Classifier Performance Evaluation," in *Proc. 2006 IEEE ICEC*, Vancouver, BC, Canada, Sep. 2006. pp. 2254-2261, doi: https://doi.org/10.1109/CEC.2006.1688586.

[242] H. ElSahar, and S. R. El-Beltagy, "Building Large Arabic Multi-Domain Resources for Sentiment Analysis," in *Proc. Int. Conf. Comput. Ling. Intell. Text Process.*, Cairo, Egypt, Apr. 2015. pp. 23-34, doi: https://doi.org/10.1007/978-3-319-18117-2_2.

[243] "Twitter Statistics for Egypt," Accessed: 21 Mar. 2020. [Online]. Available: https://www.socialbakers.com/statistics/twitter/profiles/egypt.

[244] "Egypt's Trends," Accessed: 21 Mar. 2020. [Online]. Available: https://twitter.com/EgyptTrends.

[245] "Amazon Mechanical Turk," Accessed: 21 Mar. 2020. [Online]. Available: https://docs.aws.amazon.com/mturk/index.html.

[246] "Boto API," Accessed: 21 Mar. 2020. [Online]. Available: https://github.com/boto/boto.

[247] A. Go, R. Bhayani, and L. Huang, "Twitter Sentiment Classification Using Distant Supervision," CS224N project report, Stanford 1, no. 12 (2009), 2009.

[248] H. Hakh, I. Aljarah, and B. Al-Shboul, "Online Social Media-Based Sentiment Analysis for US Airline Companies," in *Proc. New Trends Info. Tech.*, Amman, Jordan, Apr. 2017. pp. 176-181,

[249] K. F. M. Panguila, and C. J., "Sentiment Analysis on Social Media Data Using Intelligent Techniques," *Int. J. Eng. Res. Tech.,* vol. 12, no. 3, pp. 440-445, Mar. 2013.

[250] W. Y. Wang, "Liar, Liar Pants on Fire": A New Benchmark Dataset for Fake News Detection," in *Proc. 55th Annu. Meet. Assoc. Comput. Ling. (Volume 2: Short Papers)*, Vancouver, BC, Canada, Jul. 2017. pp. 422-426, doi: 10.18653/v1/P17-2067.

[251] J. Ma, W. Gao, and K. Wong, "Detect Rumors in Microblog Posts Using Propagation Structure via Kernel Learning," in *Proc. 55th Annu. Meet. Assoc. Comput. Ling. (Volume 1: Long Papers)*, Vancouver, BC, Canada, Jul. 2017. pp. 708-717, doi: 10.18653/v1/P17-1066.

[252] J. Golbeck, M. Mauriello, B. Auxier, K. H. Bhanushali et al., "Fake News vs Satire: A Dataset and Analysis," in *Proc. 10th ACM Conf. Web Sci.*, Amsterdam, Netherlands, May 2018. pp. 17-21, doi: https://doi.org/10.1145/3201064.3201100.

[253] A. Tharwat, "Classification Assessment Methods," *Appl. Comput. Informatics,* pp. 1-13, Aug. 2018. doi: https://doi.org/10.1016/j.aci.2018.08.003.

[254] "World Health Organization Official Website," Accessed: 21 Mar. 2020. [Online]. Available: https://www.who.int/.

[255] "Coronavirus disease (COVID-2019) situation reports," World Health Organization, Accessed: 20 Mar. 2020. [Online]. Available: https://www.who.int/emergencies/diseases/novel-coronavirus-2019/situation-reports/.

[256] "Coronavirus (COVID-19) Vaccinations – Statistics and Research," Our World in Data, Accessed: 10 Jun. 2021. [Online]. Available: https://ourworldindata.org/covid-vaccinations.

[257] M. Mhalla, "The Impact of Novel Coronavirus (COVID-19) on the Global Oil and Aviation Markets.," *J. Asian Sci. Res.,* vol. 10, no. 2, pp. 96-104, Apr. Jun. 2020.

[258] N. J. Gormsen, and R. S. Koijen, "Coronavirus: Impact on Stock Prices and Growth Expectations," University of Chicago, Becker Friedman Institute for Economics, Chicago, IL, USA, Working Paper No. 2020-22, Mar. 17, 2020.

[259] R. Baldwin, and B. W. D. Mauro, Economics in the Time of COVID-19, London, UK: CEPR Press, 2020, pp. 73-76. [Online]. Available: https://voxeu.org/content/economics-time-covid-19.

[260] M. Richtel, "W.H.O. Fights a Pandemic Besides Coronavirus: An Infodemic," The New York Times, Accessed : 21 Mar. 2020. [Online]. Available: https://www.nytimes.com/2020/02/06/health/coronavirus-misinformation-social-media.html.

[261] M. Cinelli, W. Quattrociocchi, A. Galeazzi, C. M. Valensise et al., "The COVID-19 Social Media Infodemic.," *arXiv preprint arXiv:2003.05004,* Mar. 2020.

[262] C. Shu, and J. Shieber, "Facebook, Reddit, Google, LinkedIn, Microsoft, Twitter, and YouTube Issue Joint Statement on Misinformation," TechCrunch, Accessed: 21 Mar. 2020. [Online]. Available: https://techcrunch.com/2020/03/16/facebook-reddit-google-linkedin-microsoft-twitter-and-youtube-issue-joint-statement-on-misinformation/.

[263] M. Kefalaki, and S. Karanicolas, "Communication's Rough Navigations: 'Fake' News in a Time of a Global Crisis," *J. Appl. Learn. Teach.,* vol. 3, no. 1, pp. 1-13, Jun. 2020. doi: https://doi.org/10.37074/jalt.2020.3.1.19.

[264] C. R. Taylor, "Advertising and COVID-19," *Int. J. of Advertising,* vol. 39, no. 5, pp. 587-589, Jun. 2020. doi: https://doi.org/10.1080/02650487.2020.1774131.

[265] B. Ansari, and M. Ganjoo, "Impact of Covid-19 on Advertising: A Perception Study on the Effects on Print and Broadcast Media and Consumer Behavior," *Purakala with ISSN 0971-2143 is an UGC CARE J.,* vol. 31, no. 28, pp. 52-62, 2020.

[266] J. Zarocostas, "How to Fight an Infodemic," *The Lancet 395,* vol. 10225, p. 676, Feb. 2020. doi: https://doi.org/10.1016/S0140-6736(20)30461-X.

[267] A. K. Cybenko, and G. Cybenko, "AI and Fake News," *IEEE Intell. Sys.,* vol. 33, no. 5, pp. 1-5, Dec. 2018. doi: https://doi.org/10.1109/MIS.2018.2877280.

[268] K. Sharma, F. Qian, H. Jiang, N. Ruchansky, M. Zhang, and Y. Liu, "Combating Fake News: A Survey on Identification," *ACM Transactions on Intelligent Systems and Technology,* vol. 10, no. 3, pp. 1-42, Apr. 2019. doi: https://doi.org/10.1145/3305260.

[269] "Coronavirus disease (COVID-19)," United Nations International Children's Emergency Fund (UNICEF), Accessed: 21 Mar. 2020. [Online]. Available: https://www.unicef.org/coronavirus/covid-19.

[270] "United Nations Official Website.," Accessed: 22 Mar. 2020. [Online]. Available: https://www.un.org/en/.

[271] I. D. Apostolopoulos, and T. A. Mpesiana, "Covid-19: Automatic Detection From X-ray Images Utilizing Transfer Learning with Convolutional Neural Networks," *Phys. Eng. Sci. Med.,* vol. 43, pp. 635-640, Apr. 2020. doi: https://doi.org/10.1007/s13246-020-00865-4.

[272] D. Singh, V. Kumar, Vaishali, and M. Kaur, "Classification of COVID-19 Patients from Chest CT Images Using Multi-Objective Differential Evolution–Based Convolutional Neural Networks," *EUR J. Clin. Microbiol. Infect. Dis.,* vol. 39, pp. 1379-1389, Apr. 2020. doi: https://doi.org/10.1007/s10096-020-03901-z.

[273] I. D. Apostolopoulos, S. I. Aznaouridis, and M. A. Tzani, "Extracting Possibly Representative COVID-19 Biomarkers from X-ray Images with Deep Learning Approach and Image Data Related to Pulmonary Diseases," *J. Medi. Bio. Eng.,* vol. 40, no. 3, pp. 462-469, Jun. 2020. doi: https://doi.org/10.1007/s40846-020-00529-4.

[274] DP. Fan, T. Zhou, G. P. Ji, Y. Zhou, et al., "Inf-Net: Automatic COVID-19 Lung Infection Segmentation from CT Images," *IEEE Trans. Med. Imag.,* vol. 39, no. 8, pp. 2626 - 2637, Aug. 2020. doi: https://doi.org/10.1109/TMI.2020.2996645.

[275] H. Kang, L. Xia, F. Yan, Z. Wan et al., "Diagnosis of Coronavirus Disease 2019 (COVID-19) with Structured Latent Multi-View Representation Learning," *IEEE Trans. Medi. Imag.,* vol. 39, no. 8, pp. 2606 - 2614, Aug. 2020. doi: https://doi.org/10.1109/TMI.2020.2992546.

[276] B. R. Beck, B. Shin, Y. Choi, S. Park et al., "Predicting Commercially Available Antiviral Drugs that may Act on the Novel Coronavirus (SARS-CoV-2) through a Drug-Target Interaction Deep Learning Model," *Comput. Struct. Biotech. J.,* vol. 18, pp. 784-790, Mar. 2020. doi: https://doi.org/10.1016/j.csbj.2020.03.025.

[277] K. Huang, T. Fu, C. Xiao, L. Glass et al., "DeepPurpose: A Deep Learning-Based Drug Repurposing Toolkit," *arXiv preprint arXiv:2004.08919,* Apr. 2020.

[278] E. Chen, K. Lerman, and E. Ferrara, "Tracking Social Media Discourse About the COVID-19 Pandemic: Development of a Public Coronavirus Twitter DataSet," *JMIR Public Health Surveill,* vol. 6, no. 2, pp. 1-9, May 2020. doi: https://doi.org/10.2196/19273.

[279] CE. Lopez, M. Vasu, and C. Gallemore, "Understanding the Perception of COVID-19 Policies by Mining a Multilanguage Twitter Dataset," *arXiv preprint arXiv:2003.10359,* Mar. 2020.

[280] K. Sharma, S. Seo, C. Meng, S. Rambhatla, et al., "COVID-19 on Social Media: Analyzing Misinformation in Twitter Conversations," *arXiv preprint arXiv:2003.12309,* Mar. 2020.

[281] L. Singh, S. Bansal, L. Bode, C. Budak, et al., "A First Look at COVID-19 Information and Misinformation Sharing on Twitter," *arXiv preprint arXiv:2003.13907,* Mar. 2020.

[282] S. Alqurashi, A. Alhindi, and E. Alanazi, "Large Arabic Twitter Dataset on COVID-19," *arXiv preprint arXiv:2004.04315,* Apr. 2020.

[283] K. Zarei, R. Farahbakhsh, N. Crespi, and G. Tyson, "A First Instagram Dataset on COVID-19," *arXiv preprint arXiv:2004.12226,* Apr. 2020.

[284] L. Cui, and D. Lee, "CoAID: COVID-19 Healthcare Misinformation Dataset," *arXiv preprint arXiv:2006.00885,* May 2020.

[285] "Twitter Streaming API," 2017. Accessed: 21 Mar. 2020. [Online]. Available: https://github.com/spatie/twitter-streaming-api.

[286] "Hydrator: Turn Tweet IDs Onto Twitter JSON & CSV From Your Desktop," 2019. Accessed: 21 Mar. 2020. [Online]. Available: https://github.com/DocNow/hydrator.

[287] "TWARC: A Command Line Tool (and Python Library) for Archiving Twitter JSON," 2019. Accessed: 21 Mar. 2020. [Online]. Available: https://github.com/DocNow/twarc.

[288] F. Haouari, M. Hasanain, R. Suwaileh, and T. Elsayed, "ArCOV-19: The First Arabic COVID-19 Twitter Dataset With Propagation Networks," in *Proc. 6th Arabic NLP Workshop*, Kyiv, Ukraine, Apr. 2021. pp. 82-91,

[289] "Instagram: Official API Graph Instagram," 2020. Accessed: 21 Mar. 2020. [Online]. Available: https://developers.facebook.com/docs/instagram-api.

[290] "UN News COVID-19.," The United Nations, Accessed: 20 Mar. 2020. [Online]. Available: https://news.un.org/en/search/covid-19.

[291] "WHO Director-General Speeches Detail.," World Health Organization, Accessed: 20 Mar. 2020. [Online]. Available: https://www.who.int/dg/speeches/detail/.

[292] "Coronavirus disease (COVID-19) news," World Health Organization, Accessed: 20 Mar. 2020. [Online]. Available: https://www.who.int/emergencies/diseases/novel-coronavirus-2019/media-resources/news.

[293] "Google Fact Check Tools API.," Google, Accessed: 20 Mar. 2020. [Online]. Available: https://toolbox.google.com/factcheck/apis.

[294] K. Sahoo, A. K. Samal, J. Pramanik, and S. K. Pani, "Exploratory Data Analysis Using Python," *Int. J. of Innov. Tech. Explor. Eng. (IJITEE),* vol. 8, no. 12, pp. 4727-4735, Oct. 2019. doi: https://doi.org/10.35940/ijitee.L3591.1081219.

[295] A. Kulkarni, and A. Shivananda, "Exploring and Processing Text Data," *NLP Recipes,* pp. 37-65, Jan. 2019. doi: https://doi.org/10.1007/978-1-4842-4267-4_2.

[296] "Plotly Python Open Source Graphing Library," 2020. Accessed: 21 Mar. 2020. [Online]. Available: https://plot.ly/python/.

[297] "Bokeh Visualization Library," 2019. Accessed: 21 Mar. 2020. [Online]. Available: https://docs.bokeh.org/en/latest/.

[298] M. Liberman, "Alphabetical List of Part-of-Speech Tags Used in the Penn Treebank Project.," Sep. 2020. Accessed: 25 Mar. 2020. [Online]. Available: https://www.ling.upenn.edu/courses/Fall_2020/ling001/penn_treebank_pos.html.

[299] "TensorFlow," Accessed : 15 May 2020. [Online]. Available: https://www.tensorflow.org.

[300] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global Vectors for Word Representation," in *Proc. 2014 Conf. Empir. meth. NLP (EMNLP)*, Doha, Qatar, Oct. 2014. pp. 1532-1543, doi: http://dx.doi.org/10.3115/v1/D14-1162.

[301] S. Paul, "Keras Sequential Api," 2018. Accessed : 5 Apr. 2020. [Online]. Available: https://medium.com/@subhamoy.paul986/keras-sequential-api-72e45c39259b.

[302] J. Huang, and C. X. Ling, "Using AUC and Accuracy in Evaluating Learning Algorithms," *IEEE Trans Knowl Data Eng,* vol. 17, no. 3, pp. 299-310, Mar. 2005.

[303] D. S. Abdelminaam, F. H. Ismail, M. Taha, A. Taha et al., "CoAID-DEEP: An Optimized Intelligent Framework for Automated Detecting COVID-19 Misleading Information on Twitter," *IEEE Access,* vol. 9, pp. 27840 - 27867, Feb. 2021. doi: https://doi.org/10.1109/ACCESS.2021.3058066.

[304] A. Y. A. Amer, T. Siddiqui, "Detection of Covid-19 Fake News text data using Random Forest and Decision tree Classifiers," *Int. J. Comp. Sci. Info. Sec.,* vol. 18, no. 12, pp. 88-100, Dec. 2020. doi: https://doi.org/10.5281/zenodo.4427204.

[305] I. Lima, and N. Chob, "Misinformation detection and rectification based on QA system and text similarity with COVID-19," in *Proc. 24th Int. Conf. IT App. Manag.*, Korea, Feb. 2021. pp. 102-111,

[306] M. Silva, F. Ceschin, P. Shrestha, C. Brant, J. Fernandes et al., "Predicting Misinformation and Engagement in COVID-19 Twitter Discourse in the First Months of the Outbreak," *arXiv preprint arXiv:2012.02164,* Dec. 2020.

[307] V. Mazzeo, A. Rapisarda, and G. Giuffrida, "Detection of fake news on CoViD-19 on Web Search Engines," *arXiv preprint arXiv:2103.11804,* Mar. 2021.

[308] S. Gundapu, and R. Mamid, "Transformer based Automatic COVID-19 Fake News Detection System," *arXiv preprint arXiv:2101.00180,* Jan. 2021.

[309] "International Committee of The Red Cross," Accessed: 21 Mar. 2021. [Online]. Available: https://www.icrc.org/en.

[310] "World Health Organization Official Twitter Account," Accessed: 21 Mar. 2021. [Online]. Available: https://twitter.com/WHO.

[311] "United Nations International Children's Emergency Fund (UNICEF) Official Twitter Account," Accessed: 21 Mar. 2021. [Online]. Available: https://twitter.com/UNICEF.

[312] "United Nations Official Twitter Account," Accessed: 21 Mar. 2021. [Online]. Available: https://twitter.com/UN.

[313] "International Committee of The Red Cross Official Twitter Account," Accessed: 21 Mar. 2021. [Online]. Available: https://twitter.com/ICRC.

[314] N. Japkowicz, "Assessment Metrics for Imbalanced Learning," in *Imbalanced Learning*, New Jersey, USA, John Wiley & Sons, Inc., May 2013, pp. 187-205.

[315] M. Bekkar, H. K. Djemaa, and T. A. Alitouche, "Evaluation Measures for Models Assessment Over Imbalanced Datasets," *J. Info. Eng. Appl.,* vol. 3, no. 10, pp. 27-38, Apr. 2013.

[316] K. H. Brodersen, C. S. Ong, K. E. Stephan, and J. M. Buhmann, "The Balanced Accuracy and its Posterior Distribution," in *Proc. 2010 20th Int. Conf. Pattern Recognit.*, Istanbul, Turkey, Aug. 2010. pp. 3121-3124, doi: 10.1109/ICPR.2010.764.

[317] G. Cohen, M. Hilario, H. Sax, S. Hugonnet, and A. Geissbuhler, "Learning from Imbalanced Data in Surveillance of Nosocomial Infection," *AI Medicine,* vol. 37, no. 1, pp. 7-18, May 2006. doi: https://doi.org/10.1016/j.artmed.2005.03.002.

[318] V. García, R. A. Mollineda, and J. S. Sánchez, "Index of Balanced Accuracy: A Performance Measure for Skewed Class Distributions," in *Proc. Iberian Conf. Pattern Recognit. Img. Anal.*, Berlin, Heidelberg, Jun. 2009. pp. 441-448, doi: https://doi.org/10.1007/978-3-642-02172-5_57.

[319] C. Ferri, J. Hernández-Orallo, and R. Modroiu, "An Experimental Comparison of Performance Measures for Classification," *Pattern Recognit. Lett.,* vol. 30, no. 1, pp. 27-38, Jan. 2009. doi: https://doi.org/10.1016/j.patrec.2008.08.010.

[320] S. Branders, "Regression, Classification and Feature Selection from Survival Data: Modeling of Hypoxia Conditions for Cancer Prognosis," UCL Ph.D. Thesis, la-Neuve, Belgium, Sep. 2015.

[321] I. H. Witten, E. Frank, M. A. Hall, and C. J., Data Mining: Practical Machine Learning Tools and Techniques, Massachusetts, USA: Elsevier, 2016.

[322] J. Cohen, "A Coefficient of Agreement for Nominal Scales," *Educ. Psychol. Meas.,* vol. 20, no. 1, pp. 37-46, Apr. 1960. doi: https://doi.org/10.1177/001316446002000104.

[323] R. Delgado, and X. Tibau, "Why Cohen's Kappa should be Avoided as a Performance Measure in Classification," *PLoS ONE,* vol. 14, no. 9, pp. 1-26, Sep. 2019. doi: https://doi.org/10.1371/journal.pone.0222916.

[324] N. V. Chawla, D. A. Cieslak, L. O. Hall, and A. Joshi, "Automatically Countering Imbalance and its Empirical Relationship to Cost," *Data Min. Knowl. Disc.,* vol. 17, pp. 225-252, Feb. 2008. doi: https://doi.org/10.1007/s10618-008-0087-0.

[325] E. B. Fowlkes, and C. L. Mallows, "A Method for Comparing Two Hierarchical Clusterings," *J. American Stat. Assoc.,* vol. 78, no. 383, pp. 553-569, Sep. 1983. doi: 10.1080/01621459.1983.10478008.

[326] M. Kubat, and S. Matwin, "Addressing the Curse of Imbalanced Training Sets: One-Sided Selection," in *Proc. 14th Int. Conf. Mach. Learn.*, California, USA, Jul. 1997. pp. 179-186, doi: https://dl.acm.org/doi/proceedings/10.5555/645526?id=31.

[327] R. Batuwita, and V. Palade, "Adjusted Geometric-Mean: a Novel Performance Measure for Imbalanced Bioinformatics Datasets Learning," in *Proc. 2009 Int. Conf. Mach. Learn. Appl.*, Florida, USA, Dec. 2009. pp. 545-550, doi: 10.1109/ICMLA.2009.126.

[328] R. Real, and J. Vargas, "The Probabilistic Basis of Jaccard's Index of Similarity," *Syst. Biol.,* vol. 45, no. 3, pp. 380-385, Sep. 1996. doi: 10.2307/2413572.

[329] D. Powers, "Evaluation: From Precision, Recall and F-Factor to ROC, Informedness, Markedness & Correlation," School of Informatics and Engineering Flinders University of South Australia, Adelaide, Australia, Dec. 2007.

[330] J. R. Thornbury, D. G. Fryback, and W. Edwards, "Likelihood Ratios as a Measure of the Diagnostic Usefulness of Excretory Urogram Information," *Radiology,* vol. 114, no. 3, pp. 561-565, Mar. 1975. doi: https://doi.org/10.1148/114.3.561.

[331] W. J. Youden, "Index for Rating Diagnostic Tests," *Cancer,* vol. 3, no. 1, pp. 32-35, Jan. 1950. doi: https://doi.org/10.1002/1097-0142(1950)3:1<32::AID-CNCR2820030106>3.0.CO;2-3.

[332] J. A. Hanley, and B. J. McNeil, "The Meaning and Use of the Area Under a Receiver Operating Characteristic (ROC) Curve," *Diagnostic Radiology,* vol. 143, no. 1, pp. 29-36, Apr. 1982. doi: https://doi.org/10.1148/radiology.143.1.7063747.

[333] R. Engler, A. Guisan, and L. Rechsteiner, "An Improved Approach for Predicting the Distribution of Rare and Endangered Species from Occurrence and Pseudo-Absence Data," *J. applied ecology,* vol. 41, no. 2, pp. 263-274, Apr. 2004. doi: https://doi.org/10.1111/j.0021-8901.2004.00881.x.

[334] T. Stéphane, Data Mining et Statistique Décisionnelle: l'intelligence des Données, Paris, France: Editions Technip, Aug. 2012.

Appendix A

Performance Evaluation Metrics Detailed Overview

Many researchers have grouped the used performance evaluation metrics used to evaluate supervised learning algorithms into three main groups from two points of view, as shown in Figure A.1.

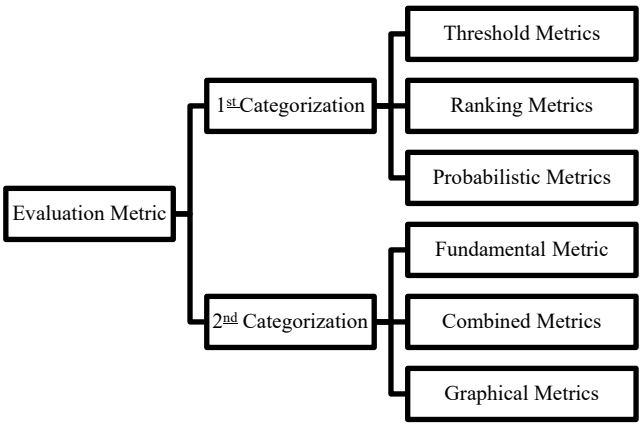


Figure A.1 Categories of Performance Evaluation Metrics.

The groups that represent the first points of view are Threshold Metrics (TM) (e.g., GM and F1-Score), Ranking Metrics (RM) (e.g., receiver operating characteristics (ROC) analysis and AUC), and Probabilistic Metrics (PM) (e.g., root-mean-squared error (RMSE)) [314]. While from the second point of view, the groups are: Fundamental Metric (FM) (e.g., ACC and Err, and P), Combined Metrics (CM) (e.g., GM and F1-Score, and MCC), and Graphical Metrics (GRM) (e.g., ROC, and Cumulative Gains Curve (CGC)) [315].

In this dissertation, we argue that the second categorization is more comprehensive and clearer than the first one. Hence, we will discuss the evaluation metrics from this point of view, as follows.

A.1 Fundamental Performance Evaluation Metrics (FM)

FM are those evaluation metrics that are basic and calculated directly from the generated confusion matrix for each of the used classification algorithms.

A.1.1 Accuracy (ACC) and Error Rate (ERR) [219] [220]*:*

ACC is a very basic and straightforward metric used for performance evaluation. It measures the overall effectiveness of the algorithm by obtaining the probability of the true value of the class label, as follows.

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \tag{A.1}$$

On the other hand, ERR is the complement of the ACC. It measures the misclassification probability according to the model prediction. The highest possible value for ACC is 1, which means that the classifier was able to correctly classify all the tested instances, and hence the ERR is 0. In other words, for a classifier to be good in terms of ACC, it must have a value that tends to 1 or ERR that tends to 0, as follows.

$$\mathrm{ERR} = 1 - \mathrm{ACC} = \frac{\mathrm{FP} + \mathrm{FN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}} \tag{A.2}$$

The ACC and the ERR measures, despite being easy to compute with less complexity, have limitations in the evaluation of a classifier and discrimination process. One of the main limitations of ACC is that it produces less distinctive and less discriminable values. Consequently, its ability in selecting and determining the best classification algorithm is diminished. Besides, ACC is also less informative and biased towards minority class instances.

A.1.2 Precision (P) and Recall (R) [219] [220]*:*

P (also known as Positive Predictive Value (PPV)) is the fraction of relevant positive instances among the retrieved instance, as follows.

$$\mathrm{P} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} = \mathrm{PPV} \tag{A.3}$$

False Discovery Rate (FDR) is the complement of P; it represents the fraction of positive instances that are incorrectly classified as negative instances, as follows.

$$\mathrm{FDR} = 1 - \mathrm{P} = \frac{\mathrm{FP}}{\mathrm{TP} + \mathrm{FP}} \tag{A.4}$$

While R (also known as True Positive Rate (TPR) or Sensitivity (SN)) measures the proportion of positives that are correctly identified, as follows.

$$\mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{A.5}$$

False Negative Rate (FNR) (also known as Miss Rate (MR)) is the complement of R; It is the proportion of positive instances that are incorrectly classified, as follows.

$$\mathrm{FNR} = 1 - \mathrm{TPR} = \frac{\mathrm{FN}}{\mathrm{FN} + \mathrm{TP}} \tag{A.6}$$

The ideal value of both P and R that indicates the best performance of a classifier is 1 and the worst possible value would be 0.

A.1.3 Negative Predictive Value (NPV) and True Negative Rate (TNR) [253]*:*

To evaluate the classifiers' performance when dealing with negative class NPV and TNR are used. NPV is the fraction of relevant negative instances among the retrieved instance, as follows.

$$NPV = \frac{TN}{TN + FN} \tag{A.7}$$

False Omission Rate (FOR) is the complement of NPV; it represents the fraction of negative instances that are incorrectly classified as negative instances, as follows.

$$FOR = 1 - NPV = \frac{FN}{FN + TN} \tag{A.8}$$

While TNR (also known as Selectivity (SL), or Specificity (SP)) measures the proportion of negative that are correctly identified, as follows.

$$TNR = \frac{TN}{TN + FP} \tag{A.9}$$

False Positive Rate (FPR) (also known as Fall-Out (FO)) is the complement of TNR; It is the proportion of negative instances that are incorrectly classified, as follows.

$$FPR = 1 - TNR = \frac{FP}{TN + FP} \tag{A.10}$$

The ideal value of both NPV and TNR that indicates the best performance of a classifier is 1 and the worst possible value would be 0. We should note that these metrics are highly affected by the change in the distribution of data on the classes, and hence are not suitable to be used with imbalanced datasets.

A.2 Combined Performance Evaluation Metrics (CM)

CM are those evaluation metrics that are obtained by combining two or more of the FMs, as shown in Table 2.6. We discuss these metrics in alphabetical order, as follows.

A.2.1 Balanced Accuracy (BA [316]*, Mean-Class-Weighted Accuracy (MCW)* [317]*, Index of Balanced Accuracy (IBA)* [318]*, and Balance Error Rate (BER)* [253]*:*

It is the average of TPR and TNR (SN and SP). It can be defined also as the average accuracy obtained in either class. It is also known as Macro-Averaged Accuracy (MAA) [319], or Balanced Classification Rate (BCR) [320], as follows.

$$BA = \frac{TPR + TNR}{2} = \frac{SN + SP}{2} \tag{A.11}$$

In other words, it measures the geometric average of the partial accuracies of each class. If the classifier performs equally well in either class, this term reduces to the ACC from eq. 1. In contrast, if the ACC is high only because the classifier takes advantage of good classification results on the majority class, then the BA will drop.

MCW is similar to BA, but it deploys a weight (ω), between 0 and 1, used to increase the importance given to one class over the other, as follows.

$$MCW = \omega \times SN + (1 - \omega) \times SP \quad (A.12)$$

Where, ω is a value between 0 and 1, which represents the weight assigned to the positive class.

IBA is used to evaluate the classification accuracy for binary classification algorithms when using imbalanced datasets. This metric combines both the ACC and a measure about how dominant the class with the highest ACC rates of each class (TPR and TNR) is, as follows.

$$IBA = ACC \times \left(1 + \propto \times (TPR - TNR)\right) \quad (A.13)$$

Where: $\propto$ is a weighting factor designed to reduce the influence of the dominance on the resulted ACC.

High IBA values are obtained when the accuracies of both classes are high and balanced. Unlike most metrics, the IBA aims to promote classifiers with better results on the positive class (which is in most cases considered to be the most important class). The IBA can be generalized to any other metrics, as follows.

$$IBA_{\alpha}(M) = M \times \left(1 + \propto \times (TPR - TNR)\right) \quad (A.14)$$

Where: M is any metric, and $\propto$ is a weighting factor designed to reduce the influence of the dominance on the result of a particular metric M.

Finally, BER (also known as Half Total Error Rate (HTER)) is the complement of BA; It is the proportion of negative instances that are incorrectly classified, as follows.

$$BER = 1 - BA = 1 - \frac{TPR + TNR}{2} \quad (A.15)$$

A.2.2 Cohen Kappa Metric (κ) [321]*:*

It was first introduced in 1960 by Cohen [322] to serve as a measure of agreement between two judges in the field of psychology. Then, in 2006, it was reintroduced by Witten et al. [321] as a performance measure for classification problems; κ is used as a measure of agreement between observed and predicted or inferred classes for cases in testing/validation datasets, but it performs bad with imbalanced datasets [323], as follows.

$$\kappa = 1 - \frac{1 - p_o}{1 - p_e} \quad (A.16)$$

Where: p_o is the relative observed agreement among raters (ACC), and p_e is the hypothetical probability of chance agreement, as follows.

$$p_e = \left(\frac{TP + FP}{TP + TN + FP + FN} \times \frac{TP + FN}{TP + TN + FP + FN}\right) + \left(\frac{TN + FP}{TP + TN + FP + FN} \times \frac{TN + FN}{TP + TN + FP + FN}\right) \quad (A.17)$$

A.2.3 Discriminant Power (DP) [315]*:*

It is used to evaluate how good a classification algorithm is in distinguishing between instances of positive and negative classes, as follows.

$$\mathrm{DP} = \frac{\sqrt{3}}{\pi}(\log X + \log Y) \tag{A.18}$$

Where: $\mathrm{X} = \frac{\mathrm{TPR}}{\mathrm{FNR}}$, and $\mathrm{Y} = \frac{\mathrm{TNR}}{\mathrm{FPR}}$

If the obtained DP is less than 1, this means that the classification algorithm is a poor discriminant. If the obtained DP is between 1 and 2, this means that the classification algorithm has a limited discrimination ability. If the obtained DP is between 2 and 3, this means that the classification algorithm has a fair discrimination ability. If the obtained DP is greater than 3, this means that the classification algorithm has a good discrimination ability.

A.2.4 Fbeta-Score (F_β) [324]*:*

It is a cost-based harmonic mean between P and R. It uses a β weight to adjust the relative importance of P versus R which depends on the cost ratio between them. Misclassifying instances that belong to the minority class becomes more costly relative to instances that belong to the majority class. Consequently, this will make R have a higher impact on F1 than P. Therefore, the classification accuracy of instances of the minority class becomes more important as the costs become more divergent, where decreasing β leads to a reduction of precision importance, as follows.

$$\mathrm{F}_\beta = \frac{(1+\beta^2) \times \mathrm{P} \times \mathrm{R}}{\beta^2 \times \mathrm{P} + \mathrm{R}}, \quad \beta = \frac{\mathrm{C}(1,0)}{\mathrm{C}(0,1)} \tag{A.19}$$

Where: C (1,0) is the cost associated with the prediction of FN, C (0,1) is the cost associated with the prediction of FP.

β typically takes the values of 0.5, 1, or 2. When β =0.5, this means that that P weighs twice as much as R. When β =1, this means that both P and R have an equal impact on the performance of the classification algorithm. In the case of setting $\beta = 1$, this will result in getting the equivalent value to equal the F1, as follows.

$$\mathrm{F1} = \frac{2 \times \mathrm{P} \times \mathrm{R}}{\mathrm{P} + \mathrm{R}} \tag{A.20}$$

When β =2, this means that R weighs twice as much as P. We should note that, F1 provides more insight into the functionality of a classification algorithm than ACC.

A.2.5 Fowlkes-Mallow's index (FMI) [325]*:*

It is used to either compute the similarity between two hierarchical clusters or a predefined cluster and the resulted classes obtained from a classification algorithm. It represents the geometric mean between P and R. In other words, this measure is used to compare either two cluster label sets or a cluster label set with a true label set, as follows.

$$\text{FMI} = \sqrt{(\text{P} \times \text{R})} = \frac{\text{TP}}{\sqrt{(TP + FN)(\text{TP} + \text{FP})}} \tag{A.21}$$

The higher the value for the FMI the greater is the similarity between the clusters and the benchmark classifications.

A.2.6 Geometric Mean (GM) [326] *and Adjusted Geometric Mean (AGM)* [327]*:*

GM is a measure that combines both sensitivity and specificity measures. It indicates the balance between the classifier's performances in the majority and minority classes. It helps in voiding the overfitting to the negative class and the degree to which the positive class is marginalized [315], as follows.

$$\text{GM} = \sqrt{\text{SP} \times \text{SN}} \tag{A.22}$$

A poor performance in the classification of the positive instances will lead to a low GM value, even if the negative instances are correctly classified per the model. The GM may lead to picking a model that lowers SP too much, which is not good especially when misclassification costs are not exactly known. To overcome these problems, AGM was introduced in [327].

AGM is a measure to deal with the type of class imbalance problems whose purpose is to increase the SN while reducing SP to as minimum as possible. It is more sensitive to variations in SP than in SN. Furthermore, it is slightly affected by the variations in the class distribution, as follows.

$$\text{AGM} = \begin{cases} \dfrac{\text{GM} + N_n \times TNR}{1 + N_n}, & TPR > 0 \\ 0, & TPR = 0 \end{cases} \tag{A.23}$$

Where: N_n is the count of negative instances.

A.2.7 Jaccard Index (J) [328]*:*

It is a verification measure of binary classification performance. It is equal to the total number of correctly classified positive instances divided by the total number of positive instances in addition to the positively misclassified instances. This metric is not considering performance in classifying the negative class instances. i.e., J is a biased score that is dependent upon the frequency of the positive class, as follows.

$$\text{J} = \frac{\text{TP}}{\text{TP+FN+FP}} \tag{A.24}$$

A.2.8 Markedness (MK) [329]*:*

It is used to measure the reliability of positive and negative classifications that are obtained by a classifier. MK is defined based on PPV and NPV metrics as follows.

$$\mathrm{MK} = \mathrm{PPV} + \mathrm{NPV} - 1 \tag{A.25}$$

PPV and NPV are highly affected by the change in the distribution of data on the classes. Accordingly, MK is not suitable to be used with class imbalanced datasets.

A.2.9 Matthews Correlation Coefficient (MCC) [225]*:*

It is a correlation coefficient between the observed and predicted binary classifications. MCC is defined as follows.

$$\mathrm{MCC} = \frac{\mathrm{TP} \times \mathrm{TN} - \mathrm{FP} \times \mathrm{FN}}{\sqrt{(\mathrm{TP} + \mathrm{FP})(\mathrm{TP} + \mathrm{FN})(\mathrm{TN} + \mathrm{FP})(\mathrm{TN} + \mathrm{FN})}} \tag{A.26}$$

It returns a value between −1 and +1. A coefficient of +1 represents a perfect prediction, 0 is no better than random prediction and −1 indicates total disagreement between prediction and observation. The MCC is generally regarded as being the most informative single score to determine the quality of a binary classifier prediction in a confusion matrix context [223]. It has the advantage of being less sensitive to change in class distribution as it considers correct classification rates and error rates in both classes. Hence, MCC is suitable to deal with class imbalanced datasets [224]. The main limitation is that in some cases assumptions are being made, by arbitrarily setting the denominator to one when any of the four sums in the denominator is zero; this results in an MCC=0, which can be shown to be the correct limiting value. Moreover, it is unable to assess the performance of the classifier when classifying instances of a particular class.

A.2.10 Negative Likelihood Ratio (LR (-)) and Positive Likelihood Ratio (LR (+)) [330]*:*

LR (-) and LR (+) are probability ratios that are used to measure the impact of the obtained result on the probability correct classification for both negative and positive classes, respectively. LR (-) measures how much the odds of the positive class decrease when a classification test is negative, as follows.

$$\mathrm{LR(-)} = \frac{\text{1-TPR}}{\mathrm{TNR}} \tag{A.27}$$

Similarly, LR (+) how much the odds of the positive class increase when a classification test is positive, as follows.

$$\mathrm{LR(+)} = \frac{\mathrm{TPR}}{\text{1-TNR}} \tag{A.28}$$

Both LR (-) and LR (+) are merged into one measure named the Diagnostic Odds Ratio (DOR). DOR summarizes the performance of the test, as follows.

$$\mathrm{DOR} = \frac{\mathrm{LR(+)}}{\mathrm{LR(-)}} = \frac{\mathrm{TP} \times \mathrm{TN}}{\mathrm{FP} \times \mathrm{FN}} \tag{A.29}$$

These measures are suitable when dealing with both balanced and imbalanced datasets. It has a limitation that the denominator can be equal to 0, in some cases, which will lead to a mathematical error. E.g., if the classifier has TNR=0, TNR=1, or if either FP or FN is equal to 0.

A.2.11 Optimized Precision (OP) [241]*:*

OP was first introduced in 2006 by Ranawana and Palade [241] to overcome the limitation of P when dealing with imbalanced datasets. It is used to optimize both TPR and TNR, as follows.

$$\mathrm{OP} = \mathrm{N_n} \times \mathrm{TNR} + \mathrm{N_p} \times \mathrm{TPR} - \frac{|\mathrm{TNR} - \mathrm{TPR}|}{\mathrm{TNR+TPR}} \tag{A.30}$$

Where: $\mathrm{N_n}$ and $\mathrm{N_p}$ are the count of negative and positive instances, respectively.

A.2.12 Youden's index (γ) [331]*:*

γ (also known as Bookmaker Informedness (BM)) is used to evaluate the ability of a classifier to avoid failure. It is derived from a linear transformation of the mean TPR and TNR, and it denotes a linear correspondence BA, as follows.

$$\gamma = \mathrm{TPR} + \mathrm{TNR} - 1 \tag{A.31}$$

$$\gamma = 2 \times \mathrm{BA} - 1 \tag{A.32}$$

The values of γ ranged from 0 to 1. The higher the value of γ the better the classifier's ability to avoid failure. It is suitable when dealing with imbalanced datasets. One of the main shortcomings of this metric is that γ equally weights its performance on positive and negative examples, and it does not change regarding the differences between the TPR and TNR of the test.

A.3 Graphical Performance Evaluation Metrics

To compare the performance of different classification models, many graphical techniques are used to represent the behavior of the classifiers, such as Receiver Operating Characteristic (ROC), Gain Chart, or Lift chart. ROC curve gives a visual display of both TPR and FPR (SN and SP) for all possible cut-offs in a single plot, which is much better and more powerful than using a series of tables. The displayed TPR and FPR are for different cut-off values for probability (If the probability of positive response is above the cut-off, we predict a positive outcome, if not we are predicting a negative one). Each cut-off value defines one point on the ROC curve, ranging cut-off from 0 to 1 will draw the whole ROC curve, as shown in Figure A.2.

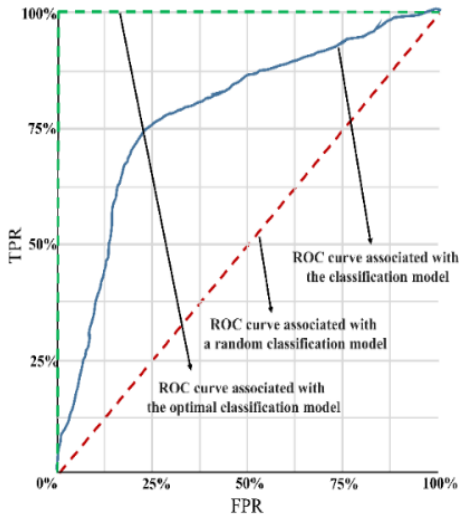


Figure A.2 ROC Curve Example.

Gain Chart (typically called a Cumulative Gains Chart (CGC)) represents the positive relative to the percentage of targeted class according to score deciles, as shown in Figure A.3. While for the Lift Chart, each point represents the ratio between the percentage positive to the percentage of the targeted population, as shown in Figure A.4. It should be noted that the lift chart is directly derived from the gain chart for the same classifier on the same dataset.

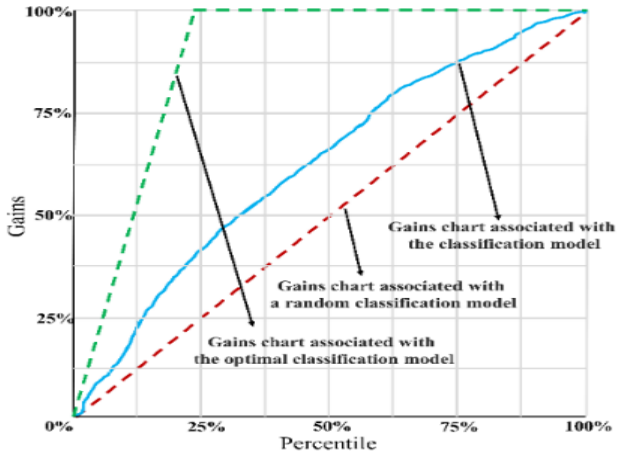


Figure A.3 Gain Chart Example.

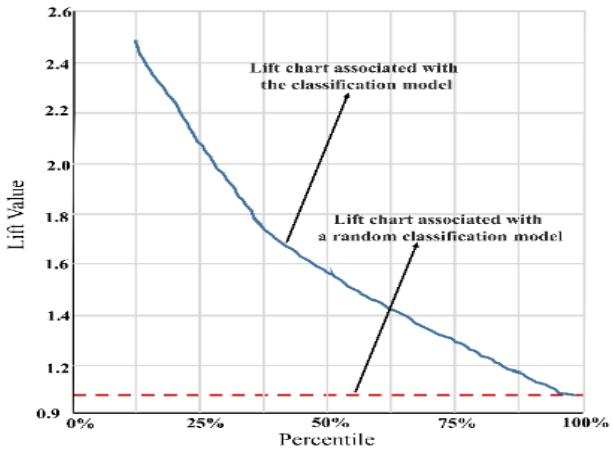


Figure A.4 Lift Chart Example.

The shape of curves expresses the behavior of the classification model under investigation. It clearly shows how superior is this model compared to the model with random performance. Moreover, these curves show how far the model under investigation is from the optimal model which is practically unachievable. These curves can help in obtaining the best cut-off value that will be used to classify instances into either positive or negative.

It should be noted that these curves are used to visualize the performance of the classification model in predicting only one class at a time (whether for positive or negative classes). Typically, these curves are used to show the ability of the classifier when classifying positive instances. If the primary objective of the classifier is to classify negative instances, then the values used in the curves will be adapted to be able to visualize the behavior of the classifier in dealing with the classification of negative class (i.e., we will plot the values of FNR and TNR in the x-axis and the y-axis in the ROC curve, respectively).

A.3.1 Area Under the Curve (AUC) [332]*:*

It is a measure used to summarize the performance of a classifier into a single metric. It can be estimated either by graphically calculating the area under the Receiver Operating Characteristic (ROC) curve using the Trapezoidal Method (TM). TM is based on the linear interpolation between each point on the ROC curve and it has a value ranging between 0.5 and 1. Figure A.5 shows a sample ROC curve and the AUC. The higher the AUC value, the better the classification ability of the classifier. For simplicity and in binary classification problems, AUC can be approximately calculated based on TPR and TNR, as follows.

$$\mathrm{AUC} = \frac{1}{2}(\mathrm{TPR} + \mathrm{TNR}) \tag{A.33}$$

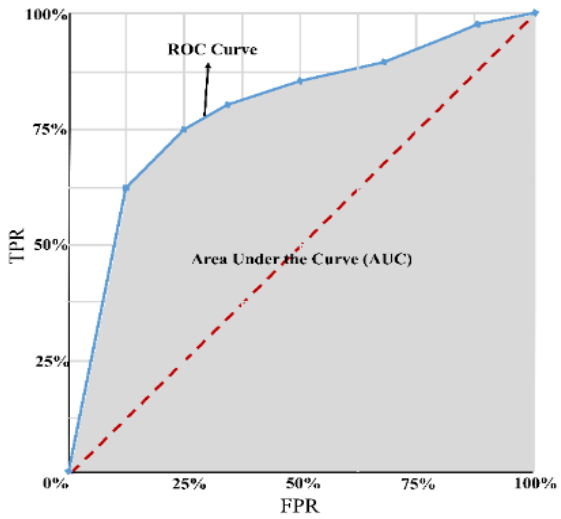


Figure A.5 ROC and AUC Example.

It should be remarked that AUC only concerns with the performance of the system when dealing with positive class. Hence, it positively can, sometimes, give a misleading insight into the classifier's performance. Moreover, it has been proven empirically and empirically that the AUC is much better than ACC, and F1 when evaluating the performance of classification algorithms and choosing the optimal solution during the model building process [375]. Additionally, although the performance of AUC is excellent for evaluation and discrimination, its computational cost is high, especially when dealing with large datasets [222].

A.3.2 Gini Coefficient (GC) [333]*:*

It is a metric used to compare the quality of different models and evaluate their classification power. GC is defined as the ratio between the area within the model curve and the random model line (A) and the area between the perfect model curve and the random model line (A+B), as follows.

$$GC = \frac{A}{A+B} = 2 \times AUC - 1 \quad (A.34)$$

Figure A.6 shows an illustrative example of GC and different regions that affect its calculation.

A.3.3 Area Under Lift (AUL) [334]*:*

It is similar to AUC, but it is derived from the Cumulative Gains Curve (CGC). CGS represents the positive relative to the percentage of the targeted class according to score deciles. While for the lift curve, each point represents the ratio between the percentage positive to the percentage of the targeted population. It can be used to get insights into the performance of different classifiers when dealing with an imbalanced dataset, as shown in Figure A.7.

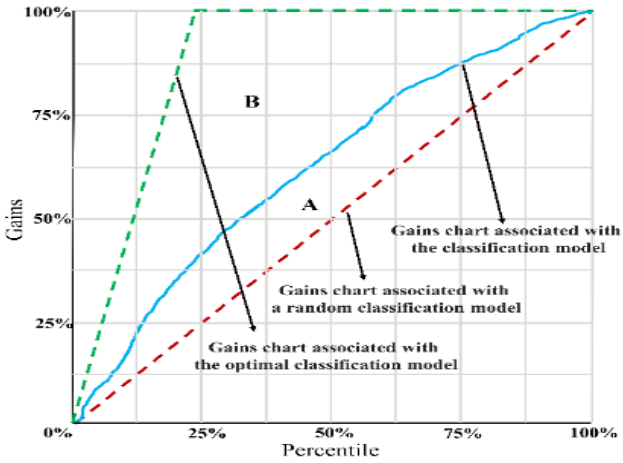


Figure A.6 GC Example.

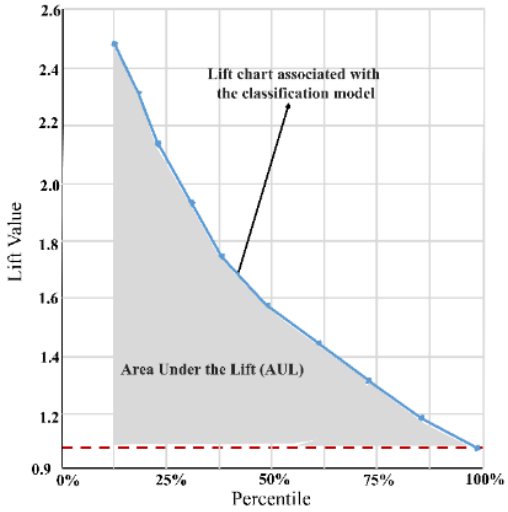


Figure A.7 AUL Example.

The value of AUL ranges from 0.5 (random performance) to1 (best performance), and it is calculated as follows.

$$\mathrm{AUL} = \frac{\mathrm{P}}{2} + (1 - \mathrm{P}) \times \mathrm{AUC} \qquad \text{(A.35)}$$

Where: P = Prior probability of positive observation on the population.